

A holistic agent-based approach: combining data architectures and job-shop scheduling

Marco Wölfel, Adrian Pfeiderer, Bernhard Bauer

Angaben zur Veröffentlichung / Publication details:

Wölfel, Marco, Adrian Pfeiderer, and Bernhard Bauer. 2026. "A holistic agent-based approach: combining data architectures and job-shop scheduling." In *Proceedings of the 28th International Conference on Enterprise Information Systems (ICEIS 2026)*, May 22-24, 2026, Benidorm, Spain; volume 2, edited by Joaquim Filipe, Michał Śmiałek, Alexander Brodsky, and Slimane Hammoudi, 1280–91. Setúbal: SciTePress. <https://doi.org/10.5220/0015050200004018>.

Nutzungsbedingungen / Terms of use:

CC BY-NC-ND 4.0



A Holistic Agent-Based Approach: Combining Data Architectures and Job-Shop Scheduling

Marco Wölfel, Adrian Pfeiderer and Bernhard Bauer

Software Methodologies for Distributed Systems, University of Augsburg, Universitätsstraße 6a, Augsburg, Germany

Keywords: Data Architecture, Job-Shop Scheduling, Predictive Analytics, Agents, Digital Twin, Unified Namespace, Manufacturing, Closed-Loop Scheduling Feedback, Resilience, Domain Independence.

Abstract: Manufacturing systems must respond to volatile demand, heterogeneous data, and frequent disruptions while maintaining quality and throughput. This paper presents an agent-based framework that connects predictive analytics, job-shop scheduling, and data collection through a digital twin data architecture. At the core of the data architecture is a unified namespace structured according to the proposed data model, enabling continuous information exchange between shop floor execution, predictive analytics, and scheduling. The approach models machines, human operators, and software services as agents, with a central data access broker supporting the closed-loop feedback between execution data and scheduling decisions. This feedback loop allows historical and real-time process data to inform future scheduling decisions, parameter adaptation, and resilience strategies. The framework is designed for extensibility, domain independence, and variable modeling granularity, supporting both long-term planning and detailed execution control. A scenario-based evaluation in a collaborative manufacturing use case demonstrates how the architecture integrates operational technology, information technology, and predictive services to support adaptive, self-correcting, and resilient production.

1 INTRODUCTION

Manufacturing is undergoing a fundamental transformation driven by increasing market volatility, cost pressure, and the demand for high-quality, just-in-time production under ever shorter lead times (Masudin and Kamara, 2018; Babai et al., 2024). At the same time, production is shifting from rigid mass manufacturing toward highly flexible environments capable of supporting customization down to lot size one. In such settings, competitiveness depends not only on faster execution but also on the ability of production systems to adapt intelligently to changing conditions.

To support this transformation, Artificial Intelligence (AI) and data-driven methods have become integral to modern manufacturing (Babai et al., 2024; Ding et al., 2023). Predictive analytics play a central role in this development, particularly in closed-loop production settings supported by Digital Twins (DTs) (Besigomwe, 2025; Singh et al., 2026). Applications such as predictive maintenance (Rai et al., 2021; Tsallis et al., 2025), demand forecasting (Douaioui et al., 2024; Feizabadi, 2022), and automated quality control (Rai et al., 2021) are increasingly used

to anticipate system states and ensure product standards. These capabilities, however, require data architectures that can persist and contextualize heterogeneous process data and provide the basis for a DT supporting monitoring, simulation, and optimization (Singh et al., 2026; Besigomwe, 2025).

In parallel, operational planning in such environments relies on Job-Shop Scheduling (JSS). Yet, schedules derived in isolation often diverge from shop-floor reality once disruptions such as machine degradation, rework, or changing process conditions occur (Destouet et al., 2025; Wang et al., 2019). Moreover, manufacturing steps are frequently interdependent: observations or parameter settings in one step may constrain scheduling decisions and machine configurations in subsequent steps. As a result, scheduling and predictive analytics should not be treated as separate concerns, but as tightly coupled elements of adaptive production control.

This paper therefore addresses the following overarching question: How can predictive analytics, heterogeneous process data, and operational JSS be integrated in a unified architecture to support resilient production? To structure the investigation, we refine this question into three research questions:

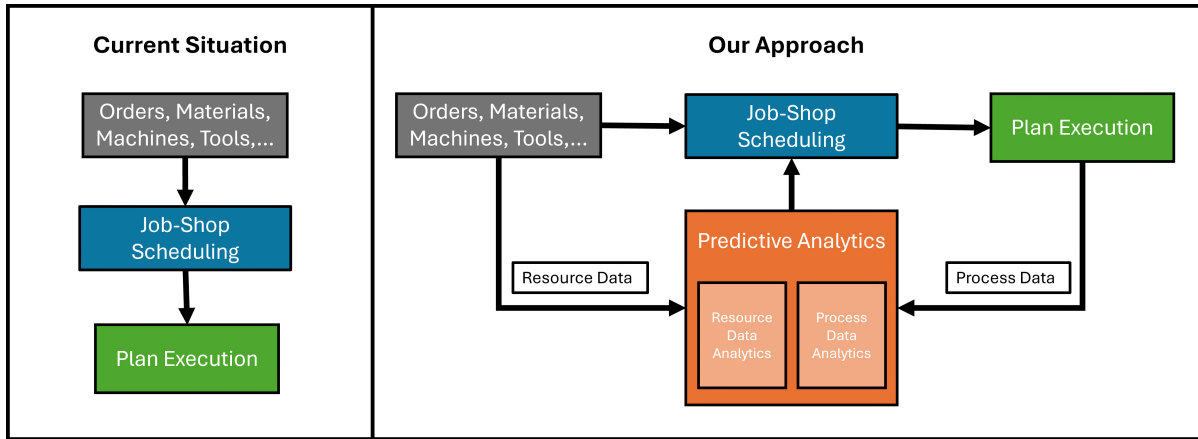


Figure 1: An overview of how the current situation compares to our approach.

RQ1) What constitutes an effective agent-based data architecture for persisting and processing heterogeneous process data within a DT? **RQ2)** How can dependencies between manufacturing steps be modeled and resolved in an agent-based system to enable efficient JSS? **RQ3)** How can predictive analytics, such as predictive maintenance and demand forecasting, improve the optimization potential of production schedules?

To answer these questions, we propose a holistic agent-based approach that models machines, transport systems, human operators, and software services as agents within a shared DT. The approach combines a heterogeneous graph-based data model, a broker-mediated system architecture, predictive analytics, and JSS to establish a closed feedback loop between shop-floor execution and future scheduling decisions. In contrast to static planning, this enables schedules to be continuously informed by operational and predictive data, thereby improving adaptability and resilience.

Figure 1 summarizes this shift from a linear scheduling workflow toward a closed-loop architecture in which execution data, predictive analytics, and scheduling continuously influence one another.

2 BACKGROUND AND RELATED WORK

The paradigm of Industry 4.0 has transformed manufacturing from isolated production systems into interconnected and data-driven environments. Originally framed around the integration of Cyber-Physical Systems (CPSs), Industry 4.0 emphasizes interconnection, information transparency, and increasingly autonomous decision support in smart factories (Kager-

mann et al., 2013; Lasi et al., 2014; Hermann et al., 2016). Recent reviews further stress that smart manufacturing is not only about automation, but about flexible, adaptive production supported by real-time data and predictive analytics (Deokar et al., 2025).

The technical realization of such environments is closely linked to CPSs, which integrate computational and physical processes and enable real-time monitoring and control across industrial value chains (Oks et al., 2024). In the context of mass personalization, CPSs also provide the basis for self-organizing manufacturing networks that can adapt to changing demands and production conditions (Qin and Lu, 2021). At the same time, the growing number of heterogeneous data sources creates a strong need for shared information-system architectures. To avoid data silos and support interoperability, architectures based on the Unified Namespace (UNS) have been proposed as integration layers for shop-floor data (Salcher et al., 2024; Díaz et al., 2025). Building on this foundation, DTs enable closed-loop manufacturing in which operational data, predictions, and optimization results can directly inform physical production (Besigomwe, 2025; Singh et al., 2026).

Implementing such systems also raises engineering and infrastructure challenges. Data-centric AI projects must cope with evolving datasets, dependencies between code and data, and the traceability of intermediate artifacts (Stieler and Bauer, 2024). In addition, computationally demanding services such as dynamic scheduling and analytics benefit from dedicated platform architectures that support the broader AI lifecycle in industrial settings (Weigell et al., 2025).

One important challenge for learning-based manufacturing services, including predictive maintenance and quality-related analytics, is limited availability of rare failure or anomaly data. Synthetic-data meth-

ods can therefore complement real production data and improve the robustness of predictive models (Mei et al., 2025; Buggineni et al., 2024). This can also support smart-factory improvement through controlled experimentation and hybrid training strategies for data-scarce scenarios (Hummel and Schuhmacher, 2023; Pfeleiderer and Bauer, 2025).

At the operational level, adaptive production requires JSS methods that can react to disruptions and changing shop-floor conditions. In dynamic manufacturing environments, scheduling approaches that optimize only once before execution begins are often insufficient because they cannot adequately respond to evolving shop-floor states (Qin and Lu, 2021; Ding et al., 2023). While the Flexible Job-Shop Scheduling Problem (FJSP) has been studied extensively with respect to constraints and objectives (Dauzère-Pérès et al., 2024), recent work increasingly relies on Graph Neural Networks (GNNs) and Deep Reinforcement Learning (DRL) to represent complex relationships between jobs, operations, and machines (Song et al., 2023; Wölfel and Bauer, 2026). Related work on the Continuous Dynamic Flexible Job-Shop Scheduling Problem (C-DFJSP) further highlights the relevance of multi-agent approaches for handling unpredictable job arrivals and urgent orders over extended operating horizons (Johnson et al., 2024).

Finally, practical scheduling rarely optimizes only a single criterion. Dynamic shop-floor control must often balance throughput, robustness, energy, or cost, which motivates multi-objective optimization (Ullah et al., 2025; Hayes et al., 2022). In the context of Reinforcement Learning (RL), recent comparisons show the relevance of multi-objective learning strategies in complex decision problems (Kemper et al., 2025).

Taken together, prior work provides important foundations for UNS-based data integration, DT-based feedback, predictive services, and adaptive JSS. However, their joint integration into a shared, extensible, agent-based architecture for closed-loop production remains comparatively underexplored. This gap motivates the framework proposed in this paper.

3 APPROACH

This section presents the proposed framework from concept to implementation. It first introduces the underlying concept through the data model and system architecture, then summarizes the resulting contributions, and finally describes how the main architectural modules are realized in the prototype.

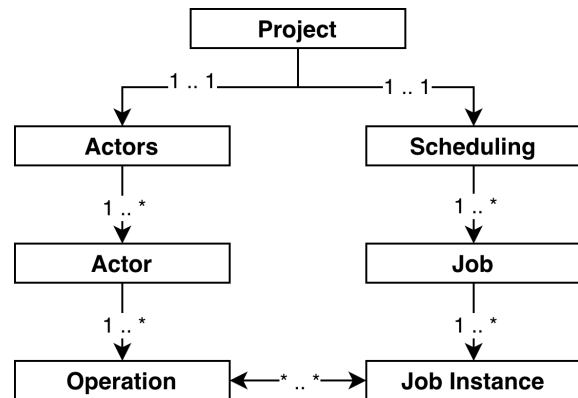


Figure 2: The data model enables feedback between collected actor data and scheduling decisions.

3.1 Concept

The core idea is to represent the data as a heterogeneous graph. The elements of this heterogeneous graph have key-value attributes and references to the collected data. The overall concept consists of a data model defined by a heterogeneous graph schema and a detailed architecture that covers the different components and agents. At its core, this architecture unifies the collection of process, predictive analytics, and scheduling data to establish a continuous closed-loop feedback mechanism. This bidirectional relationship arises because the scheduler allocates tasks to specific actors, thereby shaping the generation and collection of future shop-floor data. Conversely, the collected data is used to compute metrics that evaluate past scheduling performance and actor parameters. These metrics are then used to refine future scheduling decisions as well as actor parameters. While originally designed for traditional JSS, the concept has evolved to incorporate a broader range of actors such as software agents and human operators. In this work, an agent denotes any decision-making system that interacts with data or an environment through observations and actions, and may adapt its behavior based on feedback, experience, or an explicit objective (Sutton and Barto, 2020).

3.1.1 Data Model

The data model establishes the structural foundation for the scheduling feedback loop and bridges the gap between shop-floor data collection and algorithmic scheduling to create a seamless closed-loop system. At the highest level, the model is organized into two primary categories, namely Actors and Scheduling. The hierarchy is depicted in figure 2.

Actors. Actor nodes represent the functional entities of the shop floor, including physical actors such as machines and vehicles as well as software agents and human workers. These nodes store specific capability profiles, which allow the central scheduling agent to dynamically match available resources to required operations. They also include technical integration protocols that enable data collection for the actor in question. Operation nodes represent ongoing or completed work and serve as the primary repository for all data collected during a specific timeframe. To ensure traceability, these operation nodes are simultaneously linked to job instance nodes and explicitly map individual actions to broader work orders.

Scheduling. The scheduling category focuses on the logic of production, beginning with job nodes that function as abstract templates for workpieces. These nodes define the necessary workflows, their internal dependencies, and the required capabilities. Job instance nodes represent the concrete instantiation of these abstract jobs. By linking these instances back to the operational data captured by the actors, the architecture ensures that every action on the shop floor is accounted for within the global scheduling logic.

3.1.2 System Architecture

The system architecture defines how the production environment is coordinated as a closed loop. The overall structure is depicted in figure 3. It describes the responsibilities of the modules that work together through a central Data Access Broker.

Actors. Actors provide production data and carry out assigned work. They include human operators, physical machinery, and software agents. Each actor can be added independently, which keeps the system extensible without requiring a redesign of the overall control logic.

Planning and Scheduling. Planning and Scheduling agents set production objectives and translate them into executable plans. The module allocates work to available actors, reconciles priorities and constraints, and updates plans based on feedback. It uses insights from analytics and shop-floor observations to keep execution aligned with global goals.

Predictive Analytics. Predictive Analytics agents turn resource and process signals into decision support. They identify emerging risks and opportunities, propose adjustments to operating parameters, and recommend plan updates. These insights guide

the scheduler so that throughput and stability are preserved under changing conditions.

Data Access Broker. The Data Access Broker is the communication and data hub. It decouples producers and consumers, stores multimodal information, and provides uniform access. This hub closes the loop between planning and execution and enables decentralized growth because new actors and agents can join without changes to the core logic.

Control Dashboard. The Control Dashboard is the human interface to the system. It provides monitoring of states and the ability to issue high-level commands and configuration updates. All interactions flow through the broker so that decisions and data remain consistent across modules.

3.2 Contributions

This work contributes a heterogeneous graph-based data model and a broker-mediated architecture for closed-loop coordination between shop-floor execution, predictive analytics, and JSS. Its focus is not a fixed set of agents, but an architectural foundation for integrating heterogeneous actors within a shared data model, addressing the still limited joint integration of UNS, DTs, and JSS in prior work (Salcher et al., 2024; Besigomwe, 2025; Ding et al., 2023).

3.2.1 Core Contributions

Our approach addresses the described gap and provides the following core contributions.

Closed-Loop Scheduling Feedback. The main achievement of this architecture is the synergistic combination of data collection and production scheduling. In contrast to static planning systems, our approach naturally allows for the collection of real-time actor data from physical machines or software agents, which is then used to continuously refine and adapt the scheduling process. This is where the concepts introduced in section 2 coalesce into a functional unit. The UNS serves as a central paradigm that allows every entity in the organization to access the required data in real time. This directly addresses the first research question. JSS then uses this data to dynamically assign tasks to the actors. Furthermore, the DT acts as an abstract convergence layer that wraps and combines both the UNS and JSS. This fusion establishes a closed-loop feedback mechanism, enabling the system to use historical performance data,

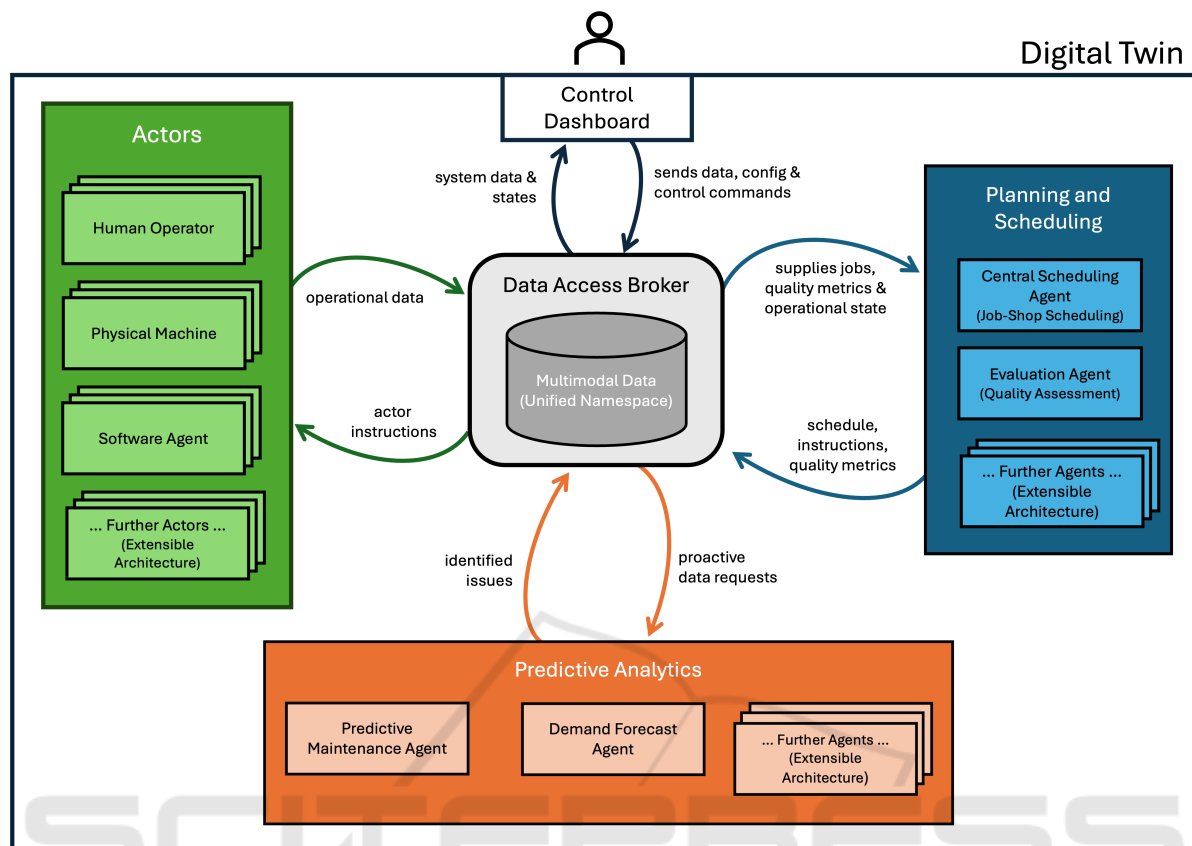


Figure 3: The system architecture shows how the data access broker enables various actors and agents to interact with and enforce the data model. It also emphasizes how humans can control the system through a dashboard.

material demand, and current state deviations to realize multi-objective optimization and make future schedules significantly more resilient.

Extensibility and Agent Integration. In order to cope with the volatility of modern manufacturing environments, the system architecture prioritizes extensibility to ensure future upgrade paths. The developed data model is highly flexible and supports seamless schema evolution, which means that future additions do not break current expectations. This inherent openness directly addresses the resolution of complex dependencies between manufacturing steps raised in the second research question. The architecture not only allows the system to continuously improve its core functions but also enables it to be extended flexibly for different purposes.

Resilience. Achieving operational resilience within this architecture depends on the ability to anticipate and mitigate disruptions before they impact production. By utilizing Predictive Analytics agents, the system can foresee potential setbacks such as material

shortages or excessive tool wear. These predictive insights are then relayed back to the JSS module, allowing the central scheduling agent to dynamically adjust plans to account for factors such as material delivery times or required tool changes. In this way, the architecture directly addresses the third research question by showing how predictive analytics can influence the optimization potential of the production schedule.

Beyond scheduling adjustments, the closed-loop system enables fine-grained parameter adaptation to safeguard the manufacturing process. This allows real-time modifications to specific process variables, such as pressure or temperature within infusion or injection molding stages, to compensate for identified irregularities. By integrating these predictive insights with automated feedback, the architecture ensures that production remains stable and efficient even in the face of unforeseen operational challenges.

3.2.2 Further Contributions

Beyond the main research questions, further requirements are considered to tackle the overall goal.

Domain Independence. The data model and system architecture are designed to handle multiple domains and combine Information Technology (IT) and Operational Technology (OT). By employing an abstracted metamodel, the system decouples core architectural logic from domain- or entity-specific constraints. This approach treats production hardware and software services as equivalent functional entities, enabling semantic interoperability without requiring structural re-engineering for new use cases.

Dynamic Granularity. To bridge the operational gap between long-term production planning and real-time JSS, the data architecture supports variable-resolution modeling. The proposed system enables dynamic abstraction and allows entities to be modeled at the fidelity required for the specific decision-making horizon. This multiscale approach ensures that computational resources are not wasted on processing high-resolution job details for planning tasks far in the future.

Usability. Driven by the requirements of distributed manufacturing environments, the system architecture prioritizes decentralized accessibility over centralized configuration. As the system may rely on input from both human operators and autonomous software agents, it implements a dual-interface strategy. A Graphical User Interface (GUI) enables shop-floor workers to intuitively instantiate and manage inputs without programming expertise, while a standardized Application Programming Interface (API) layer facilitates the integration of autonomous agents and automated scripts. One such autonomous agent is a scheduler that handles JSS.

3.3 Implementation

The following prototype instantiates the proposed data model and architecture in the selected use case and illustrates how different agents can be integrated through the same architectural interfaces. It consists of the Data Access Broker and API, plus the Control Dashboard, Actors, Predictive Analytics, and Planning and Scheduling modules.

Actors. Actors are represented in the data model and behave as agents that emit data and execute tasks. Data ingestion supports multiple integration patterns such as uploads through the GUI, automation scripts, Open Platform Communications Unified Architecture (OPC UA) subscriptions, and Representational State Transfer (REST) endpoint polling. In addition to

equipment and human operators, software agents can be scheduled to generate synthetic data, start new calculations, and trigger orders for materials and tools. All interactions with actors use the Data Access Broker as the mediation point for data and instructions.

Planning and Scheduling. The JSS module focuses on the data required to compute efficient sequences. The planning logic uses GNNs and DRL as described in (Wölfel and Bauer, 2026). The JSS agent publishes dispatching and sequencing results through the broker so that relevant actors can execute them. The Evaluation Agent for quality assessment was developed in earlier industrial projects. In this use case, a placeholder component is connected to validate interfaces while the full process is being established. The design allows the addition of further specialized agents through the broker.

Predictive Analytics. Predictive Analytics agents process resource data and real-time process data to improve robustness. Typical services include predictive maintenance that anticipates tool wear and machine failures and demand forecasting that projects workload changes. Insights flow to the scheduler through the broker to drive parameter adaptation. Examples include adjustments to process parameters such as temperature and pressure and changes to scheduling buffers so that plans remain feasible and throughput remains high.

Data Access Broker and API. The Data Access Broker is the hub for data and control flows across actors, analytics, and planning. The storage foundation is Shepard (Haase et al., 2025), which is a multi-database solution for heterogeneous information. Shepard organizes raw signals in containers and captures context in collections that form graphs. Graph nodes can reference data inside containers, other nodes, and external sources through the native Shepard API. To preserve the structure of the data model for actors and scheduling, an additional wrapper API enforces model-specific constraints. This wrapper is the integration surface for services and frontends. It also forms the basis for connecting the system to a broader DT or UNS.

Control Dashboard. The Control Dashboard wraps the broker API to provide high usability for structure and data operations. For structural management, users can create, read, update, and delete instances of the data model such as projects, actors, jobs, job instances, and operations. For data

management, users can upload, download, modify, and delete raw signals that belong to actors and operations. Centralizing these functions lets independent participants manage their workflows while protecting the integrity of scheduling and analytics.

4 EVALUATION

This section evaluates the proposed framework in the FORinFPRO manufacturing use case. It first outlines the industrial setting and process structure, then examines the framework through three representative scenarios, and finally discusses how the observed capabilities contribute to resilient production.

4.1 Use Case – FORinFPRO

The FORinFPRO research project (Bayerische Transformations- und Forschungsförderung, 2023; Bavarian Research Alliance (BayFOR), 2023) provides the industrial setting for evaluating the proposed architecture. This subsection outlines the manufacturing process, the main challenges of the use case, and how the methodology is applied in this setting.

Context. FORinFPRO focuses on the collaborative and cross-disciplinary manufacturing of the hybrid carrier “Erlanger Träger” (Ehrenstein et al., 2003; Lehrstuhl für Kunststofftechnik (LKT), 2026). In this paper, we focus on establishing a closed feedback loop in which production data informs JSS and predictive analytics, while scheduling decisions and parameter adaptations influence subsequent data collection.

Process. The manufacturing process, shown in figure 4, comprises the four major steps wet-laid production, infusion system, injection molding, and ultrasonic welding. Wet-laid production is completed first, after which infusion system and injection molding can be executed in parallel. Their outputs are then joined in ultrasonic welding. In addition, each major step consists of several sequential sub-steps, which results in a hierarchical process with both parallel branches and strict dependencies.

Challenges. The use case is well suited for evaluation because it combines several challenges typical of collaborative manufacturing. Different project partners contribute distinct process steps, collect different

types of data, and pursue partly independent objectives. This requires support for heterogeneous and multimodal data, variable process granularity, and both automated and manual interaction. In addition, downstream observations such as testing results or maintenance indicators may influence later scheduling decisions and upstream process parameters.

Methodology. Using the proposed methodology, the process is represented through abstract jobs and nested subjobs, which separate high-level planning across partners from detailed execution within a single partner environment. Machines, software services, and human operators are treated uniformly as actors collaborating on the same workpiece. During execution, they provide operational data such as pressure, temperature, flow rates, fiber lengths, or quality measurements while also receiving task assignments and configuration updates.

Data Integration. The Data Access Broker serves as the central hub for this information exchange and provides unified access to the multimodal data generated throughout the process. On this basis, the scheduler can route workpieces through the dependency graph according to both job definitions and the current operational state. Predictive Analytics agents can identify emerging issues such as maintenance needs or quality deviations, while the Control Dashboard allows participants to inspect, preprocess, filter, and selectively share data. The dashboard supports strict authentication and role-based authorization using Keycloak (Keycloak Project, 2026).

The FORinFPRO use case therefore provides a realistic setting for assessing scheduling feedback, extensibility, dynamic granularity, and usability in a closed-loop scheduling environment. As is common in research projects, the number of manufactured parts is too limited to fully evaluate the proposed architecture, especially for machine predictive maintenance and damaged-part identification. However, synthetic data can help mitigate this limitation.

4.2 Scenario-Based Evaluation

The evaluation follows a scenario-based design because the framework targets both structural properties, such as extensibility and domain independence, and operational properties, such as scheduling feedback and usability, which a purely quantitative performance metric cannot fully capture. Three representative scenarios therefore assess the functionalities of the implemented prototype, including the Data Access Broker and its API, the Control Dashboard, and

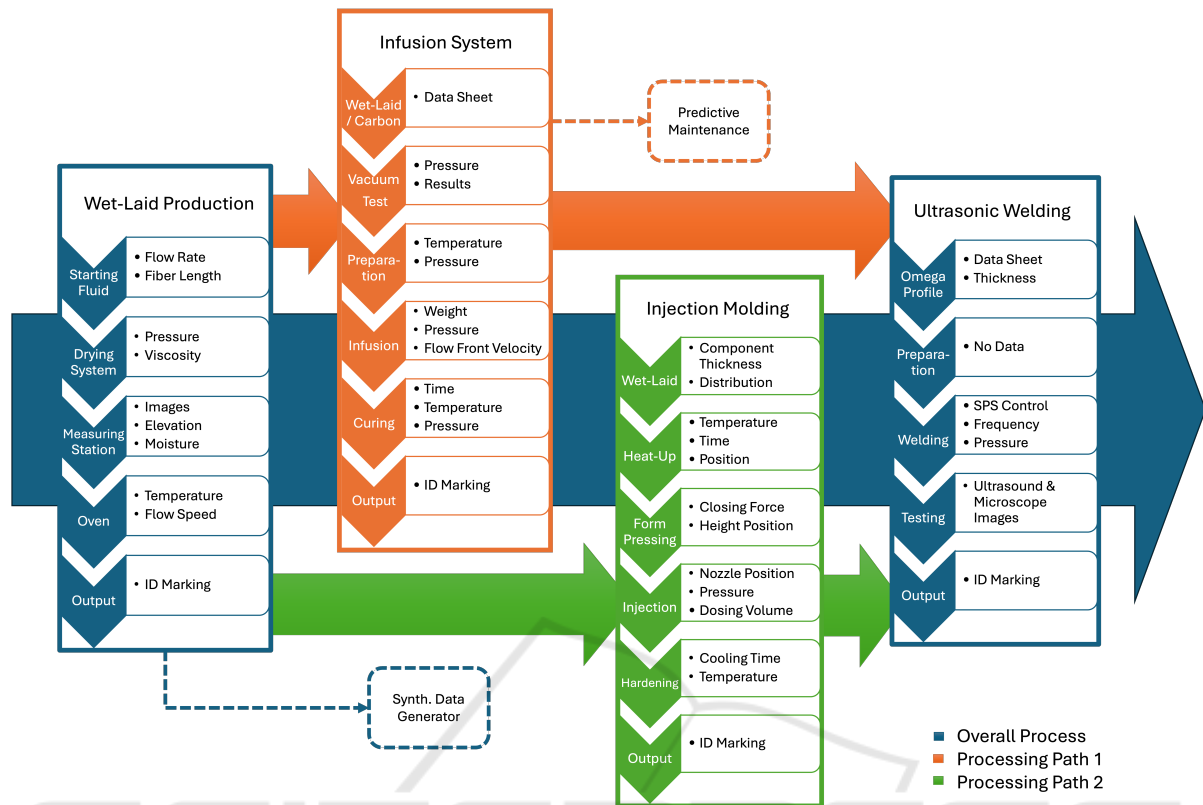


Figure 4: The scenario for the lightweight manufacturing of the hybrid carrier “Erlanger Träger”, shown in blue. The production process is divided into two independent processing paths shown in orange and green. For each sub-step of the major processing steps, the collected data is listed. The dashed fields indicate optional software agents within the process chain.

the Planning and Scheduling and Predictive Analytics modules. Each scenario is structured into setup, execution, and synthesis phases:

- Setup phases define the static preparation and initial configuration required before any active processes are initiated.
- Execution phases detail the dynamic actions and interactions involving the system, human or machine actors, and automated agents during live operation.
- Synthesis phases analyze the outcomes of the execution and demonstrate how the scenario achieves the stated research objectives.

4.2.1 Scenario 1

The first scenario evaluates the capacity of the system to establish scheduling feedback. It focuses strictly on the OT domain by simulating a traditional job-shop environment consisting of physical machines and production operations. In this scenario, we demonstrate how the architecture captures execution data from the machines and seamlessly feeds this historical performance data back into the scheduling system. This

test shows that the architecture can act as a closed-loop mechanism in which continuous data collection directly informs and optimizes future scheduling objectives and decisions.

Setup. System installation and the definition of overarching production goals form the initial state. The user defines the abstract job and its four major operations, namely wet-laid production, the infusion system, injection molding, and ultrasonic welding. This establishes the structural templates within the data model. Next, the physical machines executing these steps are defined as individual actors and configured for data collection. This collection process can operate by calculating and pushing data directly from the edge to the API. Alternatively, the central service can actively poll from REST servers and subscribe to OPC UA servers.

Execution. Dynamic instantiation of new job instances triggers the active operations. The scheduling agent assigns tasks to machines while strictly adhering to process dependencies. The infusion system and injection molding both depend on the completion of

wet-laid production. Ultrasonic welding depends on both the infusion system and injection molding. As physical processing progresses, machines or human workers update the operation states via the API or GUI, thereby signaling readiness for subsequent dependent operations.

Synthesis. Analyzing the data generated during active operation yields specific performance scores. These scores are combined with the original scheduling decisions to retrain or update the underlying scheduling algorithm. For instance, they can serve as a vital reward signal to update the policy of a DRL agent. By continuously feeding historical execution data back into the algorithmic decision process, the system actively learns, thereby showing how the architecture bridges the gap between raw data collection and dynamic scheduling feedback.

4.2.2 Scenario 2

The second scenario tests the domain independence and extensibility of the data model. Moving beyond the pure operational focus of the first scenario, this test introduces diverse and unconventional actor types into the system. Specifically, it includes autonomous software agents alongside human workers. By incorporating these new entities, we evaluate whether the abstracted metamodel can uniformly treat physical hardware, human operators, and software services without breaking current expectations or requiring structural modifications. This scenario demonstrates that the underlying heterogeneous graph model can naturally adapt to new domains.

Setup. The initial state expands to include two autonomous software agents, namely a synthetic data generation agent and a predictive maintenance agent. The underlying metamodel does not require distinct classes. Physical machines and software agents are instantiated from the exact same generic actor type and differ only in their assigned parameters. The system employs a capability-based mapping mechanism to determine which actor is qualified for specific operations.

Execution. As new job instances of the FORinF-PRO workpiece arrive, the scheduler distributes tasks across multiple domains. The software agents act as additional major steps dependent on the regular machining operations. Specifically, the synthetic data generation agent depends on wet-laid production, while the predictive maintenance agent depends on the infusion system. When scheduled, these agents

make autonomous, purpose-driven decisions. For example, if a classifier underperforms due to concept drift, the data generation agent decides to create hybrid synthetic data by fusing anomalies onto healthy examples. Simultaneously, the predictive maintenance agent evaluates infusion data to recommend timely maintenance interventions.

Synthesis. This execution demonstrates two crucial objectives. First, it shows system extensibility, as entirely new digital entities were added without modifying the underlying schema of the data model. Second, it demonstrates domain independence through the system's ability to seamlessly handle, schedule, and connect physical hardware and digital software entities as structurally identical instances within the heterogeneous graph.

4.2.3 Scenario 3

The final scenario focuses on dynamic granularity and usability, driven by the requirements of distributed manufacturing environments. It describes a shop-floor worker who interacts directly with the frontend of the system to manually insert, instantiate, and manage job data. This demonstrates the usability of the dual-interface strategy, showing that users without programming expertise can effectively manage the system without relying on centralized administration. Furthermore, this scenario highlights dynamic granularity by showcasing variable-resolution modeling. Specifically, it demonstrates how a user can initially schedule tasks using a macroscopic resolution, which represents abstract operational steps for long-term planning. Once an abstract step approaches execution, the user can seamlessly transition the model into a microscopic resolution that details the specific jobs and individual tasks required. This ensures that the system remains practical and resource-efficient across different planning horizons.

Setup. The user defines complex jobs representing the major manufacturing steps, which can now be represented with their respective sub-steps. This hierarchical configuration establishes multiple granularity levels. The overarching job represents a macroscopic resolution for high-level planning, while nested sub-jobs contain microscopic execution details. Some detailed sub-steps, including the testing step of the ultrasonic welding operation, require manual human work and can be scheduled accordingly. Furthermore, the frontend is configured to expose the full control mechanisms of the system, thereby ensuring that workers

can manage interactions without centralized administration.

Execution. Execution begins with a high-level job instantiation. For scheduling, the user or automated scheduler may decide whether to expand none, some, or all major steps into sub-steps. As a major step approaches, the system or user can switch to microscopic resolution to schedule the revealed details, including tasks assigned to human operators. At the same time, the scenario emphasizes ease of use: workers interact with the system through the frontend by issuing control commands, such as creating a new job instance for another workpiece, marking operational steps as completed, or manually uploading and preprocessing data. This manual handling applies to both human-driven and automated machine operations when automated collection is infeasible or undesired. In our use case, this occurs during ultrasonic welding. After welding, a comprehensive testing routine is performed, including destructive quality tests and microscopic analysis of the welded joints. The resulting data is manually entered via the GUI and explicitly linked to the current job instance.

These insights can directly affect other running job instances or future executions. For example, parameters in upstream stages, such as temperature or cooling time in injection molding, may be adjusted manually or automatically based on downstream observations. However, this requires sufficient domain knowledge to establish reliable relationships between testing results and the relevant process parameters.

Synthesis. This scenario successfully demonstrates two final objectives. First, it shows dynamic granularity by allowing users or the system to adjust the scheduled resolution, focusing computational resources only where they are currently relevant and enabling reusable subjob definitions. Second, it demonstrates high usability. By empowering users to control job flows, instantiate workpieces, upload data manually, and configure system structures through an intuitive graphical interface, the architecture fulfills the decentralized and user-friendly requirements of the research project.

4.3 Resilience as Outcome

Across the three scenarios, resilience emerges as a combined property of the proposed architecture rather than as an isolated function of a single module. In the context of this paper, resilience denotes the ability to maintain feasible production, acceptable quality, and coordinated decision making despite disruptions such

as machine degradation, changing process conditions, or incomplete information.

Scenario 1 contributes to this goal by establishing a closed-loop connection between execution data and future scheduling decisions. By feeding historical and real-time shop-floor data back into the scheduling process, the architecture enables the system to respond to observed deviations instead of relying exclusively on static plans. This provides the basis for adaptive behavior under changing production conditions.

Scenario 2 extends this capability by demonstrating that predictive and other software-based agents can be integrated into the same data and scheduling model as physical actors. This is particularly relevant for resilient production, since predictive maintenance, synthetic data generation, or related analytical services can identify emerging problems before they propagate through the process chain. Their seamless integration shows that the architecture can incorporate additional resilience-oriented functionality without structural redesign.

Scenario 3 complements these automated mechanisms with human-centered resilience. In collaborative manufacturing environments, not all relevant information is machine-readable, and not all corrective actions should be automated. The ability to refine job granularity, manually enter or preprocess data, and intervene through the frontend provides an important operational safeguard. Thus, usability and decentralized control are not merely convenience features, but part of the architecture's resilience strategy.

Taken together, the scenarios show that the proposed framework combines continuous data feedback, extensible integration of predictive services, and controlled human intervention in a way that supports resilient production. The evaluation therefore indicates that the architecture can enable adaptive and robust coordination in a realistic collaborative manufacturing setting.

5 CONCLUSION AND OUTLOOK

The integration of AI-driven predictions, standardized DTs, and flexible JSS into an agent-based framework defines a promising holistic approach for manufacturing. By evolving from centralized, rigid control toward a decentralized, self-organizing system, manufacturers can make more informed and adaptive trade-offs between cost, quality, and throughput. This holistic approach ensures that the production environment is not merely optimized for a perfect scenario but is fundamentally resilient to the inherent uncertainties

of the modern market. The final outcome is a production system characterized by:

- Resilience: the ability to autonomously reroute and reschedule in response to machine failures or quality deviations through real-time DTs.
- Optimization: maximizing resource utilization through DRL-based scheduling approaches that can produce high-quality decisions within seconds rather than requiring long overnight optimization runs in complex, NP-hard scenarios.
- Future-readiness: a scalable, extensible architecture that ensures long-term data sovereignty and seamless asset integration.

Looking ahead, the development of this architecture will focus on transitioning from a research prototype to a fully operational system. Future work will address the current limitations of the approach and expand the evaluation beyond the hybrid carrier “Erlanger Träger” to include a broader range of real-world use cases. By testing the framework in diverse manufacturing environments, the project aims to further demonstrate the domain independence and scalability of the agent-based approach.

A critical next step involves the full realization of all architectural components. This includes replacing current placeholder modules, such as the evaluation agent, with fully functional versions capable of high-fidelity quality assessment. Furthermore, the scheduling logic will be refined by more tightly integrating GNNs and DRL into a wider array of analytical agents to enhance the precision of parameter adaptation. A further potential extension would be to explicitly capture resources in the data model, including their use as operation inputs, outputs, and stored intermediates.

To enhance standardization and interoperability, future work might explore the Asset Administration Shell (AAS) or comparable standards. When employed, an AAS-based representation can offer a consistent digital model for each actor, whether machine or human, which eases integration into broader UNS or DT ecosystems. Taken together, these steps could strengthen the closed-loop mechanism and enable a more resilient, self-optimizing manufacturing environment across complex engineering disciplines.

ACKNOWLEDGEMENTS

The authors acknowledge the use of Google Gemini, Microsoft Copilot, and OpenAI ChatGPT throughout this article for drafting support, language improvement, and editing. All content was reviewed and approved by the authors.

This work was partially funded by the Bayerische Transformations- und Forschungsförderung and the AI Production Network Augsburg through the Bavarian state government’s Hightech Agenda program.

REFERENCES

- Babai, M. Z., Arampatzis, M., Hasni, M., Lolli, F., and Tsadiras, A. (2024). On the use of machine learning in supply chain management: a systematic review. *IMA Journal of Management Mathematics*, 36(1).
- Bavarian Research Alliance (BayFOR) (2023). New research association forinpro: 2 million euros for the development of ai-based and self-adapting manufacturing processes.
- Bayerische Transformations- und Forschungsförderung (2023). Intelligente fertigungsprozesse & closed-loop-produktion – forinpro.
- Besigomwe, K. (2025). Closed-loop manufacturing with ai-enabled digital twin systems. *Cognizance Journal of Multidisciplinary Studies*, 5.
- Buggineni, V., Chen, C., and Camelio, J. (2024). Enhancing manufacturing operations with synthetic data: a systematic framework for data generation, accuracy, and utility. *Frontiers in Manufacturing Technology*, Volume 4 - 2024.
- Dauzère-Pérès, S., Ding, J., Shen, L., and Tamssaouet, K. (2024). The flexible job shop scheduling problem: A review. *European Journal of Operational Research*, 314(2).
- Deokar, S., Bansode, V., Wankhede, S., and Kharche, P. (2025). Industry 4.0 moving towards smart manufacturing: A comprehensive review. *International Journal of Science and Research Archive*, 16(3).
- Destouet, C., Tlahig, H., Bettayeb, B., and Mazari, B. (2025). Systematic review and future directions in dynamic flexible job shop scheduling: a decade of research. *Journal of Intelligent Manufacturing*.
- Díaz, J. A., Rey, J. M., Hernández, I., Duarte, N., and Reynolds, W. (2025). Unified namespace: A new paradigm for industrial iot data integration. *IEEE Internet of Things Magazine*.
- Ding, J., Chen, M., Wang, T., Zhou, J., Fu, X., and Li, K. (2023). A survey of ai-enabled dynamic manufacturing scheduling: From directed heuristics to autonomous learning. *ACM Computing Surveys*, 55(14s).
- Douaioui, K., Oucheikh, R., Benmoussa, O., and Mabrouki, C. (2024). Machine learning and deep learning models for demand forecasting in supply chain management: A critical review. *Applied System Innovation*, 7(5).
- Ehrenstein, G. W., Amesöder, S., Fernández Díaz, L., Niemann, H., and Deventer, R. (2003). Werkstoff- und prozessoptimierte herstellung flächiger kunststoff-kunststoff und kunststoff-metall-verbundbauteile. In *Robuste, verkürzte Prozessketten für flächige Leichtbauteile. Tagungsband zum Berichts- und Industriekolloquium des SFB 396*. Meisenbach.

- Feizabadi, J. (2022). Machine learning demand forecasting and supply chain performance. *International Journal of Logistics Research and Applications*, 25(2).
- Haase, T., Dr. Glück, R., Kaufmann, P., and Willmeroth, M. (2025). shepard - storage for heterogeneous product and research data.
- Hayes, C. F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., Verstraeten, T., Zintgraf, L. M., Dazeley, R., Heintz, F., Howley, E., Irissappane, A. A., Mannion, P., Nowé, A., Ramos, G., Restelli, M., Vamplew, P., and Roijers, D. M. (2022). A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26.
- Hermann, M., Pentek, T., and Otto, B. (2016). Design principles for industrie 4.0 scenarios. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*.
- Hummel, V. and Schuhmacher, J. (2023). Method for semi-automated improvement of smart factories using synthetic data and cause-effect-relationships. In *Proceedings of the 5th Conference on Production Systems and Logistics (CPSL 2023-2)*. publish-Ing.
- Johnson, D., Chen, G., and Lu, Y. (2024). Multi-agent scheduler for the continuous dynamic flexible job shop scheduling problem. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*.
- Kagermann, H., Wahlster, W., and Helbig, J. (2013). Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry. Final report of the industrie 4.0 working group, acatech – National Academy of Science and Engineering.
- Kemper, N., Heider, M., Pietruschka, D., and Hähner, J. (2025). A comparative study of multi-objective and neuroevolutionary-based reinforcement learning algorithms for optimizing electric vehicle charging and load management. *Applied Energy*, 391:125890.
- Keycloak Project (2026). Keycloak. <https://www.keycloak.org/>.
- Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., and Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering*, 6(4).
- Lehrstuhl für Kunststofftechnik (LKT) (2026). Plastic / metal hybrid technology.
- Masudin, I. and Kamara, M. (2018). Impact of just-in-time, total quality management and supply chain management on organizational performance: A review perspective. *Jurnal Teknik Industri*, 19:11.
- Mei, R.-S., Li, G., Jia, S., Huang, L., Shih, L.-W., Arinez, J., Abell, J., and Shao, C. (2025). Synthetic data generation in smart manufacturing applications: A systematic review.
- Oks, S. J., Jalowski, M., Lechner, M., Mirschberger, S., Merklein, M., Vogel-Heuser, B., and Möslin, K. M. (2024). Cyber-physical systems in the context of industry 4.0: A review, categorization and outlook. *Information Systems Frontiers*, 26(5).
- Pfleiderer, A. and Bauer, B. (2025). Fused hybrid training samples through synthetic anomaly generation for optimized model training. In *2025 International Conference on Advanced Machine Learning and Data Science (AMLDS)*.
- Qin, Z. and Lu, Y. (2021). Self-organizing manufacturing network: A paradigm towards smart manufacturing in mass personalization. *Journal of Manufacturing Systems*, 60.
- Rai, R., Tiwari, M. K., Ivanov, D., and Dolgui, A. (2021). Machine learning in manufacturing and industry 4.0 applications. *International Journal of Production Research*, 59(16).
- Salcher, F., Finck, S., and Hellwig, M. (2024). A smart shop floor information system architecture based on the unified namespace. In *2024 IEEE International Conference on Engineering, Technology, and Innovation (ICE/ITMC)*.
- Singh, P., Krishnan, K., and Boldsai Khan, E. (2026). A review of production scheduling with artificial intelligence and digital twins. *Journal of Manufacturing and Materials Processing*, 10(1).
- Song, W., Chen, X., Li, Q., and Cao, Z. (2023). Flexible job-shop scheduling via graph neural network and deep reinforcement learning. *IEEE Transactions on Industrial Informatics*, 19(2).
- Stieler, F. and Bauer, B. (2024). Enhancing collaboration and agility in data-centric ai projects. In *Evaluation of Novel Approaches to Software Engineering*. Springer Nature Switzerland.
- Sutton, R. S. and Barto, A. G. (2020). *Reinforcement Learning - An Introduction*. The MIT Press, second edition.
- Tsallis, C., Papageorgas, P., Piromalis, D., and Munteanu, R. A. (2025). Application-wise review of machine learning-based predictive maintenance: Trends, challenges, and future directions. *Applied Sciences*, 15(9):4898.
- Ullah, W., Rashid, M. F. F. A., and Hammad, M. Y. (2025). Multi-objective dynamic job shop scheduling optimization in manufacturing systems: A short review. *International Journal of Science and Research Archive*, 17(2).
- Wang, Z., Zhang, J., and Si, J. (2019). *Dynamic Job Shop Scheduling Problem with New Job Arrivals: A Survey*. Springer Singapore.
- Weigell, B., Stieler, F., and Bauer, B. (2025). All you need is an ai platform: A proposal for a complete reference architecture. In *2025 IEEE/ACM 4th International Conference on AI Engineering – Software Engineering for AI (CAIN)*.
- Wölfel, M. and Bauer, B. (2026). Toward dynamic flexible job-shop scheduling combining graph neural networks and deep reinforcement learning on heterogeneous graph representations. In *Innovative Intelligent Industrial Production and Logistics*. Springer Nature Switzerland.