Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 253 (2025) 2377–2388

Procedia Computer Science

www.elsevier.com/locate/procedia

6th International Conference on Industry 4.0 and Smart Manufacturing

# Enhancing flexibility in intralogistics 4.0 by using Services, Capabilities, and Skills

Benjamin Blumhofer[a,*], Philipp Richard[a], Tatjana Legler[a,b], Martin Ruskowski[a,b]

[a]German Research Center for Artificial Intelligence GmbH (DFKI), Kaiserslautern, Germany
[b]Chair of Machine Tools and Control Systems. University of Kaiserslautern-Landau (RPTU)

**Abstract**

Manufacturing companies face a dynamic environment shaped by various factors that profoundly affect their production capacities. These factors encompass trends like shortened product life cycles, a surge in product variants, and subsequent reductions in batch sizes. Consequently, companies must adapt their operational capabilities, ensuring existing machinery remains versatile while seamlessly integrating new equipment into their production facilities, following the "plug and produce" approach. These shifts also reverberate through intralogistics, altering flexibility requirements and methods for individualized goods handling. Despite significant progress in modeling production information using the Capability-Skill-Service model, its application in intralogistics is relatively limited to date. However, given the potential benefits, the integration of this model into both intralogistics and production promises to address one of the key challenges of Industry 4.0: the harmonization of planning and execution processes in production and intralogistics. Closing this gap, this paper proposes an architectural framework that includes core components and information models for a Capability-Skill-Service-based Intralogistics 4.0 application. This framework not only facilitates the seamless integration of intralogistics with production planning and execution, but also lays the foundation for greater flexibility and efficiency in manufacturing. With an implementation the framework and the information models are validated.



*Keywords:* Industry 4.0; Intralogistics 4.0; Asset Administration Shell; CSS-model

## 1. Introduction

In the rapidly evolving landscape of modern manufacturing, embodied by the term of Industry 4.0, the shift towards modular and dynamically adaptable production systems is inevitable. These systems are engineered to swiftly respond to evolving market demands while accommodating diverse product specifications. As production systems embrace

* Corresponding author. Tel.: +49-631-205757055
*E-mail address:* benjamin.blumhofer@dfki.de


10.1016/j.procs.2025.01.298

increased modularity and flexibility, the parallel evolution of intralogistics systems becomes not just advantageous, yet imperative.

The term ”Intralogistics” was officially introduced by the German Mechanical Engineering Industry Association (VDMA) during CEMAT 2005, announced on June 30, 2003. It was defined as ”the organization, control, execution, and optimization of the internal material and information flow as well as the goods handling in industry, trade, and public institutions” [1].

While the overarching definition of intralogistics includes several phases, like planning, execution and optimization, the focus of this publication is on the initial phases of planning and the interfaces to execution, rather than on the execution process itself. Within the planning phase, three types of sequences are created with an extending level of detail [2]. These phases can be further subdivided into sequences, with the production planning phase comprising technology, process and manufacturing sequences [2].

The *technology sequence* represents a machine-independent sequence of required technologies to manufacture a product. It defines the fundamental steps necessary, regardless of the specific machine configuration. The *Process Sequence* extends the *technology sequence* by providing additional information about the machines on which these steps are executed. This incorporates specific machine settings and parameters of the manufacturing process. The *Manufacturing Sequence* complements the previous sequences by encompassing all non-value-added processes necessary for the production of the end product. These may include logistics and material handling activities essential to manufacturing but not directly contributing to value creation.[2]

Historically, intralogistics planning has operated in isolation from production planning, resulting in inefficiencies and underutilization of resources. A significant contributing factor to this disconnection is the absence of a unified descriptive framework that seamlessly integrates intralogistics within the entire production planning process. This disjointedness presents a challenge in the Industry 4.0 era, where seamless interoperability and holistic optimization are paramount [3].

In the current industrial environment, reliance on a uniform fleet of transport systems, often limited to transport systems from specific manufacturers, poses a barrier towards flexibility and integration. Integrating new fleet members into such systems is a laborious and costly endeavor, particularly in the context of smaller batch sizes and diversified product portfolios. Static modeling approaches are not well suited due to the dynamic requirements of individualized products and fluctuating production demands. Addressing these challenges and enhancing production flexibility necessitate innovative solutions.

The terms used in this paper are based on the Capability-Skill-Service (CSS) reference model, which is under development in a working group of the platform industry 4.0. The CSS model can be implemented in the Asset Administration Shell (AAS), Web Ontology Language (OWL) or other technologies. The CSS model has been published and used in [4]. It extends the established Product-Process-Resource (PPR) model [5] by including the concept of *function* in the context of production. A *function* is formalized by adding three further model elements at three different levels of abstraction. This new approach can be transferred to the field of intralogistics and help to fill specific gaps. For example, [6] mentions the possibility of finding alternative production processes by separately comparing required and offered functionalities. By mapping the functionalities of intralogistics alongside those of the production facilities, the possible solution space becomes visible, which can lead to a holistic optimization of the production process.

The CSS model emerges as a promising framework for meeting the demands in intralogistics. Originally developed for production modeling, the CSS model exhibits potential applicability in intralogistics [7]. Leveraging the CSS model, factories can streamline the integration of diverse fleet members from various manufacturers and facilitate the joint planning of intralogistics and production tasks, as long as all fleet members utilize the shared information model.

This paper advocates for an architectural framework outlining the core functionalities of an intralogistics systems based on the CSS model. The authors explain the necessary information models for the core functionalities and their interaction. They also explore various implementations of services, capabilities and, skills. In addition to that, this publication offers insights into extending the architecture to meet special requirements while ensuring interoperability with the CSS model. Additionally, the proposed architecture addresses the need for centralized access to complex and decentralized information, such as overall inventory management.

The contribution of this paper revolves around addressing the following research questions:

1. Identification of required capabilities, services, and skills: What are the fundamental capabilities, services, and skills necessary for effective intralogistics system operation in the context of Industry 4.0?
2. Integration of Intralogistics Task Planning within Production Planning: How can intralogistics task planning be seamlessly integrated with production planning using the CSS model, ensuring efficient resource utilization and adaptability?
3. Interoperability and Information Exchange: What information is essential for intralogistics systems, and how can the modeled components interact effectively with each other using technologies like AAS and OPC Unified Architecture (OPC UA)?

By using existing standards and integrating findings from research publications, this paper aims to guide the development of intralogistics systems towards a better integration of production and logistics to achieve greater adaptability, and efficiency within the Industry 4.0 landscape. This publication is structured as follows: In section 2, the state of the art in the development of the CSS model and related work in the field of intralogistics are presented. This is followed by the concept for integrating intralogistics into a production system using the CSS model. In section 4, the implementation of the proposed concept is demonstrated. The paper concludes with a summary of the results and an outlook on future work..

## 2. Related Work

In this section, the authors present the current state of research. The first subsection deals with the CSS model and its applications. This is followed by a subsection generally describing the AAS. The third subsection examines the topic of intralogistics in research and the perspective of its applicability to capability modeling. The subsection ends with the definitions for important terms used in section 3. An extensive literature review on the advancements the css model can offer in the field of intralogistics, is done in [7].

### *2.1. CSS-Model*

The terms capability, skill, and service follow the definitions from [8] and the extensions from [6] and [9]. A capability is therefore described as *'an implementation-independent specification of a function in industrial production to achieve an effect in the physical or virtual world'*. A skill is defined as *'an executable implementation of an encapsulated (automation) function specified by a capability'*. A skill implements the capability on a machine-specific and technology-specific basis, whereby a capability may comprise several skills. Unlike a skill, a capability is implementation-independent. A service is defined as *'a description of the commercial aspects and means of provision of offered capabilities'*. Unlike capabilities, services contain commercial data and support the marketplace concept. Offers and requests can therefore be exchanged at the service level. This differentiation forms the basis for the comparison and matching of services and capabilities to identify suitable resources. Implementations like [10] include the description of required services and capabilities in the product AAS. The resources include offered capabilities. The features described in the product can be derived to required capabilities, which can be matched during the production planning phase. The updated Version of the CSS model from 2024 adds more functionality, especially to realize constraints to the model. This enables the user of the AAS submodel to define dependencies on external information, states, or components, e.g., certain parts.

### *2.2. Asset Administration Shell*

In addition to the CSS-model, there are overall 85 published submodel templates for the AAS by the Industrial Digital Twin Association e.V. (IDTA) which can be used to model information in capability-based production and intralogistics systems. [11] demonstrates an approach for utilizing AASs in the combination with capability-based engineering for flexible production lines and was extended in [12]. The authors use the process and product AASs to match the required and offered capabilities. They developed a modelling tool to generate the AAS, including the capabilities. The modelling process is further investigated by [13]. [14] provides a mapping of digital twin models and AAS submodels. They further propose an Industrial Internet of Things (IIoT) architecture for communication of the physical asset and the digital twin using the OPC UA.

### *2.3. Applications of AAS and CSS-model in intralogistics*

[15] presents an AAS-based service infrastructure for Industrie 4.0 testbeds, enabling a standardized digital representation of production assets and resources. The proposed system was applied to use cases in adaptive production planning and functional safety, leveraging a microservices' architecture with centralized components for AAS management and process control. The containerized implementation is available as open-source, promoting interoperability and integration for production environments. [15] focuses on modelling the layout and the control component. They also build up on the capability submodel of the AAS. The authors utilize VDA 5050 for the communication with the Automated Guided Vehicle (AGV). Their approaches are advanced in [16], extending the infrastructure to human centered online planning for intralogistics and manufacturing processes. [17] proposed an implementation of skill-based flexible transport systems using OPC UA which is implemented on a rail-mounted transport system but can be adapted also to AGVs and Autonomous Mobile Robot (AMR). This model is extended by [18] to be compatible with the Planning Domain Definition Language (PDDL). [19], [20] introduce necessary components for intralogistics 4.0 systems. Extending this results, [21] implements intralogistics use cases utilizing AASs without taking the CSS model into consideration. Assuming a joint production planning is implemented, [22] shows an approach on how to handle joint scheduling from production and intralogistics. Infrastructure objectives like fleet sizing and a decentralized environment are explained in [23]. As the approaches [23] and [22] have already provided possible solutions that can be readily adapted for the proposed architecture, these topics are not further explored in this publication. In addition, [24] shows how an independent AAS can be used to handle hazardous goods individually. Especially in the age of electrification and the great leaps in energy storage technologies, such as the development of batteries, the handling of dangerous goods is becoming increasingly important.

## 3. Concept

In this section, we present our conceptual framework for integrating intralogistics into a CSS based factory architecture. This integration aims to facilitate the joint planning and execution of production and intralogistics capabilities, thereby enhancing performance and flexibility [7]. To provide an overview of the concept, Figure 1 illustrates the entire pipeline for generating an executable manufacturing sequence based on product features. The pipeline typically commences with a given product, including a product description, which may be a CAD model. From this, product features, such as geometric features in a milling process, can be extracted to create semantically described features or required capabilities [25]. These required capabilities, constrained by specific parameters, form the basis for generating a capability sequence, ultimately resulting in a technology sequence. Subsequently, the capability matching process assigns each capability to an appropriate Cyber-Physical Production Module (CPPM) by comparing the required capabilities with the available capabilities of the CPPMs within the factory.

Following the capability matching for all required value-adding capabilities, the resulting sequence is termed the process sequence.

Our proposed approach incorporates intralogistics processes into this sequence, using the capability matching and resulting in the manufacturing sequence. This sequence delineates which machine executes each capability, with references for each capability to the machine-specific skills required for its realization.

In the capability matching process, a notable distinction exists between capabilities utilized in production and those in intralogistics. While the required capabilities and their parameters can be directly derived from the product feature description in production, this is not applicable to intralogistics as the required transports are a consequence of the chosen machines for the value-adding processes. Moreover, intralogistics includes processes that are not directly linked to products. We present a conceptual framework in this chapter detailing how product-linked capabilities can be generated and automatically parameterized. Additionally, we delineate the non-product-related services and capabilities required, alongside digital thread services, which make decentralized information centrally accessible.

### *3.1. Architectural concept of the CSS-Model*

Building upon the core functionalities of an intralogistics system outlined in section 2, we integrate them into the architectural framework depicted in Figure 2 and Figure 3. In this concept, the components interact distinctively with

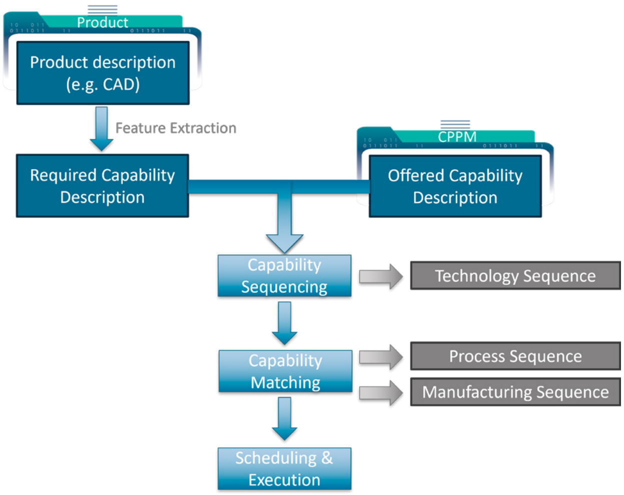


Fig. 1. Simplified planning and execution pipeline based on capabilities

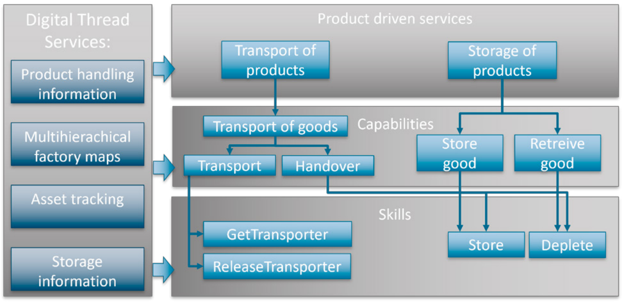


Fig. 2. Application of the CSS model to product driven core functionalities

each other, serving as templates for additional services. Product-driven services are directly linked to manufactured products, whereas factory-driven services are not. This distinction is significant for cost calculation purposes, as costs from product-driven services are directly associated with the product, whereas factory-driven costs are indirect and must be distributed across all products. The arrows between services, capabilities, and skills show which elements are used to realized the element above.

The core services in this concept encompass the transport and storage of products, while factory-internal services include storage balancing, load carrier management, and battery charging. The storage balancing service manages minimum and maximum inventory levels across storage units, intelligently distributing goods based on the order situation. Load carrier management facilitates the management of load carriers during production, necessitated by repackaging requirements. The battery charging service, primarily designed for AGVs or AMRs, schedules charging to maintain the requisite number of fleet members available at all times. These services exemplify the various interactions within the CSS model, contributing to enhanced monitoring of costs and streamlined operations.

These services are founded upon four major capabilities: transport of goods, storage, and depletion of goods, and charging. The transport capability encompasses transport and handover situations, allowing compositions of multiple transports and handovers. These capabilities are realized through four skills: *GetTransporter*, *ReleaseTransporter*, *Store* and *Deplete*, tailored to the unique requirements of each transport system. In addition to these, three skills cater to other capabilities, such as *PackIntoCarrier* and *UnpackFromCarrier* for *ChangeLoadcarrier*, and Charge for *BatteryCharging*. These skills encapsulate multiple functions, often composed of system-specific atomic skills.

To fulfill intralogistics tasks, supplementary information beyond capability descriptions is required, as elaborated in subsection 3.3.

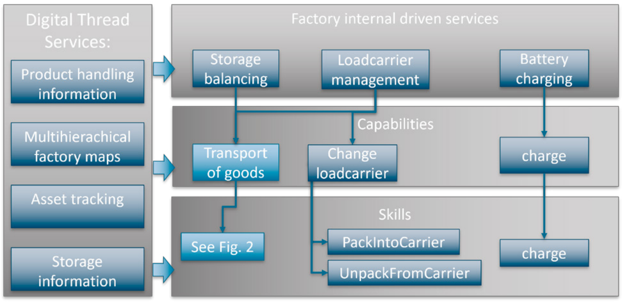


Fig. 3. Application of the CSS model to factory internal driven core functionalities

### 3.2. Capability-Based Planning

This section delineates a methodology for generating and dynamically parameterizing transport tasks based on capabilities. Illustrated in Figure 4, the approach contains the creation of a process sequence derived from the product's required capabilities, incorporating information on all value-adding processes, such as Manual Assembly.

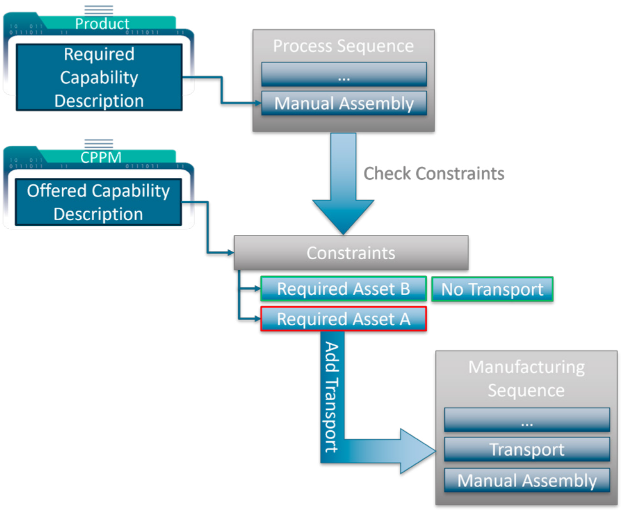


Fig. 4. Generation of Required Transport capabilities

Each CPPM has its own offered capability description including constraints for those capabilities, which are checked during the production planning. To be able to successfully use a capability, the constraints must be fulfilled. These constraints can be, e.g., two Assets which are required for the assembly. As CPPMs can contain an own buffer stock, assets like Asset B in the example can be available, so no transport is needed. In this scenario, Asset A is not in stock on the production module, which leads to the generation of a required transport in the manufacturing sequence. The generated transport-capability only contains the information on the destination and the identity of the product. To be able to assign the transport capability to one or multiple transport systems, more information is required.

This information can be taken from the AAS of the product, the CPPM, and the digital thread services as shown in Figure 5. Often, transport tasks require multiple transport systems and handover stations to get to their destination. To decide which stations and transport systems are used, this approach utilizes a combination of capability matchmaking and capability decomposition. Initially, the system tries to find one transport system which can fulfill the constraints of the required transport capability. If this is not possible, the transport is decomposed into multiple transport tasks, which can fulfill the constraints of the required capability and realize parts of the entire transport route. the matchmaking tries to find a suitable transport system or handover station per task. This process continues until the entire transport task is matched with at least one Cyber-physical Transportation Module (CPTM) or if no further decomposition of

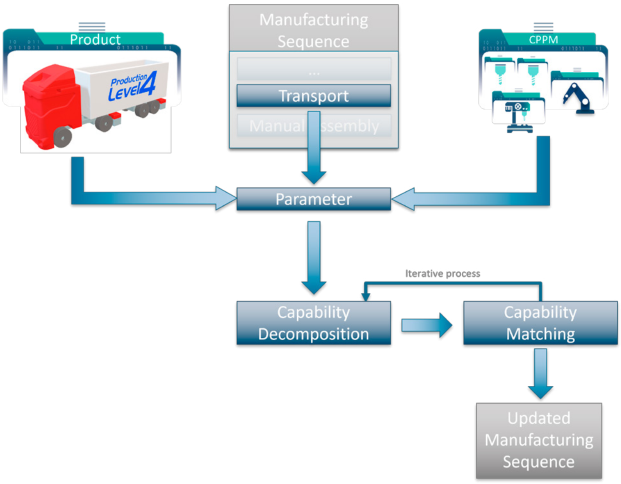


Fig. 5. Parameterization of transport capability

non-matched tasks is possible. The results are added to the manufacturing sequence, which can be executed by the scheduling and execution system. The factory internal-driven storage balancing service can utilize the same planning pipeline as the product-driven transport because of the usage of the same capabilities. This scenario only differs that instead of the required capability generation based on the product, the storage balancing creates the same yet minimal parameterized (only goal station and product ID) transport capability and afterward utilizes the same parameterization and matching pipeline as the transport product service.

### 3.3. Digital Thread Services

In addition to the information models of the assets, Cyber-Physical Systems (CPS) and products, the intralogistics system requires extended access to general information to effectively integrate and optimize the proposed architecture in the dynamic Industry 4.0 environment. The particular challenge in a flexible production environment characterized by modularization, which can be potentially reconfigured several times a day, lies in the availability and centralization of actual information. This information, which is spread across various systems and modules, must be carefully aggregated, processed and be made available in various levels of detail to meet the different requirements of the production processes. Key information that is essential for optimized production planning and control includes, for example, multi-hierarchical maps that not only support navigation and route planning within the production facility, but also provide real-time information about obstacles and the condition of the production environment. Accurate asset tracking is just as critical as providing real-time data on the location and condition of production equipment and goods to efficiently manage material flows and respond to unforeseen events. Warehouse information plays another key role by providing detailed data on inventory levels and conditions, enabling effective inventory management and cross-module optimization of warehouse resources. A central source of information that acts as a data hub is essential for this. This source continuously accumulates data from different areas of the production environment, prepares it for use, and makes it accessible to users and automated systems. Another important part of this information network is the collection of specific product handling information, which includes special requirements for the handling or storage of products, such as maximum acceleration for fragile items. Such fine-grained information is critical to ensure that products can be moved effectively and safely through the production and storage processes.

### 3.4. Cyber-physical components

The VDI 2510 defines several types of materials transport devices [26]. These can be divided into continuous conveyors and discontinuous material transport systems. This work focuses on the latter, as no conveyor systems were accessible for the evaluation. This concept is still applicable to these systems. The only modifications needed would be in the implementation on the skill level for the transport capabilities. The detailed model of the used skills for transport systems is described in [18]. This model is applicable to operator-controlled systems like a conventional forklift as well, as AMRs , only the skill implementation differs, but the interface to the skill remains the same.

The paper currently outlines that AGVs and AMRs can be categorized based on two distinctions: load-pulling versus load-carrying capabilities, and active versus passive load handling systems. Whether the transport system pulls or carries the product only influences the capability description itself, the differentiation of active and passive load handling influences which capabilities can be offered from the transport system. Active load handling systems result in the transport system having a handover capability and store and deplete skills. Passive transport systems only have the transport capability and *GetTransporter* and *ReleaseTransporter* skill. Equivalently, each storage equipped with a load handling system has in addition to the *storing goods* and *retrieving goods* capabilities, the *handover* capability and the according skills. It is also possible to have a storage without the *handover* capability and the matching skills, which requires the transport system to have this functionality. In this case, neither the storage nor the transport system has this capability, a third specialized handling system must be involved. In many production environments, products are not directly handled but are instead affixed to standardized load carriers like pallets or standardized load carriers (SLCs). These carriers themselves lack capabilities, but significantly influence the execution of capabilities. Additionally, there are components like doors that require control but do not interact with the product directly. These components can be integrated into the system by implementing a skill Interface, such as an *open-door* skill.

## 4. Implementation

In this section, we present a comprehensive implementation of a transport system featuring a robotic arm assembly. Our objective is to demonstrate the integration of transport and handover capabilities along with their corresponding skills within a cyber-physical environment. The implementation encompasses the entire spectrum of AASs, like key components such as storage units, charging stations, and packaging facilities. This holistic approach ensures the depiction of a realistic operational scenario within an Industry 4.0 framework. For detailed access to the implemented system, the complete set of AASs, is accessible through the provided link to our external repository [27]. As shown in Figure 6, there are restrictions on what carriers are acceptable and what dimensions the product can have. Based on e.g., [26, 28] further properties of the capability can be modelled to improve precision of automated matching systems. An expanded version of the capability descriptions is also available at [27] as it would exceed the spacial limitations of this publication. In the property containers are valid ranges modelled using the *SubmodelElement Range*. Depending on the type of property, it can be either *Lists*, *Properties*, or *Ranges*. Within the *Submodel*, the *SubmodelElementCollection RealizedBy* refers to the skills *GetTransporter* and *ReleaseTransporter*. A further explanation of the constraints of the model is proposed in [18]. The skill submodel of the AAS is shown in Figure 7. Besides the address and description of how to access the skill, there is a description of the required input and result parameters. All skills are implemented in OPC UA following [6]. The parameters of the *GetTransporter* are needed to know where to move, or from where to store or deplete the products in the *Store* and *Deplete* skills. More precise information can be requested from the Digital Thread Services during skill execution. All skills, except the *ReleaseTransporter* skill, contain information about elapsed time, costs, energy consumption, and a list of occurred alarms as result information. The list of alarms can be used for fault detection and maintenance. Pre- and postconditions of the skills are discussed in [18]. The Concept of Precondition- and FeasibilityCheck are adapted and used from [29]. How the skills, proposed in this paper, are implemented on the machines varies. However, in most of the scenarios, this skill composes multiple atomic skills. Depending on the route the transport system takes, the *GetTransporter* skill can also call external skills to open a door to access a closed area, which increases the autonomy of the transport system itself.

### 4.1. Digital Thread Services

Additional information is required to carry out the tasks and the associated skills. This can be requested via a corresponding Digital Thread Service, which can provide the information centrally and in various levels of detail for the respective task. These services have standardized interfaces so that the information can be accessed by autonomous systems and machines. By setting up descriptive AAS for each service, which contains detailed interfaces and descriptions, a standardized interface is created that supports such interaction. These AAS are designed to provide not


Fig. 6. Capability Model of a AMR with active load handling


Fig. 7. Skill Submodel of a AMR with active load handling

only static information about the assets, but also dynamic data such as current positions and states, which are essential for navigation and process execution. Figure 8 shows an example of a Digital Thread Service AAS using the

Multihierarchical Map Service. A practical example of this is the navigation task of a AMR, which was requested by the *GetTransporter* skill and must perform efficient path planning. This task requires information about the environment. The robot can use its skills to call a service that grants access to the grid maps. These grid maps, which are provided via ROS2 topics, represent the factory layout as a dynamically updated map. This provides the robot with information about obstacles and other relevant aspects of the layout at the level of detail required for its current task. In addition to the navigation services, the central factory topology map, modeled as a comprehensive graph, enables detailed planning and execution of intralogistics tasks. skills can call up this service to obtain a complete overview of the production site and plan optimized routes and resource allocations based on this. The advanced services that go beyond the basic functions, such as real-time inventory monitoring and machine condition monitoring, can also be called up as a Digital Thread Service. These ensure that the intralogistics system can react proactively to changes by accessing real-time data provided via the Digital Thread Services.

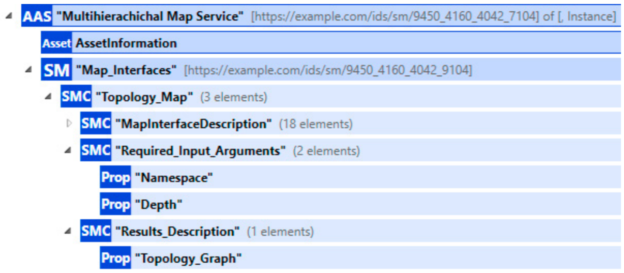


Fig. 8. AAS implementation of the Multihierachical Map Service as a sample of a Digital Thread Service

## 5. Conclusion & Outlook

In this paper, we have demonstrated the applicability of the Capability-Skill-Service (CSS) model to intralogistics systems, aiming to enhance flexibility and facilitate collaborative planning between production and intralogistics domains.

Our contributions lie in identifying the essential capabilities, services, and skills for intralogistics systems based on industry standards. Additionally, we have proposed a conceptual framework for planning and executing intralogistics tasks using the CSS model, which includes the dynamic generation and parameterization of transport capabilities. Moreover, we have discussed the vital role of Digital Thread Services in enabling the proposed architectural concept, alongside providing an implementation example featuring an autonomous mobile robot with active load handling capabilities and corresponding skill descriptions.

While the implementation example presented demonstrates the feasibility of applying the CSS model to intralogistics, it is currently limited to addressing simpler use cases. More complex scenarios and advanced functionalities have yet to be fully explored. The framework presented is limited to production system utilizing the CSS model.

By addressing the research questions outlined in the introduction, our approach sets a solid foundation for future endeavors. Moving forward, we intend to delve deeper into the detailed modeling of skills and refining the matching process. Furthermore, we aim to elaborate on the Digital Thread Services required to support the proposed architecture fully. Lastly, we plan to evaluate the overall concept through real-world use cases, thereby advancing the development of flexible and adaptive intralogistics systems, often referred to as Intralogistics 4.0, within the broader context of Industry 4.0.

## Acknowledgements

This work was sponsored/funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) in the context of the project ”Digitaler Zwilling und Künstliche Intelligenz in der vernetzten Fabrik für die integrierte Nutzfahrzeugproduktion, Logistik und Qualitätssicherung (TWIN4TRUCKS, 13IK010F). The authors are grateful for this support.

## References


[1] D. Arnold (Ed.), Intralogistik: Potentiale, Perspektiven, Prognosen, VDI-Buch, Springer, Berlin Heidelberg, 2006.

[2] G. Trommer, Methodik zur konstruktionsbegleitenden Generierung und Bewertung alternativer Fertigungsfolgen, no. 2001,15 in Berichte aus der Produktionstechnik, Shaker, Aachen, 2001.

[3] H. Kagermann, W. Wahlster, Ten Years of Industrie 4.0, Sci 4 (3) (2022) 26. doi:10.3390/sci4030026.

[4] A. Köcher, A. Belyaev, J. Hermann, J. Bock, K. Meixner, M. Volkmann, M. Winter, P. Zimmermann, S. Grimm, C. Diedrich, A reference model for common understanding of capabilities and skills in manufacturing, at - Automatisierungstechnik 71 (2) (2023) 94–104. doi:10.1515/auto-2022-0117.

[5] M. Schleipen, A. Lüder, O. Sauer, H. Flatt, J. Jasperneite, Requirements and concept for Plug-and-Work: Adaptivity in the context of Industry 4.0, at - Automatisierungstechnik 63 (10) (2015) 801–820. doi:10.1515/auto-2015-0015.

[6] C. Diedrich, A. Belyaev, R. Blumenfeld, Juergen Bock, S. Grimm, J. Hermann, T. Klausmann, A. Köcher, M. Maurmaier, K. Meixner, J. Peschke, M. Schleipen, Siwara Schmitt, B. Schnebel, G. Stephan, M. Volkmann, A. Wannagat, K. Watson, M. Winter, P. Zimmermann, Information Model for Capabilities, Skills & Servicesdoi:10.13140/RG.2.2.30098.53440.

[7] B. Blumhofer, M. Simon, A. Ritter, L.-M. Weil, T. Legler, M. Ruskowski, Capability Skill Service Model as enabler for Intralogistics 4.0: A Review, presented at 2024 7th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS) (2024).

[8] A. Köcher, C. Hildebrandt, L. M. Vieira Da Silva, A. Fay, A Formal Capability and Skill Model for Use in Plug and Produce Scenarios, in: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Vienna, Austria, 2020, pp. 1663–1670. doi:10.1109/ETFA46521.2020.9211874.

[9] A. Köcher, A. Belyaev, J. Hermann, J. Bock, K. Meixner, M. Volkmann, M. Winter, P. Zimmermann, S. Grimm, C. Diedrich, A Reference Model for Common Understanding of Capabilities and Skills in Manufacturingdoi:10.48550/ARXIV.2209.09632.

[10] M. Simon, S. Jungbluth, A. Farrukh, M. Volkmann, J. Hermann, M. Ruskowski, Implementation of Asset Administration Shells in a Shared Production Scenario with Gaia-X, in: 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Sinaia, Romania, 2023, pp. 1–8. doi:10.1109/ETFA54631.2023.10275583.

[11] Y. Huang, S. Dhouib, J. Malenfant, An AAS Modeling Tool for Capability-Based Engineering of Flexible Production Lines, in: IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society, IEEE, Toronto, ON, Canada, 2021, pp. 1–6. doi:10.1109/IECON48115.2021.9589329.

[12] S. Dhouib, Y. Huang, A. Smaoui, T. Bhanja, V. Gezer, Papyrus4Manufacturing: A Model-Based Systems Engineering approach to AAS Digital Twins, in: 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Sinaia, Romania, 2023, pp. 1–8. doi:10.1109/ETFA54631.2023.10275523.

[13] C. Bouter, M. Pourjafarian, L. Simar, R. Wilterdink, Towards a Comprehensive Methodology for Modelling Submodels in the Industry 4.0 Asset Administration Shell, in: 2021 IEEE 23rd Conference on Business Informatics (CBI), IEEE, Bolzano, Italy, 2021, pp. 10–19. doi:10.1109/CBI52690.2021.10050.

[14] X. Ye, W. Xu, J. Liu, Y. Zhong, Q. Liu, Z. Zhou, W. S. Song, S. H. Hong, Implementing Digital Twin and Asset Administration Shell Models for a Simulated Sorting Production System, IFAC-PapersOnLine 56 (2) (2023) 11880–11887. doi:10.1016/j.ifacol.2023.10.600.

[15] A. Luxenburger, D. Porta, S. Knoch, J. Mohr, T. Schwartz, 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), IEEE, Sinaia, Romania, 2023, pp. 1–8. doi:10.1109/ETFA54631.2023.10275335.

[16] A. Luxenburger, J. Mohr, D. Merkel, S. Knoch, D. Porta, C. Paul, J. Widenka, P. Schäfers, M. Baumann, S. Lehnhoff, J. Schwab, Interactive Digital Twins for Online Planning and Worker Safety in Intralogistics and Production, in: 2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR), IEEE, Los Angeles, CA, USA, 2024, pp. 66–74. doi:10.1109/AIxVR59861.2024.00016.

[17] S. Jungbluth, T. Barth, J. Nußbaum, J. Hermann, M. Ruskowski, Developing a skill-based flexible transport system using OPC UA, At-Automatisierungstechnik 71 (2) (2023) 163–175. doi:10.1515/auto-2022-0115.

[18] A. T. Bernhard, B. Blumhofer, M. Ruskowski, A. Luxenburger, D. Porta, Leverage AAS to support condition-based planning languages, manuscript submitted for publication (2024).

[19] A. Bozkurt, M. Hagg, R. Schulz, Innovative logistics concepts for a versatile and flexible manufacturing of lot size onedoi:10.15488/9645.

[20] A. Bozkurt, T. Tasci, R. Schulz, A. Verl, Designing of Smart Logistics Modules as Cyber-physical systems for Load carriers, Volume 2021 (2021) Issue 17doi:10.2195/LJ_PROC_BOZKURT_EN_202112_01.

[21] A. Bozkurt, R. Schulz, Asset Administration Shell and Knowledge Graph-based Production Logistics Planning in Fluid Manufacturing Systems, Volume 2023 (2023) Issue 1doi:10.2195/LJ_PROC_BOZKURT_EN_202310_01.

[22] J. Popper, V. Yfantis, M. Ruskowski, Simultaneous Production and AGV Scheduling using Multi-Agent Deep Reinforcement Learning, Procedia CIRP 104 (2021) 1523–1528. doi:10.1016/j.procir.2021.11.257.

[23] J. Gödeke, P. Detzner, A Simulative Approach to AMR Fleet Sizing in Decentralized Multi-Robot Task Allocation, in: 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Sinaia, Romania, 2023, pp. 1–8. doi:10.1109/ETFA54631.2023.10275717.

[24] Technologie-Initiative SmartFactory KL e.V., Trustworthiness: Safety Meets Security in Industry 4.0 Ecosystems Whitepaper SF-3.5 (Apr. 2024).

[25] M. Volkmann, T. Legler, A. Wagner, M. Ruskowski, A CAD feature-based manufacturing approach with OPC UA skills, Procedia Manufacturing 51 (2020) 416–423. doi:10.1016/j.promfg.2020.10.059.

[26] VDI 2510, Fahrerlose Transportsysteme (FTS) (Oct. 2005).

[27] Technologie-Initiative SmartFactory KL e. V., Shell repository: http://shells.smartfactory.de.
[28] VDI 3589, Auswahlkriterien für die Beschaffung von Flurförderzeugen (2014).
[29] A. Wagner, S. Lamoth, M. Volkmann, J. Hermann, M. Ruskowski, Interaction between FeasibilityCheck, PreconditionCheck and SkillExecution in skill-based machining, in: 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Sinaia, Romania, 2023, pp. 1–8. doi:10.1109/ETFA54631.2023.10275520.