Future Generation Computer Systems 153 (2024) 505–520

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Dynamic context-aware workflow management architecture for efficient manufacturing: A ROS-based case study

William Ochoa *, Jon Legaristi, Felix Larrinaga, Alain Pérez

Electronics and Computer Engineering department, Mondragon Unibertsitatea, Mondragon, 20500, Gipuzkoa, Spain

ARTICLE INFO

Keywords:
Context-aware
Workflow management
Industry 4.0
Robot Operating System (ROS)
Business Process Modeling and Notation (BPMN)
Asset Administration Shell (AAS)

ABSTRACT

The manufacturing industry of the future requires innovative approaches to optimize operational efficiency and adaptability. Integrating context-awareness into workflow management systems has emerged as a promising avenue to enhance efficiency in modern manufacturing processes. This research presents an innovative context-aware workflow management architecture designed to address industry-related challenges and overcome current limitations in the state-of-the-art. The architecture leverages Industry 4.0 standards for asset representation and workflow notation while incorporating a Context Analyzer component for real-time context interpretation. The effectiveness of the proposed solution is demonstrated in a real-world manufacturing setting, specifically in the scenario of collecting work order materials using the Robot Operating System (ROS) technology for robot navigation. The evaluation showcases improvements in task completion rate, resource utilization, and task completion time. These outcomes exemplify the potential benefits of incorporating context-awareness into manufacturing workflows, providing insights for further improvements. Contributions include advancing the understanding of context-aware workflow management, a review of the challenges that cap its adoption in the manufacturing domain, a qualitative comparison of similar approaches, practical implementation of the proposed architecture, evaluation of the context-aware component, and provision of the source code and datasets to the community for future advancement and reproducibility.

1. Introduction

The manufacturing industry of the future faces numerous challenges in optimizing operational efficiency while ensuring adaptability. With the increasing complexity and dynamism of modern production processes, traditional workflow management systems often struggle to keep pace. Efficient workflow management plays a critical role in coordinating assets, streamlining tasks, and ensuring smooth operations throughout the workflow management life cycle [1]. To address these challenges, context-awareness has emerged as a promising approach to enhance the reactiveness of manufacturing systems [2]. Context-aware workflow management systems leverage real-time data about environmental and situational information to dynamically adapt workflows. This way, manufacturing companies can optimize resource utilization and improve the overall efficiency of manufacturing operations [3]. This paper specifically focuses on applying context-aware workflow management to the domain of Industry 4.0 and to emphasize its application, a use case involving robots is provided.

In the Industry 4.0 context, workflow management requires standardization in coordinating and orchestrating industrial assets [4]. Assets encompass a wide range of units that perform tasks, including robots, machines, sensors, and other industrial equipment [5]. Orchestration on the counterpart involves defining how a set of tasks or activities within a workflow should be executed to achieve a specific goal [6]. In this regard, the Business Process Management (BPM) discipline offers tools and techniques for modeling and managing workflows, including the widely adopted BPMN (Business Process Modeling and Notation) language [7].

In the manufacturing domain, workflows encompass various objectives, including material transformation, asset allocation, and information processing [8]. Within the scope of this paper, workflows primarily comprise tasks related to robot manipulation, with the goal of using these robots to collect materials for designated work orders.

A key challenge in this domain is the coordinated asset allocation to ensure smooth operations in dynamic and variable manufacturing environments, wherein traditional systems often fall short in handling such scenarios [1]. This challenge has led industries to adopt standards for asset interoperability. Standardization of digital assets has gained significant attention, with the Asset Administration Shell (AAS), which aligns with the principles of the Reference Architectural Model for Industry 4.0 (RAMI 4.0) [9]. AAS simplifies asset interoperation and identification by providing technical and operational data.

* Corresponding author.
E-mail address: wsochoa@mondragon.edu (W. Ochoa).

https://doi.org/10.1016/j.future.2023.12.024
Received 13 August 2023; Received in revised form 16 November 2023; Accepted 21 December 2023
Available online 23 December 2023

However, to truly address the challenges in manufacturing, context-awareness needs to be integrated into workflow management systems. Context-awareness introduces new capabilities and opportunities for improving efficiency, adaptability, and responsiveness in manufacturing processes [10]. Context-aware workflow management systems can dynamically adapt and optimize processes by analyzing real-time information and performing intelligent decision-making [11]. These systems leverage contextual factors such as resource availability, resource status, and environmental changes [12] to make decisions on dynamically adjusting the workflow during execution. Resulting in a reduction in delays and improved manufacturing efficiency.

For instance, consider a scenario in a smart manufacturing facility, which employs robots to perform various tasks. These robots have characteristics such as battery level, position, uncertainty, and payload capacity (the maximum weight the robot can carry). In a context-aware workflow, when a new task arises, the system can analyze the real-time contextual information. If one robot has a low battery level, another is closer to the working location, and a third has a high payload capacity, the context-aware system can intelligently decide which robot is best suited for the task, ensuring efficient resource utilization, timely task completion, and increased task completion rate.

This research proposal aims to contribute to advancing context-aware workflow management in the manufacturing domain. Building upon existing literature and research, the proposed study focuses on developing a context-aware workflow management architecture and evaluating its context-aware component in a real-world manufacturing setting. The proposed architecture leverages dynamic asset allocation and provides intelligent decision-making on adjusting the workflow during runtime. The architecture aims to address the challenges of manufacturing operations, optimizing the utilization of resources and streamlining the overall manufacturing process.

Establishing Research Questions (RQs) is crucial to determine the focus of research [13]. In this study, two RQs guide the development and evaluation of the proposed architecture:

- RQ1: How does the proposed context-aware workflow management architecture improve operational efficiency and responsiveness within manufacturing processes?
- RQ2: What challenges and limitations arise when integrating the proposed architecture in real-world manufacturing environments, and how can they be addressed?

The structure of this paper is organized as follows: Section 2 analyzes industry-related challenges that hinder the adoption of workflow management systems in manufacturing companies. Section 3 reviews the state-of-the-art and investigates the challenges and limitations of integrating context-awareness into workflow management systems by comparing existing solutions in a qualitative manner. Section 4 summarizes the literature findings and research gaps, highlighting the current state of context-aware workflow management. In Section 5, the architecture for context-aware workflow management is presented. Section 6 assesses the effectiveness of the proposal for improving operational efficiency, responsiveness, and resource utilization in a real-world manufacturing scenario on collecting work order resources where multiple robots are employed. Section 7 discusses the results obtained from the testing phase and explains the challenges addressed by this approach. Finally, Section 8 presents the concluding remarks, along with a suggestion for potential future research. The findings of this research will contribute to the field of context-aware workflow management, enabling researchers and practitioners to enhance manufacturing processes and drive the industry toward dynamic adaptability.

2. Industry-related challenges for efficient workflow management

A comprehensive analysis of industry-related challenges that hinder the adoption of workflow management systems is presented in [14], these are categorized based on the workflow management life cycle. In this section, the most critical challenges are identified and analyzed. These include:

1. **Workflow Modeling:** It involves architectural design and workflow representation. The requirements for workflow representation include: fitting for collaborative context, supporting workflow generation, compactness, compositionality, open semantics, and extensibility. These requirements can be addressed through the use of formal modeling languages such as BPMN, BPEL (Business Process Execution Language), state machines, Petri Nets, and YAWL (Yet Another Workflow Language) [15], with BPMN being ratified as the standard language by the Object Management Group (OMG) for business process design[16].

2. **Heterogeneous Infrastructure Scale:** Designers face the challenge of finding the right level of aggregation/abstraction for composing workflows in a heterogeneous infrastructure [17]. They need to balance the decreasing unit of execution in edge environments with the requirement for interaction with central cloud orchestrator systems. Scalability is achieved through hierarchical models and abstract representations of units, considering numerous devices and their specific functionalities [18].

(a) Chaining Data From Heterogeneous Functions: In an IoT architecture, the collaboration between cloud and fog devices introduces challenges in handling fragmented information and diverse data locations [19]. Designers need to chain this fragmented functionality into cohesive workflows to ensure proper workflow management.

(b) Feasibility: Assessing the feasibility of workflows involves verifying technical feasibility as well as considering contextual factors [20]. Designers must determine if there are services of sufficient quality available for each task in the workflow and check for any policies that may restrict the execution of certain services within the workflow.

3. **Collaboration:** Workflow Management Systems need to operate in collaboration with industrial designers, machine operators, their resources, and other systems [21]. This requires standardizing output, communication, syntax, and semantics. Orchestrating workflows across multiple organizations while handling heterogeneous edge-cloud implementations poses architectural constraints.

4. **Parallel Execution Capabilities:** This challenge involves executing multiple instances of the same workflow concurrently and handling scenarios where multiple workflows request the same service simultaneously or a single workflow requests multiple services simultaneously [22]. It requires considering the various possibilities for parallelism and determining which actions can be performed simultaneously in each scenario.

5. **Asynchronous Task Execution:** Enabling workflows to handle asynchronous task communications, allowing them to proceed without waiting for immediate replies, is a significant challenge [23]. Such mechanisms listen for replies from workflow tasks in order to continue workflow execution. In contrast, synchronous communication requires tasks to pause until a reply is received.

6. **Dynamic Nature of Microservice Architectures:** Microservice architectures introduce dynamic phenomena such as constant changes and dynamism given by mixed cloud, edge, and IoT devices, requiring dynamic adaptation mechanisms for runtime configuration and deployment [24]. The volatility and rotation of edge resources pose reliability challenges, as functionality and deployment conditions quickly become obsolete [25]. Microservice-based architectures offer fast response times and rapid deployment but require workflow orchestrators to handle the speed of deployment and failures effectively [26]. Additionally, the discovery of available services at runtime becomes crucial due to the constantly changing availability of services at the edge, demanding efficient device and service registration for optimal workflow execution efficiency [17].

These industry-related challenges discern the multifaceted complexities that pose obstacles to the adoption of workflow management systems in the industry domain. Building upon this foundation, the next section delves into advancements, limitations, and challenges within the state-of-the-art, specifically examining approaches to context-aware workflow management. This analysis will provide an understanding of the existing landscape and pave the way for the development of

Table 1
Comparison of context-aware workflow management approaches.

REF	Workflow format	Design		Orchestrator		Context-aware integration		Context data		Evaluation		
		User-centric	Data-driven	Central	Edge	Design time	Runtime	Calculated	Sensor	Case study	Real-world	Metrics
[27]	BPEL, State machine	×	✓	✓	×	×	✓	Availability and Execution time of service operations. These variable values are calculated by doing a ping periodically.	×	Workflow composition	×	Execution time and memory consumption.
[28]	BPMO (Business Process Modeling Ontology)	×	✓	✓	×	×	✓	Cost efficiency, product stock availability.	×	Sales orders	× Synthetic dataset	Task failure rate
[29]	HTN (Hierarchical Task Network)	±	✓	✓	×	×	✓	Non-specified. But assuming those that come in the WSC dataset: availability and throughput.	×	Loan approval	× Synthetic dataset	Precision, Recall, and F-Score on service discovery and selection.
[30]	Control diagram	×	✓	✓	×	×	✓	Frequency, response time, memory, CPU, and precision.	Temperature, humidity, location, connection Media (WiFi), transmission Latency, and remaining system lifetime.	Car-seat FabLab	±	Cost time of policy aggregation and matching (SPARQL query built). Three epochs from 1 to 10 context variables.
[31]	BPMN, DMN	✓	✓	✓	×	×	✓	×	Temperature, Humidity, Smoke, Weight, and GPS.	Pickup cargo	×	×
[32]	BPMN	± Semantic expert is required	✓	✓	×	✓	✓	Service unavailability, execution cost, energy consumption, precision, outcome rate, acquisition cost, time, delivery rate, setup cost, and expected revenues.	×	Bicycle manufacturing	✓	Quality of final composition
[33]	APFL (Adaptable Pervasive Flow Language)	± Semantic expert is required	✓	✓	×	✓	✓	Service unavailability, status, type, and location.	×	Process chain of the car logistics in a harbor	✓	CPU performance vs. Number of services involved in a workflow composition. Number of compositions resolved within N seconds.
[34]	BPMN	×	✓	✓	×	✓	×	Energy consumption	×	Workflow composition	× Synthetic dataset	Running time to achieve service composition plan
[35]	Not specified	×	✓	✓	×	×	✓	Cost, availability, reliability, and reputation	×	Workflow composition	×	The score of the optimal composition, execution time, and deviation.
[36]	BPMN	✓	✓	✓	×	×	✓	Arrival time	Temperature, location	Smart irrigation system, Ventilation system, Health care system	×	×
[37]	BPMN	±	✓	✓	×	×	✓	×	Machine status (whether it is on or off)	Fischertechnik smart factory	✓	Composition time vs. number of services. Search time for the best service.
Ours	BPMN	✓	±	✓	✓	±	✓	Average success rate, network latency, response time. More can be added.	Temperature, humidity, proximity, battery level, payload capacity, and positional uncertainty. More can be added.	ROS-based robots in a pick-and-deliver warehouse scenario.	✓	Task completion rate, resource utilization, and task completion time.

✓: The approach entirely covers the feature
±: The approach covers the feature with limitations
×: The approach does not cover the feature.

the solution presented in this work, which aims at addressing these challenges and advancing the state-of-the-art in dynamic context-aware workflow management for Industry 4.0.

3. Related work

This section examines existing literature and research pertaining to context-aware workflow management. Additionally, it builds upon a comprehensive systematic literature review conducted on semantic workflow management systems in [38]. This section provides an updated revision and deepens into the strengths, limitations, and gaps identified in the current body of work.

The characteristics of existing approaches in the field of context-aware workflow management are compared in Table 1. The table includes columns such as "Workflow Format", which denotes the notation language employed for workflow design; "Design" indicates whether the platform implements user-centric designs that prioritize user interactions and preferences [39], or data-driven designs that rely on data analysis techniques [40] to automatize workflow modeling. The "Orchestrator" column describes whether the approach adopts a central and/or edge workflow executor, indicating how the workflow tasks are distributed and managed within the system [41].

Additionally, the "Context-Aware Integration" column highlights how context-awareness is integrated, either during the workflow design phase, where context is pre-defined, or at runtime, where real-time context data is utilized to adapt workflows dynamically [42]. The "Context Data" column identifies the type of context data considered, which can be derived from sensors, such as sensors for proximity, humidity, temperature, etc. or calculated based on historical data and analytics [43]. Furthermore, the evaluation aspects are presented through the "Evaluation" column, encompassing three sub-columns: "Case Study" specifying the context in which the approach was tested; "Real-World" indicating whether the approach was tested in an actual manufacturing scenario with real robots and resources; and "Metrics" detailing the evaluation metrics employed by each approach to measure performance and efficiency.

The works compared in Table 1 propose innovative approaches that leverage a diverse range of techniques and frameworks to address the challenges faced in workflow management. It is worth mentioning that there are two types of properties used in these approaches: Functional Properties (FPs) and Non-Functional Properties (NFPs). FPs define the functionality of a system and its components, while NFPs encompass Quality of Service (QoS) properties that determine how well the tasks or services deliver results [44].

The following enumeration analyzes the approaches by classifying them considering the critical challenge they addressed. These categories are:

(I) **Workflow composition and recomposition**

One common theme among the reviewed studies is the use of context-awareness for workflow composition and recomposition, allowing to handle exceptional situations by adapting workflows in response to context changes. For instance, the work of Bucchiarone et al. [33] and Alférez and Pelechano [27], emphasize the importance of capturing context for composing and adapting workflows dynamically. Bucchiarone et al. proposed an AI (Automated artificial intelligence) planning-based composition framework. This framework enables service discovery, selection, composition, and deployment of workflows. With their approach, activities (workflow tasks) can be annotated with preconditions and effects at design time, and workflow compositions are created using a planner module. AI is employed as the reasoning mechanism to minimize the search space by considering knowledge from previous executions and analyzing context for the reuse of smaller workflows in the final composition.

Similarly, Alférez and Pelechano proposed a tool-supported context-aware framework to guide autonomic adjustment of service compositions at runtime. Their proposal implements the components of IBM MAPE-K (Monitor, analyze, plan, execute, and knowledge) [45], a widely used framework for building autonomous systems. The MAPE-K-based framework allows for dynamic adaptation in response to exceptional situations that may arise when executing a task of a workflow.

(II) **Semantic web-based workflows**

Another aspect explored in the literature is the integration of semantic technologies into workflow management, enabling more intelligent and automated workflow adaptation. In this regard, Arul and Prakash [29] and Mazzola et al. [32] focus on adding semantics to web services and workflows to improve automatic service composition. Arul and Prakash's framework uses ontology-based search to convert syntactic service definitions into semantic representations, enabling the creation of optimal abstract and concrete-executable workflows. Similarly, Mazzola et al. employ semantic annotations and pattern-based algorithms to facilitate the semantic composition of business processes. In addition, Bekkouche et al. [35] developed an automatic semantic web service composition approach that replaces services within workflows at runtime. Services are rated using the Harmony Search (HS) algorithm, which considers QoS constraints to select the most suitable service.

(III) **Real-time context data for adaptive workflows**

The utilization of real-time context information for intelligent decision-making capabilities is also a prominent theme in the reviewed works. Kir and Erdogan [28] propose an intelligent business process management framework that captures social aspects and employs agents and ontologies to handle process exceptions. Their approach provides cognitive capabilities and supports knowledge workers in decision-making tasks. Valderas et al. [36] focus on modeling IoT characteristics in workflows and utilizing contextual knowledge for adaptive decision-making. Their approach performs decision-making by injecting high-level events while maintaining the workflow complexity.

Furthermore, several studies address the challenges and limitations associated with integrating IoT-derived context data into workflow management systems. Song et al. [31] emphasize the importance of considering IoT data and context ontologies to enhance business process decision-making. They propose a context-aware BPM ecosystem that enables adaptive processes at both design time and runtime. Malburg et al. [37] propose an architectural solution and implementation proposals for adaptive workflow management in smart factories, addressing issues related to process monitoring, adaptation, and compatibility with other running processes. Similarly, Lyu et al. [30] propose a context manager module that continuously analyzes the system environment using IoT devices and sensors to control the system behavior. The module makes decisions on whether a service should be kept, tuned, or changed on the fly. The architecture consists of IoT devices that are described semantically by means of FPs and NFPs. A microservice layer is integrated to select the best device and service to invoke it properly based on the context at runtime.

(IV) **QoS scheduling approaches for workflow applications**

In contrast to the previous studies, several literature approaches emphasize evaluating QoS criteria for optimal solution selection from a massive pool of candidates and constraints, using various algorithms to allocate resources of the highest quality for scheduled execution. For instance, in [46], a Quality-of-Service fault-tolerant workflow management system (QFWMS) is introduced by Ahmad et al. It employs QoS-aware scheduling for scientific workflows in cloud computing. The QoS criteria evaluation considers parameters like make-span – time taken to complete a job, cost – resources consumed by a job, deadline and budget – time and resource constraints, and SLA violation – whether the service level agreements are unmet. This approach outperforms by efficiently assigning tasks to the nearest available resources. Similarly, Ambursa et al. introduced LAPSO [47], a particle swarm optimization and min–max-based workflow scheduling algorithm. LAPSO focuses on balancing six critical QoS workflow scheduling objectives: time, cost, reliability, availability, security, and reputation. It offers an effective solution for scenarios with strict constraints.

Furthermore, Sharma et al. proposed an ant colony-based optimization model for QoS-based task scheduling in cloud computing environments [48]. Their multi-objective optimization approach evaluates three primary factors: response time, throughput, and reliability. The study addresses various QoS constraints, such as maximum response time and minimum throughput, aiming to identify an optimal solution that adheres to these constraints while accommodating the dynamic nature of QoS criteria. Similarly, Yu et al. proposed a QoS-based workflow management system for service grids [49], which enables users to specify QoS requirements, including deadline and budget, for workflow execution. It employs a scheduling algorithm that minimizes execution costs while ensuring the deadline is met, considering measurements of time constraints and execution costs. This approach demonstrates its adaptability to dynamic situations through runtime rescheduling.

It is worth mentioning that AAS could play a key role in context-aware workflow management, as previously highlighted in Section 1. AAS provides a standardized means for digitally representing and describing machines. For instance, the TechnicalData [50], NamePlate [51], and CapabilitiesSkillsServices [52] AAS submodels provide a consistent way to describe the technical and operational characteristics of machines. Specifically, the capabilities, skills, and services of assets would facilitate context-aware workflow management systems to discover and orchestrate assets dynamically. As an example, consider a context-aware workflow management system that leverages these AAS submodels to identify robots with the capability to collect work order resources. This system can then orchestrate these robots to accomplish the task. Although the CapabilitiesSkillsServices AAS submodel is still under development, its potential to transform workflow management in Industry 4.0 environments is promising.

The examination of these various approaches has revealed different strategies and technologies in the field of context-aware workflow management, raising the Industry 4.0 domain. However, this examination has also revealed challenges that still need to be addressed to advance toward dynamic context-aware workflow management in the manufacturing domain.

4. Literature findings and research gaps

The reviewed studies on context-aware workflow management revealed several strengths in addressing the challenges of workflow management. These studies propose innovative approaches, leveraging techniques such as AI planning, semantic technologies, and real-time context information to improve workflow composition, adaptation, and decision-making. The use of context-awareness in workflow composition and re-composition allows for dynamic adaptation to changing circumstances, while the integration of semantic technologies enhances

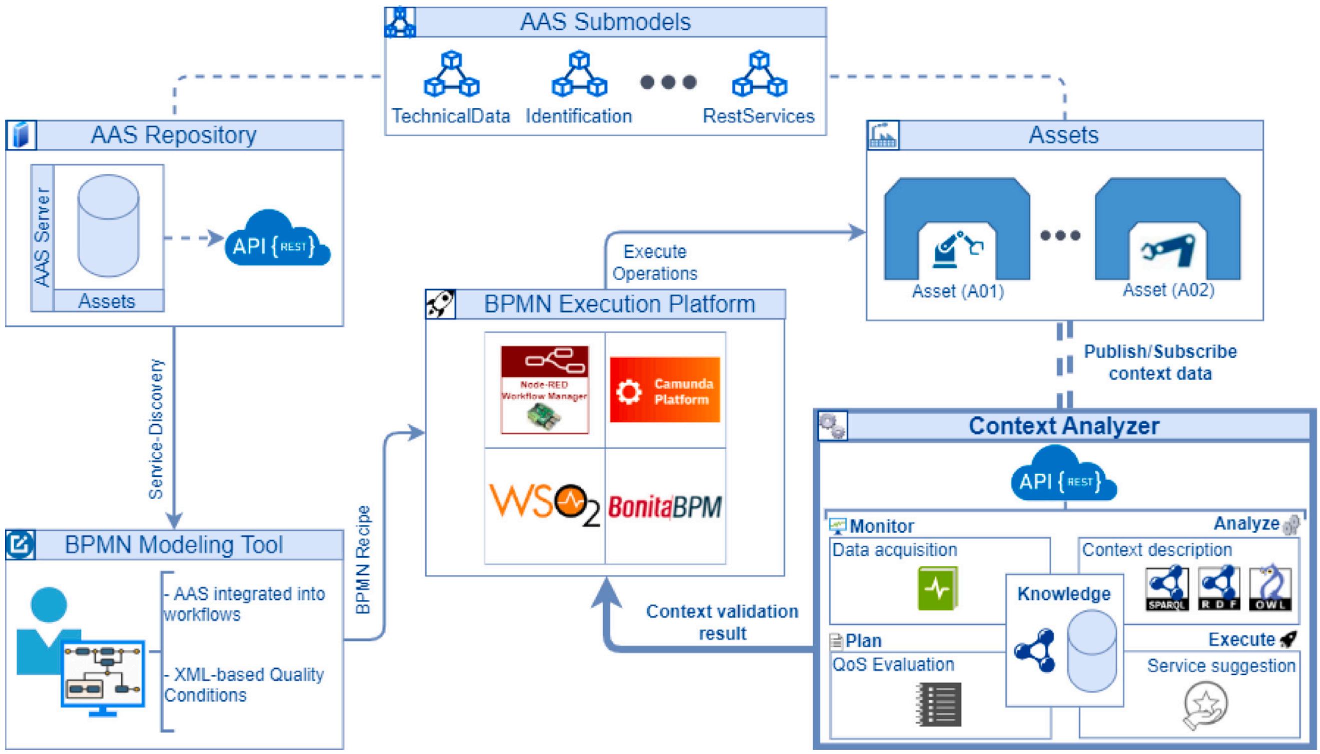


Fig. 1. Architecture for context-aware workflow management: An asset administration shell-based approach.
Source: Extended from [53].

automatic service composition and workflow design. Additionally, the utilization of real-time context information enables intelligent decision-making capabilities, supporting knowledge workers and improving overall workflow management. Furthermore, the studies address the challenges of integrating IoT-derived context data into workflow management systems, proposing architectural solutions and implementation proposals for adaptive workflow management in smart factories. Overall, these strengths contribute to advancing the state-of-the-art in context-aware workflow management and highlight the practical implementation challenges that need to be addressed.

However, the exploration of existing literature in the field of context-aware workflow management has also revealed several areas with opportunities for further research and improvement. These include (1) Need for real-world implementations, (2) Consideration of a broader range of context variables, including both calculated and sensor-derived data, and (3) Utilization of industry-oriented standards and standardized workflow formats. Additionally, there is also (4) Need for user-centric design approaches and migration from abstract workflows to executable workflows. Another notable area is (5) Need for decoupled systems, as current approaches often lack decoupling between various components such as the workflow modeling software, the workflow execution software, and the context-aware component. This lack of decoupling restricts their compatibility with a wider variety of components and systems. For instance, decoupling the context-aware component from the workflow management system would offer the advantage of easy integration with already existing workflow management systems.

Inspired by these limitations, this work proposes a context-aware workflow management architecture that differs from existing approaches in the following aspects: (1) Tested in a realistic manufacturing environment with ROS-based robots and the provision of metrics demonstrating improved manufacturing efficiency, (2) Utilization of semantic web technologies to allow the inclusion of a broader range of context variables, including both calculated and sensor-derived data, within the definition and evaluation of quality conditions, and (3) Utilization of industry-oriented standards and standardized workflow formats within a decoupled architecture to improve compatibility with existing systems.

5. Architecture for context-aware workflow management

This section presents details of the context-aware workflow management architecture. Outlines key components, algorithms, and methodologies for achieving dynamic context-aware workflow management. This section also explains the compatibility of the architecture with the manufacturing environment.

The architecture, as illustrated in Fig. 1 and described briefly in Table 2, is designed to facilitate the orchestration of machine/device services at both the central and edge levels, providing a more distributed and flexible workflow. The decoupled nature of the architecture allows for easy implementation of any individual component into existing workflow management systems. This work develops an updated version of the proposed architecture previously introduced in [53], with a particular focus on the novel context-aware component. An additional advantage of this architecture and all its components is the Apache-2.0 license, which allows for community use and further enhancements.[1]

5.1. Workflow design and execution process

This architecture orchestrates asset services by first describing the technical and operational data of assets, which are represented as .aasx files. These digital descriptions are stored in the AAS Repository. The next step is to design the workflow recipe using any BPMN Modeling Tool. In order to ease the modeling of AAS-based workflows, a Camunda Modeler plugin called "AAS Web Service Discoverer" is included. The plugin lists the assets and services offered by the administration shell server. Users can include them in the workflow by performing drag and drop into the canvas.

[1] https://github.com/MUFacultyOfEngineering.

Table 2
Components of the architecture for context-aware workflow management.

Component	Description
Assets	AAS is utilized to digitize I4.0 physical machines/devices at plant level and represent them as I4.0 digital assets [54].
AAS Repository	Contains the AAS Server that stores administration shell data. The data can be queried and/or maintained using the AAS Server API (Application Programming Interface). The AAS Server can be any including Basyx [55], NOVAAS [56], Admin-shell-io (AASX Server).
AAS Submodels	Describes technical and operational data of assets [57,58]. The RestServices AAS submodel, in particular, characterizes attributes of REST services, including URL, name, method, IsAsync, RequestBody, and Response. This submodel, initially introduced in [53], is used to facilitate a Service-Discovery mechanism within Camunda Modeler.
BPMN Modeling Tool	It can be any BPMN modeling software. In order to ease the modeling of workflows, a plugin for Camunda Modeler called "AAS Web Service Discoverer" is provided. The plugin enables Camunda Modeler to discover services from a chosen AAS Repository. With this tool, users can design manufacturing business processes out of asset services in BPMN format. In addition, the plugin provides an interface to set quality conditions using the quality properties of the assets.
BPMN Execution Platform	Comprehends any workflow executor software that can understand BPMN recipes. There are several BPMN executor software options available, including Camunda Platform, WSO2, Bonita BPM, and Node-RED Workflow Manager (Node-RED WM) [59]. The latest one is a workflow management system that interprets and runs workflow recipes written in BPMN. It can be installed in embedded systems with low resource requirements and is pre-programmed to make decisions in response to context changes by communicating with the API of Context Analyzer.
Context Analyzer	Employs semantic web technologies for context mapping and the MAPE-K (Monitor, Analyze, Plan, Execute, and Knowledge) reference model for autonomous systems [60]. Its goal is to enhance workflow dynamism during runtime.

Once the design of the workflow is completed, the corresponding XML representation is uploaded to the BPMN Execution Platform. This can be any workflow manager that can read BPMN. In this case, Node-RED WM is proposed as a lightweight workflow manager that can operate at central and edge levels. Furthermore, Node-RED WM is pre-programmed to automatically make decisions based on the recommendations provided by the Context Analyzer. When a process is initiated, the Context Analyzer is queried by Node-RED WM each time a Service-Task is scheduled for execution. Context Analyzer evaluates whether a service should be replaced or not on-the-fly by analyzing context variables and quality conditions.

In summary, the decoupled nature of this architecture allows for more flexibility and integration into existing workflow management systems. The next subsection describes in detail how the architecture and Context Analyzer work, including the algorithm that performs the selection of the best device/service during workflow execution.

5.2. Context analyzer

Context recognition is crucial for manufacturing systems to react correctly and enable dynamic changes to the workflow during runtime based on context data [61]. Semantic Web technologies have demonstrated advantages in effectively describing and inferring context data [38]. The Context Analyzer serves as a pivotal component in the proposed context-aware workflow management architecture. It leverages semantic web technologies for context description to provide device and service recommendations.

The Context Analyzer not only interprets contextual information but also incorporates elements of Recommendation Systems (RS). Recommendation Systems utilize intelligent algorithms to analyze preferences and behaviors to provide suggestions for products, services, or content [62]. Service and device re-selection is crucial to perform workflow adaptations in response to context changes [63]. In this work, the Context Analyzer acts as a specialized RS by leveraging contextual data and quality conditions to recommend optimal devices or services that can perform workflow tasks with higher efficiency. Furthermore, this component offers an API REST interface, which makes it a detached component. This feature allows for easy integration into existing workflow management systems.

The component is built using a combination of semantic web technologies and the widely adopted MAPE-K model for autonomous systems [45]. MAPE-K-based systems are particularly suited to address exceptional situations that may arise during workflow execution [60]. In the context of smart manufacturing facilities involving robots, exceptional situations might include scenarios where a robot encounters a sudden obstacle in its path, experiences a drop in battery level, or faces unexpected environmental changes that affect its operation. These situations demand real-time adaptive responses to ensure the smooth progression of the workflow.

Furthermore, the combination of semantic web technologies and the MAPE-K model offers capabilities to effectively describe administration shells, devices, sensors, and services. The MAPE-K model is employed as the architectural reference, providing a framework for handling context data including specialized modules for capturing context data, analyzing and storing it in a knowledge base, composing a plan to improve the workflow or overcome problems, and applying adaptations to the workflow at runtime.

Fig. 2 provides an overview of the technology stack and subcomponents that comprise the Context Analyzer. This comprehensive technology stack empowers the architecture to effectively interpret and leverage contextual information, enabling intelligent recommendations, decision-making, and adaptive workflow management.

5.2.1. MAPE-K modules

The following itemization briefly explains the MAPE-K modules of Context Analyzer and their interactions with other components of the proposed architecture. Context Analyzer includes modules that gather data, analyze it, and deliver a recommendation object for service replacement. These modules make use of semantic web technologies to enable the representation and interpretation of data in a structured machine-readable format.

- **Monitor:** Comprehends a module called Context Monitor, which continuously gathers real-time data from devices and sensors at plant-level. This module makes use of AAS to gather relevant information about connectivity options to gather real-time data from assets. Context Monitor offers various connectivity options, including OPCUA, HTTP, ROS Topics, MQTT, and more. The collected data is stored in a semantic repository. It can be any semantic web repository that supports RDF, in this case GraphDB is employed. To insert the data, Context-Monitor follows the schemas defined in the DeviceServiceOnt ontology to dynamically build insert statements. This ontology provides semantic descriptions of administration shells, devices, services, and quality properties.

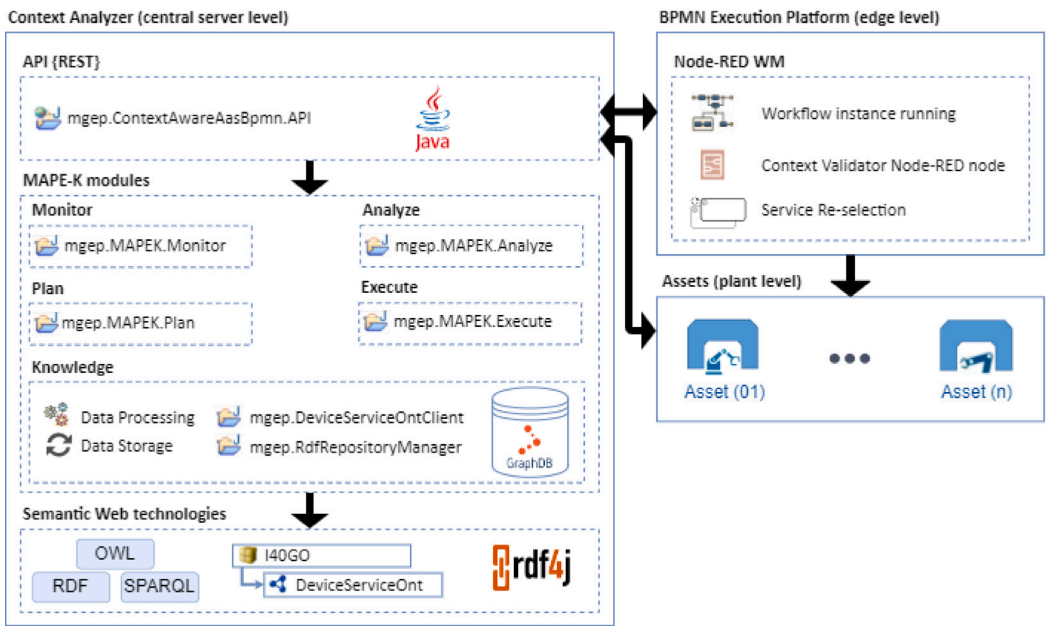


Fig. 2. Context analyzer technology stack and interaction with other components in this architecture.

- **Analyze:** Details about administration shells, assets, services, and quality conditions are received for the computation phase. SPARQL queries are built dynamically considering the DeviceServiceOnt ontology and the received quality conditions. The queries are then executed in the semantic repository. (See Algorithm 1).
- **Plan:** The resultset returned by the semantic repository is used to prepare a final response, a JSON object representing a recommendation for service replacement.
- **Execute:** The recommendation object is delivered to the API consumer, which will decide whether to accept the suggestion or not. In this proposal, Node-RED WM serves as the API consumer. Node-RED WM is designed to always accept the suggestion, resulting in service replacement during workflow runtime.
- **Knowledge:** Stores triplets about administration shells, devices/machines, services, inputs, outputs, and quality parameters using the DeviceServiceOnt ontology and GraphDB as the semantic repository. The DeviceServiceOnt ontology leverages semantic web standards, such as RDF and OWL, to provide a formal and expressive representation.

5.2.2. Semantic web integration

The Context Analyzer component incorporates semantic web technologies to enable the representation and interpretation of data in a structured and machine-readable format. Key technologies of this integration include Web Ontology Language (OWL), Resource Description Framework (RDF), and Simple Protocol and RDF Query Language (SPARQL), which collectively empower the architecture with semantic capabilities [64].

OWL is the language for ontology design that facilitates the interpretability of information by machines [65]. In this work, OWL is employed to design the DeviceServiceOnt ontology, which provides the vocabulary for describing the various entities within manufacturing processes and their relationships. RDF, on the other hand, is a data model for expressing information about resources in a graph-like format using subject–predicate–object triples [66]. In this work, RDF is employed to define and connect entities within the manufacturing workflow, such as administration shells, devices, services, inputs, outputs, and quality parameters. SPARQL serves as the query language for RDF [67], enabling the Context Analyzer to gather semantically meaningful information in real-time from the semantic repository.

Furthermore, reusing ontologies ensures interoperability and alignment with established standards, facilitating integration with existing systems and enabling semantic interoperability across domains [68]. Thereby, the design of the DeviceServiceOnt ontology, as illustrated in Fig. 3, incorporates relevant classes and properties from I40GO,[2] a global ontology for Industry 4.0 applications.

I40GO restructures and categorizes the knowledge contained in various Industry 4.0 ontologies in various layers and modules. Constructed through a fusion of MODDALS [69] and NeOn [70] methodologies, I40GO promotes compatibility across applications. It adheres to the FAIR principles — Findability, Accessibility, Interoperability, and Reusability [71]. Its modular structure segregates knowledge into various abstraction layers, each of which contains specific modules and classes. The strength of I40GO lies in its ability to unite and harmonize knowledge derived from a variety of existing ontologies, including MASON, Digital Reference, GENIAL, RAMI 4.0, I40KG, AutomationML, and OntoCAPE. This amalgamation ensures that domain-specific ontologies like DeviceServiceOnt can draw from a rich, shared resource, thus advancing knowledge representation and interoperability within the Industry 4.0 domain.

The incorporation of these semantic web technologies in the Context Analyzer component ensures a consistent and structured representation of the data, enabling dynamic querying and inference capabilities.

5.2.3. Best service/device selection algorithm

The Context Analyzer component performs the best device/service selection by executing the steps described in Algorithm 1. This algorithm takes the quality conditions into account and treats them as minimal constraints that a service/device must meet to become a candidate for selection. The quality conditions are also later used to choose the best service/device from the pool of candidates. To achieve this, the algorithm within the Context Analyzer component creates SPARQL queries based on the service name and quality conditions expressed as conditional expressions. These queries are then forwarded to the semantic repository, and the resultset is returned to the Context Analyzer for further processing.

Initially, the algorithm assigns weights to all the instances in the repository to determine which of them fully or partially satisfies the

[2] https://purl.org/i4go.

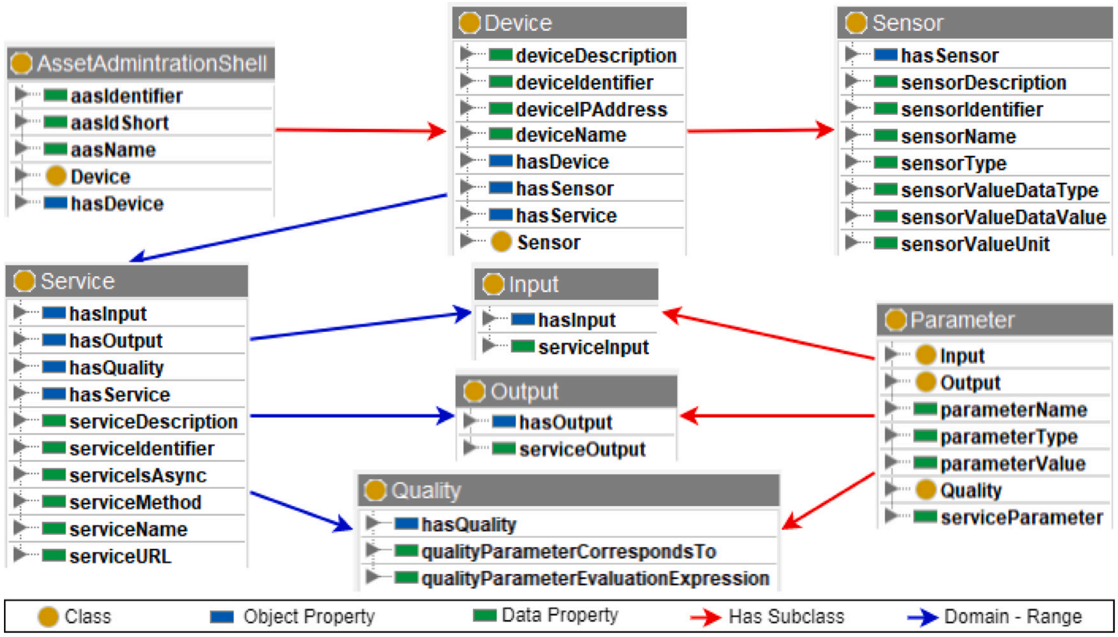


Fig. 3. DeviceServiceOnt ontology classes and relations.

quality conditions. For instance, if there are five quality conditions and one instance meets all five, the "Conditions Met Rate" is 100%. However, if an instance meets only two, the "Conditions Met Rate" is 40%.

The following enumeration delves into details on the possible scenarios that may arise in the selection process for the best device or service, these cases are influenced by the QoS-Aware service recommendation technique explained in [62,72] and are:

Case 1. When a single device/service satisfies all quality conditions, the selection process concludes by returning and recommending that instance for task execution.

Case 2. When multiple devices/services meet all quality conditions, a sorting operation is employed to identify the device/service with the optimal quality values among the pool of candidates. This entity is then recommended for task execution.

Case 3. When none of the devices/services satisfies all quality conditions but some partially meet these conditions, a sorting operation is executed. The device/service with the best quality values among the pool of candidates is recommended for task execution, accompanied by warnings.

Case 4. When none of the devices/services aligns with any quality condition, a sorting process identifies the device/service with the most favorable quality values. However, executing the task is not advisable, as it may not reach completion.

The sorting operation for determining the best service/device employs two sorting strategies: (1) "the lower the quality value, the better" and (2) "the higher the quality value, the better". These strategies are subject to the conditional symbols ($>$, $\geq$, $<$, $\leq$) provided in the conditional expression. For instance, $HUMIDITY \leq 52$ indicates that less humidity is better, while $SuccessRate > 90$ means a higher success rate is better. Thus, sorting sub-queries are built considering the priority and the conditional symbols established in the quality conditions. These sorting sub-queries are then executed accordingly in the semantic repository. Finally, the first instance within the resultset is taken and considered the best service/device.

In summary, the Context Analyzer component relies on the MAPE-K modules and semantic web technologies that work together to gather data, analyze it, and deliver a recommendation for service/device replacement. With this understanding, the next section evaluates the context-aware component and demonstrates how it can improve manufacturing efficiency.

Algorithm 1 Context Analyzer Best Service/Device Selection

```
Require: Service Name and List of Quality Conditions
  function SelectBestService(ServiceName, QualityConditions)
    resultSet ← ∅
    while resultSet = ∅ and length(QualityConditions) ≥ 1 do
      query ← buildSparqlQuery(ServiceName, QualityConditions)
      sortingQuery ← empty
      for condition ∈ QualityConditions do
        symbol ← extractConditionalSymbol(condition)
        pName ← extractPropertyName(condition)
        sortingQuery ← sortingQuery + buildSortingQuery(symbol, pName)
      end for
      query ← query + sortingQuery
      resultSet ← executeQuerySemanticRepository(query)
      if resultSet = ∅ then
        QualityConditions ← removeLastCondition(QualityConditions)
      end if
    end while
    return getFirst(resultSet)
  end function
```

6. Evaluation

This section presents the evaluation of the proposed context-aware workflow management architecture using a case study on a warehouse scenario that includes ROS-based robots. The evaluation focuses on assessing the performance of the Context Analyzer, which leverages the quality properties of robots gathered by the Context Monitor through subscriptions to ROS topics. Key metrics such as task completion rate, resource utilization, and task completion time are used to evaluate the effectiveness of the Context Analyzer in enhancing manufacturing efficiency. With these results, a discussion of the practical applicability of the architecture is conducted.

6.1. Experiment setup

The experiment was conducted in a simulated scenario using Nvidia Isaac Sim,[3] a widely adopted scalable robotics simulation application

[3] https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/overview.html.

(a) Carter V1 Front

(b) Carter V1 Back

Fig. 4. Front and back side of the Nvidia Carter V1 robot.

used by industry leaders such as Amazon Warehouse, Fraunhofer, and Festo. Nvidia Isaac Sim provides photorealistic and physically accurate virtual environments, enabling realistic evaluation of robotic systems. ROS Noetic served as the controller software for robot navigation, providing the necessary libraries. The simulation utilized a Google Cloud Virtual Machine equipped with 12 vCPUs, 28 GB of memory, an NVIDIA Tesla T4 GPU, and Ubuntu 20.04 for computational resources.

The warehouse scenario includes 10 Carter v1 robots.[4] Fig. 4 displays the front and back views of the robot used in the experiment. Carter v1 is based on a differential drive and uses a lidar sensor and a camera to perceive the world.

Visualization was enabled using RViz, a complementary ROS tool that enables the robots to perceive the world, also allowing users to monitor the state of the robots. Fig. 5 illustrates the simulation environment in both Isaac Sim and RViz, with the 10 robots positioned at their respective start positions. Two transit cones and a wet floor sign were included as obstacles for the robots.

In addition to the simulation setup, the components of the architecture described in Section 5 were properly deployed and configured. The AAS Server (Basyx) and the semantic repository (GraphDB) were installed using Docker containers on the same Google machine. Leveraging Docker containers ensured seamless deployment, while the computational capacity of the Google machine allowed smooth management of the semantic repository.

On the user's computer, the BPMN Modeling tool (Camunda Modeler with the AAS Service Discoverer plugin) and the workflow executor software (Node-RED WM) were installed. The provision of these tools enables the user to create and execute AAS-based manufacturing workflows, as well as to define quality conditions.

[4] https://docs.nvidia.com/isaac/archive/2020.2/doc/tutorials/carter_hardware.html.

The Context Analyzer component, responsible for real-time monitoring and context interpretation, was also deployed on the Google machine, adjacent to the AAS Server and semantic repository. This close proximity to the simulation allowed stable subscription to ROS topics and low-latency data processing. Thus, enabling retrieval and analysis of quality properties from the ROS-based robots in real-time. The component then stored and processed this data using semantic web technologies and the DeviceServiceOnt ontology.

6.2. Case study: Collect work order resources

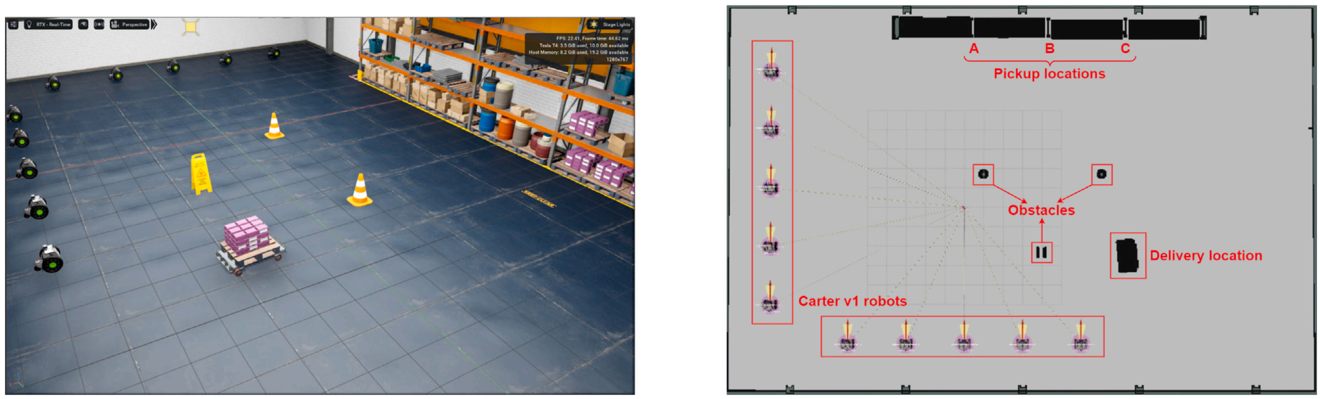
The case study was conducted using a BPMN diagram designed for a representative manufacturing scenario that involves 10 robots picking up and delivering bins. As depicted in Fig. 6, the user is required to input the work order materials, providing a list specifying the color and quantity of materials to be dispatched. A conveyor system facilitates material transportation, with bins dispatched to positions A, B, and C based on their colors (red, yellow, and blue). The Context Analyzer component is tasked with selecting one robot from the pool of 10 to pick up each dispatched bin and deliver it to the corresponding palette. This process continues until the required number of resources is dispatched.

Each robot in the simulation has different quality property values. These include *PROXIMITY_PICKUP* with values ranging from 0 m to 30 m, where 0 m represents the robot being at the pickup location and 30 m indicates a significant distance from the pickup location. The *POSITIONAL_UNCERTAINTY* quality property can contain values ranging from 0.0 to 100.0, representing the level of uncertainty the robot has about its current position. A value of 0 indicates that the robot precisely knows its location, while a value of 100 means the robot has significant uncertainty about its position, making it more likely to be lost. The *BATTERY* quality property can contain values ranging from 0% to 100%, reflecting the remaining battery capacity. And the *PAYLOAD_CAPACITY* quality property can have values ranging from 0.3 kg to 2.0 kg, indicating the maximum weight each robot can carry.

During the design phase, quality conditions are defined using the "AAS Web Service Discoverer" plugin for Camunda Modeler, as shown in Fig. 7. This plugin provides a list of available quality properties from the administration shell, allowing the user to establish the quality conditions for each task. Thus, the quality conditions established for the "Collect Bins" task are: $PROXIMITY\_PICKUP \leq 20.0 \,\&\&\, BATTERY \geq 25 \,\&\&\, POSITIONAL\_UNCERTAINTY \leq 0.90 \,\&\&\, PAYLOAD\_CAPACITY \geq 0.60$.

To provide a glimpse of the scenario and illustrate the execution of the "Collect Bins" task, Fig. 8 provides a step-by-step demonstration. Initially, in Fig. 8(a), the 10 robots are located at their start positions, while in Fig. 8(b), they are placed at random positions. Subsequently, in Fig. 8(c), the Context-Aware Workflow Manager is executed, and the Context Analyzer selects the best device to perform the Collect Bins task. The selection is based on real-time quality properties and the rules provided during design. In Figs. 8(d), 8(e), and 8(f) the selected robot executes the task.

To explain which robots could be capable of completing the designated "Collect Bins" task, a set of instances is presented in Table 3. For instance, the Carter10 robot would not be able to complete the designated task due to its Proximity of 26 m (relative to the pickup location), its Battery level of 13%, its Positional Uncertainty of 2.37, and its Payload Capacity of 0.5 kg. These values do not meet the conditions established for the completion of this job. In contrast, the Carter2 and Carter9 robots are well-equipped for the job. In this case, Context Analyzer would choose the Carter2 as the robot with the best quality values among the 10 robots, with a Proximity of 5 m, a Battery level of 96%, a Positional Uncertainty of 0.73, and a Payload Capacity of 1.9 kg, meeting the conditions established for the completion of this task.



(a) Isaac Sim: Robots at initial positions

(b) RViz: Robots at initial positions

Fig. 5. Simulation environment.

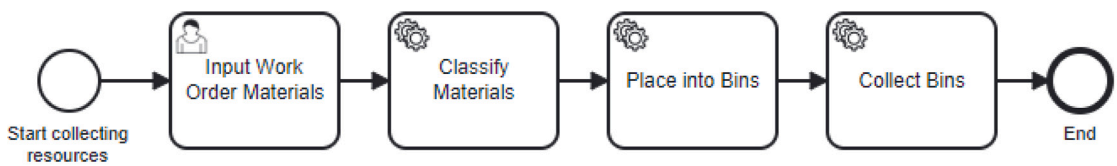


Fig. 6. Collect work order resources - BPMN diagram. This is a simplified version, the full version can be found here.

Table 3
Quality evaluation examples for the Collect Bins task.

Asset	Battery	Proximity	Positional uncertainty	Payload capacity	Num. conditions met	Conditions met rate
Carter1	75%	3 m	3.12	1.6 kg	3/4	75%
Carter2	96%	5 m	0.73	1.9 kg	4/4	100%
Carter3	20%	23 m	2.93	2.0 kg	1/4	25%
Carter4	62%	12 m	0.62	0.5 kg	3/4	75%
Carter5	14%	27 m	4.12	0.9 kg	1/4	25%
Carter6	15%	13 m	0.37	1.6 kg	3/4	75%
Carter7	2%	5 m	2.37	1.2 kg	2/4	50%
Carter8	3%	11 m	0.97	0.7 kg	2/4	50%
Carter9	42%	16 m	0.64	1.3 kg	4/4	100%
Carter10	13%	26 m	0.94	0.5 kg	0/4	0%

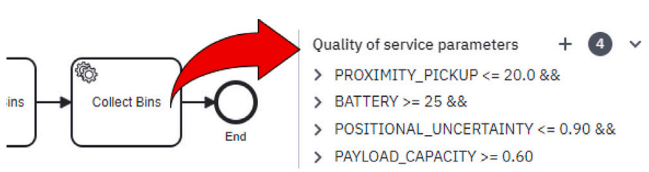


Fig. 7. Defining quality conditions using the "AAS Web Service Discoverer" plugin for Camunda Modeler during the workflow design phase.

This simulation was iterated 100 times to evaluate the performance and effectiveness of the Context Analyzer. The evaluation metrics selected align with the goals and objectives of context-aware workflow management and have been commonly employed in similar approaches, as revised in Section 3. Therefore, task completion rate, resource utilization, and task completion time were chosen as the key performance indicators to assess the performance of the Context Analyzer.

6.2.1. Reliability and responsiveness

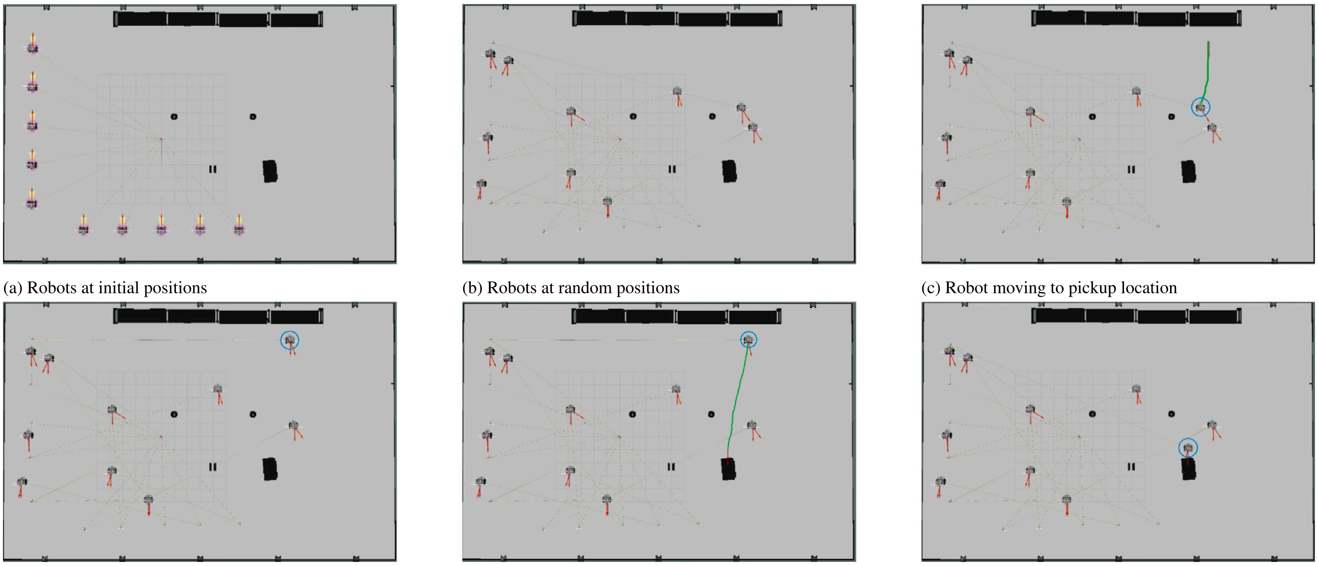
This metric evaluates the performance of the Context Analyzer by scrutinizing the task completion rate during incremental testing. The task completion rate is calculated using the formula:

$$SuccessRate = \frac{\sum_{i=1}^{n} SuccTask_i}{n} \times 100$$

In the formula, *SuccTask* represents the number of successfully completed tasks, and n denotes the total number of tasks executed. The summation symbol $\sum$ indicates the sum of the individual success rates of each task from $i = 1$ to $i = n$. The resulting value is divided by n and then multiplied by 100 to obtain the success rate as a percentage.

Fig. 9 compares the task completion rates with Context Analyzer (With CA) and without Context Analyzer (Without CA). The analysis involves comparing task completion rates starting from 5 robots up to 10 robots, and each rate is calculated using the formula described previously. Each iteration was executed 100 times to provide statistics on the reliability of the Context Analyzer. In the figure, task completion rate reflects the percentage of successfully completed tasks out of the total assigned tasks. With the integration of the Context Analyzer (With CA), the task completion rate consistently outperforms scenarios without the Context Analyzer (Without CA). Context Analyzer selects robot configurations based on context and quality conditions, leading to higher task completion rates. As the number of robots increases, the improvement in task completion rate becomes more evident. With 5 robots, the task completion rate increases from 42% without the Context Analyzer to 84% with it. Similarly, with 10 robots, the task completion rate reaches 96% with the Context Analyzer, compared to 57% without it.

In addition, the conditions met rate represents the likelihood of the selected robot configurations meeting all quality conditions required for



(a) Robots at initial positions

(b) Robots at random positions

(c) Robot moving to pickup location

(d) Robot picking up bins

(e) Robot moving to delivery location

(f) Robot delivering bins

Fig. 8. Step-by-step case study simulation in RViz.

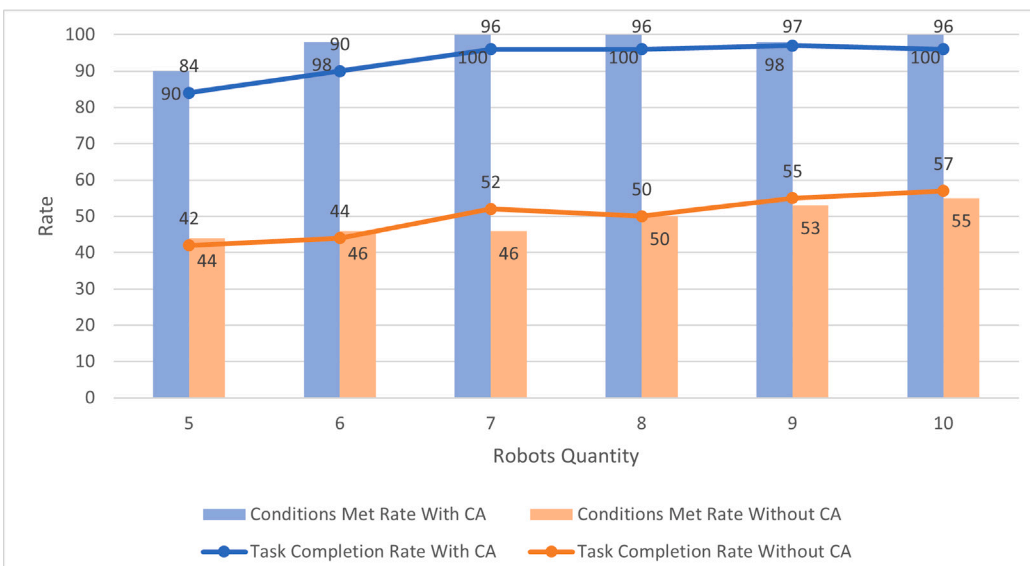


Fig. 9. Task completion rate comparison.

successful task execution. The Context Analyzer identifies robot configurations that satisfy quality conditions. With 5 robots, the conditions met rate improves from 44% without the Context Analyzer to 90% with it. Similarly, with 10 robots, the conditions met rate reaches 100% with the Context Analyzer, compared to 55% without it.

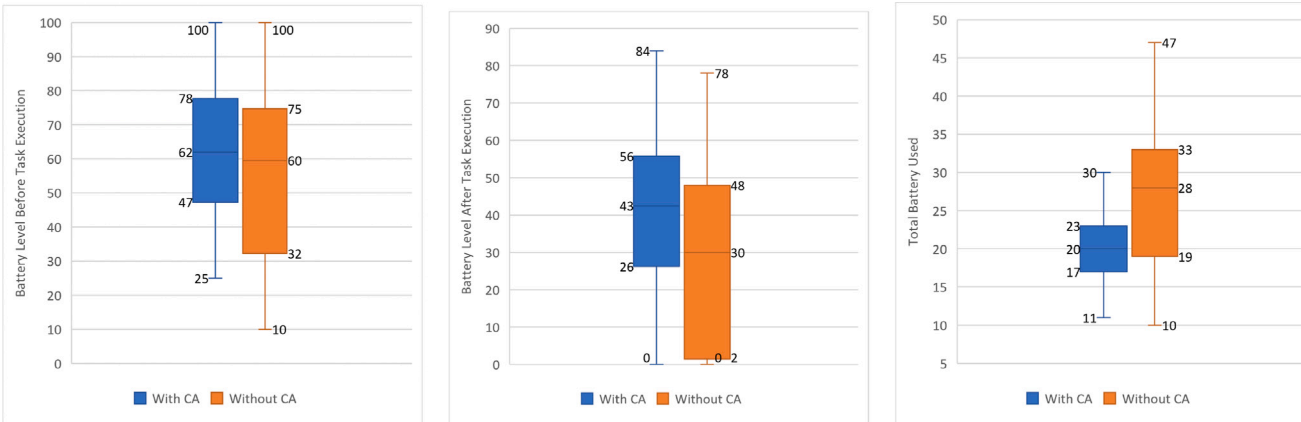
The high task completion rates and conditions met rates achieved with the Context Analyzer demonstrate its reliability and responsiveness. Context Analyzer intelligently selects devices by identifying robot configurations, based on real-time context data and quality conditions. This capability ensures a higher likelihood of successful task completion. However, it is important to note that while the Context Analyzer plays a crucial role in selecting the best device/service and enhancing task completion rates, it cannot guarantee task completion in all scenarios. Task completion ultimately depends on the conditions set during the design phase of the manufacturing process, which should align with the specific goals for task completion. Furthermore, external factors such as ineffective robot navigation may influence the outcome.

6.2.2. *Resource utilization*

This metric evaluates how efficiently the system utilizes resources, particularly in terms of battery consumption. Fig. 10 presents a comparison of the battery consumption between the robot selected by the Context-Analyzer and the robot selected randomly to complete the task.

As observed in Fig. 10(a), the Context Analyzer tends to select robots with higher battery levels for the execution of the designated task. Although the median battery levels in both cases are similar (62 With CA and 60 Without CA), a significant difference emerges in the lower quartile, with 47 for the robots selected With CA, compared to 32 Without CA. This indicates that the Context Analyzer intelligently selects robots with more charge at the start of the task.

After task completion (Fig. 10(b)), another convincing observation is that the robots chosen by the Context Analyzer maintain higher battery levels, with a lower quartile of 26 With CA, compared to only 2 Without CA. This result highlights the selection process of the Context Analyzer, which takes into account both, the battery levels and the proximity of robots to the pickup location. By including both



(a) Battery level before task execution
(b) Battery level after task execution
(c) Total battery used

Fig. 10. Energy consumption comparison.

factors in the conditional expression, the Context Analyzer ensures that the selected robots remain well-charged even after completing their assignments.

Furthermore, the devices selected by the Context Analyzer exhibit lower energy consumption compared to those chosen randomly. Fig. 10(c) depicts the median battery consumption in both cases, showing 20 for the robots selected With CA and 28 Without CA. Signifying a median on energy saving of 8%.

In summary, this metric demonstrates the capability of the Context Analyzer in efficiently utilizing battery power. The Context Analyzer contributes to improving the system performance by intelligently selecting well-charged and nearby robots. This way, optimizing energy consumption and opening up opportunities for additional task assignments.

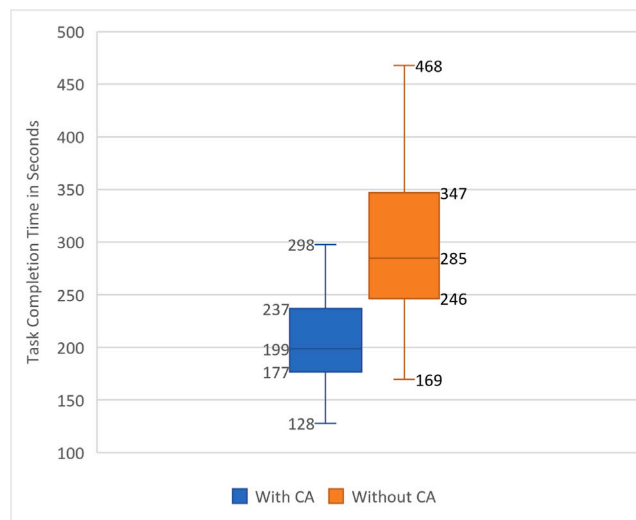
6.2.3. Task completion time

This metric quantifies the duration taken by the selected robot to complete the task, which encompasses moving to the pickup location, picking up bins, moving to the delivery location, and delivering the bins. Fig. 11 illustrates the duration, in seconds, taken by the selected robot to complete the task, including the aforementioned steps. The comparison is made between the scenarios With CA (utilizing the best device selection) and Without CA (employing random device selection).

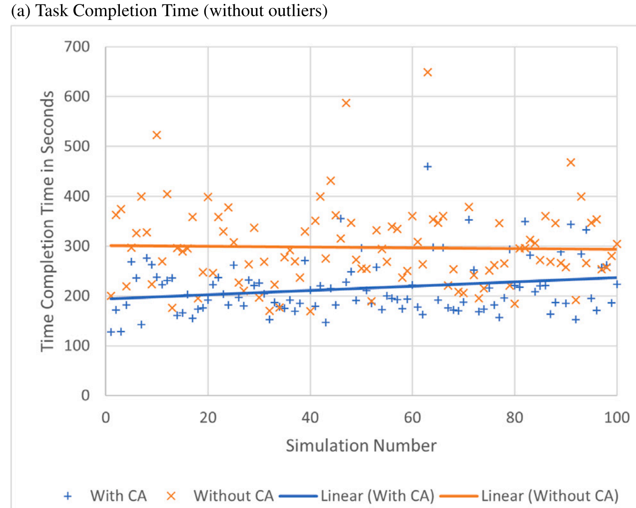
Fig. 11(a) depicts a boxplot comparison of task completion time in seconds. As seen in the figure, the devices selected by the Context Analyzer exhibit shorter task completion times. The median task completion time With CA is 199 s, a notable improvement compared to the median time of 285 s Without CA. This reduction in task completion time, averaging 30%, showcases the effectiveness of the Context Analyzer in optimizing task execution. Furthermore, the upper quartile, representing the maximum completion time is equivalent to the median completion time Without CA, with 298 and 285 s, respectively. This observation indicates a notable improvement in the worst-case scenario.

Additionally, the dispersion of the data, as depicted in Fig. 11(b), is noticeably smaller With CA compared to Without CA. This reduction in dispersion signifies a more consistent and reliable performance. The ability of the Context Analyzer to reduce task completion time variability ensures a more predictable and efficient workflow.

For the sake of replicability, detailed proofs of this experiment and the dataset can be accessed at https://github.com/MUFacultyOfEngineering/ContextAnalyzer/tree/main/SimulationProofs. In addition to the experimental results, the dataset generated from this study is also a valuable contribution. It encompasses information from 100 incremental simulations conducted with varying numbers of robots, ranging from 5 to 10. This dataset captures essential data such as



(a) Task Completion Time (without outliers)



(b) Dispersion Task Completion Time

Fig. 11. Task completion time comparison.

the positions of the robots, quality properties of all devices prior to selecting the best and random devices, the time taken by the selected device to complete the Collect Bins task, and the remaining battery level upon task termination. This dataset not only provides valuable insights into the experimental outcomes but also serves as a foundation for further research and analysis in the field of context-aware workflow management in manufacturing environments. Furthermore, a comprehensive video showcasing the steps conducted during the experiment is provided in the same repository, offering a visual representation of the methodology and procedures employed.

7. Discussion

In this section, the answers to the RQ established in Section 1 are presented. The responses serve as an opportunity to discuss the results of the testing phase and explain the advantages and features of the proposed architecture.

Answer to RQ1. "How does the proposed context-aware workflow management architecture improve operational efficiency and responsiveness within manufacturing processes?": The answer to this question encompasses how the challenges identified in Section 4 are addressed by this proposal. These challenges are:

1. **Need for real-world implementations**: To assess the effectiveness of the proposed architecture, a real-world case study scenario involving ROS-based robots was conducted. Three key metrics were employed: Task completion rate, resource utilization, and task completion time. The feasibility of the implementation was demonstrated by the results of the evaluation, indicating that operational efficiency and responsiveness in manufacturing processes are effectively enhanced by the proposed architecture.
2. **Consideration of a broader range of context variables, including both calculated and sensor-derived data**: The utilization of semantic web technologies allows for the inclusion of a wider range of context variables within the definition of quality conditions. To provide context-awareness, the architecture incorporates a robust Context Analyzer component built using semantic web technologies within a MAPE-K-based architecture. The system is granted the capability to gather and analyze context variables using a wide array of connectivity options for gathering context values from various sensors, including OPCUA, HTTP, ROS Topics, and MQTT. The flexibility in connectivity options allows for the integration of diverse sensor data sources, enhancing the accuracy and relevance of the context information used in decision-making. Context data is then stored within a flexible semantic repository, with its core being the DeviceServiceOnt domain-specific ontology. The ontology, in turn, reuses knowledge from a well-defined Industry 4.0 global ontology named I40GO.
3. **Utilization of industry-oriented standards and standardized workflow formats**: The proposal leverages industry-oriented standards, including BPMN, and uses standardized workflow formats within a decoupled architecture to improve compatibility with existing systems. The workflow modeling challenge is effectively tackled by leveraging BPMN as the formal workflow modeling language, ensuring a standardized and comprehensive representation of manufacturing workflows. The utilization of Asset Administration Shell as the Industry 4.0 standard contributes to addressing this challenge by managing the rotation of assets at plant-level. The proposal also leverages AAS in combination with BPMN offering orchestration of AAS assets over REST, thereby tackling the flexibility and adaptability required to handle the dynamic nature of modern industrial environments within a microservice-oriented architecture. A heterogeneous infrastructure scale is achieved through the utilization of Node-RED WM, allowing for seamless integration and interaction between edge devices. Node-RED WM also offers parallel and asynchronous execution capabilities, enabling concurrent execution of multiple instances of the same workflow.
4. **Need for user-centric design approaches and migration from abstract workflows to executable workflows**: One key advantage of this proposal lies in its compatibility with the widely recognized BPMN as the standard workflow format. By embracing BPMN and AAS, the architecture facilitates user-centric design by providing a familiar and intuitive environment for manufacturing workflow design with a dedicated component that offers available devices and their corresponding services within the modeler palette, enabling users to conveniently drag and drop device services into their workflows.

   Furthermore, integrating quality conditions into the architecture is a critical aspect that underscores the importance of human expertise in manufacturing process design. Quality conditions are set during the workflow design phase, enabling the inclusion of specific criteria and constraints. These conditions can incorporate sensor-derived data, such as humidity, temperature, and proximity, as well as calculated-derived data like response time, network latency, and success rate. By considering these quality conditions, the proposed architecture facilitates the selection of the most suitable device or service for executing a desired task based on the real-time context.
5. **Need for decoupled systems**: The proposal decouples components such as the context-aware module from the workflow management system, enhancing compatibility with a variety of components and systems. Another key strength of the proposed architecture is its ability to execute workflows efficiently in different computing environments. The workflow executor component is designed to run smoothly on both Central and Edge environments, requiring minimal resources. This versatility enables the architecture to adapt to various deployment scenarios, ensuring optimal performance and responsiveness in resource-constrained environments.

Answer to RQ2. "What challenges and limitations arise when integrating the proposed architecture in real-world manufacturing environments, and how can they be addressed?": The integration of the proposed architecture in a real-world manufacturing setting can present certain challenges and limitations that need to be addressed to optimize efficiency. During the experimental phase, it became evident that the conditions set during the design phase of the manufacturing process play a crucial role in determining task completion. If the quality conditions are not properly defined or aligned with the specific goals, it may not guarantee task completion in all scenarios.

To address this challenge, it is essential to develop an automatic mechanism that can identify and suggest appropriate quality conditions during the workflow design phase. This mechanism would leverage historical data, machine learning algorithms, and expert knowledge to recommend optimal quality conditions based on specific manufacturing requirements. By incorporating intelligent algorithms into the design phase, manufacturers can ensure that the quality conditions are accurately defined, leading to improved task completion rates and overall performance.

Furthermore, a potential limitation arises from the focused search space and task-by-task optimization inherent in the proposed approach. While this design enhances efficiency for the current task at hand, it may pose challenges in scenarios where inter-task dependencies or global optimization across the entire workflow are critical. This could limit the adaptability of the approach in workflows with intricate dependencies and complex interactions among a large number of assets and tasks. To address this limitation, future iterations of the proposed approach could explore the integration of optimization algorithms commonly utilized in QoS-based scheduling approaches. By incorporating such algorithms, the system could extend its scope to consider global workflow dynamics.

Overall, the proposed architecture stands out as a comprehensive and adaptable solution for context-aware workflow management in the manufacturing industry. Its compatibility with BPMN and AAS, seamless service discovery, diverse connectivity options, efficient execution, integration of quality conditions, real-time context analysis, and favorable evaluation results collectively contribute to its effectiveness and viability.

8. Conclusions

This paper presented a context-aware workflow management solution that addresses industry-related challenges and incorporates state-of-the-art advancements. This solution optimizes manufacturing processes by leveraging BPMN as the standard for workflow design, Asset Administration Shell for asset representation, and a Context Analyzer Component for real-time context interpretation.

The experimentation and evaluation phase have demonstrated the effectiveness and efficiency of the proposed solution in improving workflow management in the Industry 4.0 domain. The context-awareness component provides intelligent service selection and dynamic workflow adaptation, resulting in significant improvements in task completion time, resource utilization, task completion rate, and overall manufacturing efficiency. The utilization of semantic web technologies and the MAPE-K model has played a pivotal role in enabling context interpretation and successful service selection during runtime, effectively incorporating context-awareness into the architecture.

Furthermore, the architecture offers flexibility by providing decoupled components, making it scalable and applicable to diverse workflow systems and company configurations. The Context Analyzer, in particular, empowers users to define rules that apply to diverse industrial processes with varying quality conditions, making it adaptable to different manufacturing scenarios. These capabilities of this solution empower organizations to tailor the workflow management system to their specific needs, optimizing the utilization of resources and services based on unique quality criteria.

In future work, we aim to enhance the Context Analyzer component by integrating a notification mechanism for cases where tasks cannot be completed due to the unavailability of devices or services meeting the required quality conditions. Furthermore, we plan to explore the integration of optimization algorithms commonly utilized in QoS-based scheduling approaches to extend the scope of the system. This enhancement will enable our approach to consider global workflow dynamics, addressing the challenge posed by the focused search space and task-by-task optimization. Additionally, we intend to develop an automatic mechanism for identifying and suggesting appropriate quality conditions during the design phase. These combined advancements will reinforce the practical applicability and benefits of this solution, facilitating its adoption and contributing to the improvement of manufacturing operations.

CRediT authorship contribution statement

William Ochoa: Investigation, Methodology, Writing – original draft, Writing – review & editing, Software. **Jon Legaristi:** Investigation, Writing – review & editing, Software. **Felix Larrinaga:** Conceptualization, Supervision, Software. **Alain Pérez:** Conceptualization, Supervision, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The source code and datasets utilized for evaluation phase of this proposal have been provided in the manuscript as a link to GitHub.

Acknowledgments

This research work has been funded by the Department of Education, Universities and Research of the Basque Government under the SIIRSE project (KK-2022/00007) and the Ikerketa Taldeak program (IT-IT1519-22).

References

[1] R.F. Da Silva, R. Filgueira, I. Pietri, M. Jiang, R. Sakellariou, E. Deelman, A characterization of workflow management systems for extreme-scale applications, Future Gener. Comput. Syst. 75 (2017) 228–238.

[2] S. Jeschke, C. Brecher, T. Meisen, D. Özdemir, T. Eschert, Industrial internet of things and cyber manufacturing systems, in: Industrial Internet of Things: Cybermanufacturing Systems, Springer International Publishing, Cham, 2017, pp. 3–19, http://dx.doi.org/10.1007/978-3-319-42559-7_1, Ch. 3.

[3] M. Moghaddam, M.N. Cadavid, C.R. Kenley, A.V. Deshmukh, Reference architectures for smart manufacturing: A critical review, J. Manuf. Syst. 49 (2018) 215–225, http://dx.doi.org/10.1016/j.jmsy.2018.10.006.

[4] H. Yang, S. Kumara, S.T. Bukkapatnam, F. Tsung, The internet of things for smart manufacturing: A review, IISE Trans. 51 (11) (2019) 1190–1216.

[5] P. Marcon, C. Diedrich, F. Zezulka, T. Schröder, A. Belyaev, J. Arm, T. Benesl, Z. Bradac, I. Vesely, The asset administration shell of operator in the platform of industry 4.0, in: 2018 18th International Conference on Mechatronics-Mechatronika (Me), IEEE, 2018, pp. 1–5.

[6] T.N. BV, J. van Veelen, Productive4. 0, in: Productive4.0 Project Consortium, 2017, 2022.

[7] OMG, Business Process Management (BPM), 2022, URL https://www.omg.org/bpm/.

[8] J. Davis, T. Edgar, R. Graybill, P. Korambath, B. Schott, D. Swink, J. Wang, J. Wetzel, Smart manufacturing, Annu. Rev. Chem. Biomol. Eng. 6 (2015) 141–160.

[9] S. Bader, E. Barnstedt, H. Bedenbender, B. Berres, M. Billmann, M. Ristin, Details of the Asset Administration Shell-Part 1: the Exchange of Information Between Partners in the Value Chain of Industrie 4.0 (Version 3.0 RC02), White paper, Federal Ministry for Economic Affairs and Climate Action (BMWK), 2022.

[10] K. Boloor, R. Chirkova, Y. Viniotis, T. Salo, Dynamic request allocation and scheduling for context aware applications subject to a percentile response time SLA in a distributed cloud, in: 2010 IEEE Second International Conference on Cloud Computing Technology and Science, IEEE, 2010, pp. 464–472.

[11] Y. Chen, P. Yu, Z. Zheng, J. Shen, M. Guo, Modeling feature interactions for context-aware QoS prediction of IoT services, Future Gener. Comput. Syst. 137 (2022) 173–185, http://dx.doi.org/10.1016/j.future.2022.07.017, URL https://www.sciencedirect.com/science/article/pii/S0167739X2200245X.

[12] A. Carrera-Rivera, F. Larrinaga, G. Lasa, Context-awareness for the design of Smart-product service systems: Literature review, Comput. Ind. 142 (2022) 103730, http://dx.doi.org/10.1016/j.compind.2022.103730, URL https://www.sciencedirect.com/science/article/pii/S0166361522001270.

[13] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Systematic literature reviews, in: Experimentation in Software Engineering, Springer, 2012, pp. 45–54.

[14] J.G. Represa, F. Larrinaga, P. Varga, W. Ochoa, A. Perez, D. Kozma, J. Delsing, Investigation of microservice-based workflow management solutions for industrial automation, Appl. Sci. 13 (3) (2023) http://dx.doi.org/10.3390/app13031835.

[15] S. Ivanov, A. Kalenkova, Comparing process models in the BPMN 2.0 XML format, in: Proceedings of the Spring/Summer Young Researchers' Colloquium on Software Engineering, Vol. 27, 2015, pp. 255–266.

[16] OMG, Business process model and notation (BPMN), 2023, URL https://www.omg.org/spec/BPMN/2.0/PDF.

[17] L.M. Vaquero, F. Cuadrado, Y. Elkhatib, J. Bernal-Bernabe, S.N. Srirama, M.F. Zhani, Research challenges in nextgen service orchestration, Future Gener. Comput. Syst. 90 (2019) 20–38, http://dx.doi.org/10.1016/j.future.2018.07.039.

[18] Y. Jiang, Z. Huang, D.H.K. Tsang, Challenges and solutions in fog computing orchestration, IEEE Netw. 32 (3) (2018) 122–129, http://dx.doi.org/10.1109/MNET.2017.1700271.

[19] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, M. Rovatsos, Fog orchestration for internet of things services, IEEE Internet Comput. 21 (2) (2017) 16–24, http://dx.doi.org/10.1109/MIC.2017.36.

[20] D. Hollingsworth, U. Hampshire, Workflow Management Coalition: The Workflow Reference Model, Vol. 19, Document Number TC00-1003, 1995, p. 224, (16).

[21] S. Du, P. Wu, G. Wu, C. Yao, L. Zhang, The collaborative system workflow management of industrial design based on hierarchical colored petri-net, IEEE Access 6 (2018) 27383–27391.

[22] M. Tanaka, O. Tatebe, Pwrake: A parallel and distributed flexible workflow management tool for wide-area data intensive computing, in: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, 2010, pp. 356–359.

[23] P. Zhao, L. Di, G. Yu, Building asynchronous geospatial processing workflows with web services, Comput. Geosci. 39 (2012) 34–41.

[24] C. Rotsos, D. King, A. Farshad, J. Bird, L. Fawcett, N. Georgalas, M. Gunkel, K. Shiomoto, A. Wang, A. Mauthe, N. Race, D. Hutchison, Network service orchestration standardization: A technology survey, Comput. Stand. Interfaces 54 (2017) 203–215, http://dx.doi.org/10.1016/j.csi.2016.12.006, SI: Standardization SDN & NFV.

[25] L. Velasco, A. Castro, D. King, O. Gerstel, R. Casellas, V. Lopez, In-operation network planning, IEEE Commun. Mag. 52 (1) (2014) 52–60, http://dx.doi.org/10.1109/MCOM.2014.6710064.

[26] S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A.C. Arpaci-Dusseau, R.H. Arpaci-Dusseau, Serverless computation with OpenLambda, in: Proceedings of the 8th USENIX Conference on Hot Topics in Cloud Computing, HotCloud '16, USENIX Association, USA, 2016, pp. 33–39.

[27] G. Alférez, V. Pelechano, Achieving autonomic Web service compositions with models at runtime, Comput. Electr. Eng. 63 (2017) http://dx.doi.org/10.1016/j.compeleceng.2017.08.004.

[28] H. Kir, N. Erdogan, A knowledge-intensive adaptive business process management framework, Inf. Syst. 95 (2021) 101639, http://dx.doi.org/10.1016/j.is.2020.101639.

[29] U. Arul, S. Prakash, Toward automatic web service composition based on multilevel workflow orchestration and semantic web service discovery, Int. J. Bus. Inf. Syst. 34 (1) (2020) 128–156.

[30] M. Lyu, H. Benfenatki, F. Biennier, P. Ghodous, Control as a service architecture to support context-aware control application development, IFAC-PapersOnLine 52 (2019) 1085–1090, http://dx.doi.org/10.1016/j.ifacol.2019.11.340.

[31] R. Song, J. Vanthienen, W. Cui, Y. Wang, L. Huang, Context-aware BPM using IoT-integrated context ontologies and IoT-enhanced decision models, in: 2019 IEEE 21st Conference on Business Informatics (CBI), Vol. 1, IEEE, 2019, pp. 541–550.

[32] L. Mazzola, P. Waibel, P. Kaphanke, M. Klusch, Smart process optimization and adaptive execution with semantic services in cloud manufacturing, Information 9 (11) (2018) 279.

[33] A. Bucchiarone, A. Marconi, M. Pistore, H. Raik, A context-aware framework for dynamic composition of process fragments in the internet of services, J. Internet Serv. Appl. 8 (1) (2017) 1–23.

[34] T. Baker, M. Asim, H. Tawfik, B. Aldawsari, R. Buyya, An energy-aware service composition algorithm for multiple cloud-based IoT applications, J. Netw. Comput. Appl. 89 (2017) 96–108.

[35] A. Bekkouche, S.M. Benslimane, M. Huchard, C. Tibermacine, F. Hadjila, M. Merzoug, QoS-aware optimal and automated semantic web service composition with user's constraints, Serv. Orient. Comput. Appl. 11 (2) (2017) 183–201.

[36] P. Valderas, V. Torres, E. Serral, Modelling and executing IoT-enhanced business processes through BPMN and microservices, J. Syst. Softw. 184 (2022) 111139, http://dx.doi.org/10.1016/j.jss.2021.111139.

[37] L. Malburg, M. Hoffmann, R. Bergmann, Applying MAPE-K control loops for adaptive workflow management in smart factories, J. Intell. Inf. Syst. (2023) 1–29.

[38] W. Ochoa, F. Larrinaga, A. Pérez, Context-aware workflow management for smart manufacturing: A literature review of semantic web-based approaches, Future Gener. Comput. Syst. (2023) http://dx.doi.org/10.1016/j.future.2023.03.017, URL https://www.sciencedirect.com/science/article/pii/S0167739X23000948.

[39] H. Nguyen Ngoc, G. Lasa, I. Iriarte, Human-centred design in industry 4.0: case study review and opportunities for future research, J. Intell. Manuf. 33 (1) (2022) 35–76.

[40] C.O. Klingenberg, M.A.V. Borges, J.A.V. Antunes Jr., Industry 4.0 as a data-driven paradigm: a systematic literature review on technologies, J. Manuf. Technol. Manag. 32 (3) (2021) 570–592.

[41] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P.J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny, K. Wenger, Pegasus, a workflow management system for science automation, Future Gener. Comput. Syst. 46 (2015) 17–35, http://dx.doi.org/10.1016/j.future.2014.10.008, URL https://www.sciencedirect.com/science/article/pii/S0167739X14002015.

[42] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: Recommender Systems Handbook, Springer, 2010, pp. 217–253.

[43] C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, Context aware computing for the internet of things: A survey, IEEE Commun. Surv. Tutor. 16 (1) (2013) 414–454.

[44] M. Junghans, S. Agarwal, Web service discovery based on unified view on functional and non-functional properties, in: 2010 IEEE Fourth International Conference on Semantic Computing, IEEE, 2010, pp. 224–227.

[45] IBM, An architectural blueprint for autonomic computing, 2006, URL https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf.

[46] Z. Ahmad, B. Nazir, A. Umer, A fault-tolerant workflow management system with Quality-of-Service-aware scheduling for scientific workflows in cloud computing, Int. J. Commun. Syst. 34 (1) (2021) e4649.

[47] F.U. Ambursa, R. Latip, A. Abdullah, S. Subramaniam, A particle swarm optimization and min–max-based workflow scheduling algorithm with QoS satisfaction for service-oriented grids, J. Supercomput. 73 (2017) 2018–2051.

[48] N. Sharma, P. Garg, et al., Ant colony based optimization model for QoS-Based task scheduling in cloud computing environment, Meas.: Sens. 24 (2022) 100531.

[49] J. Yu, R. Buyya, C.K. Tham, et al., QoS-based scheduling of workflow applications on service grids, in: Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing (e-Science 2005, IEEE CS Press, Los Alamitos, CA, USA), Citeseer, 2005, pp. 5–8.

[50] P.I. 4.0, Submodel templates of the asset administration shell - generic frame for technical data for industrial equipment in manufacturing, 2022, URL https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Submodel_templates-Asset_Administration_Shell-Technical_Data.html.

[51] P.I. 4.0, Submodel templates of the asset administration shell - ZVEI digital nameplate for industrial equipment, 2022, URL https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Submodel_templates-Asset_Administration_Shell-digital_nameplate.html.

[52] P.I. 4.0, Information model for capabilities, skills & services, 2023, URL https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/CapabilitiesSkillsServices.html.

[53] W. Ochoa, F. Larrinaga, A. Pérez, Architecture for managing AAS-based business processes, Procedia Comput. Sci. 217 (2023) 217–226, http://dx.doi.org/10.1016/j.procs.2022.12.217, 4th International Conference on Industry 4.0 and Smart Manufacturing.

[54] P.I. 4.0, Details of the asset administration shell part 1, 2021, URL https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html.

[55] Basyx, BaSyx, 2022, URL https://wiki.eclipse.org/BaSyx.

[56] G. Di Orio, P. Maló, J. Barata, NOVAAS: A reference implementation of industrie4. 0 asset administration shell with best-of-breed practices from IT engineering, in: IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society, Vol. 1, IEEE, 2019, pp. 5505–5512.

[57] Z.D. Elektroindutrie, The reference architectural model RAMI 4.0 and the Industrie 4.0 component, 2018.

[58] M. Platenius-Mohr, S. Malakuti, S. Grüner, J. Schmitt, T. Goldschmidt, File- and API-based interoperability of digital twins by model transformation: An IIoT case study using asset administration shell, Future Gener. Comput. Syst. 113 (2020) 94–105, http://dx.doi.org/10.1016/j.future.2020.07.004, URL https://www.sciencedirect.com/science/article/pii/S0167739X20302600.

[59] F. Larrinaga, W. Ochoa, A. Perez, J. Cuenca, J. Legaristi, M. Illarramendi, Node-RED workflow manager for edge service orchestration, in: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2022, pp. 1–6.

[60] T.A. Nguyen, M. Aiello, T. Yonezawa, K. Tei, A self-healing framework for online sensor data, in: 2015 IEEE International Conference on Autonomic Computing, 2015, p. 1, http://dx.doi.org/10.1109/ICAC.2015.61.

[61] K. Hribernik, G. Cabri, F. Mandreoli, G. Mentzas, Autonomous, context-aware, adaptive Digital Twins—State of the art and roadmap, Comput. Ind. 133 (2021) 103508.

[62] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, Recommender Systems: An Introduction, Cambridge University Press, 2010.

[63] H. Gao, W. Huang, X. Yang, Y. Duan, Y. Yin, Toward service selection for workflow reconfiguration: An interface-based computing solution, Future Gener. Comput. Syst. 87 (2018) 298–311.

[64] B. DuCharme, Learning SPARQL: Querying and Updating with SPARQL 1.1, " O'Reilly Media, Inc.", 2013.

[65] D.L. McGuinness, F. Van Harmelen, et al., OWL web ontology language overview, W3C Recomm. 10 (10) (2004) 2004.

[66] F. Manola, E. Miller, B. McBride, et al., RDF primer, W3C Recomm. 10 (1–107) (2004) 6.

[67] J. Pérez, M. Arenas, C. Gutierrez, Semantics and complexity of SPARQL, ACM Trans. Database Syst. 34 (3) (2009) 1–45.

[68] E. Simperl, Reusing ontologies on the Semantic Web: A feasibility study, Data Knowl. Eng. 68 (10) (2009) 905–925.

[69] J. Cuenca, F. Larrinaga, E. Curry, MODDALS methodology for designing layered ontology structures, Appl. Ontol. 15 (2) (2020) 185–217, http://dx.doi.org/10.3233/AO-200225, URL http://www.edwardcurry.org/publications/MODDALS.pdf.

[70] M.C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, The NeOn Methodology for Ontology Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 9–34, http://dx.doi.org/10.1007/978-3-642-24794-1_2, Ch. -.

[71] M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L.B. da Silva Santos, P.E. Bourne, et al., The FAIR Guiding Principles for scientific data management and stewardship, Sci. Data 3 (1) (2016) 1–9.

[72] P. Wang, K.-M. Chao, C.-C. Lo, On optimal decision for QoS-aware composite service selection, Expert Syst. Appl. 37 (1) (2010) 440–449, http://dx.doi.org/10.1016/j.eswa.2009.05.070, URL https://www.sciencedirect.com/science/article/pii/S0957417409004898.

William Ochoa A. is a Ph.D. student in the information systems department of MU-EPS S.coop. on the subject of semantic enhanced business processes in Industry 4.0. He obtained a B.S. degree in computer systems engineering (2014) from the State University of Guayaquil (Ecuador) and obtained an M.Sc. degree in Computer Science (2017) from Eötvös Loránd University (Hungary). Since 2012 he has worked on the development of Enterprise Resource Planning (ERP) systems. His main areas of interest are ERP development, web engineering, semantic web, and business process management systems.

Jon Legaristi Labajos is a graduate in Computer Engineering from Mondragon University, having completed his studies in 2022. He is currently finishing a Master's Degree in Data Analysis, Cybersecurity, and Cloud Computing, which is expected to be completed at the beginning of 2024. Since 2020, Jon has been actively involved as a research assistant in the Department of Electronics and Informatics, within the Software and Systems Engineering research group. His work focuses on various aspects of engineering, including web development, interoperability, and Industry 4.0.

Dr Felix Larrinaga was born in Markina-Xemein (Bizkaia) Spain in 1965. He obtained his Ph.D. in Mobile Communications at Staffordshire University (UK) in 1999. Previously, he finished the B.Sc. (Hons) in Industrial Electronics at Mondragon University in 1995. Since 2000, he combines lecturing and research at Mondragon University with an extensive experience in industrial work as a project leader. He is currently a lecturer and a researcher at Mondragon University (Basque Country, Spain). He has actively participated in the creation of the new research team on Web Engineering at the Faculty of Engineering. The research activities of the group are in the areas of interoperability, web architecture, microservices, and semantic web technologies. The team works on multidisciplinary projects with other universities, technological centers, and enterprises in areas such as energy, health, or industry. At the moment, Dr. Larrinaga is working on several projects related to manufacturing, smart cities, and energy.

Alain Perez Riaño [Ph.D.] is a *Lecturer & Researcher* in the IT department of MU-EPS, a *Computer Science Engineer* (2010), and a *Ph.D.* in Computer Science with the thesis "*Semantic Web and Semantic Technologies to Enhance Innovation and Technology Watch Process*", also by MGEP-MU (2016).Since 2009 he has worked on research projects related to *Interoperability, IoT, Smart Cities, and Semantic Web*, which stand out: *Cityfied*, *Rennovates,* and *SmartEnCity*.