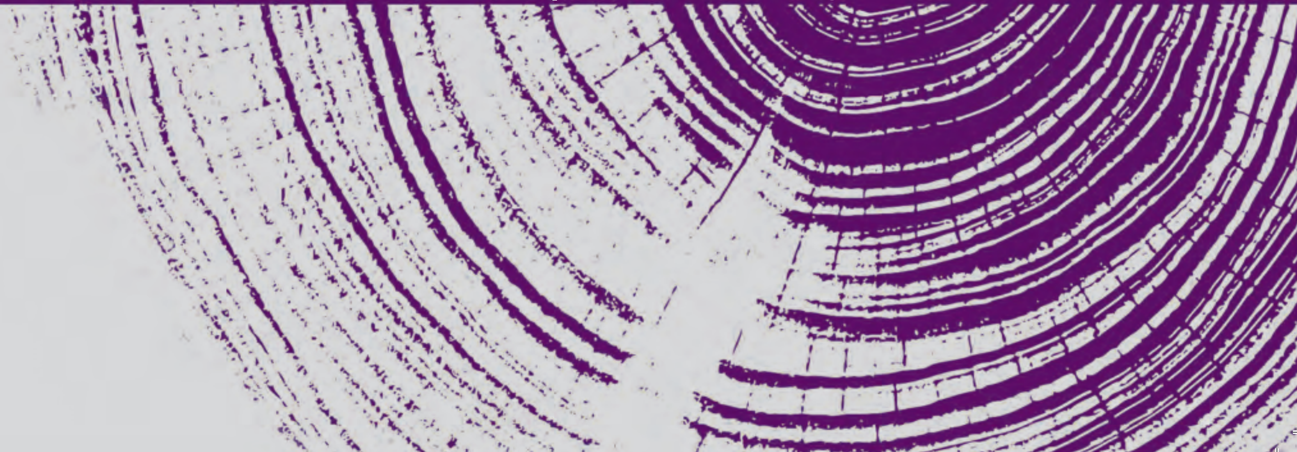
Mondragon
Unibertsitatea

DOCTORAL THESIS

ENHANCING FLEXIBILITY IN INDUSTRY 4.0 WORKFLOWS: A CONTEXT-AWARE ARCHITECTURE FOR DYNAMIC SERVICE ORCHESTRATION

WILLIAM STEVEN OCHOA AGURTO | Arrasate-Mondragón, 2024



Mondragon Unibertsitatea Goi Eskola Politeknikoa

ENHANCING FLEXIBILITY IN INDUSTRY 4.0 WORKFLOWS: A CONTEXT-AWARE ARCHITECTURE FOR DYNAMIC SERVICE ORCHESTRATION

WILLIAM OCHOA AGURTO

2024

PHD THESIS

DOCTOR IN COMPUTER ENGINEERING

THESIS PROJECT TITLE:

ENHANCING FLEXIBILITY IN INDUSTRY 4.0 WORKFLOWS: A CONTEXT-AWARE ARCHITECTURE FOR DYNAMIC SERVICE ORCHESTRATION

Author

WILLIAM OCHOA AGURTO

Supervisors

FÉLIX LARRINAGA BARRENECHEA

ALAIN PÉREZ RIAÑO

ARRASATE 2024

Declaration

Hereby I, William Ochoa Agurto declare, that this doctoral thesis is my original authorial work, which I have worked out by my own. All sources, references and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Yo William Ochoa Agurto declaro que esta tésis doctoral es original, fruto de mi trabajo personal, y que no ha sido previamente presentado para obtener otro título o calificación profesional. Las ideas, formulaciones, imágenes, ilustraciones tomadas de fuentes ajenas han sido debidamente citadas y referenciadas.

Arrasate, 2024

William Ochoa Agurto

Acknowledgments

I would like to express my sincere gratitude to the Mondragon Goi Eskola Politeknikoa (MGEP) for providing valuable opportunities and dedicated resources for this doctoral study. I sincerely thank the MGEP team with special appreciation to my esteemed directors Félix Larrinaga and Alain Pérez. Your expertise, guidance, and support have been invaluable in this journey.

I am deeply grateful to my wife, Ángela Carrera, for her unwavering support, understanding, and encouragement. Her belief in my abilities has been my driving force, and I appreciate her faithful partnership in this challenging yet rewarding journey. I would also like to express my heartfelt gratitude to my parents, Manuel Ochoa Ponce and Betty Agurto Santillán, for their support, encouragement, and sacrifices. The values and principles they instilled in me were inspiring, and I am indebted to them. I sincerely thank my brothers and sisters, whose encouragement and support have sustained me throughout this learning journey. The unconditional support of my family has been an integral part of my success, and I feel fortunate to have each and every one of them by my side.

Finally, I am grateful for the friends and colleagues I have had the opportunity to work with on campus. Your friendship, wisdom and cooperative spirit have enriched my learning experience and helped me to complete this thesis.

Abstract

The Industry 4.0 era is reshaping the manufacturing landscape, fostering highly flexible processes that swiftly adapt to operational requirements, situational events, and environmental factors. To achieve this, it is crucial to orchestrate manufacturing operations while adhering to Industry 4.0 specifications. In pursuit of standardization, the Asset Administration Shell (AAS) emerges to provide a digital representation of physical assets. Concurrently, Workflow Management, with the Business Process Model and Notation (BPMN) language, offers an approach for coordinating tasks using specialized tools.

However, achieving flexible and adaptable workflows remains a challenge. Systems with context-awareness capabilities interpret real-time data and reconfigure themselves to minimize delays and improve overall efficiency. Addressing this challenge, the Semantic Web emerges as an approach to translate raw data into semantically enriched information, enabling intelligent decisions based on real-time contextual information.

This doctoral thesis presents an industry 4.0 architecture for context-aware workflow management that leverages recent advances in the state-of-the-art and addresses existing industry-related challenges. The architecture incorporates decoupled components that: 1) Integrate AAS for asset representation, BPMN for workflow notation, and a service discovery mechanism, offering a flexible tool for manufacturing process design. 2) Operate on central servers and edge-embedded systems with minimal resource consumption. And, 3) Provide a context-aware component for runtime workflow reconfiguration, enhancing the dynamism of manufacturing operations.

Furthermore, the architecture undergoes validation through several experiments, demonstrating effective context-awareness in dynamic service re-selection during workflow execution. Results show improvements in task completion time, task completion rate, and energy utilization. Additionally, this thesis highlights the flexibility of the architecture by implementing it in diverse manufacturing scenarios without altering individual components or changing the semantic repository structure. This work establishes a foundation for efficient context-aware workflow management in Industry 4.0, offering a versatile solution for future manufacturing.

Resumen

La era de la Industria 4.0 está remodelando al panorama manufacturero, fomentando procesos altamente flexibles que se adaptan rápidamente a requisitos operativos, eventos situacionales y factores ambientales. Para lograr esto, es crucial orquestar las operaciones manufactureras cumpliendo con las especificaciones de la Industria 4.0. En busca de estandarización, emerge el concepto de Asset Administration Shell (AAS) que proporciona una representación digital de activos físicos. Concurrentemente, la Gestión de Flujos de Trabajo, con el lenguaje Business Process Model and Notation (BPMN), ofrece un enfoque para coordinar tareas mediante herramientas especializadas.

Sin embargo, lograr flujos de trabajo flexibles y adaptables sigue siendo un desafío. Los sistemas con capacidades de conocimiento del contexto interpretan datos en tiempo real y se reconfiguran para minimizar retrasos y mejorar la eficiencia general. Para abordar este desafío, emerge la Web Semántica como un enfoque que permite traducir datos crudos en información enriquecida semánticamente, permitiendo decisiones inteligentes basadas en información contextual en tiempo real.

Esta tesis doctoral presenta una arquitectura de Industria 4.0 con capacidad de conocimiento del contexto para la gestión de flujos de trabajo. La propuesta aprovecha los avances recientes en el estado del arte y aborda desafíos existentes en la industria. La arquitectura incorpora componentes desacoplados que: 1) Integran AAS para la representación de activos, BPMN para la notación de flujos de trabajo y un mecanismo de descubrimiento de servicios, ofreciendo una herramienta flexible para el diseño de procesos de fabricación. 2) Operan en servidores centrales y sistemas embebidos en el edge con un mínimo consumo de recursos. Y, 3) Proporcionan un componente consciente del contexto que recopila datos en tiempo real y reconfigura en tiempo de ejecución los flujos de trabajo, mejorando la flexibilidad de las operaciones de fabricación.

Además, la arquitectura se valida a través de varios experimentos, demostrando una capacidad efectiva de conocimiento del contexto para la reselección dinámica de servicios durante la ejecución de flujos de trabajo. Los resultados muestran mejoras en el tiempo de finalización de tareas, la tasa de finalización de tareas y la utilización de energía. Además, esta tesis destaca la flexibilidad de la arquitectura al implementarla en diversos escenarios de fabricación sin alterar componentes individuales ni cambiar la estructura del repositorio semántico. Este trabajo establece una base para la eficiente gestión de flujos de trabajo conscientes del contexto en la Industria 4.0, ofreciendo una solución versátil para la manufactura del futuro.

Laburpena

4.0 Industriaren aroa fabrikazioaren panorama birmoldatzen ari da, eta prozesu oso malguak sustatzen ditu, eskakizun operatiboetara, egoera-gertaeretara eta ingurumen-faktoreetara azkar egokitzen direnak. Hori lortzeko, funtsezkoa da fabrikazio-eragiketak orkestratzea, 4.0 Industriaren zehaztapenak betez. Normalizazioaren bila, Asset Administration Shell-ek (AAS) aktibo fisikoen irudikapen digitalak sortzen ditu. Aldi berean, Lan-Fluxuen Kudeaketan, Business Process Model and Notation (BPMN) hizkuntzarekin, zereginak koordinatzeko ikuspegi bat ahalbidetzen da tresna espezializatuak erabiliz.

Hala ere, lan-fluxu malguak eta moldagarriak lortzea erronka izaten jarraitzen du. Testuingurua ezagutzeko gaitasunak dituzten sistemek denbora errealeko datuak interpretatzen dituzte eta berriro konfiguratzen dituzte atzerapenak minimizatzeko eta eraginkortasun orokorra hobetzeko. Erronka horri aurre egiteko, Web Semantikoak datu gordinak semantikoki aberastutako jakintzan bihurtzen ditu, denbora errealeko testuinguruko informazioan oinarritutako erabaki adimentsuak ahalbidetuz.

Doktorego tesi honek 4.0 Industriaren arkitektura bat aurkezten du, lan-fluxuak era adimentsu batean kudeatu ditzakeena testuingurua ulertuz. Arkitekturak osagai desakoplatuak ditu helburu hauek lortzeko: 1) AAS aktiboen irudikatzaileak, BPMN lan-fluxuaren notazioak eta zerbitzuen aurkikuntza-mekanismo bat integratzea, fabrikazio-prozesuen diseinurako tresna malgu bat eskaintzeko. 2) baliabide gutxiko kontxumoa duenez, zerbitzari zentraletan eta ertzean txertatutako sistemetan funtzionatzea. Eta, 3) exekuzio-denborako lan-fluxua birkonfiguratu ahal izatea, testuingurua ezagutzen duen konponente baten bitartez, fabrikazio-eragiketen dinamismoa hobetzeko.

Gainera, arkitektura, hainbat esperimenturen bidez testuinguruaren jakintzaren eraginkortasuna balioztatzen du. Emaitzek hobekuntzak erakusten dituzte zereginak burutzeko denboran, zereginen burutze tasan eta energiaren erabileran. Arkitekturaren malgutasuna nabarmentzen da ere, fabrikazio-eszenatoki ezberdinetara moldatuz, konponenteak edo biltegi semantikoa egitura aldatu gabe. Lan honek 4.0 Industria testuinguruan oinarritutako lan-fluxuen

kudeaketa eraginkorraren oinarriak ezartzen ditu, etorkizuneko fabrikaziorako irtenbide polifazetikoa eskainiz.

Contents

List of Figures

List of Tables

Acronyms

AAS	Asset Administration Shell
AI	Artificial Intelligence
API	Application Programming Interface
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Modeling and Notation language
CPS	Cyber-Physical Systems
FP	Functional Properties
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technologies
IoT	Internet-of-Things
MAPE-K	Monitor - Analyze - Plan - Execute - Knowledge
NFP	Non-Functional Properties
OMG	Object Management Group
OWL	Web Ontology Language
QoS	Quality of Service
RAMI 4.0	Reference Architecture Model for Industry 4.0
RDF	Resource Description Framework

REST	REpresentational State Transfer
ROS	Robotic Operating System
SLR	Systematic Literature Review
SMS	Smart Manufacturing Systems
SPARQL	SPARQL Protocol and RDF Query Language
WfMS	Workflow Management Systems
XML	eXtensible Markup Language
YAWL	Yet Another Workflow Language

1

Introduction

The manufacturing landscape is undergoing a transformation driven by the principles of Industry 4.0, heralding the era of future manufacturing [2]. This transformation envisions highly flexible manufacturing processes capable of swift adaptation to dynamic operational requirements, situational events, and environmental factors. Smart manufacturing emerges as a fundamental pillar in this evolution, wherein manufacturing operations experience an underlying integration with Information and Communication Technologies (ICT) [3]. Smart Manufacturing Systems (SMS) are intelligent systems that leverage devices, sensors, real-time data, and sophisticated algorithms for automating operations [4].

The scope of the topics and concepts discussed in this thesis are depicted in Fig. 1.1. It all starts with SMS which must adhere to architectural concerns to effectively orchestrate manufacturing operations while dealing with the dynamic requirements of customers, situational events, and environmental factors. The principles outlined in the Reference Architecture Model for Industry 4.0 (RAMI 4.0) [5] emerge by introducing a framework for shaping the landscape of modern industrial systems. At its core, RAMI 4.0 embodies a hierarchical and systematic approach to the integration of Cyber-Physical Systems (CPS) within the broader context of Industry 4.0 [6]. CPS represent the synergy between computational elements and the physical world, allowing for more interconnected operations [7]. Within RAMI 4.0, the Microservice Architecture philosophy takes centre stage as a pillar of flexibility [5].

The Microservice Architecture philosophy advocates for the decomposition of monolithic services into smaller and independent units [8], transforming centralized systems into more decoupled and versatile systems. These granular, distributed services cater to the heterogeneity challenge posed by diverse devices and machines in the manufacturing ecosystem [7]. This promotes the autonomy and decentralization of SMS, laying the groundwork for service-driven manufacturing processes.

Orchestration of these granular services can be achieved through Workflow Management, a standardized approach to coordinating tasks using specialized tools to achieve business goals [9]. Workflow Management Systems (WfMS) encompass tools for maintaining workflows during

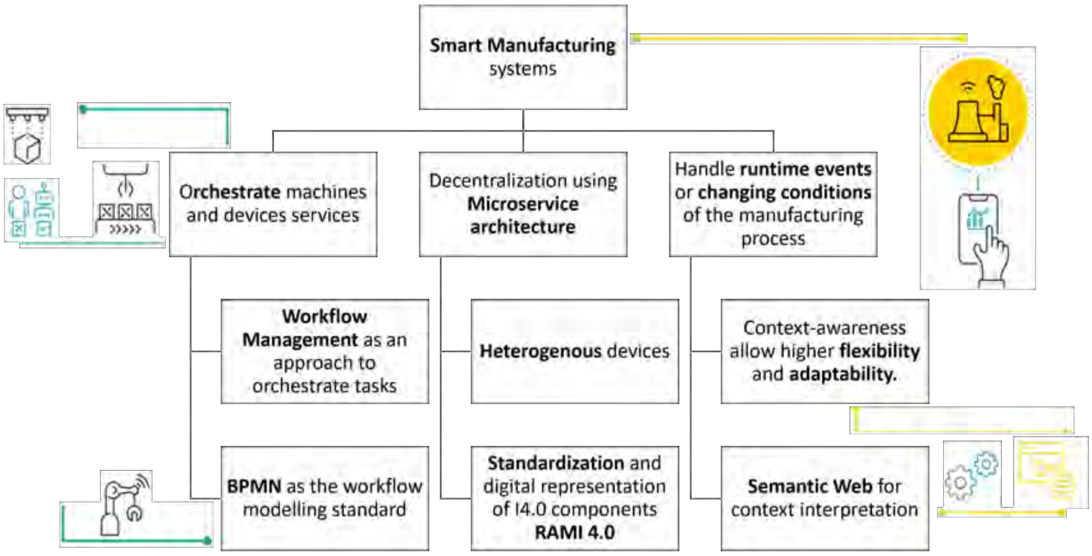


Figure 1.1: Overview of the key concepts involved in this thesis

their entire life cycle — design, execution, and monitoring phases [10]. WfMS rely on notation languages for representing processes, with BPMN (Business Process Model and Notation) being the globally recognized standard that facilitates comprehensive process modelling [11]. Unlike traditional production lines, BPMN provides a standardized language that promotes clarity in a human-readable format [12].

However, achieving highly flexible and adaptable workflows presents a persistent challenge. One approach to addressing this challenge is through context-awareness [13]. In general terms, a system is context-aware if it has the ability to interpret context, incorporating real-time data and environmental and situational factors to reconfigure itself [14]. Similarly, smart manufacturing systems featuring context-awareness exhibit the capability to make automatic or semi-automatic adjustments to production processes in response to unexpected or changing conditions [15], resulting in a reduction in delays and improved manufacturing efficiency.

The scenario described previously poses context interpretation as a crucial challenge, and the Semantic Web emerges as a potential solution, providing a framework for translating raw data into semantically enriched and meaningful data [16]. Moreover, the semantic web addresses the heterogeneity of machines and devices with its Industry 4.0 ontologies and linked data [17], empowering machines to comprehend and interpret context within the manufacturing process.

1.1 Research Context

Future manufacturing envisions an increasingly flexible and adaptable production landscape [2]. The dynamic orchestration and reconfiguration of processes are crucial, allowing real-time adjustments to respond to both, changing conditions and varying customer demands [18].

These dynamic adjustments to the process during runtime are essential to optimize production efficiency and resource utilization [19].

To realize the vision of dynamic and adaptive manufacturing, smart manufacturing systems are increasingly adopting Microservice-oriented architectures [20]. According to the Microservice architecture philosophy, the creation of smaller independent service units allows heterogeneous systems to interact, reduce the complexity of managing individual components, and support distributed deployments [8]. As such, Microservice architecture has become the primary approach for achieving automation and digitization in systems and platforms [21]. However, the growing number of decentralized machine services poses challenges in terms of organization and control [22, 23].

Additionally, interoperability among heterogeneous devices poses another challenge, primarily due to CPS and IoT devices using distinct communication protocols [24]. This complexity makes translating messages between devices cumbersome. To address interoperability issues, the Reference Architectural Model for Industrie 4.0 (RAMI 4.0) was introduced by the German Electrical and Electronic Manufacturers' Association (ZVEI) in [1]. RAMI 4.0 standardizes and digitally represents Industry 4.0 assets using a three-dimensional layer model. The goal is to facilitate interoperation and asset identification within the network by providing technical and operational data. RAMI 4.0 introduces the Asset Administration Shell (AAS) [25], an industrial concept for mapping physical machines into their digitized versions (digital twins).

Furthermore, workflows in the Business Process Management (BPM) domain are conceptually similar to manufacturing processes, dealing with executing several tasks to achieve a goal, such as producing products. The BPM discipline resolves around the workflow design, execution, and monitoring phases using specialized tools [26]. These tools leverage standards to address the workflow management life cycle, with notation languages like BPMN playing a primordial role in this field [27].

In this rapidly evolving landscape of Industry 4.0, the ability to dynamically, efficiently, and adaptively orchestrate manufacturing processes has become the cornerstone of success. The impact of context-awareness in future manufacturing underscores a key research area for enhancing the flexibility and adaptability of manufacturing processes to ever-changing conditions.

1.2 Motivation and Problem

To ground the exploration in real-world applicability, this section begins by immersing in diverse scenarios across industries. These scenarios serve as windows into the challenges faced by modern industries, shaping the understanding of the pressing need for advanced context-aware workflow management solutions.

1.2.1 Scenarios Across Industries

The purpose of this subsection is to underscore the pragmatic utilization of context-aware systems in diverse scenarios and exemplify how these systems can dynamically respond to real-time information. The objective is to explain how context-aware systems contribute to achieving optimal production outcomes, thereby emphasizing the significance of this research.

- **Smart Warehouse:** Consider a scenario in a smart manufacturing facility where robots are employed 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 manufacturing system, when a new operation arises, the system analyzes real-time contextual information. Suppose one robot has a low battery level, another is closer to the working location, and a third has a high payload capacity. In that case, the context-aware system intelligently decides which robot is best suited for the task, ensuring efficient resource utilization, timely task completion, and an increased task completion rate.
- **Automotive:** Suppose an automotive manufacturing environment where adaptive workflows integrate into the production floor. Autonomous robots, equipped with sensors and context-aware capabilities for assembling vehicles, navigate the manufacturing space to perform tasks such as welding, screwing, painting, and folding. The context-aware system adjusts and optimizes the production process in real-time, considering factors like the type of car being produced, component specifications, parts availability, robot locations, conveyors availability, and even changes in the production schedule. This dynamic adaptation to the manufacturing process ensures resource utilization and accelerates production processes.
- **Agriculture:** Imagine a futuristic agricultural setting where autonomous robots, equipped with sensors and context-awareness capabilities, navigate through a plantation of trees to harvest a variety of fruits—ranging from apples and oranges to berries. In this context-aware agricultural system, these robots utilize computer vision to assess tree conditions, recognize ripe fruits, and precisely determine their locations. Analyzing real-time contextual information, the system optimizes the harvesting process through dynamic adaptations during runtime. This mechanism assigns robots to harvest fruits based on multiple characteristics, including the type of fruit, robot proximity to ripe ones, quantity of fruits, current robot battery level, cutting precision, and operational status. This scenario illustrates the need for optimal resource utilization as the system should intelligently direct robots to ensure efficient harvesting. Thus, it exemplifies the transformative potential of Industry 4.0 within a context-aware agricultural system.

These scenarios illustrate the adaptability and flexibility challenges of Industry 4.0 that our research addresses. Addressing these challenges requires dealing with the intricate interoperability challenge inherent in modern manufacturing operations where heterogeneous machines and devices govern.

1.2.2 Manufacturing Challenges

Manufacturing industries today grapple with multifaceted challenges, including interoperability among machines and systems and the dynamic adaptation of production processes in real-time [28]. Resolving these challenges effectively will harness the full potential of Industry 4.0, thereby satisfying the dynamic demand of customer needs.

Previous works have researched algorithms, frameworks, systems, and components that deal with the interoperability and adaptability challenges posed by the context-awareness feature in the workflow management domain. A systematic literature review of these proposals is presented in Chapter 3. While these proposals contribute to advancing context-awareness in the workflow management field, they present diverse opportunities for improvement including, (1) Consideration of a broader range of context variables, including both calculated and sensor-derived data, (2) Utilization of industry-oriented standards and standardized workflow formats, (3) Adoption of user-centric design approaches, transitioning from abstract workflows to executable workflows, and (4) Development of decoupled components and systems, providing more versatility and integration with existing systems.

In the manufacturing domain, addressing workflow flexibility and adaptability through context-awareness has received limited research attention. As discussed in Chapter 3, existing works incorporate diverse techniques, including machine learning algorithms, semantic reasoners, fuzzy logic algorithms, decision tables, and optimization algorithms. However, implementing these approaches in various manufacturing scenarios demands significant effort and expert knowledge, hindering compatibility and interoperability with existing systems.

In the Industry 4.0 domain, RAMI 4.0 promotes the usage of architectural standards for designing manufacturing systems. RAMI 4.0 introduces the Asset Administration Shell (AAS) as an approach to achieving enhanced interoperability, promoting the digital representation of physical machines and devices [29]. AAS, when integrated into manufacturing systems, maintains assets in a machine-readable format that can be stored in a repository. AAS software implementations provide well-defined Application Programming Interfaces (API) to maintain the repository, offering the possibility to integrate other systems due to its Microservice-Oriented design.

Furthermore, RAMI 4.0 advocates knowledge representation formalism to enhance the digital representation of physical assets [30]. Concerning knowledge representation, the semantic web provides mechanisms to turn raw data into semantically enriched and meaningful data [31], which can be later used for providing real-time context to systems about the manufacturing process.

Hence, this doctoral thesis aims to address the interoperability and adaptability challenges posed by the context-awareness feature in manufacturing processes. A context-aware solution for Industry 4.0 should leverage existing advancements in the field while addressing current challenges that impede higher flexibility, such as context interpretation. Possible application fields of this proposal could range in various fields such as smart warehouse, automotive, and agriculture, where context data plays a pivotal role in diverse processes. Thus, the proposal should be capable of adapting workflows using context information and should encompass the key aspects of the workflow life cycle, including design, runtime, and management, focusing on:

1. **Designing Manufacturing Business Processes:** Utilizing standards for industry 4.0 asset representation and workflow notation, leveraging the Asset Administration Shell for standardized digital representations and globally-recognized notation languages like BPMN for workflow recipes.

2. **Context-Awareness:** Incorporating semantic web-based components that can sense and interpret real-world context, allowing the system to dynamically adapt manufacturing operations based on real-time contextual data.

3. **Execution Across Edge and Cloud:** Enabling workflow execution in both edge and cloud environments, offering advantages such as real-time data analysis, low-latency decision-making, and access to contextual information near manufacturing machines.

In summary, this research identifies challenges and opportunities through a meticulous analysis of the impact of context awareness and standards on the entire workflow management lifecycle within the manufacturing and Industry 4.0 domains. The idea is to propose an architecture for workflow management that integrates standards, supports edge and cloud execution with minimal resource consumption and most importantly includes a context-aware component that enables manufacturing workflow reconfiguration at run-time.

1.3 Research Objectives

The main research objectives (O) and their corresponding requirements (REQ) for the development of this thesis are described as follows:

O1: Identify industry-related challenges for efficient workflow management.
O2: Identify current challenges in the state-of-the-art regarding the implementation of context-awareness to support workflow management through a systematic literature review, with a special focus on semantic web-based approaches.
O3: Propose and develop an architecture for context-aware workflow management that takes all the benefits from current approaches and addresses active challenges.

REQ 3.1: The solution should incorporate Industry 4.0 standards for asset representation to enhance interoperability among existing machines and systems.

REQ 3.2: The proposal should be designed following the Microservice Architecture philosophy to provide decoupled components that can work independently.

O4: Evaluate the proposal using several case studies on diverse manufacturing scenarios.

REQ 4.1: Evaluate each component separately to demonstrate independence and self-functioning.

REQ 4.2: Evaluate the context-awareness feature using stress tests to validate the performance of the semantic repository in terms of response time by using growing data volumes.

REQ 4.3: Design a simulated scenario to test the overall performance and effectiveness of the proposal, highlighting flexibility in adapting workflows during process runtime.

O5: Analyze the results and findings from the evaluation phase.

1.4 Research Questions

Establishing Research Questions (RQ) is crucial to determine the focus of research [32]. In this study, the following RQs are aligned with the aforementioned objectives to guide the actual research process systematically and thoroughly. It is important to note that RQ2 is further divided into three sub-questions to facilitate the identification and analysis of relevant literature.

RQ1: What are the key industry-related challenges associated with workflow management?

RQ2: How do existing semantic web-based workflow management approaches contribute to enhancing workflow adaptability and flexibility?

RQ2.1: What active challenges and emerging perspectives are associated with semantic web-based solutions supporting the workflow management life cycle?

RQ2.2: What tools and algorithms are commonly employed in the development of semantic web-based solutions supporting the workflow management life cycle?

RQ2.3: What evaluation factors are considered when measuring the performance of semantic web-based solutions supporting the workflow management life cycle?

RQ3: What are the existing challenges and limitations in the state-of-the-art with respect to context-aware workflow management approaches and their impact on the industry?

RQ4: Which components and features should be integrated into a context-aware workflow management solution to address the identified challenges while leveraging advantages from existing approaches?

RQ5: How does the performance of a context-aware workflow management solution that incorporates the features found in RQ4, manifest in a manufacturing scenario, and what are the key findings drawn from its evaluation, including limitations and potential improvements?

1.5 Hypotheses Formulation

Hypotheses (H) are defined with respect to the objectives and the motivation described previously. The research during this doctoral thesis focuses on developing an approach that will demonstrate the following hypotheses.

H1: A context-aware workflow management architecture and components incorporating Industry 4.0 standards and semantic web for context interpretation demonstrate improved manufacturing efficiency in terms of task completion time, task completion rate, and energy utilization.
This hypothesis aligns with research objective O3 and emphasizes the integration of Industry 4.0 standards through the Microservice Architecture philosophy. The hypothesis focuses on assessing the overall improvement in manufacturing efficiency brought by the context-awareness capability of the architecture. By incorporating context-awareness, the architecture enables real-time adaptation of manufacturing workflows based on changing environmental factors and context data. It also aligns with the research objectives O4 and O5 by testing the workflow adaptability capability using evaluation metrics such as task completion rate, task completion time, and energy utilization. Thus, demonstrating that the context-aware workflow management architecture results in quantifiable enhancements in manufacturing efficiency.

H2: A context-aware workflow management architecture and components that leverage the Microservice Architecture philosophy can be implemented in diverse manufacturing scenarios without necessitating alterations to the individual components or changing the semantic repository structure, demonstrating the flexibility of the proposal.
The last hypothesis revolves around the versatility of the proposed context-aware workflow management architecture, which at the same time is in line with the research objective O4. This hypothesis posits that the architecture can be implemented into various manufacturing environments without necessitating alterations to the architecture, including the context-aware component and its semantic repository. Thus, eliminating the need for extensive reconfiguration or customization efforts. It underscores the architecture's potential to serve as a versatile solution applicable to a wide array of manufacturing scenarios.

1.6 Research Methodology

This section defines the tasks (T) carried out during the doctoral thesis. These tasks are thought out to accomplish the objectives set out earlier. The methodology is divided into three main tasks, and in turn, each task is divided into sub-tasks.

T1: Study applications of context-awareness for workflow management in the Industry 4.0 domain. Inside this task we have defined the following sub-tasks:

- T1.1: Initial idea formulation: The very first stage of the project involved discussing the formulation of the idea. The main objective was to explore the potential benefits of using semantic web technologies to improve manufacturing efficiency and flexibility.
- T1.2: Review background concepts: Background concepts were reviewed to be clear about definitions to carry out an efficient search for studies in the literature. Definitions and concepts directly related to semantic web technologies and workflow management.
- T1.3: Review the state-of-the-art: A systematic literature review regarding state-of-the-art proposals that utilize semantic web technologies for enhancing any of the stages of the workflow life cycle.
- T1.4: Reinforcement courses: To ensure a proper understanding of the literature review, it was important to review and strengthen knowledge on topics that were not fully covered before. This was achieved by studying courses on research methodologies and scientific writing, as well as technical courses on business process management and semantic web technologies.
- T1.5: Problem formulation: Identification of current challenges and limitations in the state-of-the-art for the formulation of a problem to solve. The hypotheses and objectives were formulated as well.

T2: Research development to demonstrate hypotheses

- T2.1: Initial prototype design: Based on the analysis of the state-of-the-art, best practices and existing challenges were considered for the design of a new prototype for context-aware workflow management. This sub-task includes (a) The design of architectural concerns, (b) The selection of development tools, (c) The selection of a business process modeller tool, and (d) The selection or creation of a business process executor tool.
- T2.2: Prototype development: The development of the prototype is divided into three phases: The first phase includes the development of AAS submodels for the representation of asset functionalities. The second phase involves developing a plugin for a workflow modeller tool to allow discovery and offer asset services for the user to

design manufacturing processes. The third phase includes incorporating context-awareness capabilities into the architecture to enable workflow reconfiguration at runtime.

- T2.3: Experimental phase: A testing phase using several case studies was conducted. The source code of the developed tools and datasets for the simulation were also provided.
- T2.4: Results presentation / Publications: Several articles introducing our architecture showcasing the results were published in conferences and journals.

T3: Dissemination and dissertation

- T3.1: Compilation of contents and results: Generated during this research project and elaboration of conclusions.
- T3.2: Thesis side documents: Elaboration of required documents for the thesis dissertation.
- T3.3: Thesis dissertation: The thesis defence is planned at the very last steps.

1.7 Research Contributions

This section presents the significant contributions and scientific publications resulting from the research efforts documented in this doctoral thesis. It provides a comprehensive overview of the advancements, including innovative methodologies, software implementations, and datasets. These contributions are not only valuable to the academic community but also provide practical assets that can help drive the industry towards the forefront of the state-of-the-art in Industry 4.0 and manufacturing workflow management.

1.7.1 Contributions

The contributions of this doctoral thesis range from methodology for conducting literature reviews, to the actual literature review, as well as software and dataset contributions that allow the community to take our work and continue advancing to the state-of-the-art. The following enumeration delves into these contributions and is sorted by importance, being the most notable contribution first within the list.

1. **Architecture for Context-Aware Workflow Management:** During the literature revision, several challenges were identified including (1) Consideration of a broader range of context variables, including both calculated and sensor-derived data, (2) Utilization of industry-oriented standards and standardized workflow formats, (3) Adoption of user-centric design approaches, transitioning from abstract workflows to executable workflows, and (4) Development of decoupled components and systems, providing

more versatility and integration with existing systems. In response, an architecture was proposed to overcome these limitations, with the aim of improving manufacturing efficiency and adaptability of processes. The architecture and its components were delivered iteratively, these components are:

(a) **Node-RED Workflow Manager for Edge Service Orchestration:** Through the correlation analysis conducted during the literature revision, we encountered a need for a workflow manager that can operate at both, edge and central environments. Thus, this contribution embodies a BPMN-based workflow manager called Node-RED Workflow Manager[1]. It was first delivered within a scientific publication in [33].

(b) **AAS Enhanced Business Process Modeler:** The literature review also revealed a need for standardization in the design of manufacturing processes. Thus, this contribution encompasses an enhanced workflow modelling tool[2] that integrates Asset Administration Shell as the Industry 4.0 standard and BPMN as the workflow modelling standard. The tool features a service-discovery mechanism that facilitates the design of manufacturing processes by providing users with the available services from the AAS repository. Furthermore, quality conditions can also be established during the design phase to constrain and optimize the execution of a given task within the workflow by selecting the best available service or device. This component was first presented in a scientific publication in [34].

(c) **Context Analyzer component:** Building upon the previously described contribution, the Context Analyzer component[3] serves as the core element within an architecture for context-aware workflow management by allowing workflow adaptations during runtime. These adaptations are executed through the utilization of the MAPE-K reference model (Monitor, Analyze, Plan, Execute, and Knowledge), a structured framework for building autonomous systems. Alongside the MAPE-K model, semantic web technologies are employed for context interpretation. Furthermore, the Context Analyzer was presented through a scientific publication at the 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM) 2023, held in Lisbon-Portugal from November 22 to 25 2023. Furthermore, this article was awarded the **"Best Service Innovation Paper Award"** during the conference and is currently under publication process. A pre-print is available in [35].

Each component within the architecture was validated through individual experiments, demonstrating effectiveness in specific areas such as workflow design, execution, monitoring, and adaptability during runtime. In addition, a consolidated experiment was

---

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

[2] `https://github.com/MUFacultyOfEngineering/AASBPM`

[3] `https://github.com/MUFacultyOfEngineering/ContextAnalyzer`

conducted to validate the overall performance of the architecture. This consolidated experiment is enlisted next as a contribution for applicability purposes, including both, the source code for the simulation scenario and datasets.

2. **Realistic Simulation on a Warehouse Scenario:** A realistic warehouse scenario involving ROS-based robots was conducted to evaluate the overall performance of the architecture. The simulation was conducted using Nvidia Isaac Sim, a photo-realistic simulation tool and navigation of the robots was handled using ROS (Robotic Operating System) in combination with RViz (ROS Visualization). In this scenario, there are 10 robots with different quality values. They are randomly positioned within the environment, which also includes obstacles that make navigation difficult for the robots. There are bins for the robots to pick up and deliver to a specific location, and a pallet where the robots should deliver the bins. Multiple quality conditions are considered during process execution to coordinate robot allocation dynamically, and the context-awareness feature of the proposed architecture is evaluated.

   The results showcased the actual benefits of the architecture in terms of task completion time, task completion rate, and resource utilization. These results were delivered through a scientific article in [36], and the customized warehouse simulation scenario is provided in a GitHub repository[4].

3. **Datasets:** Along with the simulation phase, datasets are also provided for reproducibility purposes[5]. For this, data concerning robot positions, uncertainty rate, battery level, payload capacity, and proximity were collected during 100 simulations and over an increasing number of robots (from 5 to 10). This dataset is available together with a video showcasing the steps to implement and test the proposal. The provision of this dataset, coupled with the comprehensive tutorial video, fosters transparency and facilitates the community in validating and building upon our research.

Additional contributions include a domain-specific ontology, methodology for conducting systematic literature reviews in the scope of computer engineering, and the actual literature reviews in this domain:

4. **DeviceServiceOnt: An Ontology for Device and Service Representation:** Within the development of the Context Analyzer component, the need for a specialized ontology to represent devices and services became evident. To address this, the DeviceServiceOnt ontology[6] was created. This ontology, derived from the broader I40GO (A Global Ontology for Industry 4.0), is tailored to specifically capture the nuances of devices and services within the manufacturing context. Utilizing the modular and layered structure of I40GO, DeviceServiceOnt provides a dedicated layer for device and service

---

[4] `https://github.com/MUFacultyOfEngineering/RosIsaacSimWarehouseCarterx10`

[5] `https://github.com/MUFacultyOfEngineering/ContextAnalyzer/tree/main/SimulationProofs`

[6] `https://github.com/MUFacultyOfEngineering/ContextAnalyzer/blob/main/DeviceServiceOnt.owl`

representation. The development and utilization of DeviceServiceOnt are detailed in a scientific paper currently under review process, with a pre-print available in [37].

5. **Methodology for conducting Systematic Literature Reviews:** The vast and rapidly evolving domain of Industry 4.0 manufacturing workflows necessitated the development of a robust methodology for systematic literature reviews. Addressing this need, our innovative methodology, presented in [38], provides a structured approach tailored to the intricacies of computer science research. It serves as a valuable guide, enabling researchers to navigate the expansive literature, extract meaningful insights, and contribute to the cumulative knowledge in the field.

6. **Systematic Literature Review on Semantic Web-Based Context-Aware Workflow Management:** In the pursuit of unravelling the intricacies of context-aware workflow management within the Industry 4.0 landscape, our systematic literature review, detailed in [39], stands as a comprehensive resource. This contribution surveys the state-of-the-art and meticulously identifies prevailing challenges, at the same time it offers a roadmap for future enhancements. Researchers and practitioners alike can leverage this review to gain a nuanced understanding of the current landscape and chart their course toward more effective and adaptive workflow management.

7. **Review on Microservice-Based Workflow Management Solutions for Industrial Automation:** Addressing the industrial challenges obstructing the seamless integration of workflow management in Industry 4.0, our review, presented in [40], undertakes a thorough examination of microservice-based frameworks. This contribution sheds light on the hurdles hindering adoption and provides a comparative analysis of existing frameworks.

1.7.2 Scientific Publications

The works covered in this dissertation have either been published or submitted for review in various peer-reviewed international journals and conferences. The scientific publications directly related to the work in this thesis are:

Journal papers:

1. (Published) **Ochoa, W.**, Larrinaga, F. and Pérez, A., 2023. Context-aware workflow management for smart manufacturing: A literature review of semantic web-based approaches. Future Generation Computer Systems. SJR Quartile: Q1.

2. (Published) Carrera-Rivera, A., **Ochoa-Agurto, W.**, Larrinaga, F. and Lasa, G., 2022. How to conduct a systematic literature review: A quick guide for computer science research. MethodsX, p.101895. SJR Quartile: Q2.

3. (Published) Represa, J.G., Larrinaga, F., Varga, P., **Ochoa, W.**, Perez, A., Kozma, D. and Delsing, J., 2023. Investigation of Microservice-Based Workflow Management Solutions for Industrial Automation. Applied Sciences, 13(3), p.1835. SJR Quartile: Q2.
4. (Published) **Ochoa, W.**, Legaristi, J., Larrinaga, F. and Perez, A., Dynamic context-aware workflow management architecture for efficient manufacturing: A ROS-based case study. Future Generation Computer Systems. SJR Quartile: Q1.
5. (Under Review) **Ochoa, W.**, Cuenca J., Larrinaga, F. and Perez, A., I40GO: a Global Ontology for Industry 4.0. Expert Systems with Applications. SJR Quartile: Q1.

Conference papers:

1. (Published) **Ochoa, W.**, Larrinaga, F. and Pérez, A., 2023. Architecture for managing AAS-based business processes. Procedia Computer Science, 217, pp.217-226. Presented at the 4th International Conference on Industry 4.0 and Smart Manufacturing (ISM) 2022, held in Linz-Austria from November 2 to 4 2022.
2. (Published) Larrinaga, F., **Ochoa, W.**, Perez, A., Cuenca, J., Legaristi, J. and Illarramendi, M., 2022, April. Node-red workflow manager for edge service orchestration. In NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium (pp. 1-6). IEEE.
3. (Accepted) **Ochoa, W.**, Larrinaga, F., Pérez, A. and Cuenca J., 2023. Enhancing Flexibility in Industry 4.0 Workflows: A Context-Aware Component for Dynamic Service Orchestration. Procedia Computer Science. Presented at the 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM) 2023, held in Lisbon-Portugal from November 22 to 24 2023. This work was awarded the **"Best Service Innovation Paper Award"** among 323 accepted scientific papers.

This collection of contributions and publications showcases the various advancements made in the field of Industry 4.0 and manufacturing workflow management. The various advancements presented here include valuable methodologies, tangible software implementations, and datasets. These contributions promote academic discussion, encourage the community to build upon our work, and serve to validate the research conducted for this doctoral thesis.

1.8 Thesis Organization

The structure of this work is as follows: Chapter 2 reviews fundamental concepts pertinent to the domain of this thesis. The research commences in Chapter 3, which delves into a systematic literature review of semantic web-based workflow management. This chapter encompasses the gathering of advancements, best practices, and the identification of prevailing challenges, with a particular emphasis on context-awareness. Building upon this foundation, Chapter 4 proposes a context-aware workflow management architecture tailored to address the challenges identified during the literature revision. Additionally, it evaluates the architecture and its components through several experiments. Chapter 5 then elaborates on the context-awareness capability of the architecture for enhancing the flexibility and adaptability of workflows. A unified case study on a realistic warehouse scenario employing ROS-based robots is conducted in Chapter 6. It also discusses the results of the experiment while highlighting the benefits of the proposal in improving manufacturing efficiency. Finally, Chapter 7 presents the conclusions of this work and sets the foundation for future work.

2

Background Concepts

In the rapidly evolving landscape of modern manufacturing, a paradigm shift driven by Industry 4.0 principles is transforming the way production processes are orchestrated and managed. This chapter explores fundamental concepts crucial for understanding the intricate interplay of Industry 4.0, Workflow Management, Context-Awareness, and Semantic Web, all of which are essential for the domain of this thesis.

2.1 Industry 4.0 (I4.0)

I4.0 relies on the adoption of digital technologies to gather data in real-time and to analyze it, providing useful information to the smart manufacturing system [41]. Smart Manufacturing, or Advanced Manufacturing, embodies the idea that systems should exhibit the flexibility to autonomously adapt production processes for various product types and changing conditions [15]. Furthermore, I4.0 encompasses the exchange of information through technologies embedded in final products, often referred to as Smart Products or Embedded Systems, which form an integral part of the broader I4.0 landscape [3].

A convergence of emerging technologies, particularly the Internet of Things (IoT), cloud services, big data, and analytics, has enabled the realization of this concept, giving rise to the cyber-physical system paradigm of I4.0 [42].

2.1.1 Cyber Physical Systems (CPS)

CPS represents a significant advancement in the realms of computer science and information and communication technologies [43]. CPS are systems that collaborate with and connect to the physical world and its ongoing processes. They provide and utilize data-accessing and data-processing services available within the network [43].

CPSs are physical devices that integrate computational processes, often incorporating feedback loops where physical sensors and actuators influence computations, and computations, in turn, impact actuators. In essence, CPS employ computations and communication deeply embedded with physical processes to introduce new capabilities into systems [44]. The scope of CPS ranges from products featuring embedded systems, such as miniature pacemakers, to large-scale entities like national power grids and driverless vehicles [41].

2.1.2 Microservice Architecture Philosophy

In modern industrial systems, the integration of CPS poses architectural challenges, necessitating a paradigm shift. The microservice architecture philosophy offers a compelling solution by advocating for the decomposition of monolithic services into independent, modular units [8]. These microservices operate independently, promoting decentralization, flexibility, and efficient utilization of the diverse machines and devices within the manufacturing ecosystem [7].

At the core of the Microservice Architecture Philosophy are principles that reshape the traditional landscape of industrial systems [45]. These include (a) the decomposition of monolithic services into smaller ones, (b) the independence of services for greater autonomy and agile development, (c) interoperability achieved through well-defined interfaces, and (d) scalability enabled by the modular nature of microservices for improved versatility of the systems. This philosophy poses a transformative approach to architectural concerns in CPS, offering a modular and decentralized design that aligns with the evolving demands of modern manufacturing systems.

2.1.3 Reference Architectural Model for Industrie 4.0 (RAMI 4.0)

RAMI 4.0 combines the essential elements of Industry 4.0 within a three-dimensional layer model, offering a comprehensive depiction of all critical facets of I4.0. This approach facilitates the breakdown of complex interrelations into more manageable and comprehensible clusters, ensuring mutual understanding among all stakeholders engaged in I4.0 discussions [46]. By leveraging this framework, I4.0 technologies can be systematically categorized and further developed [5]. RAMI 4.0 was developed by the German Electrical and Electronic Manufacturers' Association (ZVEI) to support I4.0 initiatives [1]. Key characteristics of RAMI 4.0 encompass the following:

1. Embodies a service-oriented architecture.
2. Unifies all elements and IT components within a layered lifecycle framework. Combines all elements and IT components in a layer and life cycle.
3. Deconstructs intricate processes into easily digestible packages, encompassing data privacy and IT security models.
4. Advocates for flexible systems and machines.
5. Functions are distributed across the network.

6. Participants interact across hierarchy levels.
7. Communication occurs seamlessly among all participants.
8. Products are integral components of the network.

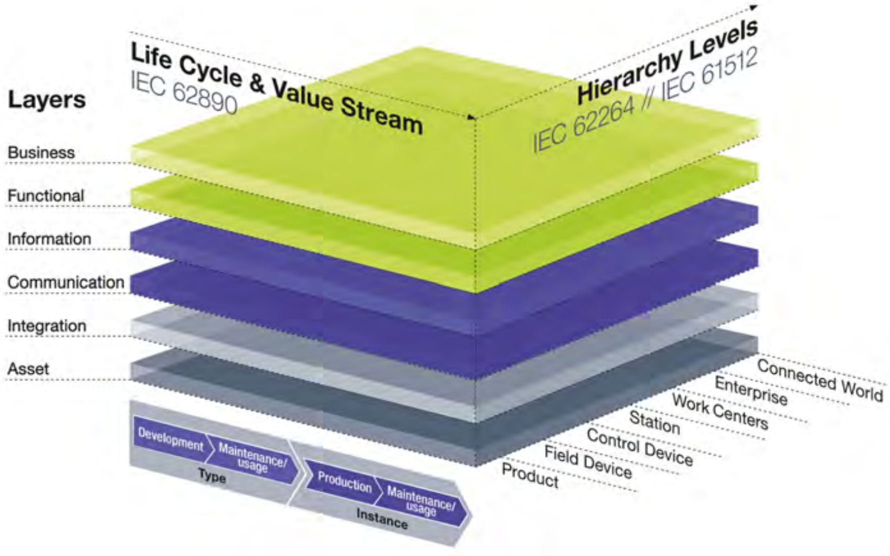


Figure 2.1: Reference Architectural Model for Industrie 4.0 (RAMI 4.0) (Taken from [1])

As depicted in Figure 2.1, RAMI 4.0 encompasses three main axes [1]:

1. **Hierarchy Levels**, illustrated on the right horizontal axis, corresponding to the hierarchy levels defined in IEC 62264, the international standards series for enterprise IT and control systems. These levels represent various functionalities within factories or facilities. Within the Industry 4.0 context, these functionalities encompass workpieces, denoted as "Product" and their connection to the Internet of Things and services, labelled as the "Connected World".
2. **Life Cycle Value Stream**, depicted on the left horizontal axis, outlines the life cycle of facilities and products. This concept draws from IEC 62890, which pertains to life-cycle management for systems and products used in industrial-process measurement, control, and automation.
3. **Layers**, visible on the vertical axis, consists of six layers that detail the mapping of a machine and its properties into a virtual representation. For instance, the "Business" layer is concerned with organizational and business processes, while the "Functional" layer focuses on asset functions. The "Information" layer handles necessary data, the "Communication" layer manages information access, the "Integration" layer oversees the transition from the real to the digital world, and the "Asset" layer involves physical entities in the real world.

Within these three axes, RAMI 4.0 allows for the mapping of all critical aspects of Industry 4.0. This facilitates the classification of objects such as machines according to the model.

RAMI 4.0 supports the description and implementation of highly flexible Industry 4.0 concepts, enabling a gradual transition from the present into the world of Industry 4.0 [1].

RAMI 4.0 facilitates the digitization of the assets for virtual representation [1]. An asset can be i.e. physical components, hardware, documents, software, etc [6]. Human beings are also part of the Asset Layer and are connected to the virtual world via the Integration Layer [5].

2.1.4 Asset Administration Shell (AAS)

AAS represents a standardized digital representation of assets, serving as the cornerstone of interoperability between applications managing manufacturing systems [47]. It identifies the Administration Shell and the assets it represents, maintaining digital models of various aspects (submodels) and describing the technical functionality exposed by the Administration Shell or respective assets [47]. For the success of Industrie 4.0, it is essential that information about not only entire machines but also specific parts and components be stored in the administration shell [29].

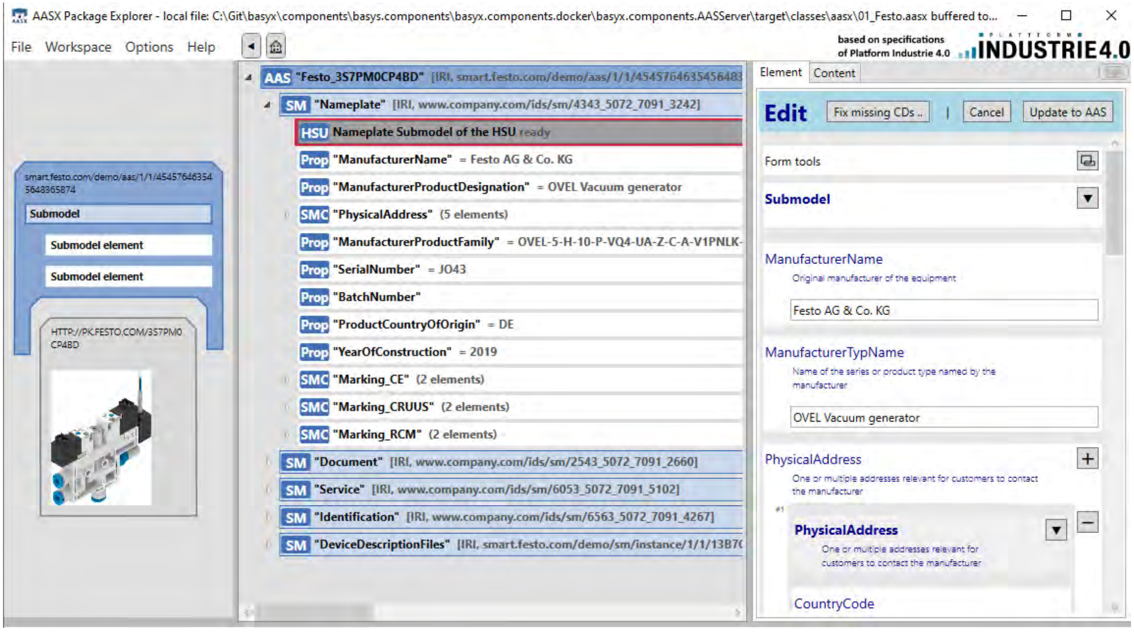
According to the publication “Details of the Asset Administration Shell” [47] by Plattform Industrie 4.0 and ZVEI, assets extend beyond machines and their components to include supply materials, individual products, documents, software, and any entity requiring a connection. Additionally, assets may possess certain fundamental characteristics, including an intrinsic value, ownership, computing resources, and relations with other assets (via the AAS registry) [25, 48].

For the domain of smart manufacturing, assets must be uniquely identified worldwide using identifiers (IDs) [47]. The Administration Shell itself also possesses a unique ID. In batch-based production, batches are treated as assets and are described by a corresponding Administration Shell. When a set of assets requires description by an Administration Shell, a unique ID for the composite asset must be created [29].

2.1.4.1 AAS Submodels

An AAS typically contains or references several submodels, which define properties and services while implementing a reflexive interface. Submodels offer high-level information, such as details regarding services offered by assets, asset status models, or plant topology models. These submodels may comprise properties, functions, events, references, relationships, files, web-references, and BLOBs (binary large objects) [49]. This diversity enables the provision of a wide range of data as submodels and the incorporation of numerous data sources. Submodels can serve as common points of interaction in modern Industrie 4.0 systems, offering a facade to data sources that once required the use of proprietary interfaces [49].

Within each submodel template, property definitions’ identifiers for use as semantic references are predefined. To instantiate such a submodel, properties are to be created, each with a semantic reference to the property definition, and attach a value. Typically, this process involves generating an identifier for the submodel instance itself, typically in the form of a URI/URL [50]. For example, the Nameplate submodel, depicted in Figure 2.2, provides properties for registering an asset's nameplate. This enables users to query information like the physical address, serial number, year of construction, manufacturer name, and more [51]. Additional submodel templates can be found in [52], [53], and [54].



AAS "Festo_3S7PM0CP4BD" [IRI, smart.festo.com/demo/aas/1/1/4545764635456483
SM "Nameplate" [IRI, www.company.com/ids/sm/4343_5072_7091_3242]
HSU Nameplate Submodel of the HSU ready
Prop "ManufacturerName" = Festo AG & Co. KG
Prop "ManufacturerProductDesignation" = OVEL Vacuum generator
SMC "PhysicalAddress" (5 elements)
Prop "ManufacturerProductFamily" = OVEL-5-H-10-P-VQ4-UA-Z-C-A-V1PNLK-
Prop "SerialNumber" = JO43
Prop "BatchNumber"
Prop "ProductCountryOfOrigin" = DE
Prop "YearOfConstruction" = 2019
SMC "Marking_CE" (2 elements)
SMC "Marking_CRUUS" (2 elements)
SMC "Marking_RCM" (2 elements)
SM "Document" [IRI, www.company.com/ids/sm/2543_5072_7091_2660]
SM "Service" [IRI, www.company.com/ids/sm/6053_5072_7091_5102]
SM "Identification" [IRI, www.company.com/ids/sm/6563_5072_7091_4267]
SM "DeviceDescriptionFiles" [IRI, smart.festo.com/demo/sm/instance/1/1/13B7

ManufacturerName
Original manufacturer of the equipment
Festo AG & Co. KG

ManufacturerTypName
Name of the series or product type named by the manufacturer
OVEL Vacuum generator

PhysicalAddress
One or multiple addresses relevant for customers to contact the manufacturer
#1
PhysicalAddress
One or multiple addresses relevant for customers to contact the manufacturer
CountryCode

Figure 2.2: The Nameplate AAS Submodel

From a technical standpoint, an AAS implementation may support diverse data serialization formats such as XML, JSON, RDF, AML, or OPC UA node-set [55].

2.1.4.2 AAS Implementations

As of this writing, we have collected available AAS implementations that support the creation and operation of an AAS. It may not be necessarily complete and other implementations might exist. Thus, Table 2.1 lists the existing AAS implementations, including information such as software license type, communication protocols supported by the implementation, the link to the source code, and some additional remarks about the implementation.

2.2 Workflow Management

Task orchestration is a fundamental aspect of Smart Manufacturing systems, guiding them in producing final products based on predefined workflow recipes [15]. Within this context, Workflow Management assumes a pivotal role by formalizing these recipes. It all commences

Table 2.1: Asset Administration Shell implementations

Name	Remarks	License	Protocols	Link to Source Code
AASX Package Explorer [56]	A desktop implementation for creating, editing, and browsing asset administration shells as .aasx packages. Supporting serialization in XML and JSON.	Apache License 2.0	HTTP, MQTT, OPC-UA	`https://github.com/admin-shell-io/aasx-package-explorer`
BaSyx (Previously BaSys) [57]	An open-source software platform for the asset administration shell, provides APIs as the communication interface between end-users, assets, and its registry. Provides software development kits (SDKs) for C++, C#, and Java.	Eclipse Public License 2.0	HTTP, MQTT, and TCP/UDP.	`https://github.com/eclipse-basyx/basyx-java-sdk`
PyI40AAS [58]	An open-source Python module for manipulating and validating AAS. Provides an implementation of the Asset Administration Shell (AAS) for Industry 4.0 Systems, compliant with the meta model and interface specification of Plattform Industrie 4.0.	Dual-licensed: Eclipse Public License 2.0 and Apache License 2.0	-	`https://git.rwth-aachen.de/acplt/pyi40aas`
SAP AAS Service [59]	Provides a system based on Docker images implementing the RAMI 4.0 reference architecture (including AAS).	Apache License 2.0	HTTP	`https://github.com/SAP/i40-aas`
NOVAAS [60]	An open-source implementation and execution environment of AAS. Provides an implementation of the Digital Twin concept by using JavaScript and Low-code development platform (LCDP) Node-Red.	EUROPEAN UNION PUBLIC LICENCE 1.2	MQTT, HTTP	`https://gitlab.com/novaas/catalog/nova-school-of-science-and-technology/novaas`

with the Workflow Management discipline, responsible for the design, execution, and monitoring of workflow recipes as part of the workflow lifecycle. A workflow, in this context, is defined as a “systematic organization of resources into processes that transform materials, provide services, or process information”, representing an orchestrated and repeatable pattern of business activities [61]. Workflows centre around automating procedures and are often associated with re-engineering, involving the analysis, modelling, and redefinition of business activities [62].

2.2.1 Workflow Management Systems (WfMS)

A WfMS is a system that defines, manages, and executes workflows [62]. Three are the main function key areas of WfMS: design-time, run-time, and management. These function areas primarily focus on task coordination within and across workflows [63]. Figure 2.3 provides an illustration of these function areas, which are typical within workflow management systems.

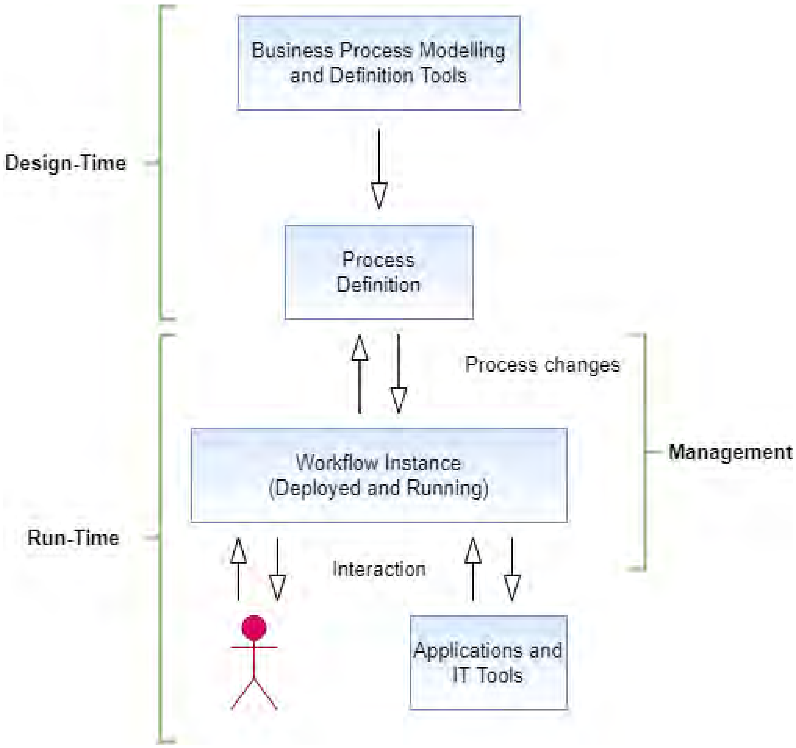


Figure 2.3: Main function areas of a typical Workflow Management System

- **Design-time**: Concerned with defining and modelling workflow processes and activities. During this stage, a business process is converted from the real world into a computer processable definition (process definition) using analysis, modelling and definition techniques. Some workflow systems may allow modifications to process definitions during runtime, as pointed out by the arrows in Figure 2.3. According to David Hollingsworth [62], a Process Definition typically consists of discrete activity steps, encompassing associated computer and/or human operations and rules dictating the flow of the process across these activity steps.

- **Run-time**: This phase is centred on interactions with both human users and IT applications, which are frequently integrated into activity steps. Here, actionable workflows are enacted by the execution system, translating workflow elements into resource invocation data, scheduling activity steps within the process, and requesting the most suitable human and IT application resources [61]. Human resources may include tasks like form filling and necessitating support for interaction as control transitions between activities. In some WfMS, runtime control functions are included, permitting changes during the runtime.

- **Management**: Composed of control functions that allow the management of workflow processes and sequencing the activities of processes. It often incorporates monitoring tools to detect and resolve workflow issues before they can negatively impact critical business processes. During this phase, workflow instances are executed, and the system should provide an interface for monitoring workflow execution and tools to address deviations [61]. The necessary monitoring information is typically outlined in the Service Level Agreement (SLA) within the workflow, ensuring quality and delivering alerts in the event of failures [64].

Workflow recipes can be formalized into business processes through the use of standardized notation languages that support microservices in the form of service tasks. The management of those formal recipes leads us to the Business Process Management (BPM) domain which includes the systems and notation languages to design, execute, and manage business processes.

2.2.2 Business Process Management (BPM)

BPM, as defined by the Object Management Group (OMG), comprises a collection of techniques aimed at continuously and iteratively improving all processes essential for running a business [65]. These techniques have proven effective in error reduction, cost savings, and enhanced productivity across diverse industries, including manufacturing, telecommunications, insurance, government agencies, and more. BPM's success lies in its capacity for process automation and the execution of error-free manual processes [65].

2.2.2.1 Standardized Notation Languages for Workflow Design

Several notation languages for modelling business processes have been identified, each with unique concepts, advantages, and disadvantages.

1. **Business Process Modeling and Notation (BPMN)**: As recognized by OMG, BPMN has emerged as the standard for business process diagrams [66]. It is designed to be used by stakeholders responsible for designing, managing, and realizing business processes. Moreover, it is precise enough to facilitate the translation of BPMN diagrams into software process components. BPMN employs a flowchart-like notation that is environment-agnostic and highly user-friendly [66]. It offers a graphical standardized notation that enables business process modelling through workflows, bridging the communication gap between design and implementation. This graphical notation aids in comprehending performance collaborations and business transactions between organizations, ensuring clear self-awareness and understanding among businesses and their partners. Additionally, it allows organizations to swiftly adapt to new internal and B2B business scenarios [67].

2. **Web Services Business Process Execution Language (WS-BPEL)**: WS-BPEL, often abbreviated as BPEL, serves as a standardized language for the orchestration of web services. It is an XML-based notation language designed to facilitate large-scale programming [68]. Through an XML BPEL file, a business analyst can represent the logic and

associated elements, including web services and BPEL process logic. BPEL takes charge of orchestrating the entire process, establishing the sequence and timing of processes. Despite the widespread acceptance of web services in distributed applications, BPEL, by default, lacks human interaction capabilities, a significant gap in many real-world business processes. To address this, BPEL4PEOPLE was created to enable the modelling of human interactions, from simple approvals to complex scenarios such as separation of duties [69].

3. **Yet Another Workflow Language (YAWL)**: YAWL is an open-source workflow language that includes software with an execution engine and a graphical editor. It builds upon the foundation of PETRI networks, extending and formalizing this framework [70]. According to The YAWL Foundation, YAWL offers support for complex workflows, thanks to its formal semantics, enabling robust design-time workflow checks. It is based on an XML notation language.

There are numerous systems that can manage business processes written in different notation languages. Often, business processes are managed by Business Process Management Systems (BPMS). These are systems that offer more robust tools for designing, executing, and monitoring workflows.

2.2.3 Intelligent Business Process Management Suites (iBPMS)

iBPMS is a term coined by Gartner to designate Smart Business Process Management Suites that support the full life cycle of business processes including discovery, design, analysis, implementation, execution, decision, monitoring, and optimization [71]. iBPMS is the natural evolution of BPM systems, it is an integrated set of technologies that coordinates people, machines, and things [72]. It enables the improvement and transformation of business processes through a collaborative platform [73]. Gartner identified a set of key capabilities that must meet an iBPMS [71], they are summarized next.

- **Process Discovery**: Capability to discover processes and identify opportunities for improvement and automation.
- **Business Process Automation**: Automation of tasks and their orchestration into processes.
- **Business Rules and Decision Management**: Capability to define business rules, recommendations or decision management engines, which provide guidance for making human or automated operational decisions.
- **Analytics**: An iBPMS provides insights to make better decisions by analyzing various aspects of the process data.
- **Integration**: Pre-built adapters to enable integration and orchestration with external applications and services though native connectors, APIs, events or connecting databases.

- **Human Task Management and Collaboration**: Enable stakeholders to collaborate in a business process. Collaboration may be facilitated through annotations, redactions, real-time chat, instance documents, co-browse sessions or integration with telephony and video chat services.
- **Low Code Application Development**: Enables users to build model-driven applications, which automate workflows that change frequently and are constrained to a single business function or team.
- **Context and Behavior History**: Enables management and analysis of contextual data around processes. This data may come from external applications, databases or event streams. It is to enhance the intelligence and effectiveness of the system.

A collection of the most relevant workflow systems on this domain is presented in Table 2.2. Columns on top are to compare the key capabilities offered by the iBPMS. The table complements the BPM systems shown by Gartner in [72].

Enriching business processes involves integrating context-awareness capabilities that interpret changes in context and enable decision-making. The next section explores the concept of context-awareness, featuring an architecture proposed by IBM for building autonomous systems. It also delves into available technologies for mapping and processing IoT sensor data, with a primary focus on semantic web technologies.

2.3 Context-Awareness

Context is any data that can be used to understand the current state (situational or locational) of an entity [74]. A system is context-aware if adaptations are performed when dynamic changes occur in the environment where it executes. These adaptations are tailored to actual user needs, preferences, and expectations [75]. Dynamic adaptations are those reconfigurations that can be applied during the system runtime. However, a context-aware system does not necessarily imply automation or real-time processing, rather it refers to the ability to respond to context [76].

2.3.1 Context-Aware Systems

Context-aware systems are computer systems or software applications designed to perceive and respond to the surrounding context or environment in which they operate [77]. These systems use information about the asset's location, preferences, activities, and other relevant factors to provide more personalized and relevant services or content [78]. The goal of context-aware computing is to enhance user experiences and streamline interactions with technology by adapting to changing circumstances [79].

Key characteristics and components of context-aware systems include [77]:

Table 2.2: iBPMS comparison by key capabilities

iBPMS	Edition	Written in	Remarks	Business Process Automation	Notation Languages	Analytics	Integration	Human Task Management and Collaboration	Low Code Application Development	Gartner peerinsights 5/5
Airflow	Community Edition	Phyton	-	-	DAG, Pipelines	Interface to visualize pipelines, monitor progress and troubleshoot issues	Plugins + API	-	-	-
Nintex	Commercial	-	bpmn files can be imported after conversion to xpdl	Promapp, RPA	XPDL	Nintex Analytics: Reports, dashboards, KPIs	A gallery of pre-built connectors, API	Y - Promapp	Nintex, RPA gallery templates, Nintex DocGen, Nintex Forms	4.9 4 Reviews
Nintex-K2	Commercial	-	Acquired by Nintex in October 2020	RPA templates on the gallery	XPDL	Nintex Analytics: reports, dashboards, KPIs	A gallery of pre-built connectors, API, and SDK to create connectors	-	Nintex K2 SmartForms	4.4 25 Reviews
Camunda	Commercial	Java	30 days trial	RPA tasks, RPA scripts, Connector for UiPath, RPA analytics and insights	BPMN 2.0, DMN	Cockpit (Full feature set), Camunda Optimize	Plugins + API	Camunda Tasklist, Camunda Cawemo	Camunda Embedded Forms	-
Camunda	Community Edition	-	Does not have a connector for RPA. But can be done on UiPath separately	RPA can be done using UiPath and Camunda's API	BPMN 2.0, DMN	Cockpit (Just basic feature set)	Plugins + API	Camunda Tasklist	Camunda Embedded Forms	-
ProcessMaker	Commercial	PHP	Does not have a connector for RPA. But can be done on UiPath separately	RPA can be done using UiPath and ProcessMaker's API	BPMN 2.0	Business Activity Monitoring, Dashboards, reports	Plugins + API	Through form builder	Dynaform: Form Builder	4.4 20 Reviews
ProcessMaker	Community Edition	-	-	RPA can be done using UiPath and ProcessMaker's API	BPMN 2.0	N	Plugins + API	Through form builder	Dynaform: Form Builder	-
Bonita	Commercial	Java	-	RPA through UiPath connector	BPMN 2.0	Process Monitoring, Reporting and Analytics	Plugins + API	-	Bonita UI Form Designer	4.5 18 Reviews
Bonita	Community Edition	-	-	RPA through UiPath connector	BPMN 2.0	Process Monitoring, Reporting and Analytics	Plugins + API	-	Bonita UI Form Designer	-
Creatio (formerly bpm'online)	Commercial	C# .Net Core	14 days trial. They do not have pre-built connectors for RPA	RPA can be done using UiPath and Creatio's API	BPMN 2.0	Reporting, Dashboards, AI and ML Tools	Plugins + API	Jointly designer	System designer	5 22 Reviews
Creatio (formerly bpm'online)	Free	-	"Studio Free" is free but is just the workflow modeler.	-	BPMN 2.0	-	-	-	-	-
Newgen	Commercial	-	Version control of processes and views	RPA Designer, Process Analytics, Case management	BPMN 2.0	Reporting, Dashboards, SLA	Plugins + API	Jointly designer	UI designer, application templates	4.3 27 Reviews
Bizagi	Commercial	JavaScript, .Net	Can import diagrams from Microsoft Visio, IBM Bluewoks, XPDL, and BPMN	AI and RPA integrated natively	BPMN 2.0	Business process simulator, Bizagi Diagnostics	Plugins + API + Can create custom connectors	Jointly designer	Bizagi Studio and application templates	4.5 95 Reviews
Bizagi	Free	-	Modeler is free for up to 1 user	-	BPMN 2.0	-	-	-	-	-

- Context Sensing: These systems rely on various sensors and data sources to gather information about the user and their surroundings. Common sensors include GPS, accelerometers, gyroscopes, cameras, microphones, and environmental sensors.
- Context Processing: Once the data is collected, context-aware systems process and analyze it to extract meaningful information. This can involve techniques such as data fusion, machine learning, and pattern recognition to understand the context.
- Context Inference: Inference mechanisms use the processed data to make informed decisions or predictions about the user's situation or needs. For example, inferring that a user is driving based on GPS data and accelerometer readings.
- Adaptation: Context-aware systems adapt their behaviour or output based on the inferred context. This adaptation can take various forms, such as adjusting the user interface, suggesting relevant content, changing device settings, or triggering specific actions.
- Personalization: Context-aware systems often aim to provide personalized experiences by tailoring recommendations or services to individual preferences and behaviours. This can include suggesting music, news articles, or restaurant recommendations based on past user interactions and current context.

Context-aware systems have numerous applications in fields like marketing, healthcare, transportation, manufacturing, and more, offering the potential for more efficient and user-centric experiences [79]. Relevant examples include [80]:

- Location-Based Services (LBS): Mobile apps that offer location-specific information, like maps, nearby points of interest, and real-time traffic updates.
- Context-Aware Recommender Systems: E-commerce platforms and streaming services that suggest products, movies, or songs based on a user's previous behaviour and current preferences.
- Smart Home Automation: Systems that adjust lighting, heating, and security settings based on the occupants' presence, preferences, and the time of day.
- Healthcare Applications: Wearable devices and apps that monitor a user's health and provide timely alerts or recommendations based on their activity, vital signs, and medical history.
- Context-Aware Messaging: Messaging apps that automatically select the appropriate communication method (e.g., text, voice, video) based on factors like the user's location and network conditions.

However, they also raise architectural concerns, as they often require data located in diverse devices and external systems to later process that data and make intelligent decisions [79]. To address this issue, IBM has made progress on software architectural concerns with the MAPE-K framework.

2.3.2 MAPE-K Architectural Model

Context-aware systems require a structured architecture to facilitate decision-making based on environmental data [79]. IBM has introduced the MAPE-K (Monitor, Analyze, Plan, Execute, and Knowledge) framework to construct autonomic and self-adaptive systems [81] (see Figure 2.4). MAPE-K serves as the foundation for developing components that can: 1) capture product and process information, 2) analyze the collected data to generate output and store relevant information in a knowledge base, 3) formulate strategies to address situations that may disrupt normal system behaviour, and 4) deploy these plans to make real-time decisions.

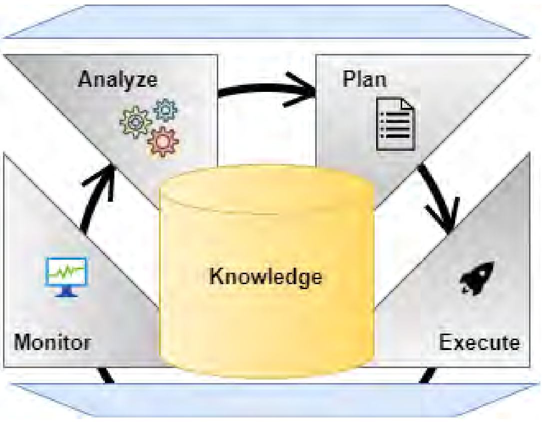


Figure 2.4: The IBM's MAPE-K reference model

Table 2.3 provides an overview of the MAPE-K modules, along with techniques and technologies applicable to each phase, this summary is based on [81, 82],

As delineated in Table 2.3, there are many technologies available for constructing autonomous systems within the MAPE-K reference architecture. Nevertheless, the Semantic Web has garnered significant attention within the scientific community. Its primary appeal lies in its ability to model IoT sensor data as ontological entities, facilitating the application of inference techniques [83, 84].

2.4 Semantic Web Technologies

The Semantic Web grants information a well-defined meaning, fostering collaboration between computers and humans [31]. The core idea behind the Semantic Web is to enhance the existing web, which primarily consists of human-readable text and unstructured data, by adding a layer of structured and semantically rich information that computers can understand and process [85]. This semantic transformation of data involves the creation of ontological entities, which are structured and standardized knowledge representations [31]. Ontologies are constructed using a language that employs URIs to establish connections between entities, commonly referred to as "triples" [86]. Furthermore, the Semantic Web permits the definition of constraints and the application of semantic reasoning to deduce new knowledge by evaluating facts and rules [87].

Table 2.3: Available techniques and technologies for each MAPE-K module

MAPE-K Module	Description	Technique/Technology
Monitor	Capture product and process information from IoT sensors.	Publish/Subscribe through OPC-UA, MQTT, REST
Analyze	Map/Model data into machine-understandable data.	Ontology Based Key-Value pairs Graph-Based
Plan	Apply reasoning techniques to infer new information. The output is a plan that will overcome any situation that may affect the normal behaviour of a system.	Machine Learning Semantic Reasoner Fuzzy Logic Decision tables Heuristics Mixed
Execute	Deploy the plan to allow decision-making at runtime.	Depends on the technologies used in both, the execution system and the produced plan.
Knowledge	Store data to be analyzed, already analyzed, and the produced plan in a knowledge base.	Relational databases Graph databases Key-value storage Wide-column storage

Key components and concepts of the Semantic Web, as outlined in [88], include:

- Web Ontology Language (OWL): OWL is used for defining ontologies and specifying relationships between terms. It enables the creation of formalized knowledge structures.
- Resource Description Framework (RDF): According to W3C, RDF serves as a "standard model for data interchange on the Web" [86]. It excels at merging data with varying schemas and accommodating schema evolution without necessitating changes in data consumers [86]. RDF utilizes URIs to establish relationships between entities, often referred to as "triples", forming labelled graphs that visually represent these relationships.
- SPARQL Protocol and RDF Query Language (SPARQL): SPARQL is a query language designed for querying RDF data. It enables powerful searches and retrievals of structured data from Semantic Web sources.

While the Semantic Web has made steady progress, its full realization remains an ongoing research endeavour. Achieving widespread standards adoption and constructing extensive, interconnected knowledge bases are key challenges in this field.

The exploration of these key concepts delves into advanced manufacturing paradigms where robust frameworks and proposals emerge. Industry 4.0 propels manufacturing into a new era, Workflow Management structures operational efficiency, iBPMS orchestrates business processes intelligently, Context-Awareness adds adaptability, and Semantic Web Technologies infuse structured meaning. This collective knowledge forms the backdrop, setting the stage for an in-depth exploration of context-aware workflow management in the chapters to follow.

3

State-of-the-art Revision on Semantic Web and Context-Awareness in Workflow Management

This chapter presents a revision of the latest advancements in context-aware workflow management with a special focus on semantic web-based approaches. Its primary objective is to analyse the existing tools, techniques, and active challenges associated with achieving flexible and adaptable workflow management within the Industry 4.0 domain.

Moreover, this chapter aims to answer the research questions "RQ1: What are the key industry-related challenges associated with workflow management?" and "RQ2: How do existing semantic web-based workflow management approaches contribute to enhancing workflow adaptability and flexibility?". This is accomplished by thoroughly discussing the open opportunities revealed during the literature review process, coupled with a keyword correlation analysis. The chapter also addresses the "RQ3: What are the existing challenges and limitations in the state-of-the-art with respect to context-aware workflow management approaches and their impact on the industry?" through a discussion of the open opportunities discovered during the literature revision.

Furthermore, this chapter aligns with two research objectives: "O1: Identify industry-related challenges for efficient workflow management" and "O2: Identify current challenges in the state-of-the-art regarding the implementation of context-awareness to support workflow management through a systematic literature review, with a special focus on semantic web-based approaches".

It is worth mentioning that this state-of-the-art revision contributed to several scientific publications, including:

- **[38]:** A methodology for conducting systematic literature reviews in computer science research, which explains the steps followed for conducting this literature revision.
- **[39]:** A systematic literature review on semantic web-based context-aware workflow management approaches for smart manufacturing, which analyzes the current body of work

by categorizing the approaches and identifying advantages and disadvantages.

- **[40]:** A survey on microservice-based workflow management solutions for industrial automation, wherein an analysis of workflow management challenges within the Industry 4.0 domain is presented.

3.1 Introduction

Workflow management, integral to modern organizational operations, orchestrates a series of tasks towards predefined objectives [89]. These objectives span material transformation, service provision, and information processing, all under the purview of the Workflow Management discipline [62].

Workflow recipes can be formalized into business processes using standard-globally known notation languages such as BPMN (Business Process Modeling and Notation language), BPEL (Business Process Execution Language), and YAWL (Yet Another Workflow Language) [90]. These languages, fundamental to Business Process Management (BPM), wield a suite of tools and techniques aimed at error reduction, cost containment, and heightened productivity across diverse sectors [65].

However, the orchestration of services often encounters interoperability hurdles due to divergent communication protocols and technology stacks within the IoT ecosystem [91]. To address this, researchers have turned to semantic web technologies, fostering effective communication among IoT devices [92]. For example, Mehdi et al. [93] proposed a semantic web-based solution tailored for smart city environments, aimed at strengthening interoperability across heterogeneous IoT devices and services, with potential applications in various domains.

Semantic web technologies have the ability to interpret contextual information and infer new knowledge, which can enhance the dynamic nature of workflows [94]. Providing manufacturing systems with semantic capabilities can turn them into context-aware systems, capable of self-configuration and exhibiting cognitive responses to evolving scenarios [92]. This contextual awareness fosters dynamic adaptations during workflow design and runtime, streamlining the management of service level agreements (SLAs) and ensuring heightened service availability and reliability to meet evolving consumer demands [92].

Given the critical nature of these systems in the context of Industry 4.0 manufacturing, a critical examination of the state of the art would appear timely and essential. Although a number of pertinent studies have been identified, no research has been found that focuses on studying semantic context-aware workflow management approaches for the smart manufacturing domain. For instance, Breslin et al. [95] (2010) presented a literature review on the semantic web and its implications in the Industry domain. However, their work, published over a decade ago, primarily emphasized the benefits of the semantic web in industry without surveying state-of-the-art solutions.

Similarly, Pauwels et al. [96] (2017) reviewed the applications and use cases of semantic web technologies in the Architecture, Engineering, and Construction (AEC) domain. While timely and comparative in nature, their review did not delve into approaches featuring context-aware capabilities, merely mentioning it as a concept without in-depth exploration.

Asghari et al. [97] (2018) presented an in-depth survey of IoT-based service composition approaches, including semantic web-based approaches within a broader classification. However, their analysis focused on a limited number of semantic web-based approaches, with context-awareness mentioned as a future challenge rather than a core aspect of investigation.

In another study, Bazan and Estebez [98] (2021) analyzed the state-of-the-art in business process management for Industry 4.0, incorporating context awareness to enhance business process reactions. While insightful, their analysis was limited to only two semantic web-based approaches, leaving considerable scope for further exploration.

This chapter addresses the aforementioned gaps by analyzing challenges, techniques, and proposals not fully covered in previous studies and is organized into several phases. Phase 0, outlined in Section 3.2, serves as a preliminary exploration into industry-related challenges that workflow management approaches need to address, setting the stage for the subsequent phase. Phase 1, depicted in Section 3.3, explains the methodology for conducting the literature review and presents the review itself. The literature revision categorizes existing approaches using a technical taxonomy while assessing their strengths and limitations. From this section, the approaches that specifically feature context-awareness capabilities are considered for further analysis within the next phase. Phase 2, detailed in Section 3.4, offers a comparison of workflow management approaches specifically featuring context-awareness capabilities. Here, active challenges requiring attention to enhance flexibility in context-aware workflow management are listed and analyzed. Finally, the conclusions of the chapter are drawn in Section 3.5.

3.2 Phase 0: Industry-Related Challenges for Efficient Workflow Management

This section serves as a comprehensive analysis of the foundational industry-related challenges for effective workflow management. Furthermore, this section answers the RQ1 “What are the key industry-related challenges associated with workflow management?” by identifying and analyzing the most critical challenges based on the workflow management life cycle.

3.2.1 Analysis of the Critical Challenges

The comprehensive examination of challenges related to achieving efficient workflow management is extensively covered in [99] and [40]. Here, a condensed version is provided to highlight the fundamental aspects crucial for workflow management in modern manufacturing settings. These include:

1. **Workflow Modeling**

This challenge involves architectural design and workflow representation concerns. 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 modelling languages such as BPMN, BPEL (Business Process Execution Language), state machines, Petri Nets, and YAWL (Yet Another Workflow Language) [100], with BPMN being ratified as the standard language by the Object Management Group (OMG) for business process design [12].

2. **Heterogeneous Infrastructure Scale**

Designers face the challenge of finding the right level of aggregation/abstraction for composing workflows in a heterogeneous infrastructure [99]. 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 [101].

(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 [102]. 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 [103]. 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 [104]. This requires standardizing inputs, outputs, communication, syntax, and semantics of workflows. 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 [105]. 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 [106]. 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 [107]. The volatility and rotation of edge resources pose reliability challenges, as functionality and deployment conditions quickly become obsolete [108]. Microservice-based architectures offer fast response times and rapid deployment but require workflow orchestrators to handle the speed of deployment and failures effectively [109]. 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 [99].

3.2.2 Discussion and Findings

This subsection discusses the challenges identified earlier and proposes potential solutions. For instance, the workflow modelling challenge can be tackled by leveraging formal workflow modeling languages like BPMN, BPEL, and YAWL [100], offering a standardized and thorough representation of manufacturing workflows.

Mitigating the challenge of heterogeneous infrastructure scale could involve the utilization of edge-embedded operating systems such as Raspberry Pi and Arch Linux ARM [110], this could be complemented by edge workflow managers requiring minimal resources. By executing and combining these techniques, integration and interaction between edge devices can be enhanced.

Integrating the RAMI 4.0 architectural model as the Industry 4.0 standard presents an opportunity to manage asset rotation at the plant level [111], contributing to overcoming infrastructure scalability challenges. In addressing dynamic microservice architectures, workflow managers can leverage RAMI 4.0 alongside BPMN, BPEL, or YAWL to orchestrate devices and their services via REST, meeting the architectural demands of modern and dynamic industrial environments.

In summary, these industry-related challenges discern the complexities that workflow management systems should meet in the industry domain. Building upon this foundation, the next phase delves into advancements, limitations, and challenges within the state-of-the-art by systematically examining approaches to semantic web-based workflow managers.

3.3 Phase 1: Systematic Literature Review on Semantic Web-based Workflow Management Approaches

This section delves into a Systematic Literature Review (SLR) aimed at examining the landscape of Semantic Web-based workflow management approaches. The industry-related challenges identified in Phase 0 are considered foundational challenges that today's workflow management proposals should address [99]. Therefore, the studies analyzed in Phase 1 undergo careful scrutiny using a methodological approach.

This phase encompasses the methodology employed for conducting the literature review, the actual process of reviewing pertinent literature, and the findings derived from this comprehensive analysis. By scrutinizing existing research in this domain, this phase aims to identify key trends, tools, algorithms, challenges, and advancements in workflow management, with a special focus on semantic web-based approaches. Additionally, opportunities for further research are garnered through a keyword correlation analysis, informing subsequent phases of this chapter.

3.3.1 Literature Review Methodology

This literature review followed the guidelines to perform SLRs found in [112]. Figure 3.1 illustrates the methodology of this literature review, comprising two main phases: Planning and Conducting. Each of these phases includes several sub-steps. The Planning is to define the protocol and the Conducting is to execute the search string into the selected digital libraries and to perform refinement of articles.

3.3.1.1 Planning

The Planning phase entails defining the protocol for the systematic literature review. This is a critical stage in conducting SLR as it establishes the procedures and serves as a record of planned activities [32]. The step-by-step procedure outlined in [112] was adhered to for conducting this SLR. Utilizing tools aids in systematically executing the procedure, ranging from basic spreadsheets to specialized SLR platforms like Parsifal[1] or Covidence[2]. Parsifal and Covidence are online platforms tailored to support researchers, particularly in the Software Engineering field. Parsifal, chosen for this study, aligns with the guidelines presented in [112], offering collaborative features and comprehensive documentation capabilities [113]. The following steps describe in detail the Planning phase.

Planning Step 1: Define PICOC elements and synonyms

PICOC stands for Population, Intervention, Comparison, Outcomes, and Context [114]. It

---

[1] https://parsif.al/

[2] https://www.covidence.org/

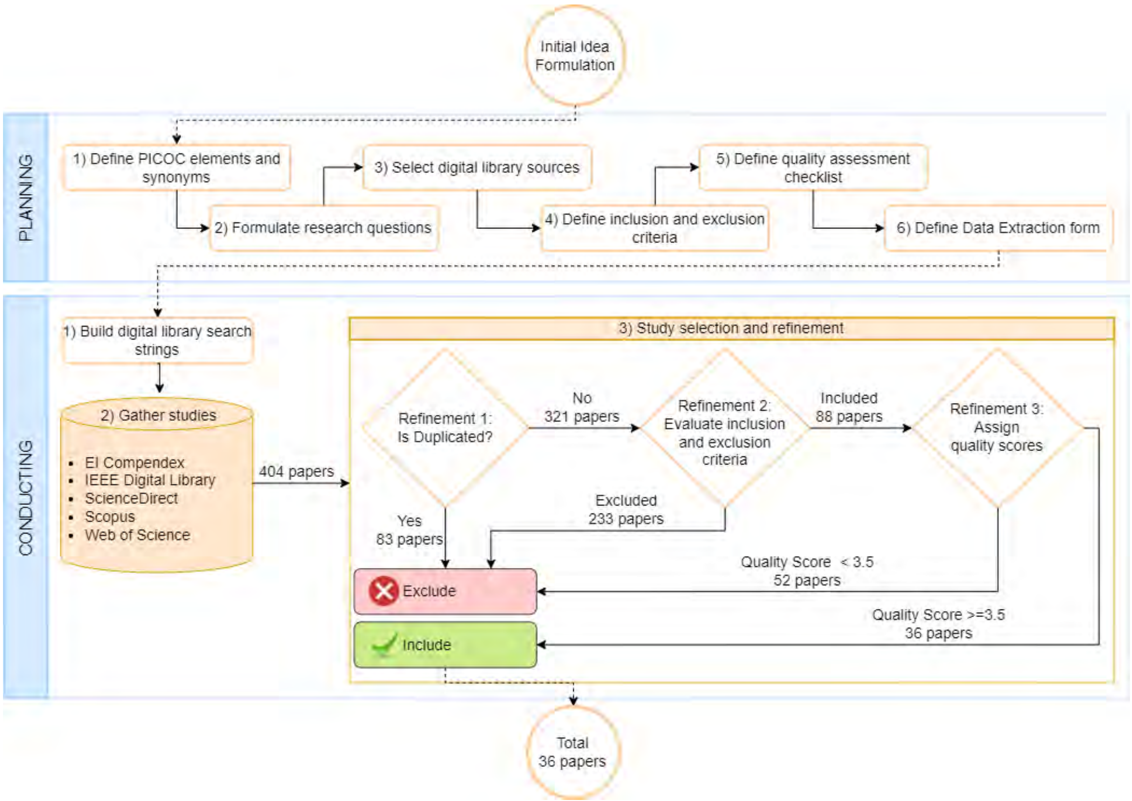


Figure 3.1: The research methodology for the literature review

serves as a research method for delineating the scope of literature reviews. This approach is frequently utilized to identify the different components of a topic, facilitating the formulation of research questions accordingly. Table 3.1 presents the definitions of each PICOC element along with the selected keywords and synonyms. As keywords are derived from the PICOC elements, it is essential to consider synonyms that can be used in queries executed in digital libraries.

Planning Step 2: Formulate research questions

Involves defining the research questions that set the focus for study identification and data extraction [32]. RQs must include the PICOC elements to establish the research focus and assist in identifying the primary studies [114].

In this task, the RQ2 of this thesis is considered as a base to formulate more specific research questions that guide the literature revision. Thus, the RQ2 defined as "How do existing semantic web-based workflow management approaches contribute to enhancing workflow adaptability and flexibility?", was split into three additional questions:

RQ2.1: What active challenges and emerging perspectives are associated with semantic web-based solutions supporting the workflow management life cycle?

RQ2.2: What tools and algorithms are commonly employed in the development of semantic web-based solutions supporting the workflow management life cycle?

Table 3.1: PICOC elements, keywords and synonyms

PICOC	Description	Keyword	Synonyms
Population	The population in which the evidence is collected. Population can be a specific role, an application area, or an industry domain.	Industry 4.0	Digital Factory, Digital Manufacturing, Future Factory, Future Manufacturing, Industrie 4.0, Industry, Smart Factory, Smart Manufacturing
Intervention	Intervention is the methodology, tool, or technology that addresses a specific issue.	Process Modeling	BPEL, BPM, BPMN, Business Process Modeling, Business Process Modelling, YAWL
Comparison	Comparison is the methodology, tool, or technology in which the intervention is being compared.	Semantic Web	Ontology, Semantic, Semantic Web Service
Outcome	Outcomes relate to factors of importance to practitioners. The results that Intervention could produce.	Framework	Extension, Plugin, Tool
Context	This is the context in which the Comparison takes place. Some systematic reviews might choose to exclude this element.	Not applicable	Not applicable

RQ2.3: What evaluation factors are considered when measuring the performance of semantic web-based solutions supporting the workflow management life cycle?

Planning Step 3: Select digital library sources
This step entails selecting suitable digital libraries from which researchers can gather the studies [115]. The validity of an article is dependent on the appropriate selection of the database as it must sufficiently cover the study area. For instance, multidisciplinary databases like Web of Science[3] (WoS) and Scopus[4] provide access to literature in technology, biomedicine, and other fields, making them suitable choices for this study. Additionally, it was also important to include sources relevant to computer science, such as EI Compendex[5], IEEE[6], and ScienceDirect[7].

Planning Step 4: Define inclusion and exclusion criteria
Inclusion and exclusion criteria are established to perform the refinement step for the articles collected from the library sources. Inclusion criteria refer to articles that can be considered for the next refinement step. Exclusion criteria refer to articles that do not fit this research purpose and are therefore removed from the study. Table 3.2 provides the inclusion and exclusion criteria considered for this SLR.

Planning Step 5: Define quality assessment checklist
A quality assessment checklist refers to a set of questions, answers, and scores assigned to each article. Consequently, articles can be sorted by their score and filtered to obtain those that are more closely related to the research domain. Table 3.3 sets out the quality assessment checklist

---

[3] `https://www.webofscience.com`

[4] `http://www.scopus.com`

[5] `http://www.engineeringvillage.com`

[6] `http://ieeexplore.ieee.org`

[7] `http://www.sciencedirect.com`

Table 3.2: Inclusion and exclusion criteria

Type	Criteria
Inclusion	• Strongly related: Abstract indicates that the full text is directly dedicated to Semantic workflows in the domain of Smart Manufacturing. • Partially related: Abstract indicates that some parts of the text are related to Semantic workflows.
Exclusion	• Article is duplicated in another database library • Article is not a journal article nor a conference paper • Article is not written in English • Article was published before 2015 • Semantics and workflows are only mentioned as an example or cited expression • The proposed solution uses semantics but not for workflows

for this SLR.

Table 3.3: Quality assessment checklist

Element	Description
Questions	• Does this article use semantic web technologies for the proposed solution? • Is this article related to workflows, dataflows, or any kind of business process modelling? • Is this article oriented towards industry (Smart Manufacturing)? • Does this article incorporate context awareness as part of the proposed solution? • Does this article propose a framework, tool, or methodology?
Answers	To assign a weight for each question answered. The weight is 1.0 if the answer is "YES", 0.5 if the answer is "Partially", and 0.0 if the answer is "No".
Cutoff score	The maximum score is 5, and the cutoff score is 3.5. Papers whose total score is less than the cutoff score are not considered for the deep review phase.

Planning Step 6: Define Data Extraction form

The Data Extraction form is a visual form composed of input text fields designed to register relevant data from the articles during the review process. This step aids in synthesizing the information from each article and answering the established research questions [116]. Table 3.4 lists the input fields of the data extraction form.

3.3.1.2 Conducting

The Conducting phase involves executing the actual review by considering the previously defined protocols. First, search query strings are built using PICOC elements, then the search string is executed and studies are gathered from the various digital libraries.

Conducting Step 1: Build digital library search strings

Concerns with crafting a search string that integrates the PICOC elements and their synonyms identified earlier, tailored to each chosen database library. The search string structures the population, intervention, comparison, and outcomes within parentheses and connects them using the Boolean operator "AND", while synonyms are linked with the Boolean operator "OR".

Table 3.4: Data extraction form fields

Field Name	**Datatype**	**Value**
Solution type	Select One Field	• Framework, software, tool, algorithm • Methodological approach • None, just a survey or review article
Category Main Focus	Select One Field	• Process Modeling • Knowledge-Management System • Service Composition • Optimization of Service Composition • Reconfigurable Systems • Autonomous Computing
Their identified problem	String Field	n/a
Proposed solution	String Field	n/a
Validation of proposed solution	String Field	n/a
Their conclusion	String Field	n/a
Identified Gaps / Comments	String Field	n/a
Tools used for their proposal	String Field	n/a
Open source	Boolean Field	Yes/No
Link to Source Code	String Field	n/a

- EI Compendex:

```
((((("Industry 4.0" OR "Digital Factory" OR "Digital Manufacturing" OR "Future Factory" OR "Future Manufacturing"
OR "Industrie 4.0" OR "Industry" OR "Smart Factory" OR "Smart Manufacturing") AND ("Process Modelling" OR "BPEL"
OR "BPM" OR "BPMN" OR "Business Process Modeling" OR "Business Process Modelling" OR "Process Modeling" OR "YAWL")
AND ("Semantic Web" OR "Ontology" OR "Semantic" OR "Semantic Web Service") AND ("Framework" OR "Extension" OR "Plugin"
OR "Tool")) WN KY)) AND (((ca OR ja) WN DT) AND (english WN LA)))
```

- IEEE Digital Library:

```
("Industry 4.0" OR "Digital Factory" OR "Digital Manufacturing" OR "Future Factory" OR "Future Manufacturing" OR
"Industrie 4.0" OR "Industry" OR "Smart Factory" OR "Smart Manufacturing") AND ("Process Modelling" OR "BPEL" OR
"BPM" OR "BPMN" OR "Business Process Modeling" OR "Business Process Modelling" OR "Process Modeling" OR "YAWL")
AND ("Semantic Web" OR "Ontology" OR "Semantic" OR "Semantic Web Service") AND ("Framework" OR "Extension" OR "Plugin"
OR "Tool")
AND DOCUMENT TYPES: (Journals OR Conferences)
```

- ScienceDirect:

```
("Industry 4.0" OR "Industry") AND ("BPM" OR "Business Process Modeling") AND ("Semantic web") AND ("Framework"
OR "Tool")
```

- Scopus:

```
TITLE-ABS-KEY(("Industry 4.0" OR "Digital Factory" OR "Digital Manufacturing" OR "Future Factory" OR "Future Manufacturing"
OR "Industrie 4.0" OR "Industry" OR "Smart Factory" OR "Smart Manufacturing") AND ("Process Modelling" OR "BPEL"
OR "BPM" OR "BPMN" OR "Business Process Modeling" OR "Business Process Modelling" OR "Process Modeling" OR "YAWL")
AND ("Semantic Web" OR "Ontology" OR "Semantic" OR "Semantic Web Service") AND ("Framework" OR "Extension" OR "Plugin"
OR "Tool") ) AND ( LIMIT-TO ( DOCTYPE,"cp" ) OR LIMIT-TO ( DOCTYPE,"ar" ) )
```

- Web of Science:

```
TS=(("Industry 4.0" OR "Digital Factory" OR "Digital Manufacturing" OR "Future Factory" OR "Future Manufacturing"
```

```
OR "Industrie 4.0" OR "Industry" OR "Smart Factory" OR "Smart Manufacturing") AND ("Process Modelling" OR "BPEL"
OR "BPM" OR "BPMN" OR "Business Process Modeling" OR "Business Process Modelling" OR "Process Modeling" OR "YAWL")
AND ("Semantic Web" OR "Ontology" OR "Semantic" OR "Semantic Web Service") AND ("Framework" OR "Extension" OR "Plugin"
OR "Tool"))
AND LANGUAGE: (English) AND DOCUMENT TYPES: (Articles OR Proceedings Papers)
```

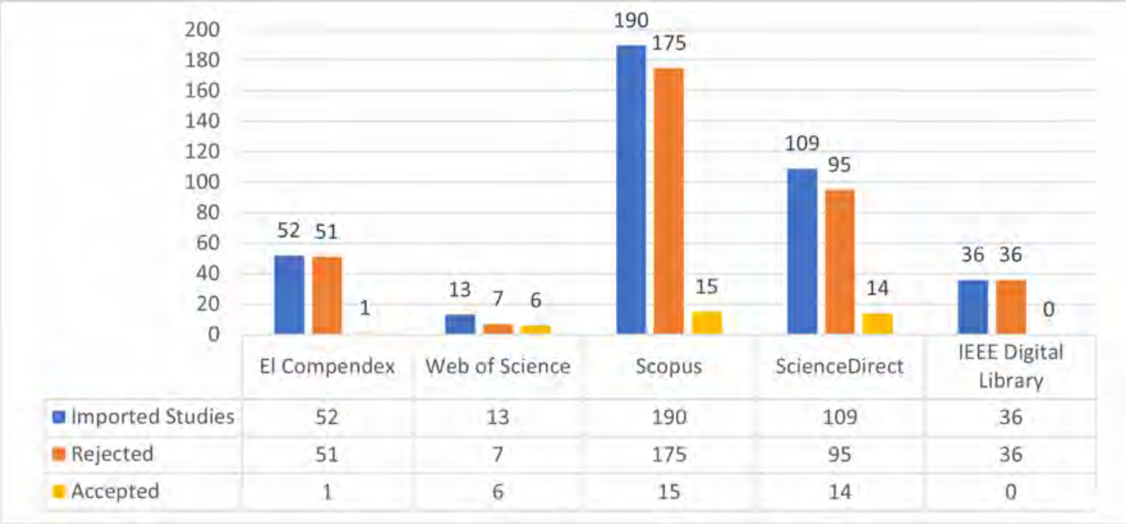
Conducting Step 2: Gather studies

Encompasses executing the formulated search strings in each digital library to retrieve relevant research articles. After executing the search strings, the search results were downloaded as CSV (Comma Separated Value) files, containing metadata such as abstracts, authors, titles, and publication years. This metadata is crucial for subsequent data extraction and both quantitative and qualitative analysis. In total, 404 papers were collected from all databases during this step.

Conducting Step 3: Study selection and refinement

This stage comprises three steps for paper selection and refinement. The first step is to identify duplicates that appear in each of the searches in the selected databases. Automatic procedures are utilized to search and exclude duplicate papers by performing title matching. In this work, the studies were imported into Parsifal to utilize its pre-built functionality *Find_Duplicates*. The tool identified 83 duplicates from the initial 404 papers. The two remaining refinement steps were then executed manually by a single author for each article. Thus, filters with respect to the inclusion and exclusion criteria were applied to the remaining 321 papers during the second step, resulting in 88 papers that passed this filter. In the last refinement step, the remaining 88 articles were assessed for quality, and weights/scores from 1.0 to 5.0 were assigned to each.

After scoring all papers, 36 with a score of 3.5 or higher were selected for the deep review phase. Fig. 3.2 illustrates the distribution of imported, accepted, and rejected studies per digital library.



	El Compendex	Web of Science	Scopus	ScienceDirect	IEEE Digital Library
🟦 Imported Studies	52	13	190	109	36
🟧 Rejected	51	7	175	95	36
🟨 Accepted	1	6	15	14	0

Figure 3.2: Distribution of imported, accepted, and rejected studies per digital library

3.3.2 In-Depth Review of Literature by Categories

The 36 selected articles present approaches that use semantic web technologies in one or more workflow management functional areas. Table 3.5 categorizes the studies based on their main focus and workflow management functional areas. The categories include Process Modeling, Service Composition, Optimization of Service Composition, Reconfigurable Systems, Autonomous Computing, and Knowledge Management systems. Each category is described briefly, giving a general idea of the focus of the studies included in that category. The number of papers in each category is also listed.

The following sections provide an in-depth review of the 36 selected articles by category. Comparison tables are also provided for each category to briefly summarize the main ideas of each paper, use cases (if any), advantages, tools used, and opportunities. This latter highlights weaknesses within the proposals and helps determine current active challenges in this research domain.

Table 3.5: Categorization of selected studies

Workflow Function Area	Category	Focus / Description	Papers
Design-Time	Process Modeling	Studies that focus on the modeling or design of workflow recipes as connected graphs (not necessarily executable).	4
	Service Composition	Studies that focus on the modeling or design of workflow recipes and service discovery and service selection for the final composition (should be executable).	10
Runtime	Optimization of Service Composition	Studies that focus on optimizing workflow recipes at runtime through service re-selection by evaluating Functional Parameters (FP) and Non-Functional Parameters (NFP), also known as Quality of Service (QoS) parameters.	4
	Reconfigurable Systems	Studies that focus on making substantial changes to the control flow through decision-making techniques to achieve the process goal (may require user intervention).	8
	Autonomous Computing	Studies that focus on automatic or dynamic re-composition, optimization, and execution of workflows without requiring user intervention.	5
Management	Knowledge Management Systems	Studies that focus on integrating a knowledge database to store semantics about devices and their services, to query, infer knowledge, and deliver that information to the user.	5

3.3.2.1 Process Modeling

Process Modeling is a discipline that aids in organizing organizational processes through various methods and techniques [117]. Among these techniques, the Flow Chart model stands out as a commonly used method. It employs graphical representations with symbols to depict operations and flow directions, facilitating the definition and analysis of business problems [118]. Table 3.6 offers a comparison of selected approaches categorized under Process

Modeling, with a narrative description provided for the most relevant ones. These studies propose prototypes and implementations that focus on the modeling of workflows enhanced by semantic web technologies.

For instance, Suri et al. [119] proposed a semantic framework for the development of IoT-aware business processes. This approach integrates IoT resources into business processes using an extended version of BPMN 2.0. The approach uses Signavio BPMS to enable semantic annotations of the process model and IoT-BPO ontology to assist the user with ontology annotation.

An approach that utilizes BPMN and ontologies for automatic data processing in gas turbine maintenance was presented by Barz et al. in [120]. The architecture processes semantically annotated data from machines to sensors and analyses data for anomaly detection. This approach employs a semantic knowledge base to store and query data and Camunda BPMN to design and control the process of gas turbine maintenance. Similarly, Kaar et al. [121] developed an ontology that integrates characteristics of the multi-perspective RAMI 4.0 (Reference Architectural Model for Industry 4.0) and proposes combining it with S-BPM (Subject-oriented Business Process Management) to encapsulate industry standards and stakeholder behavior.

Table 3.6: Comparison of process modeling approaches

Ref.	Main idea	Use case	Advantage	Opportunity	Tools used
[122]	A semantic business process modeling approach.	-	BPMN compliant	Need for prototype and experimental phase	• CODO • DOLCE • BPMNO
[121]	A RAMI 4.0-based ontology	-	RAMI 4.0 and S-BPM compliant	Need for prototype and experimental phase	• RAMI 4.0 • S-BPM • C-MAPS tools
[120]	An architecture for machine anomaly detection using a digital technique for filling reports and handwriting recognition.	Siemens gas turbine maintenance	• Automation upgrade • Complexity abstraction	Need for security	• Camunda BPMS • BPMN • MyScript • Blazegraph
[119]	A framework to support the development of IoT-aware business processes.	Monitoring the temperature of orchids	Complexity abstraction	Need for fault-tolerance	• Signavio BPMS • IoT-Lite • BPMN

3.3.2.2 Service Composition

Service composition is a technique that encompasses the discovery, selection, and execution of services to achieve the user's goal. This process involves breaking down the user goal into functional and non-functional requirements or properties (FPs and NFPs) [97]. Table 3.7 presents a comparison of the selected approaches categorized under Service Composition. A narrative description is also provided for the most relevant ones. These studies introduce prototypes and implementations centred around the composition of services leveraging semantic web technologies.

For instance, Bucchiarone et al. [123] proposed an AI (Automated artificial intelligence) planning-based composition framework. This framework enables service discovery, selection, composition, and deployment by using an extension of BPEL in what [124] called *"Adaptable Pervasive Flow Language (APFL)"*. Using APFL activities can be annotated with preconditions and effects at design time. The authors used APFL to create a composition planner module that performs service composition. The algorithm creates abstract activities that are then replaced by "fragments", which are smaller compositions. An AI planning algorithm is employed as the reasoning mechanism to minimize the search space by considering knowledge from previous executions and analyzing context for the reuse of fragments in the final composition.

Kir and Erdogan [125] designed an intelligent business process management framework called "agileBPM". This framework combines ontologies and agents to provide cognitive and exception-handling capabilities by using the Hierarchical Task Network (HTN) planner algorithm to compose services. The algorithm evaluates QoS, business goals, and business rules to perform a search and ranking of the processes that can achieve the user goal. Similarly, Mazzola et al. in their studies [126–128] proposed an architecture and components for semantic web service composition. This approach uses a web service annotation tool that follows the Everything-as-a-Service principle to annotate web services and store them in a common marketplace. BPMN models are created and enriched with annotations to define the semantic behavior of each task in terms of IOPE (Input, Output, Preconditions, and Effects). A pattern-based algorithm is used for the semantic composition of business processes and an optimization phase is included to consider non-functional aspects (QoS constraints) for solving the Constraint Optimization Problem (COP).

3.3.2.3 Optimization of Service Composition

The Optimization of Service Composition is the process of re-selecting services within an existing composition to enhance Quality of Service (QoS) parameters, which are in line with Service Level Agreements (SLA). This process is prompted by the convergence of numerous web services that perform similar functions [134]. QoS parameters, categorized as non-functional properties (NFPs), serve to evaluate services and determine the optimal service for a given task. These parameters are typically represented by numeric values ranging from 0 to 100, encompassing factors like response time, availability, throughput, success rate, reliability, compliance, best practices, latency, relevancy, and class (such as highest, lowest, platinum, or gold). Table 3.8 provides a comparison of the selected approaches categorized under Optimization of Service Composition. A narrative description for the most pertinent ones is also provided.

For instance, Baker et al. [135] developed an optimal service composition algorithm specialized in energy efficiency. The algorithm focuses on reducing the number of services used in the composition problem, which also results in a reduction in energy consumption. The approach incorporates semantic web technologies to describe services. An initial composition

Table 3.7: Comparison of service composition approaches

Ref.	Main idea	Use case	Advantage	Opportunity	Tools used
[126] [127] [128]	An architecture for semantic service annotation.	• Aluminum forging for bicycle hull body forming • Preventive maintenance	• Robust architecture • Service marketplace • Complexity abstraction • BPMN compliant • Open-source	• Complex to set-up • Need to read sensor logs	• OWL-S • Docker • Java • BPMN
[129]	A hybrid composition framework (top-down and bottom-up).	BPEL compliant	• Open-source • Interactive UI	Need to handle NFPs	• BPMN to BPEL library • Petals BPM, EasierSBS, XPath language, SAWSDL
[123]	An AI planning-based composition framework.	Process chain of the car logistics	• Context-awareness • Interactive UI	Need for security	• BPEL, ASTRO CAptEvo framework • AI planning algorithm as the reasoning mechanism
[130]	Location-aware discovery mechanism for the IoT.	A personal assistant that recruits services based on their location.	Interactive UI	• Need to handle location change scenarios • Need to capture GPS coordinates	• Schema.org, Protégé, Hydra vocabulary, JSON-LD • Phyton, rdflib, rdflib_jsonld, Elixir, Apache Jena Fuseki
[131]	A semantic Node-RED version.	FESTO Process Automation Workstation	• Complexity abstraction • Open-source • Interactive UI	Need for security	• Node-RED • iot.schema.org, JSON-LD, SPARQL
[132]	An ecosystem for the discovery and composition of IoT.	Smart city	• Open-source • Robust architecture • Security	-	• O-MI, O-DF, Schema.org, MobiVoc • SQL, NoSQL • OAuth, OpenID, SAML, LDAP, JWT
[125]	A BPM framework that combines ontologies and agent-based process execution to provide decision-making capabilities.	-	• Context-awareness • Open-source • Fault-tolerant	Need for security	• BPMO, Enterprise Strategy Ontology • SWRL, WSMO, OWL-S, OWLS-TC4 • HTN Planner, Wade platform
[133]	A framework for the modeling and execution of IoT business processes without changing the meta-model of BPMN.	-	• Context-awareness • Open-source • BPMN compliant • Microservice-oriented architecture	Need for security	• SOSA ontology, SWRL, protege, SPARQL • BPMN, API, Camunda, Java, .Net

plan is created by considering various services from different providers. The algorithm then performs optimization by ordering the list of service providers such that those with the largest number of services are placed first. The optimum individual service or composite service is then selected considering QoS conditions. Finally, the re-composition is performed by taking the optimal services.

An optimal semantic web service composition approach was presented by Bekkouche et al. in [136]. Their approach takes into account NFPs and QoS constraints for the optimization problem, which is solved by applying the Harmony Search (HS) algorithm. The approach is tested using the Web Service Challenge (WSC) 2009 dataset. Similarly, Abid et al. [137] proposed an optimal semantic web service composition framework that first classifies, discovers, and composes services. Optimization is then performed by using semantic similarity measures, considering FPs and QoS parameters to calculate the optimal services.

Table 3.8: Comparison of optimization of service composition approaches

Ref.	Main idea	Use case	Advantage	Opportunity	Tools used
[138]	Optimization based on FPs and NFPs in design-time and runtime by using the COP algorithm.	• Aluminum forging for bicycle hull body forming • Preventive maintenance	• BPMN compliant • Open-source	-	• antlr, JaCoP solver, OWL-S • BPMN • COP algorithm
[135]	Reducing the number of services for the composition problem can turn in a reduction in energy consumption.	-	-	Need to consider more evaluation factors such as memory, CPU, size of data sent and received, and execution time.	• OWL-S XPlan package • Java, NetBeans
[136]	Optimization based on NFPs and QoS constraints by using the Harmony search algorithm.	-	Considers a well set of QoS attributes: Response time, cost, availability, reliability, and reputation.	Need to standardize the recipe using notation languages	• Harmony search algorithm • WSC 2009 dataset
[137]	Optimization based on FPs and NFPs by using semantic similarity measures.	-	-	Need to standardize the recipe using notation languages	• Java, Jena, and JAXP • OWL-S, SWS Test Collection

3.3.2.4 Reconfigurable Systems

A Reconfigurable System refers to a system capable of dynamically reconfiguring its elements at runtime, including applications, platforms, system architectures, underlying infrastructures, and management facilities [139]. This reconfiguration typically occurs during system maintenance or when installing new updates, sometimes necessitating user intervention. Table 3.9 presents a comparison of the selected approaches categorized under Reconfigurable Systems, along with narrative descriptions of the most relevant studies. These studies propose prototypes and implementations focused on reconfiguring systems to enact significant changes to the control flow, aiming to achieve user goals through decision-making, adaptive case management, and machine-learning techniques.

For instance, Ciasullo et al. [140] presented a framework to support group decision-making problems in the context of business process outsourcing by selecting the most suitable provider. The framework includes a "time-aware" feature to consider the time elapsed from past executions. The framework relies on the fuzzy linguistic consensus model (presented in [141, 142]) and a context-aware plugin (presented in [143]). A reinforcement learning algorithm is implemented to learn and assign weights/scores to support the users in the decision-making process, by considering the current context and the time at which they take part in the decision-making group.

A context-aware BPM ecosystem is proposed by Song et al. in [144]. This proposal integrates IoT data into context ontologies to enhance business process decision-making. Using the framework, business processes are aware of their context at both design time and runtime and thus can adapt to dynamic situations. The architecture consists of business process models and contextual entities including their relationships. Context is sensed from a variety of sources including IoT devices and information systems, and then analyzed to output understandable contextual knowledge. Decision models are then created to take this contextual knowledge into account. For the execution phase, the business processes and decision models are deployed and executed accordingly. In a similar vein, Ordoñez et al. [145] developed a framework for automating monitoring and decision support. The framework employs Natural Language Processing (NLP) for the creation of a "problem file" which defines a sequence of actions to achieve a goal. A monitoring module gathers logs generated during execution. A decision support module then takes the logs and determines whether there are real errors or not, by using an inference engine to change the service in charge if needed.

Table 3.9: Comparison of reconfigurable systems approaches

Ref.	Main idea	Use case	Advantage	Opportunity	Tools used
[146]	An ontology for reconfigurable system integration and decision-making support.	Festo test bench	-	-	• PPRR ontology • GATE RDF editor, RDF, SPARQL
[147]	An ontology for ACM systems that allows business partners to describe business cases using natural language.	Approve an architectural model	Adaptations are immediately available to the business users	Requires coordination between business users	Papyrus Converse
[148]	A context manager module to decide whether a service should be kept, tuned, or changed on the fly.	A car-seat example picked from a FabLab	• Considers FPs and NFPs, the inputs, and outputs of services • Context-awareness • On-the-fly service replacement	Re-composition phase still needs development	• Java, Jena API, IntelliJ IDEA • Ontologies: SSN, FIPA CSIRO, SDO
[144]	A context-aware BPM framework for making adaptations in dynamic situations.	Truck pick-up cargo	BPMN and DMN compliant	Need for a standard architecture	• OWL, SWRL • DMN
[149]	An assistant for engineering change processes.	A product service system engineering process	WS-BPEL compliant	• Complex to maintain • Low scalability • Need for security	• .Net framework • Goal-oriented process modeling language (GRL)
[150]	A decision support system based on the "Ant Colony Optimization" algorithm.	• Place order in wholesale of food • Manufacture of chocolate	BPMN compliant	Need for security	• Business Field Ontology, Collaborative ontology • Java, Neo4j, Ant Colony Optimization
[140]	A context-aware framework for group decision-making in business process outsourcing.	Decision-making on who to outsource the production sub-process of an Italian footwear company.	Context-awareness	Need to consider more context-awareness properties	• Fuzzy linguistic consensus model • Reinforcement learning algorithm • OWL2, SKOS
[145]	A monitoring and decision support module to determine whether a service should be replaced.	-	Considers error logs	Need for security	• Natural Language Processing, AI planning • Java, JSLEE, ITIL • SOA Ontology, OWL, OWL-API, SWRL, Pellet reasoner

3.3.2.5 Autonomous Computing

Autonomous Computing is a research area centred on self-adaptive and autonomic computing systems capable of configuring, healing, optimizing, and protecting themselves autonomously, without human intervention [151]. Table 3.10 offers a comparison of the selected approaches categorized under Autonomous Computing, which concentrate on the automatic or dynamic composition, optimization, and execution of services. Furthermore, a narrative description of the most relevant studies is provided.

For instance, Alférez and Pelechano [152] proposed a context-aware framework for the autonomic adjustment of service compositions at runtime. The approach creates models at design time using ontologies and BPMN and WS-BPEL to support dynamic adjustments of service composition. A verification phase checks the changeability of the models and their configurations before the execution to ensure safe service re-composition. At runtime, whenever a context event happens, the previously created models are queried to perform service re-composition.

An approach to autonomous computing by applying the MAPE-K (Monitor, Analyze, Plan, Execute, and Knowledge) paradigm was designed by Lam et al. in [153, 154]. This approach relies on an "Autonomic Manager" that facilitates the development of interoperable IoT systems using semantics web Technologies. The "Autonomic Orchestration" uses the semantics of the "Autonomic Manager" to enable the dynamic orchestration of services within Arrowhead Framework[8]. The architecture integrates a Semantic Extractor (SE) that gathers information from services and systems and transforms them into semantic knowledge. The SE uses Machine Learning (ML) techniques to recognize popular semantic models and extract the necessary information. A knowledge graph is then built from the chosen ontologies and the information is stored in a knowledge base. An Autonomic Framework (AF) employs SPARQL and SWRL as reasoning mechanisms for knowledge inference and decision-making at runtime.

Arul et al. [155] proposed a framework for automatic and dynamic service composition by adding semantics to web services to satisfy the dynamic changes of user needs. They adopted an extended version of the hierarchical task network (HTN) planner, named hierarchical task network based on user constraints (HTNUC) which can understand OWL-S processes. This proposal takes user requirements and constraints to produce an abstract composite workflow that determines the execution order of tasks. An optimal abstract composite workflow is then created to satisfy the user goals. Ontology-based search is employed to convert syntactic to semantic definitions of services. Finally, the most suitable service candidates are selected using semantic web service discovery (SWSD) to produce a concrete-executable workflow from the discovered services.

3.3.2.6 Knowledge Management Systems

The primary function of Knowledge Management Systems is to capture both product and process information for later use in automating processes. These systems typically store data in a knowledge database and derive new insights by applying complex rules, heuristics, artificial intelligence, and agents [157]. Table 3.11 offers a comparison of the selected approaches categorized under Knowledge Management Systems. Furthermore, a narrative description of the most relevant studies is provided. These studies present prototypes and implementations

---

[8] https://www.arrowhead.eu/

Table 3.10: Comparison of autonomous computing approaches

Ref.	Main idea	Use case	Advantage	Opportunity	Tools used
[156]	A semantic engine for the dynamic execution of business processes.	Create Order business process	Context-awareness	-	• PrjOnt, SWRL rules • WSO2 application server • Activiti
[152]	A context-aware framework for the autonomic adjustment of service compositions at runtime.	-	Context-awareness	• Needs to consider more context-awareness properties. • Needs a revalidation step to ensure that a service is actually unavailable.	• Goal/ Question/ Metric (GQM), MAPE-K paradigm • FAMA-FW, GNU Prolog, BPMN, WS-BPEL • Java, Java VisualVM, Apache Axis2, Apache Tomcat, Apache ODE
[153] [154]	An autonomous computing framework using the MAPE-K paradigm for dynamic orchestration of services within the Arrowhead Framework.	Smart home	• Arrowhead compliant • Security (handled by Arrowhead Framework)	Needs to support human intervention for execution confirmation	• MAPE-K paradigm, Machine Learning techniques • IoT-O, Semantic Sensor Network (SSN), Semantic Actuator Network (SAN), SPARQL, SWRL, RDF, OWL
[155]	A framework for automatic and dynamic service composition adding semantics to web services and satisfying the dynamic changes of user needs.	Finding the most suitable composite solution	Context-awareness	Need for security	• HTN planner • OWL-S, WSMO, OWL-API, Pellet reasoner • Extended Finite-State Automaton (EFA), WSC 2009 dataset

that integrate knowledge databases into systems to store and process the semantics of devices and services.

For instance, Borsato [158] proposed a semantic information model based on an ontology for the calculation of energy efficiency indicators. A BPMN is first created and then divided into activities until the elementary tasks are reached (top-down approach) so that all BPMN entities can be mapped into the ontology as individuals and assertions. Then, the inference and query capabilities of Protégé and RacerPro are employed. Each task is accomplished by an operation, which corresponds to a manufacturing unit process. Their solution provides parameters to a manufacturing unit process by bringing data property assertions for a given individual.

A semantic and context-aware framework for the smart home scenario is presented by Elkady et al. in [159]. The framework gathers data from IoT devices and employs ontologies to interpret context, considering diverse types of context data such as temperature, location, date, and time. An API (Application Programming Interface) provides abstract access to the backend and also delivers contextual information. The results show that their approach can make predictions based on context changes. Similarly, Hippolyte et al. [160] proposed a semantic and context-aware framework for the energy management field. Ontologies are employed to describe the semantics of devices, and a web service generator tool creates web services from instances of the ontology. The tool embeds SPARQL queries and an API is used to facilitate data extraction from the knowledge base with HTTP/REST instead of HTTP/SPARQL.

Table 3.11: Comparison of knowledge-management systems approaches

Ref.	Main idea	Use case	Advantage	Opportunity	Tools used
[161] [162]	A semantic wiki-based system.	Austempered Ductile Iron	Open-source	Need for an API	Loki semantic wiki
[163]	A knowledge base for optimizing engineering processes.	Engineering of robot-based automation solutions at a German automotive supplier	Complexity abstraction	Complex to maintain	• Microsoft Excel • VBA programming language
[158]	An ontology and knowledge base for computing energy efficiency indicators.	A welded hull panel assembly	Inference capabilities	• Need for real-time data • Need for UI to deliver results	• Protégé • RacerPro • Bonita BPM
[159]	A framework to support the development of context-aware IoT applications.	Smart home	• Complexity abstraction • Customizable services • API • Context-awareness	-	-
[160]	An API to obtain the semantics of devices.	Energy management systems for decision support	Complexity abstraction	Need for security	• Java, JAX-RS, JAXB, OWL2, SPARQL, GraphDB • RDF/XML Validator, OWL Validator, TopBraid Composer

3.3.3 Discussion and Findings

This section examines the gaps identified during the literature review and synthesizes them into the most critical points. To achieve this, research questions RQ2.1, RQ2.2, and RQ2.3, are addressed by providing a discussion of the most important findings. Furthermore, a keyword correlation analysis is presented to uncover trends and topics warranting further exploration.

3.3.3.1 Analysis and Answers to Literature Review Research Questions

The research questions are addressed here by identifying current gaps within the studied proposals in the literature. This analysis also extends to the tools and algorithms utilized by authors in these proposals, uncovering prevalent trends. Additionally, the key metrics and use cases commonly used for evaluating the performance of these kinds of approaches are identified.

> RQ2.1: What active challenges and emerging perspectives are associated with semantic web-based solutions supporting the workflow management life cycle?

The reviewed studies pose various challenges. To answer this question, the most significant challenges drawn from the literature revision are listed and discussed next.

(a) Need for standardization in process modelling and service-oriented architectures

Many of the reviewed papers suggest incorporating new process modelling languages to enhance the interpretability of recipes for semantic reasoners and workflow management software. In contrast, the example shown in [164], demonstrates that existing process modelling languages like BPMN and BPEL can support additional features and customized elements and attributes. In addition, the work presented in [133], emphasizes that maintaining a high level of expressiveness in existing notation languages is crucial.

Regarding standardization in service-oriented architectures, many of the reviewed prototypes are presented as non-industry-oriented architectures, lacking support for the critical features that workflow managers should address such as those identified in Section 3.2. In contrast, adopting a service-oriented architecture (SOA) in combination with Industry 4.0 standards like RAMI 4.0, offers numerous advantages, including a higher degree of integration, automation, wider reachness, security, and future-readiness, as outlined in [165]. Some industrial IoT frameworks that combine Industry 4.0 architectural references and SOA are: Arrowhead Framework[9], AUTOSAR[10], and FIWARE[11]. A comparison of recent platforms and state-of-the-art industrial IoT frameworks is detailed in [166].

Furthermore, RAMI 4.0 was introduced to standardize interoperability between machines [1]. This architectural model digitally represents Industry 4.0 components through a three-dimensional layer model. To achieve this, the concept of Asset Administration Shell (AAS) is introduced, which enables the interoperation and identification of assets within the network by providing their technical and operational data. More details about AAS can be found in Plattform i40 [47].

(b) Need to assist process designers in manufacturing process modelling
In the context of BPM, the design of business processes is a task handled by process designers. Often BPM is applied to systems in the domain of Banking, Enterprise Resource Planning, or similar management software [167]. However, some authors in the literature have found the need to include process designers in the domain of manufacturing systems for the design of manufacturing business processes. Business process designers often do not possess the appropriate knowledge or tools to configure machine services or orchestrate processes using such services. Therefore, the challenge lies in providing tools to aid process designers in the design of manufacturing business processes. Considering that AAS describes machine services by using submodels [1], an AAS-based modelling tool can be developed offering asset services in the modeller palette, thereby facilitating the design of manufacturing business processes.

(c) Need for an edge-compliant process manager software
Most existing prototypes for workflow managers are delivered as central-based architectures, with data processing mainly handled in a central server. Edge computing has become a popular paradigm in the industrial domain, as it delivers low latency, more mobility, and more contextual information by connecting cloud computing facilities to end-users [131]. Therefore, it becomes crucial to develop business process management software that can orchestrate workflows in the edge environment.

---

[9] `https://www.arrowhead.eu/arrowheadtools`

[10] `https://www.autosar.org/`

[11] `https://www.fiware.org`

(d) Need to interpret contextual information to perform dynamic adaptations

This literature review analyzed studies in which the context-awareness feature was present in 16 out of 36 proposals. Context-awareness is an important characteristic of ubiquitous computing as it permits the interpretation of contextual information [168]. In a workflow instance, context-awareness can be achieved by capturing data from devices and processing it through the use of semantic web technologies [16]. Fig. 3.3 illustrates the application of context-awareness by workflow function area. For instance, in the design-time phase, IOPE is the main technique for enhancing workflow design. This was demonstrated in the work of Bucchiarone et al. [123], which achieved semi-automatic workflow modelling. In the runtime phase, optimization based on Quality of Service (QoS) is the main technique, as reported by Bekkouche et al. in [136] who replaced services within workflows at runtime. Knowledge management using semantic wikis is the most common technique in the management phase. In this regard, Elkady et al. [159] proposed a semantic-based framework that delivers context information to the user interface in the form of reminders or notifications triggered by context changes.

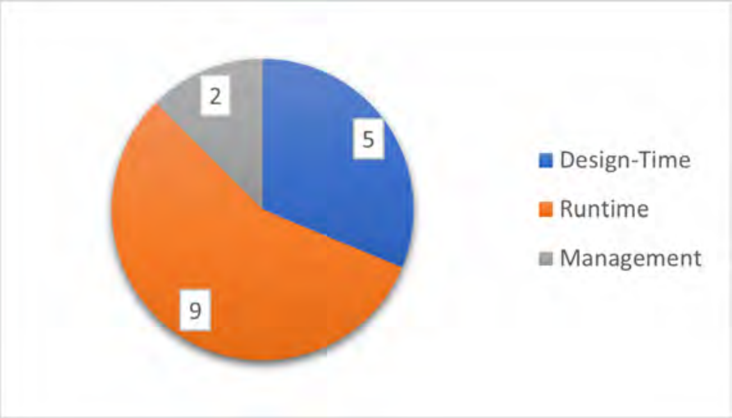


Figure 3.3: Distribution of Context-Awareness feature by Workflow Function Area

Although some authors enabled context-awareness in their proposals, this should be further leveraged by considering sensor logs. As described in Section 2.4, data delivered by sensors can be exploited using ontologies. Attributes of ontologies that can interpret contextual information include `responseTime`, `gpsPosition`, `successRate`, `networkLatency`, `hasWeight`, `hasSize`, `hasTemperature`, and `hasHumidity`.

To effectively adapt to unexpected events during the execution of tasks, it is crucial to gather and analyze data in real-time to create a plan. One approach is IBM's MAPE-K reference model for building autonomous systems, as outlined in Sub-section 2.3.2. In the monitoring phase, IoT sensors can be utilized to capture context data, which can then be processed through semantic web technologies such as OWL, RDF, SWRL, and SPARQL. During the planning phase, decision tables can be created using the Decision Model and Notation (DMN) standard. By combining DMN with BPMN, the workflow execution can be dynamically altered to ensure standardized decision-making.

RQ2.2: What tools and algorithms are commonly employed in the development of semantic web-based solutions supporting the workflow management life cycle?

The reviewed studies proposed prototypes and implementations that are built on top of semantic web technologies to support the stages of the workflow life cycle. Table 3.12 summarizes the semantic web tools employed to develop each of the proposals. The complete list of the tools and ontologies can be found in the tables published in Section 3.3.2, organized by category.

Table 3.12: Tools and techniques employed by the studied approaches

Domain	Description	Tools
Ontology	Defines and models sensors, services, the data they gather, and their domain.	- SSN - SOSA - OWL-S - IoT-O - SOA Ontology - Schema.org
Semantic reasoners	Computes context data and infers new knowledge.	- RacerPro - Hermit - Pellet - Jena - FaCT++
Other reasoners		- AI Planning - Harmony Search - Ant Colony Optimization - Constraint Optimization Problem
Semantic Graph Database	Integrates heterogeneous data from many sources and creates links between datasets. They support SPARQL queries and SWRL rules.	- GraphDB - Apache Jena Fuseki - Stardog - AllegroGraph
Workflow Management	Includes tools for workflow design, execution, and monitoring.	- Camunda - Signavio - Bonita - Activiti

RQ2.3: What evaluation factors are considered when measuring the performance of semantic web-based solutions supporting the workflow management life cycle?

To answer this question, the datasets and measurement variables that the articles under review employed for the experimental phase of the proposals were collected. Of the 36 reviewed studies, 30 presented evaluation phases of the proposals. Fig. 3.4 shows the top 5 measurement variables employed to measure the performance of their approaches. *Execution Time* scored the highest and was considered in 12 papers, followed by *Precision/ Recall/ F-Score* with 5, and *Number of Services* with 3. *Precision/ Recall/ F-Score* are performance metrics (between 0 and 1) often related to classification methods. The authors under review applied these metrics to evaluate the quality and quantity of the data retrieved from collections. A higher *precision* indicates that an algorithm returns more relevant results than irrelevant ones, and a higher *recall* means that an algorithm returns most of the relevant results (regardless of whether irrelevant results are also returned). *F-Score* provides a single score that balances both the concerns of *precision* and *recall* in one number. On the other hand, context-aware variables were not evaluated: `response time`, `location`, `availability`, `throughput`, `success rate`, `reliability`, `compliance`, `best practices`, `latency`, `relevancy`, and `class`.

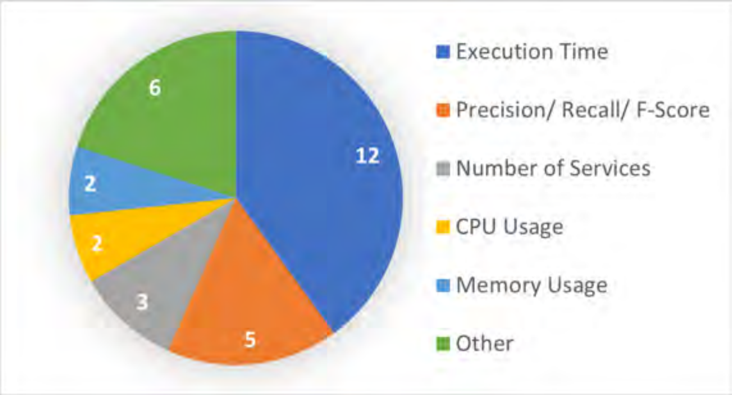


Figure 3.4: Distribution of measurement variables employed in the articles under review to evaluate the proposals

Fig. 3.5 summarizes the use cases that the authors considered for testing their proposals. For instance, in the use case "Truck pick-up cargo", Song et al. experimented with a context-aware BPM framework for making adaptations in dynamic situations. In the case of "Manufacture of chocolate", Montarnal et al. proposed and tested a decision support system. With respect to data, Table 3.13 lists the datasets utilized by the authors for the evaluation phase. These are collections of semantic web services that include QoS parameters.

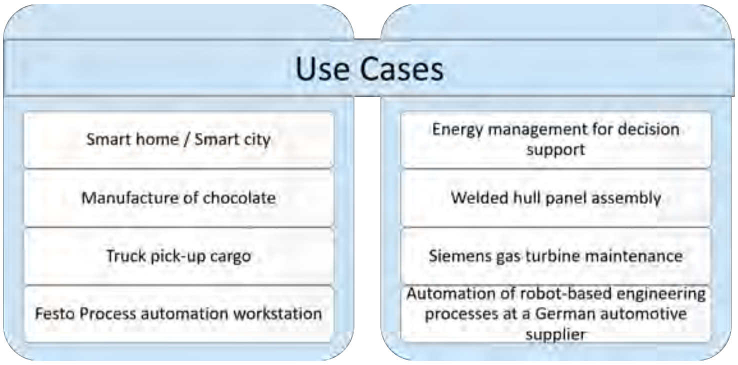


Figure 3.5: Use cases employed in evaluating the proposals under review

---

[12] A GitHub repository with shared resources for testing semantic web services technologies that contain most of the other mentioned datasets

Table 3.13: Datasets employed for the experimental phase

Proposal	Dataset	Link
[159]	Aruba dataset	`http://casas.wsu.edu/datasets/`
[136, 155]	WSC 2009 dataset	`http://www.wschallenge.org/`
[137]	SWS Test Collection[12]	`https://github.com/kmi/sws-test-collections`
[125]	OWLS-TC4	`http://projects.semwebcentral.org/projects/owls-tc/`
[135]	OWL-S XPlan package dataset	`https://www.dfki.de/klusch/owls-xplan/OWLS-XPlan-1-Manual.pdf`
[129]	SAWSDL test collection	`http://projects.semwebcentral.org/projects/sawsdl-tc/`

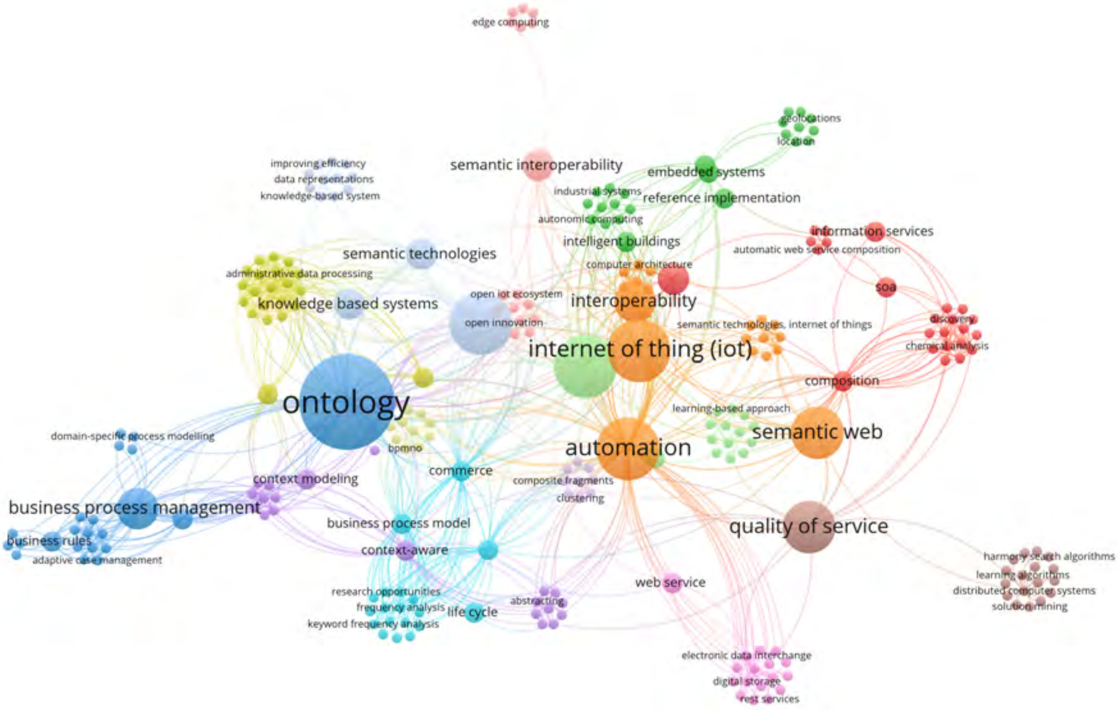


Figure 3.6: Keyword correlation analysis: Identifying the most prominent themes in the literature

3.3.3.2 Keyword Correlation Analysis

This subsection reveals topics of study in this literature review. The study utilized VOSviewer[13], a popular software in the scientific community for constructing and visualizing bibliometric networks as cluster maps. A comparison of VOSviewer and CitNetExplorer and the description of the clustering algorithms can be found in [169]. For this analysis, VOSviewer was fed with a dataset of indexed keywords of the papers that passed the 2nd refinement phase (88 papers). The data was manually polished to remove extra white spaces and then exported to a CSV file to serve as input for the software.

In VOSviewer, each circle represents a keyword, and the size of the circle indicates the number of papers that contain that keyword in their title or abstract. The connections, represented by lines, illustrate the frequency with which two keywords co-occur in the same

---

[13] `https://www.vosviewer.com/`

papers. A line between two circles signifies that these keywords are often found together in the literature. The thickness or prominence of the line corresponds to the strength of their co-occurrence, with a minimum strength set at 1. Thicker or more prominent lines indicate a higher frequency of co-appearance. This means that a line (link) is drawn between two keywords if they appear together in at least one paper.

As shown in Fig. 3.6, the most prominent theme in the literature is *ontology*, represented by the blue cluster of 22 items. This is followed by the combination of *internet of things* and *semantic web*, represented by the orange cluster of 19 items. On the other hand, *edge computing* and *semantic interoperability* (top centre, in pink), is the smallest cluster with only 7 items and limited connections to larger clusters. In addition, no direct link appears to exist between *edge computing* and *business process management*, which suggests a potential research opportunity in this area.

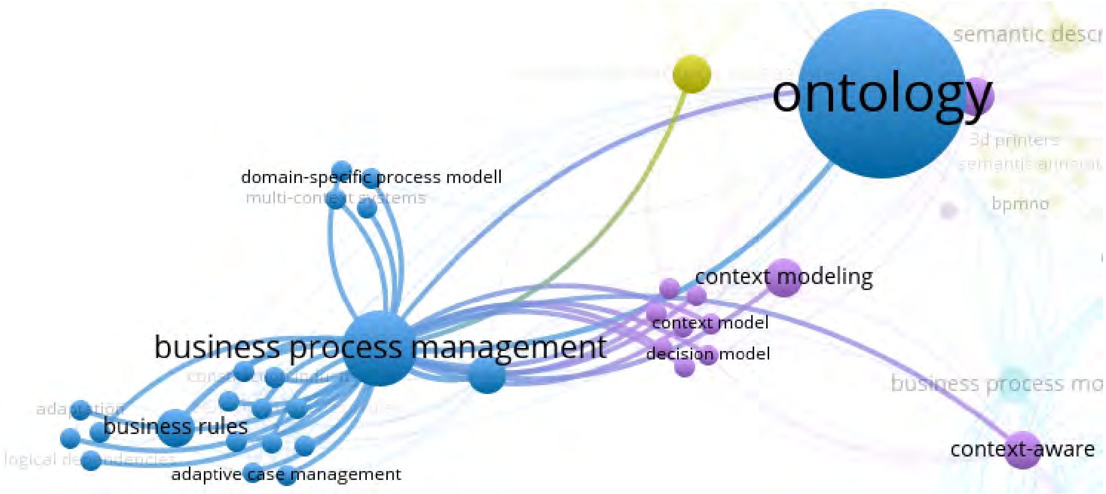
In addition, Fig. 3.7 visually represents the correlation between the "business process management" and "context-aware" clusters. Within the "business process management" cluster, there are 33 links, while the "context-aware" cluster has 25 links. Notably, there are 10 links that specifically connect both clusters, indicating research efforts on both topics combined (papers that contain keywords from both clusters). Furthermore, it is noteworthy that there exists an indirect association between the "context-aware" and "semantic web" clusters. This connection is established through the relationship between "context-aware" and "quality of service", revealing a potential intersection of research interest in combining "contex-aware" and "semantic web" through the use of "quality of service" techniques.

In summary, this review has identified the current tools and algorithms employed in the development of semantic and context-aware smart manufacturing solutions. Workflow management systems are particularly utilized to manage manufacturing processes, with BPMN as the standard notation language. Notably, the incorporation of semantic web technologies contributed to dealing with the heterogeneity of diverse machines through the use of ontologies. Nevertheless, this literature review reveals a lack of prototypes that incorporate service-oriented architectures and Industry 4.0 standards.

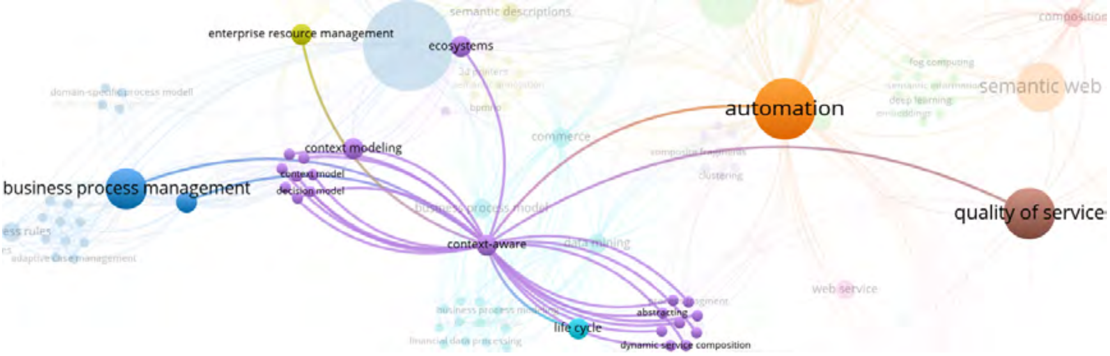
Moreover, the review underscores the significance of context-awareness as a critical feature for enhancing the flexibility and adaptability of manufacturing processes. This finding and the challenges identified during this phase serve as a basis for the subsequent phase, which offers a comparative analysis of approaches that specifically incorporate context-awareness capabilities.

3.4 Phase 2: Analysis of Context-Aware Workflow Management Approaches

This section presents a comparative analysis of context-aware workflow management approaches. Furthermore, this section aims to answer the RQ3 of this thesis established as "What are the existing challenges and limitations in the state-of-the-art with respect to context-



(a) "business process management" cluster



(b) "context-aware" cluster

Figure 3.7: Keyword correlation analysis: Identifying connections between the "business process management" and "context-aware" themes

aware workflow management approaches and their impact on the industry?". To answer the research question, this analysis builds upon the previous section wherein it was determined that the context-awareness feature is present in 16 out of 36 semantic web-based workflow management proposals. The most relevant ones are analysed in this phase.

3.4.1 Comparative Analysis

The characteristics of existing approaches in the field of context-aware workflow management are compared in Table 3.14. 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 [170], or data-driven designs that rely on data analysis techniques [171] to automatize workflow modelling. 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 [172].

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 [173]. 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 [174]. 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 the table propose innovative approaches that leverage a diverse range of techniques and frameworks to address the diverse challenges faced in workflow management. The following enumeration analyses the studies by categorizing them based on their main technique for achieving flexibility and adaptability in workflows.

(I) Workflow composition and recomposition
One common theme among the reviewed studies is the use of context-awareness for workflow composition and recomposition, allowing the system to handle exceptional situations by adapting workflows in response to context changes. 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 [176].

For instance, the work of Bucchiarone et al. [123] and Alférez and Pelechano [152], 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) [177], 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

Table 3.14: Comparison of context-aware workflow management approaches.
✓: The approach entirely covers the feature
+/-: The approach covers the feature with limitations
×: The approach does not cover the feature

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
[152]	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.
[125]	BPMO (Business Process Modeling Ontology)	×	✓	✓	×	×	✓	Cost efficiency, product stock availability.	×	Sales Orders	× Synthetic Dataset	Task failure rate
[155]	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.
[148]	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.
[144]	BPMN, DMN	✓	✓	✓	×	×	✓	×	Temperature, Humidity, Smoke, Weight, and GPS.	Pickup Cargo	×	×
[128]	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
[123]	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.
[135]	BPMN	×	✓	✓	×	✓	×	Energy consumption	×	Workflow composition	× Synthetic Dataset	Running time to achieve service composition plan
[136]	Not specified	×	✓	✓	×	×	✓	Cost, availability, reliability, and reputation	×	Workflow composition	×	The score of the optimal composition, execution time, and deviation.
[133]	BPMN	✓	✓	✓	×	×	✓	Arrival time	Temperature, location	Smart Irrigation System, Ventilation System, Health Care System	×	×
[175]	BPMN	+/-	✓	✓	×	×	✓	×	Machine Status (whether it is on or off)	Fischertechnik Smart Factory	✓	Composition time vs number of services. Search time for the best service.

workflow management, enabling more intelligent and automated workflow adaptation. In this regard, Arul and Prakash [155] and Mazzola et al. [128] 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. [136] 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 [125] 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. [133] 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. [144] 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. [175] 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. [148] 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 micro-service 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 among a pool of candidates, using various algorithms to allocate resources of the highest quality. For instance, in [178], 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 a min-max-based workflow scheduling algorithm called Look-Ahead Particle Swarm Optimization (LAPSO) [179]. 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 [180]. 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 [181], 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.

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.

3.4.2 Discussion and Findings

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 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.

It is worth mentioning that AAS could play a key role in context-aware workflow management, as previously highlighted in Section 3.3.3. AAS provides a standardized means for digitally representing and describing machines. For instance, the TechnicalData [52], NamePlate [51], and CapabilitiesSkillsServices [182] 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.

However, during this phase, 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) Adoption of user-centric design approaches, transitioning from abstract workflows to executable workflows. Another notable area is (5) Development of decoupled components and systems, providing more versatility and integration with existing systems, as current approaches often lack decoupling between various components such as the workflow modelling 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.

3.5 Conclusions

This chapter provided several phases to rigorously analyse the state-of-the-art in workflow management, serving as a starting point for this thesis.

Phase 0 provided the industry-related challenges that workflow managers should address. These challenges encompass various facets, including dealing with workflow modelling, heterogeneity, collaboration, parallel and asynchronous task execution, and microservice architectures. This phase aligns with research objective O1, “Identify industry-related challenges for efficient workflow management”, concurrently addressing RQ1, “What are the key industry-related challenges associated with workflow management?” in Section 3.2 and delineating the stage for the subsequent phase.

Phase 1 presented a methodology for conducting the literature review and executed the actual literature revision in Section 3.3. This phase scrutinized semantic web-based workflow management approaches within the smart manufacturing domain. It corresponded with research objective O2, “Identify current challenges in the state-of-the-art regarding the implementation of context-awareness to support workflow management through a systematic literature review, with a special focus on semantic web-based approaches”. Simultaneously addressing RQ2, “How do existing semantic web-based workflow management approaches contribute to enhancing workflow adaptability and flexibility?”. To facilitate conducting this study, RQ2 was further divided into three sub-questions (RQ2.1, RQ2.2, and RQ2.3). These sub-questions were addressed through exhaustive discussions and a keyword correlation analysis, leading to the identification of the most significant active challenges in the literature.

These challenges encompass various aspects, including standardization in process modelling, incorporation of industry-oriented architectures, compatibility with edge and cloud infrastructures, and context interpretation for dynamic adaptations, with the latter being deemed the most crucial. Furthermore, from the analysis emerges context-awareness as a crucial feature that workflow management architectures should integrate to enhance workflow flexibility and adaptability.

Consequently, Phase 2 compared approaches in the literature that specifically feature context-awareness, while answering RQ3, “What are the existing challenges and limitations in the state-of-the-art with respect to context-aware workflow management approaches and their impact on the industry?” in Section 3.4. Through a rigorous analysis during this last phase, several avenues for future development were discovered: 1) need for real-world implementations, 2) consideration of a broader range of context variables, including both calculated and sensor-derived data, 3) utilization of industry-oriented standards and standardized workflow formats, 4) adoption of user-centric design approaches, transitioning from abstract workflows to executable workflows, and 5) development of decoupled components and systems, providing more versatility and integration with existing systems. The subsequent chapter presents a standardized solution that leverages recent advancements and addresses the identified challenges.

4

Context-Aware Workflow Management Architecture for Dynamic Service Orchestration

In response to the challenges identified during the state-of-the-art revision, this chapter unveils a proposal for a context-aware workflow management architecture designed to address the following concerns: 1) need for real-world implementations, 2) consideration of a broader range of context variables, including both calculated and sensor-derived data, 3) utilization of industry-oriented standards and standardized workflow formats, 4) adoption of user-centric design approaches, transitioning from abstract workflows to executable workflows, and 5) development of decoupled components and systems, providing more versatility and integration with existing systems.

Furthermore, this chapter addresses the research question "RQ4: Which components and features should be integrated into a context-aware workflow management solution to address the identified challenges while leveraging advantages from existing approaches?". Additionally, it aligns with the research objective "O3: Propose and develop an architecture for context-aware workflow management that takes all the benefits from current approaches and addresses active challenges".

It is noteworthy that the development of this architecture led to several scientific publications, two of which are directly related to this chapter:

- **[34]:** This publication introduces the architecture, emphasizing its workflow modelling components. The architecture leverages Asset Administration Shell (AAS) and Business Process Modeling Notation (BPMN) language to develop a plugin for the Camunda Modeler, simplifying workflow design by integrating available asset services into the modeler palette. Additionally, this paper introduces an AAS submodel that enables the registration of asset services into the administration shell, particularly in the form of REST services.
- **[33]:** This work proposes an alternative to traditional workflow managers, which can be integrated into the architecture. The proposal is designed to operate in embedded systems like Raspberry Pi, requiring minimal hardware resources.

4.1 Introduction

In the era of Industry 4.0, future manufacturing plants will become increasingly complex and interconnected [183]. This complexity demands new approaches to managing and orchestrating production processes while enhancing flexibility and adaptability. Moreover, the discussion of the literature review in the preceding chapter (Section 3.4.2) underscored the significance of addressing the challenges posed by Industry 4.0, particularly the need to tackle the "Utilization of industry-oriented standards and standardized workflow formats".

One promising avenue to address this challenge involves leveraging AAS to represent physical devices and machines as digital twins [184]. AAS standardizes the representation of both the technical and operational data of assets [185], while BPMN provides a standard language for graphically representing business processes [12]. Workflows written in BPMN are managed using the Business Process Management (BPM) discipline, which provides tools for the design and execution of business processes [186]. The combination of AAS and BPMN can offer versatility in designing and executing manufacturing workflows using digital twins.

The literature also underscored the importance of addressing the challenge of "Adoption of user-centric design approaches, transitioning from abstract workflows to executable workflows". While abstract workflows provide a high-level conceptual view of the process, transitioning to executable workflows is crucial to gaining a clear understanding of how the process will function in practice [187]. Executable workflows enable real-time monitoring of process execution, providing insights into the performance metrics, bottlenecks detection, and identification of areas for optimization [188]. This visibility allows organizations to continuously improve processes and swiftly adapt to changing business requirements [125].

Furthermore, adopting a microservice-oriented architecture design can also tackle the challenge of "Development of decoupled components and systems, providing more versatility and integration with existing systems". Microservice-based architectures offer integration capabilities through standardized APIs and communication protocols, facilitating interoperability with existing systems [189]. The advantages of components and systems that follow the microservice-oriented design are characterized as offering flexibility, scalability, interoperability, reduced dependencies, modularity, and vendor neutrality [190].

During the literature revision, service re-selection and re-composition techniques emerged as vital strategies for achieving context-awareness within workflow systems. The service re-selection technique enables workflows to adapt in real-time in response to changing conditions, such as resource availability or environmental situations [191]. Real-time adaptiveness is essential for ensuring workflow agility and responsiveness to dynamic manufacturing environments, where conditions may change rapidly [192, 193].

However, enhancing the reactiveness of manufacturing workflows lies also in overcoming the challenge of "Consideration of a broader range of context variables, including both calculated and sensor-derived data". The consideration of diverse context variables is crucial to making more informed decisions about the manufacturing process [194]. In modern manufacturing facilities where various heterogeneous devices govern, workflow systems are required to consider the current state of machines and products being produced to adapt the process in real-time [195]. A workflow system that considers a broad range of context variables can optimize resource utilization more effectively, avoiding possible bottlenecks or over-utilization [196].

Consequently, this chapter introduces a workflow management architecture with context-awareness capabilities, leveraging Industry 4.0 and workflow standards within a decoupled solution to address the mentioned challenges. The chapter is organized as follows: Section 4.2 provides an overview of the architecture, offering brief descriptions of each component. Section 4.3 elaborates on the components tailored for enhanced BPMN workflow modelling, including a plugin for service discovery and its integration within Camunda Modeler and AAS to facilitate standardized workflow design. Following this, Section 4.4 explores various options for BPMN execution, with a focus on a proposal capable of operating in both edge and central environments, requiring minimal hardware resources. Finally, Section 4.5 presents the concluding remarks of this chapter.

4.2 Architecture Overview

The architecture, as illustrated in Fig. 4.1, is designed to facilitate the orchestration of machine/device services in an edge-to-cloud environment, providing a dynamic and flexible workflow at runtime. The architecture and tools are licensed under the Apache-2.0 license, which allows for community use[1][2][3][4].

This proposal also answers the research question "RQ4: Which components and features should be integrated into a context-aware workflow management solution to address the identified challenges while leveraging advantages from existing approaches?". The following itemization briefly describes each component:

- **Assets:** An asset is any unit of work that can perform a task [6]. Located at the top-right, this component represents Industry 4.0 physical devices at plant level. These assets undergo digitization through the AAS concept, and their digital representations are stored in a repository.
- **AAS Repository:** This repository houses the AAS Server, which stores administration shell data. Positioned at the top-left, the data within can be queried and/or maintained using the AAS Server API (Application Programming Interface). The AAS Server can be any of the implementations discussed in Section 2.1.4.2, including Basyx, NOVAAS, Admin-shell-io, etc.

---

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

[2] `https://github.com/MUFacultyOfEngineering/ContextAnalyzer`

[3] `https://github.com/MUFacultyOfEngineering/RosIsaacSimWarehouseCarterx10`

[4] `https://github.com/MUFacultyOfEngineering/NodeRedWorkflowManager`

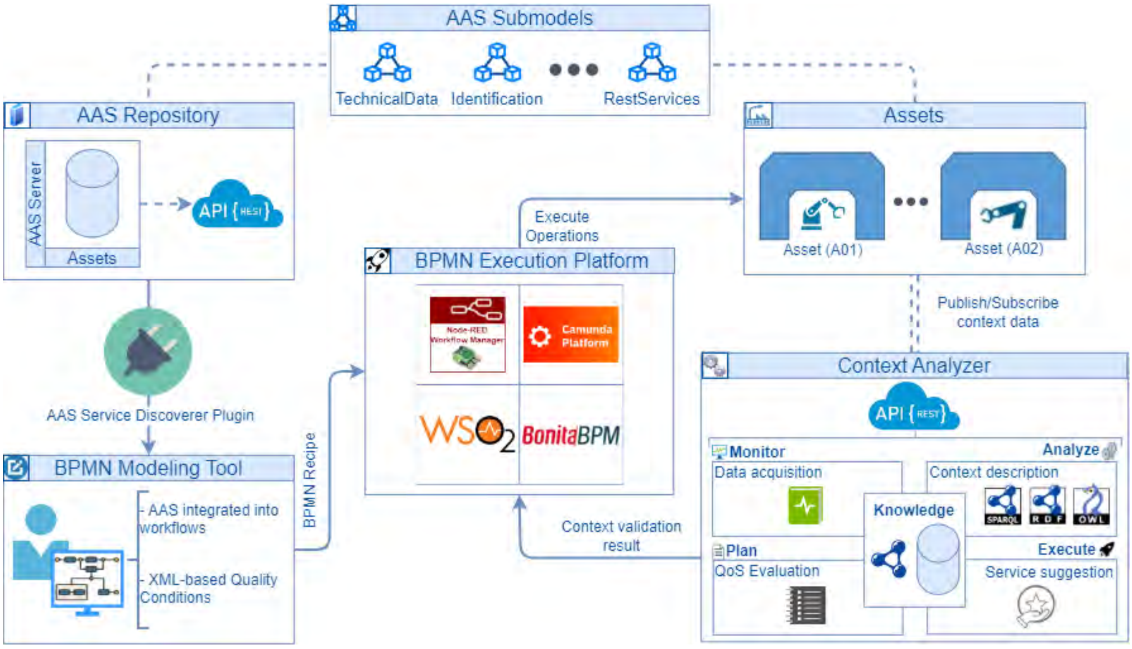


Figure 4.1: Architecture for Context-Aware Workflow Management: An Asset Administration Shell-based approach

- **AAS Submodels:** This component serves to describe technical and operational data of assets through AAS submodels [30]. Located at the top-middle, it includes a RestServices submodel that characterizes attributes of machine or device services, including URL, name, method, IsAsync, RequestBody, and Response.
- **BPMN Modeling Tool:** Encompassing the Camunda Modeler and a plugin called "AAS Web Service Discoverer", this tool enables the Camunda Modeler to discover services from a chosen AAS Repository. Positioned at the bottom-left, it facilitates the design of manufacturing business processes using asset services in BPMN format.
- **BPMN Execution Platform:** Responsible for managing the orchestration of asset services, this component is situated at the centre. Any workflow or business process executor capable of interpreting recipes written in BPMN can be employed here. Examples of BPMN execution platforms include Camunda Platform, WSO2, and BonitaBPM. As part of this proposal, Node-RED Workflow Manager (Node-RED WM) [33] is introduced as an alternative software. Node-RED WM is a workflow management system that interprets and runs workflow recipes written in BPMN and can be installed in embedded systems, requiring minimal hardware resources.
- **Context Analyzer:** Positioned at the bottom-right, this context-aware component employs semantic web technologies for context mapping in combination with the MAPE-K (Monitor, Analyze, Plan, Execute, and Knowledge) model [81]. Its primary goal is to enhance workflow dynamism during runtime. Furthermore, the Context Analyzer component serves a pivotal role in this context-aware workflow management architecture, aiming to furnish service recommendations based on real-time context and quality of service conditions. Given

the intricacies and crucial role of this component, it requires a dedicated chapter (Chapter 5) to explain its design and functionality extensively.

Now that a general understanding of the architecture and its key components has been gained, the focus shifts to exploring each component in detail. The exploration begins with components directly related to the workflow design phase, delving into the integration of AAS within the Camunda Modeler. Subsequently, it examines components associated with the workflow execution phase, exploring available options for workflow execution and highlighting Node-RED WM as an alternative. Through this detailed examination, the aim is to elucidate how these components synergize to streamline workflow design and execution.

4.3 Enhanced BPMN Modelling Tool: Integrating AAS within the Camunda Modeler

This section delves into the components of the proposed architecture specifically involved in the workflow design phase. It starts by exploring the literature on the utilization of AAS in business process orchestration. Then, it introduces the components of the proposed architecture for enhanced BPMN workflow modelling, emphasizing the integration of AAS for a service discovery mechanism. Additionally, an experiment involving two Lego robots in a piece-classifying use case is conducted to demonstrate the practical application of the enhanced workflow modelling tool.

4.3.1 Related Work

According to Gartner [72], an intelligent BPM Suite (iBPMS) is *"... a licensed software that supports the full cycle of business process and decision, discovery, analysis, design, implementation, execution, monitoring, and optimization"*. In [71], Gartner identifies the "top" 20 iBPMS vendors wherein Camunda BPMS Platform[5] is among them. -Moreover, according to Slintel [197], Camunda has 13.47% of market share in the Workflow Automation category, just behind competitors such as Apache Airflow with 30.79% and REDCap with 14.99%. Camunda is a robust open-source platform for managing the full lifecycle of business processes, offering a set of components for the design, execution, and monitoring of business processes. Among those components, there is Camunda Modeler which is a business process modeling tool featuring a user-friendly interface for the design of business processes.

Concerning AAS, an asset can be turned into a digitized I4.0 component by describing and referencing one or more submodels. An asset can be i.e. physical components, hardware, documents, software, human beings, etc [198]. Their properties can be described using submodels in a reflexive interface, providing high-level information about the functions, events, references, relationships, location, status, etc [199].

---

[5] `https://camunda.com/platform/`

Moreover, there is a lack of work done in the domain of AAS and business processes together. This is noticeable by examining the publications in each discipline. Articles in the AAS discipline deal with Cyber-Physical Systems (CPS). For instance, the authors in [200] introduced a platform for the design of production lines based on AAS by orchestrating CPS services. However, the authors designed the recipe in another format than business processes. On the other hand, papers in the domain of business processes are often related to IoT and manufacturing systems. However, without any specific proposal about using AAS since it is still an emerging topic. For instance, [201] introduced an architecture to make BPMS aware of IoT devices, allowing the orchestration of the IoT services and, at the same time, maintaining the recipe in business process format. In addition, the authors in [202], proposed an extension of BPMN to support accurately the modeling of CPS processes, adding as well resources and context data to the recipe. Similarly, the authors in [203], introduced the Web Services Business Process Execution Language (WS-BPEL) which is an extension of BPEL. WS-BPEL allows the definition of IoT-aware business processes using BPEL and runs in any business process execution engine. Another example is the work of [7], the authors introduced a framework that includes Service-Discovery and Service-Composition mechanisms for IoT service orchestration. Notwithstanding, none of these authors considered the AAS concept in their implementations.

The importance of mapping asset services into AAS comes from the importance of composing complex orchestrations in manufacturing systems [7]. Thus, the proposed component for enhanced workflow design should be able to describe the services offered by an asset in its AAS digital representation.

4.3.2 AAS-Powered Business Process Modelling Tool

This proposal enables the design of AAS-based business processes. Figure 4.2 shows the components of the architecture specifically tailored to allow AAS-based workflow modelling. First, *Assets* represents I4.0 physical devices at the plant level. These assets are digitized through the AAS concept and their digital representations are stored in the *AAS Repository*, which could be AAS Server and/or AAS Registry depending on the vendor's implementation. AAS repository implementations should offer APIs (Application Programming Interface) for the management of the stored AASs [1]. As mentioned earlier, assets can be described using submodels. In order to describe machine services, a submodel named *RestServices* is introduced. The *BPMN Modeling Tool* layer contains both, the Camunda Modeler and our novel Camunda plugin that enables the Service-Discovery mechanism. Thus, business analysts can design business processes out of asset services.

The following enumeration provides detailed information about the components comprising the AAS-powered BPMN modelling tool.

(I) **Assets:** An asset is any unit of work that can perform a task [6]. Examples of assets

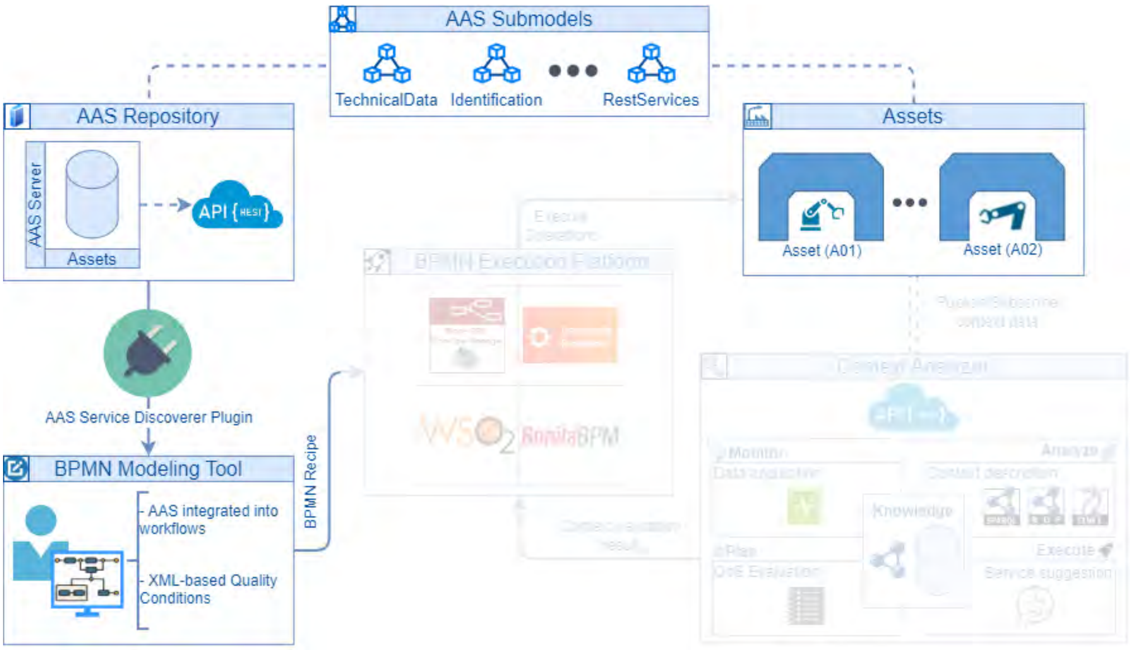


Figure 4.2: Components that allow AAS-Powered Business Process Modeling within the proposed architecture

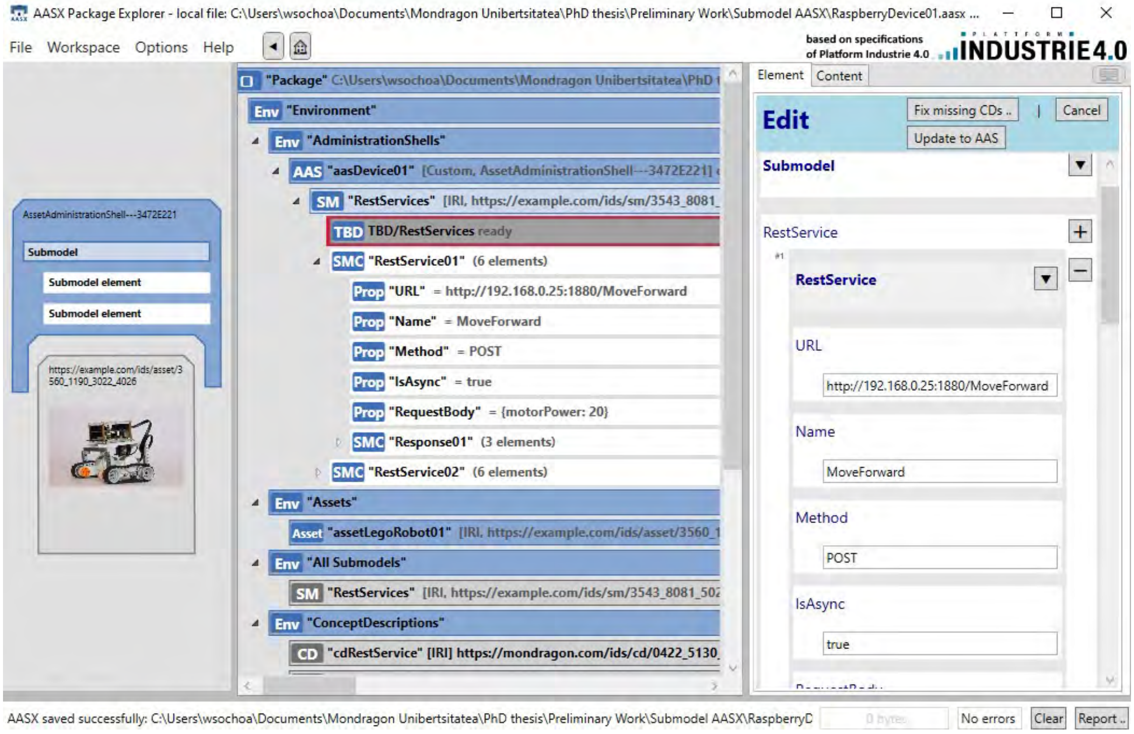
include, physical components, hardware, documents, software, and human beings [5]. An asset can be turned into a digitized I4.0 component by describing and referencing one or more AAS submodels in a reflexive interface, providing high-level information about the functions, events, references, relationships, location, status, etc [199]. For instance, the *Nameplate* submodel described in [51] provides the properties to register the nameplate of an asset. This way, parties involved can use this submodel to query the physical address, serial number, year of construction, manufacturer name, etc. Thus, enabling the provision of a wide variety of data. Other submodel templates can be seen in [52–54]. However, at the current stage of the state-of-the-art, there does not exist a submodel for representing services of devices/machines in the form of REST services.

(II) AAS RestServices Submodel: The RestServices AAS submodel is a template that allows the registration of machine services in the form of REST services. The template includes properties of REST services such as URL, Name, Method, IsAsync, RequestBody, and Response, corresponding to the basic elements of a typical REST service [21]. Furthermore, the submodel can be extended with more properties according to diverse manufacturing scenarios. Figure 4.3 shows the definition of the new submodel on the AASX Package Explorer[6].

Tables 4.1, 4.2, and 4.3 break down the attributes specifications of the *RestServices* submodel. At the top level, the *RestServices* submodel contains one property of type SubmodelElementCollection and its multiplicity is set to 1..* to allow the registration of 1 to many services of an asset. Next, the *RestService* SubmodelElementCollection allows to characterize an individual

---

[6] `https://github.com/admin-shell-io/aasx-package-explorer`


AASX Package Explorer - local file: C:\Users\wsochoa\Documents\Mondragon Unibertsitatea\PhD thesis\Preliminary Work\Submodel AASX\RaspberryDevice01.aasx ...

"Package" C:\Users\wsochoa\Documents\Mondragon Unibertsitatea\PhD t

Env "Environment"

Env "AdministrationShells"

AAS "aasDevice01" [Custom, AssetAdministrationShell---3472E221]

SM "RestServices" [IRI, https://example.com/ids/sm/3543_8081_

TBD TBD/RestServices ready

SMC "RestService01" (6 elements)

Prop "URL" = http://192.168.0.25:1880/MoveForward

Prop "Name" = MoveForward

Prop "Method" = POST

Prop "IsAsync" = true

Prop "RequestBody" = {motorPower: 20}

SMC "Response01" (3 elements)

SMC "RestService02" (6 elements)

Env "Assets"

Asset "assetLegoRobot01" [IRI, https://example.com/ids/asset/3560_1

Env "All Submodels"

SM "RestServices" [IRI, https://example.com/ids/sm/3543_8081_502

Env "ConceptDescriptions"

CD "cdRestService" [IRI] https://mondragon.com/ids/cd/0422_5130_

URL: http://192.168.0.25:1880/MoveForward

Name: MoveForward

Method: POST

IsAsync: true

AASX saved successfully: C:\Users\wsochoa\Documents\Mondragon Unibertsitatea\PhD thesis\Preliminary Work\Submodel AASX\RaspberryD

Figure 4.3: AAS RestServices Submodel template

service by using properties such as URL, Name, Method, IsAsync, RequestBody, and Response. Finally, the *Response* SubmodelElementCollection allows specifying how a service would respond to a request.

Table 4.1: RestServices Submodel Template – Full attributes specifications

idShort:	RestServices		
Class:	Submodel		
semanticId:	[IRI] `https://mondragon.com/ids/sm/7471_5180_3022_3423`		
Parent:	RestServices		
Explanation:	A submodel template for mapping device REST services@en, Una plantilla de submodelo para mapear propiedades de un servicio REST de un dispositivo@es		
[SME type] idShort	semanticId = [idType]value Description@en	[valueType] example	card.
[SMC] RestService	[IRI] `https://mondragon.com/ids/cd/0422_5130_5022_9484` Information about an individual Rest Service	[-] 5 elements	1..*

Once assets are digitized using the provided RestServices submodel, the .aasx package should be deployed into the AAS Repository where the information can be queried through the AAS Repository's API.

(III) AAS Repository: Serving as the digital storage hub for asset representations, the AAS Repository houses technical and operational data about physical assets in a standardized format. Typically equipped with APIs (Application Programming Interfaces), these repositories facilitate the management and retrieval of stored data [1]. Acting as a centralized resource, the repository enables the storage, organization, and accessibility of asset information, catering

Table 4.2: RestService SubmodelElementCollection – Full attributes specifications

idShort:	RestService		
Class:	SubmodelElementCollection		
semanticId:	[IRI] `https://mondragon.com/ids/cd/0422_5130_5022_9484`		
Parent:	RestService		
Explanation:	Information about an individual Rest Service@en, Información respecto a un Servicio Rest@es		
[SME type] idShort	**semanticId = [idType]value Description@en**	**[valueType] example**	**card.**
[Property] URL	[IRI] `https://mondragon.com/ids/cd/0241_0190_3022_9102` The REST service endpoint. Make sure it is reachable within the network.	[string] `http://127.0.0.1/myRestService/?parameter1=val1`	
[Property] Name	[IRI] `https://mondragon.com/ids/cd/7052_0190_3022_2921` The name of the REST service	[string] My REST service name	
[Property] Method	[IRI] `https://mondragon.com/ids/cd/1561_0190_3022_7753` The type of method	[string] GET POST PUT PATCH DELETE	
[Property] IsAsync	[IRI] `https://mondragon.com/ids/cd/2582_3130_5022_8052` Specify whether or not the operation is executed asynchronously	[boolean] false true	
[Property] RequestBody	[IRI] `https://mondragon/ids/cd/6321_8081_5022_2401` Payload or request body. The actual data to be sent to the HTTP REST endpoint.	[string] {} < / > 0 text	
[SMC] Response	[IRI] `https://mondragon.com/ids/cd/2262_0190_3022_6903` Service response	[-] 3 elements	1..*

Table 4.3: Response SubmodelElementCollection – Full attributes specifications

idShort:	Response		
Class:	SubmodelElementCollection		
semanticId:	[IRI] `https://mondragon.com/ids/cd/2262_0190_3022_6903`		
Parent:	Response		
Explanation:	Service response@en, Respuesta del servicio@es		
[SME type] idShort	**semanticId = [idType]value Description@en**	**[valueType] example**	**card.**
[Property] Code	[IRI] `https://mondragon.com/ids/cd/9562_0190_3022_6374` Responde code	[int] 200 201	
[Property] MediaType	[IRI] `https://mondragon.com/ids/cd/4572_0190_3022_0456` Media type	[string] application/json application/xml text/xml	
[Property] ExampleValue	[IRI] `[IRI]https://mondragon.com/ids/cd/6192_0190_3022_608` An example of response body	[string] {} < / > 0 text	

to diverse needs within manufacturing or industrial settings. A review of existing AAS implementations, including Basyx, Admin-shell-io, NOVAAS, and PyI40AAS, was conducted in subsection 2.1.4.2. In this work, both Admin-shell-io and Basyx AAS implementations underwent testing, affirming their compatibility.

After relevant data has been uploaded to the AAS Repository, it becomes accessible for querying through the API endpoints provided by the repository. In this proposal, a plugin for the Camunda Modeler takes responsibility for querying the AAS Repository to gather information about asset services.

(IV) AAS Service Discoverer plugin: It is an essential plugin integrated into the Camunda Modeler, enabling service discovery during the workflow design phase. Developed for the desktop variant of Camunda Modeler, the plugin operates as a standalone application compatible with Linux, MacOS, and Windows platforms [204]. With this plugin, users can input initial settings for the modeller to reach the AAS Repository. These settings are saved into the configuration file of Camunda Modeler under the *aasWebServiceDiscoverer* key-value-pair.

The plugin employs an algorithm to identify the type of AAS implementation hosted on the server based on the provided settings. Once identified, it utilizes the AAS Repository's API to access the required data. Concurrently, the plugin can recognize endpoints offered by Basyx [199] and Admin-shell-io [205]. The specific endpoints utilized by the plugin are detailed in Table 4.4.

Table 4.4: AAS Server API endpoints used by the AAS Web Service Discoverer Plugin

Description	AAS Server	Endpoint
1) Get the list of registered shells	admin-shell-io: Basyx:	/server/listaas /shells
2) Get details of a given submodel for a given shell	admin-shell-io: Basyx:	/aas/{aasId}/submodels/{submodelId}/complete /shells/{aasId}/aas/submodels/{submodelId}/submodel/submodelElements

The algorithm in this plugin iterates each of the AASs found and gets all submodelElements whose submodel idShort is *RestServices*. Each submodelElement represents an individual service of an asset. Thereby, all the services of all the assets can be obtained in JSON format. Finally, the algorithm converts those asset services from JSON to BPMN Service-Tasks and adds them into the modeller's palette in a fashion manner.

In summary, the components outlined in this section harness AAS and BPMN to provide an advanced workflow modelling tool. Of particular significance is the AAS Service Discoverer plugin, which greatly assists business users in designing business processes by incorporating services offered by machines. The next subsection delves into an experiment involving two Lego robots in a piece-classifying use case to showcase the practical application of this enhanced workflow modelling tool.

4.3.3 Use Case: Designing a Piece-Classifier BPMN Workflow

This experiment aims to demonstrate the feasibility of the proposal in enabling the service discovery during the workflow design phase. To conduct this experiment, the following tools are required: 1) AASX Package Explorer 2022-01-13.alpha. 2) AASX Server 2022-01-13.alpha implementation from admin-shell-io, specifically the blazor variant. 3) Camunda Modeler 5.0.0-alpha.1. 4) Camunda Platform. And, 5) the aasx packages given in the GitHub repository[7]. Additionally, two Lego robots are employed as assets. To facilitate this experiment, a Raspberry Pi HAT (Hardware Attached on Top) is installed on both to connect the Raspberry Pi with the Lego robots.

The experiment is organized into phases. Phase 1 entails the setup, which includes installing the necessary tools for conducting the experiment. Phase 2 focuses on designing a BPMN workflow recipe using the enhanced BPMN workflow modelling tool. In Phase 3, the designed BPMN recipe is validated through testing using the *Token Simulation* plugin of Camunda Modeler. Subsequently, the validated recipe is deployed into the Camunda Platform for execution.

Experiment Phase 1: Workbench and Lego robots

This phase of the experiment focuses on describing the setup of the workbench and the Lego robots involved. Figure 4.4 illustrates the workbench used for this experiment, which consists of four main elements, labelled A, B, C, and D:

- A) Lego colour sorter: This robot sorts pieces based on their colour by moving the feed tray along the conveyor belt and ejecting pieces one by one.
- B) Lego two-axis robotic arm: This robot has the capability to move its arm in various directions and control the opening and closing of its claw.
- C) Non-red pieces container: A container designated for storing pieces that are other colours than red.
- D) Red pieces container: A container specifically for storing red pieces.

To control the motors and sensors of the Lego robots, software based on Python and Node-RED is developed for both robots. This software facilitates switching between the execution environments of “Digital twins” and “Physical robots”. Consequently, the functionalities of both robots are exposed as REST services, with OPCUA invocations running in the background. Table 4.5 provides a list of the REST services offered by the devices considered in this experiment.

Experiment Phase 2: BPMN recipe design

In this phase, assets and their descriptions are created using the AASX Package Explorer, utilizing the introduced *RestServices* submodel. The generated .aasx files are then placed in the *..AasxServerBlazor/aasxs* path to enable the AASX Server to access them. Subsequently, both the Camunda Modeler and the *AAS Web Service Discoverer plugin* are launched.

---

[7] `https://github.com/MUFacultyOfEngineering/AASBPM/tree/main/aasResources/aasxs`

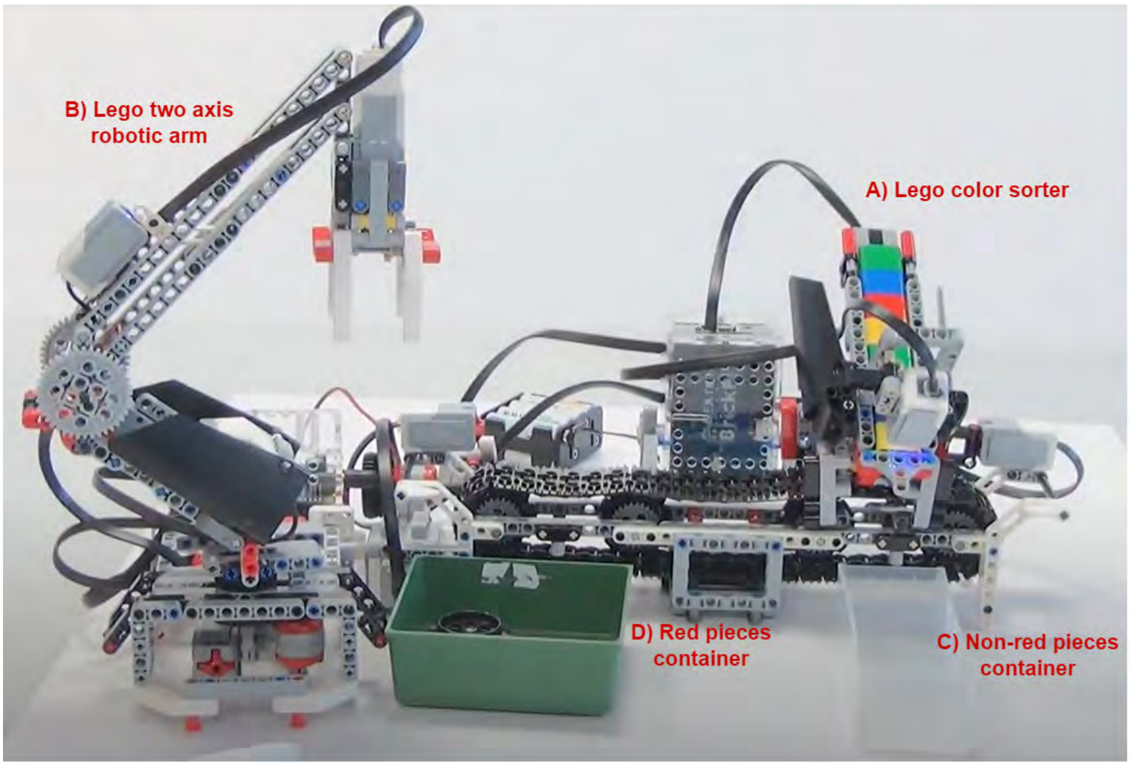


Figure 4.4: Workbench and Lego Robots

Table 4.5: Endpoints Lego robots

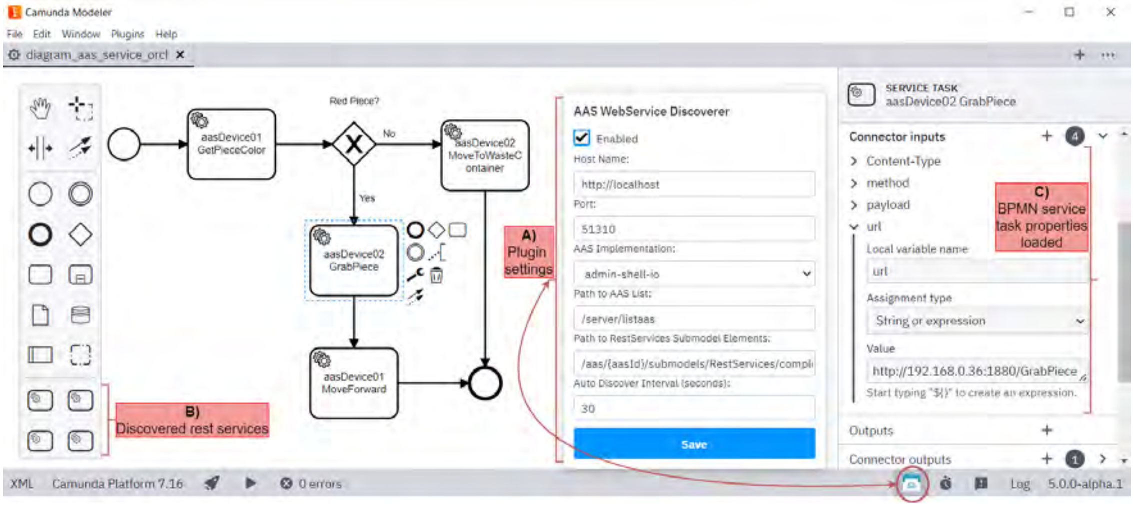
Lego Robot	EndPoint	Description	Return
Color Sorter	/GetPieceColor	Gets current piece colour	String
	/MoveLeft	Moves the feed tray through the conveyor to the far left	bool
	/MoveRight	Moves the feed tray through the conveyor to the far right	bool
	/ThrowPiece	Throws the current piece out of the feed tray	bool
Robotic Arm	/TurnLeft	Turns the arm 90° to the left	bool
	/TurnRight	Turns the arm 90° to the right	bool
	/OpenClaw	Fully opens the claw	bool
	/CloseClaw	Closes the claw	bool

Figure 4.5 illustrates the functionality of the plugin, which consists of three segments, labelled A, B, and C:

A) Represents the initial step where the user clicks on the AAS icon to reveal a modal. The modal allows the user to input settings such as the location of the AAS repository within the network, the type of AAS implementation, and the EndPoint paths to the AAS list and SubmodelElements. The settings can be saved using the *Save* button, which stores them in the Camunda Modeler's configuration file.

B) The plugin then queries the AAS repository and the discovered services are displayed at the bottom of the palette in a separate group.

C) When a user selects and drags an asset service from the discovered services onto the canvas, a BPMN service-task is automatically generated. All the property values of the service-task are loaded from the AAS, ensuring consistency between the asset service and its representation in the BPMN editor.



AAS WebService Discoverer
☑ Enabled
Host Name: http://localhost
Port: 51310
AAS Implementation: admin-shell-io
Path to AAS List: /server/listaas
Path to RestServices Submodel Elements: /aas/{aasId}/submodels/RestServices/compl
Auto Discover Interval (seconds): 30

SERVICE TASK
aasDevice02 GrabPiece
Connector inputs 4
Content-Type
method
payload
url
Local variable name: url
Assignment type: String or expression
Value: http://192.168.0.36:1880/GrabPiece
Start typing "${}" to create an expression.

Figure 4.5: Designing a BPMN workflow in Camunda Modeler and the AAS Service Discoverer Plugin

In this example, an AAS service named “GrabPiece” has been imported into the BPMN editor. Figure 4.6 illustrates the XML generated for this imported task.

```
<bpmn:serviceTask id="aasDevice02.RestService01" name="aasDevice02 GrabPiece" camunda:asyncBefore="true" camunda:asyncAfter="true">
  <bpmn:extensionElements>
    <camunda:connector>
      <camunda:inputOutput>
        <camunda:inputParameter name="url">http://192.168.0.36:1880/GrabPiece</camunda:inputParameter>
        <camunda:inputParameter name="method">POST</camunda:inputParameter>
        <camunda:inputParameter name="Content-Type">application/json</camunda:inputParameter>
        <camunda:inputParameter name="payload">${color: 'red', motorPower: 20}</camunda:inputParameter>
        <camunda:outputParameter name="responsename">${response}</camunda:outputParameter>
      </camunda:inputOutput>
      <camunda:connectorId>http-connector</camunda:connectorId>
    </camunda:connector>
  </bpmn:extensionElements>
  <bpmn:incoming>Flow_1f464wv</bpmn:incoming>
  <bpmn:outgoing>Flow_1kqcput</bpmn:outgoing>
</bpmn:serviceTask>
```

Figure 4.6: Camunda Modeler - XML GrabPiece Service-Task

During the design phase, quality conditions can be also defined using the “AAS Service Discoverer” plugin for Camunda Modeler, as shown in Fig. 4.7. This plugin provides a list of available quality properties from the administration shell, allowing the user to establish the quality conditions for each task.

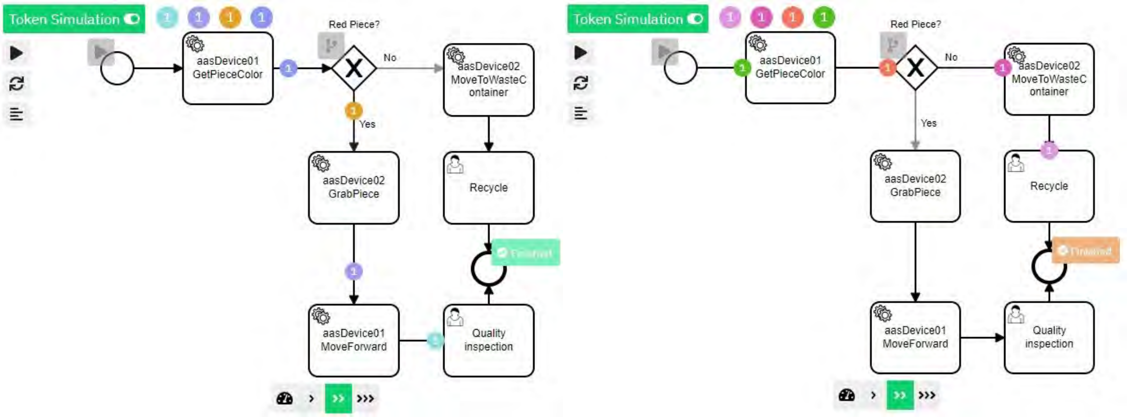
PROXIMITY_PICKUP <= 20.0 &&
BATTERY >= 25 &&
POSITIONAL_UNCERTAINTY <= 0.90 &&
PAYLOAD_CAPACITY >= 0.60

Figure 4.7: Defining quality conditions for a workflow task in Camunda Modeler and the AAS Service Discoverer plugin

Experiment Phase 3: BPMN recipe validation and deployment

This phase encompasses the validation and deployment of the previously generated BPMN recipe. The workflow recipe is validated using the *Token Simulation*[8] plugin for Camunda Modeler. This plugin allows for the virtual execution of workflows, enabling the identification and resolution of potential errors prior to actual deployment.

Figure 4.8 illustrates the validation process for both scenarios; when the current piece is red and when the current piece is other than red. After successfully validating the BPMN recipe, it is deployed into the Camunda Platform for execution. Figure 4.9 showcases the process instance in Camunda Cockpit, which is part of the Camunda Platform. It also displays the values of variables associated with each Service-Task during the execution of the workflow.



(a) Red piece

(b) Other than red piece

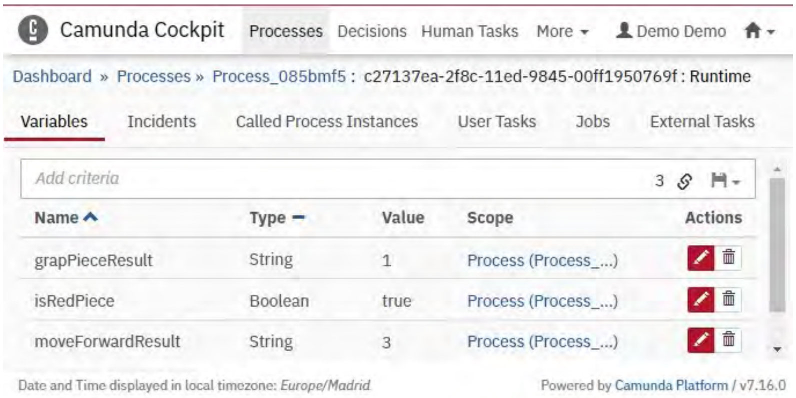
Figure 4.8: BPMN recipe validation (prior deployment)

4.4 BPMN Execution Platforms: Node-RED Workflow Manager

Following the decoupled nature of the proposed architecture, as illustrated in Fig. 4.10, the “BPMN Execution Platform” is responsible for orchestrating tasks within BPMN recipes. Furthermore, the proposed architecture offers flexibility, accommodating any workflow

---

[8] `https://docs.camunda.io/docs/components/modeler/web-modeler/token-simulation/`



Dashboard » Processes » Process_085bmf5 : c27137ea-2f8c-11ed-9845-00ff1950769f : Runtime

Name	Type	Value	Scope	Actions
grapPieceResult	String	1	Process (Process_...)	
isRedPiece	Boolean	true	Process (Process_...)	
moveForwardResult	String	3	Process (Process_...)	

Date and Time displayed in local timezone: *Europe/Madrid*

Powered by Camunda Platform / v7.16.0

Figure 4.9: BPMN recipe execution in Camunda Platform

execution platform compliant with BPMN standards. Options include the Camunda Platform, ProcessMaker, WSO2, or BonitaBPM, all of which adhere to BPMN standards. A detailed comparison of these workflow execution platforms is conducted in Section 2.2.3.

However, in response to the challenge posed by the "Need for an edge-compliant process manager software" identified during the literature review (Section 3.3.3), an approach to workflow execution platform is presented in this section. Node-RED Workflow Manager (Node-RED WM), is an edge-embedded workflow manager alternative built in Node-RED. A key characteristic of Node-RED WM is its capacity to load and execute business processes written in BPMN, requiring minimal hardware resources.

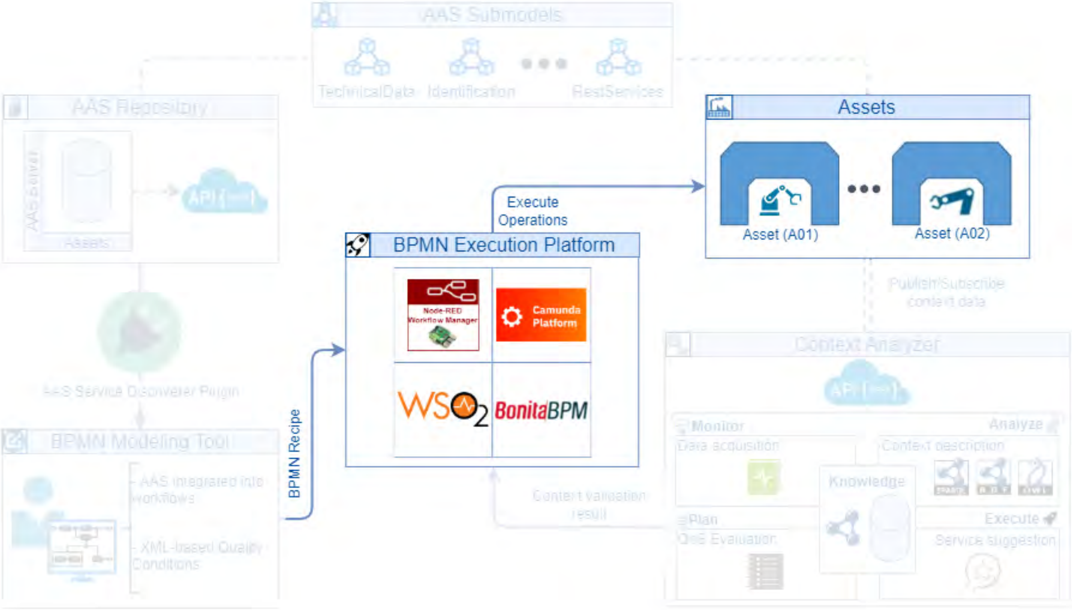


Figure 4.10: Components within the proposed architecture for BPMN Execution Platform

4.4.1 Related Work

In recent years, diverse tools have been proposed to support Business Process Management. Those tools are often tagged as Workflow Management Software (WfMS) or Intelligent Business Process Management Systems (iBPMS). These software suites are often utilized for robust collaboration, automation, and tracking of business processes [63]. Gartner defines The Intelligent business process management systems (iBPMS) as *"... a licensed software that supports the full cycle of business process and decision discovery, analysis, design, implementation, execution, monitoring, and optimization"* [71]. iBPMS is an evolution of WfMS since it incorporates a set of technologies to coordinate services, processes, machines, and people [72], improving and transforming workflow management discipline through a collaborative platform. Among the open-source iBPMS mentioned in the Gartner's Magic Quadrant [73], the following are the preferred choices: Camunda[9], Bonita[10], and Process Maker[11]. All these solutions run on a centralized server usually deployed in cloud infrastructures. However, there are some key differences between these tools.

Camunda Platform is an open-source BPM platform that is designed for complex, enterprise-grade processes. It offers a wide range of features, including a BPMN workflow engine and a task management system. Camunda is a good choice for organizations that need a powerful and scalable workflow platform. Similarly, ProcessMaker is another open-source BPM platform that is designed for ease of use. It offers a variety of pre-built templates. ProcessMaker is a good choice for organizations that need a simple and easy-to-use workflow platform. However, it may not be as powerful as Camunda for complex processes.

Other commercial but powerful options include Bonita, a BPM platform designed for business users. It offers a variety of features, including a workflow engine and a task management system. Bonita is a good choice for organizations that need a powerful and easy-to-use workflow platform. However, it can be expensive. Similarly, Creatio[12] (formerly bpm'online) is a commercial BPM platform designed for enterprise-grade processes. It offers a wide range of features, including a workflow engine, a task management system, and a CRM system. Creatio is a good choice for organizations that need a powerful and scalable BPM platform with CRM integration. However, it can also be expensive.

Newgen[13] is a commercial BPM platform that is designed for enterprise-grade processes. It offers a wide range of features, including a BPMN modeller, a workflow engine, a task management system, and a document management system. Newgen is a good choice for organizations that need a powerful and scalable BPM platform with document management integration. However, it can be expensive. Similarly, Bizagi[14] is another commercial BPM

---

[9] `https://camunda.com/`

[10] `https://www.bonitasoft.com/`

[11] `https://www.processmaker.com/`

[12] `https://www.creatio.com/`

[13] `https://newgensoft.com/`

[14] `https://www.bizagi.com/`

platform designed for enterprise-grade processes. It offers a wide range of features, including a workflow engine, a task management system, and a business intelligence system. Bizagi is a good choice for organizations that need a powerful and scalable BPM platform with business intelligence integration. However, it can be expensive.

A wide variety of BPMN workflow executor platforms are available, each with its own strengths and weaknesses. While these systems offer a range of features, they often require significant resources and may not be suitable for all organizations. Additionally, the centralized nature of these systems can limit their flexibility and scalability. To address these limitations, Node-RED Workflow Manager is proposed as a lightweight and versatile workflow management system. Built into the Node-RED platform, Node-RED Workflow Manager leverages the edge computing paradigm to enable deployment in resource-constrained environments. It also supports the execution of business processes defined in BPMN, providing a standardized approach to process modelling and execution. Furthermore, Node-RED Workflow Manager integrates with other frameworks and architectures, making it a flexible and extensible solution for a wide range of applications.

4.4.2 Node-RED Workflow Manager for Edge Service Orchestration

The Node-RED Workflow Manager (Node-RED WM) is an open-source BPMS built in Node-RED. Node-RED WM offers a set of flows and REST endpoints for managing the upload of recipes in BPMN format, their instantiation, and execution as business processes. The application developed in Node-RED offers the endpoints to communicate with the Workflow Manager, holds the code to manage the BPMN recipes, and interacts with the external systems by invoking their services or receiving their asynchronous replies.

The Node-RED Workflow Manager (Node-RED WM) is an open-source business process management system (BPMS) seamlessly integrated into the Node-RED platform. It provides a comprehensive set of flows and REST endpoints for managing the upload of BPMN-formatted recipes, their instantiation, and execution as business processes. The Node-RED WM application offers endpoints for communication with the Workflow Manager, houses the code for managing BPMN recipes, and interacts with external systems by invoking their services or receiving their asynchronous replies.

4.4.2.1 Node-RED WM Database Structure

Node-RED WM uses a relational database to store the information necessary to complete workflows such as recipes, instances, tasks, and status. This way the Node-RED WM knows what process instances are running, which is the currently active task and go through the recipe by identifying the next task to run in the process until the flow is completed. The structure of the database is shown in Fig. 4.11 where:

- The *process* table stores an identifier and name of the process and the BPMN recipe as an XML document.

- The *process_instance* table records the instances of a process and the status of each instance. Thus, there can be several instances of the same process. The *status* column indicates whether the instance is *in-Progress* or *finished*.

- The *active_task* table holds tasks information relating each task with the *instance_id* they belong to. The information stored includes: 1) *bpmn_id*: The task identifier in the BPMN document. 2) *name*: The name of the task as in the BPMN document. 3) *type*: The type of the element as for the BPMN standard such as *startEvent*, *serviceTask*, *intermediateCatchEvent*, and *parallelGateway*. 4) *status*: The status of the task can be: *Ready_To_Start*, *In_Progress*, *Finished*, or *Finished_Treated*.

Figure 4.11: Node-RED Workflow Manager database entity-relationship model

4.4.2.2 Node-RED WM Building Blocks

Node-RED WM is composed of two building blocks which are 1) Process Manager and 2) Task Executor. As seen in Fig. 4.12, these building blocks are at the same time composed of modules in charge of a variety of functionalities that cooperate for effectively executing a workflow written in BPMN. The Process Manager contains functionalities that manage BPMN recipes as a whole while the Task Executor contains functionalities in charge of executing each task.

Process Manager functionalities:

1. *Upload_Recipe*: This flow offers an endpoint to upload a new recipe, read the name of the workflow process from the XML document (process id or name) and insert a new record using that name and the whole BPMN as text format (XML document).

2. *Start_Process*: Offers an endpoint that starts the execution of a process selected by the user. It must receive the name of the process to be executed. This module finds the desired

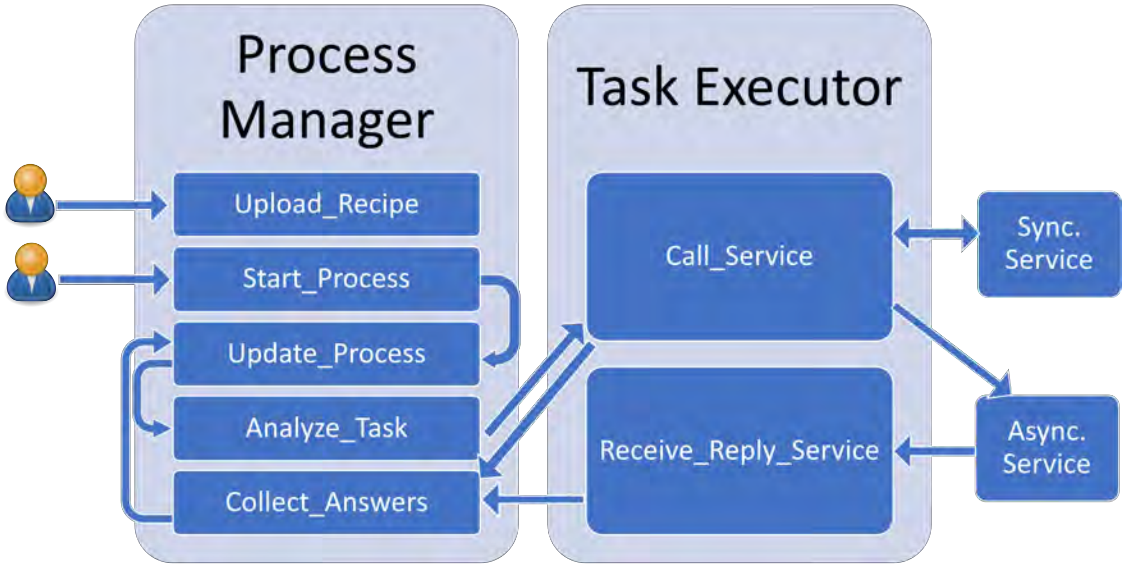


Figure 4.12: Node-RED Workflow Manager main building blocks

process and creates a new process instance in the database. Thenceforth, the module launches the process instance by reading from the recipe the *name* and *id* of the first task in the process (*startEvent*) and inserting that task in the *active_task* table. This task is then linked to its instance and complemented with the information from the recipe (*bpmn_id* = *startEvent id*, *name* = *startEvent name* and *type* = *startEvent*). The *status* is set to *Ready_To_Start*. Finally, this module calls upon the *Update_Process* flow.

3. *Update_Process*: Updates the status of a process instance by checking the status of individual tasks. This module queries the *active_task* table to get tasks with the *Ready_To_Start* and *Finished_Treated* status. Those records with the *Ready_To_Start* status indicate that the tasks can begin, while the records with the *Finished_Treated* status indicate that the tasks have already finished. This is necessary to address the merging of different paths in the recipe when using BPMN *parallelGateways*.

For each completed task, this module collects its *TargetRef*. In the BPMN standard, the *TargetRef* attribute indicates the type of the next task in the process. If the *targetRef* is *parallelGateway*, it means that two or more paths in the recipe are converged. Thus, this flow checks if all possible paths (*sequenceFlow/sourceRef* in BPMN) are finished. If that is the case, then the module updates those tasks in the *active_task* table marking them as *Finished_Treated* and collecting the next step (*TargetRef*). This should be a *parallelGateway* that is inserted in the *active_task table* as any other task with its status set to *Ready_To_Start*.

4. *Analyze_Task*: Checks a task for its type and updates the *active_task* table with new upcoming task. Depending on the task type, this flow can perform one of the following actions:

- *startEvent*: The flow finds the *startEvent id* that is associated with a *sequenceFlow* element within the BPMN recipe. The *sequenceFlow* element indicates the relation between the active task and the next task. Then, the module collects the next tasks reading the *targetRef* s attributes in the *sequenceFlow*s. There could be more than one task to continue with the process (splitting of process in several branches). The flow inserts those tasks in the *active_task* table with the status *Ready_To_Start* and finishes the *startEvent* talks by updating its status to *Finished_Treated*.

- *serviceTask*: The flow changes the status of the task to *In_Progress* and invokes the *Call_Service* flow in the Task Executor block. The call contains the *process_instance_id*, the *active_task_id* and the *message_expected* in case of an asynchronous service. *Message_expected* is a correlation flag the Node-RED WM uses to relate request responses to asynchronous services.
- *parallelGateway*: The flow performs the same actions as for *startEvent* since neither type performs any actions apart from indicating which are the next steps/tasks in the process.
- *intermediateCatchEvent*: The flow puts the process on hold until a notification is received from an asynchronous service to continue. This notification arrives in the form of a message (*message_expected*). In this case, the module changes the status of task to *In_Progress* waiting for the notification.
- *endEvent*: The flow sets the task status to *Finished_Treated* and finishes the process by setting the process instance to *Finished* in the *process-instance* table.

5. *Collect_Answers*: Is the flow where the Task Executor writes the replies received from the system/services and associates them to the proper tasks. To do that the flow collects the process *instance_id*, *task_id* and results from the replies. If the result is correct, this module changes the task status to Finished. This means that the task has finished. In order to continue with the recipe the Node-RED WM still needs to check if there are several branches that need to merge. This is done by the *Update_Process* flow.

Task Executor functionalities:

1. *Call_Service*: is a flow that Node-RED WM uses to make calls to external services (requests). The flow performs the following actions. It collects information about the service to be called such as the serviceURL, method, input, headers, type (i.e. async), etc. If the service to be called is asynchronous, the flow must collect the correlation message included in the BPMN recipe (*message_expected*) and call the service with the parameters collected and the correlation message. The call must include the parameters necessary to identify the process instance (*instance_id*) and task (*task_id*), and the *message_expected*. If the service to be called is synchronous, the service is called and the result collected. After that, the flow invokes the *Collect_Answers* endpoint providing as input the *instance_id*, *task_id*, and the result.

2. *Receive_Reply_Service*: This flow is where asynchronous external services reply once finished. The duties for this flow include the collection of the correlation message (*message_expected*) and those parameters related to process/service invocation such as *instance_id*,

task_id and results. The flow finally calls the *Collect_Answers* module by passing the information collected.

4.4.3 Use Case: Executing and Monitoring a Piece-Classifier BPMN Workflow

This subsection presents the implementation of the Node-RED Workflow Manager in a demonstrator. The demonstrator implements a workstation built with two Lego Education Mindstorms machines. The first Lego Device (LD1) is a colour sorting machine (Lego Educational Mindstorm EV3). The second Lego Device (LD2) is a crane that collects buckets with pieces and moves them to a given position. Each Lego Device brick is attached to a Raspberry Pi board. Node-RED and OPC-UA servers are installed on those boards. Lego bricks sensors and actuators are managed through the OPC-UA server. REST web services are created with Node-RED. Those endpoints are Node-RED flows that activate the Lego Devices using OPC-UA client connections. This way Lego Devices can be controlled/monitored directly using OPC-UA clients or invoking the web services offered by Node-RED.The Node-RED WM will orchestrate processes using those web services. Among the different web services deployed by each Lego, the Node-RED WM orchestrates the following services to run different processes in the use case:

- LD1_Initialized station
- LD1_Move Left
- LD1_Move Right
- LD1_Through piece
- LD1_Select pieces(array with color and number)
- LD2_Initialized station
- LD2_Move Bucket(from pos1 to pos2)

The Node-RED WM as described in Subsection 4.4.2 is deployed in one of the embedded systems (Raspberry Pi board). Node-RED WM orchestrates service-oriented tasks included in BPMN workflows. BPMN recipes are designed using BPMN modelling tools. Fig. 4.13 presents a BPMN recipe created with Eclipse WSO2 Integration Studio 8.0.0 (other tools are also valid). BPMN recipes include Service_Tasks to invoke REST endpoints offered by the Lego devices, Parallel_Gateways to split or join process branches, and Message_Catching_Events to wait for asynchronous replies from the Lego devices.

The BPMN recipe can be uploaded to Node-RED WM as a raw XML file through the */upload_recipe* flow endpoint. The recipe is stored in the database and is ready to be activated using the Node-RED WM */start_process* flow endpoint or through a more user-friendly

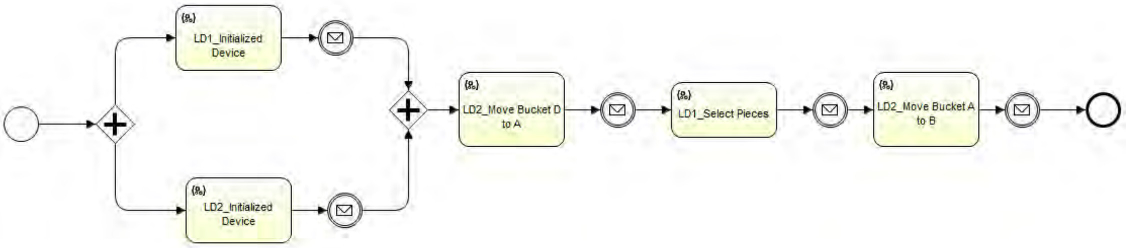
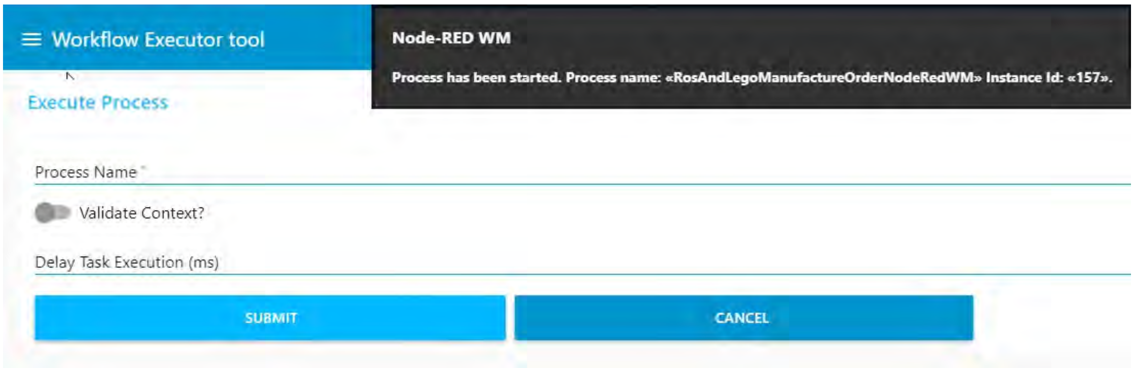


Figure 4.13: BPMN Recipe for orchestrating functionalities of a classifier robot

interface called "Workflow Executor" which is shown in Fig. 4.14. By executing a workflow process, an instance of the process is created. Node-RED WM creates separated instances of a process recipe if invoked several times and assigns a unique identifier to each instance. Several recipes can be uploaded simultaneously enabling the orchestration of different processes at the same time. Process in parallel and multiple instances of the same process can be run if the workstation resources enable it. Node-RED WM orchestrates workflows by reading the BPMN file and calling the services written in the BPMN process. The Node-RED WM goes through the recipe reading different BPMN elements. It begins with the *startEvent* element, merging and splitting process branches with the *parallelGateway* element or waiting on events with the *messageCatchingEvent* element.



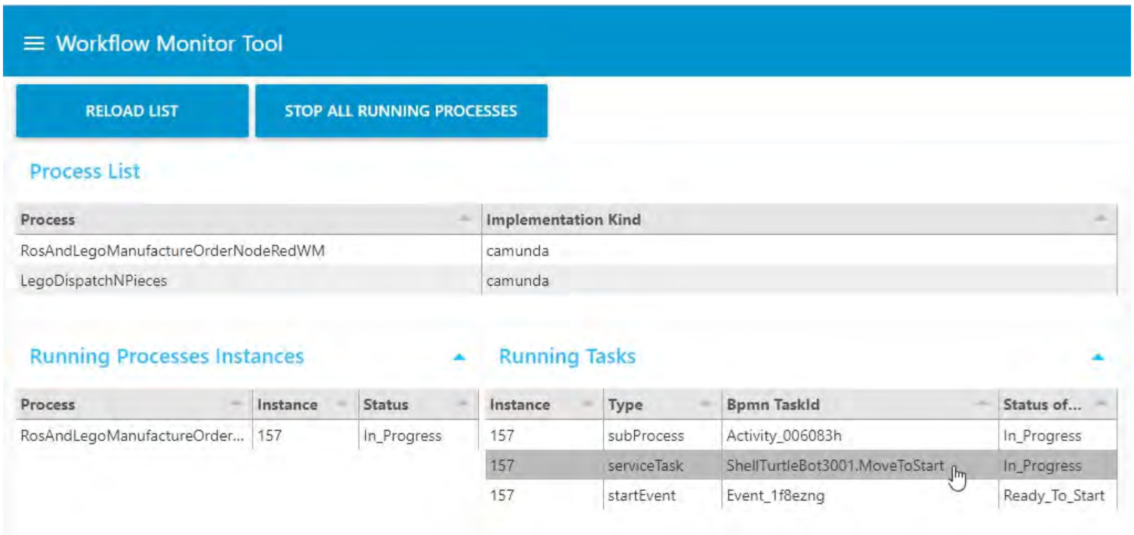
Node-RED WM

Process has been started. Process name: «RosAndLegoManufactureOrderNodeRedWM» Instance Id: «157».

Figure 4.14: Workflow Executor User Interface in Node-RED WM

In this way, Node-RED WM manages the process by controlling which task node is next in the flow until it reaches the BPMN *endEvent* node. The status of each instance and its active task are managed using the database or through a more user-friendly interface called "Workflow Monitor" which is shown in Fig. 4.15. This Workflow Monitor offers the possibility to monitor the current status of processes being executed in real-time.

In summary, Node-RED WM presents the potential to execute and monitor BPMN workflow recipes effectively. This alternative workflow manager, developed using Node-RED technology, offers opportunities for enhancement through targeted development efforts. The solution can be improved in several key areas. Firstly, it can be adapted to accommodate various recipe formats, enhancing its flexibility and usability. Secondly, integration with message



Process List

Process	Implementation Kind
RosAndLegoManufactureOrderNodeRedWM	camunda
LegoDispatchNPieces	camunda

Running Processes Instances

Process	Instance	Status
RosAndLegoManufactureOrder...	157	In_Progress

Running Tasks

Instance	Type	Bpmn TaskId	Status of...
157	subProcess	Activity_006083h	In_Progress
157	serviceTask	ShellTurtleBot3001.MoveToStart	In_Progress
157	startEvent	Event_1f8ezng	Ready_To_Start

Figure 4.15: Workflow Monitor User Interface in Node-RED WM

brokers, databases, and protocols like OPC-UA can broaden the system's capabilities, enabling seamless integration with existing infrastructure. Lastly, the inclusion of context data from sensors can significantly improve the system's adaptability and responsiveness to real-time changes.

Within the proposed architecture, a dedicated component is assigned the task of collecting and analyzing context data, a topic that will be explored in detail in the next chapter.

4.5 Conclusions

This chapter outlined the thesis proposal for a context-aware workflow management architecture, specifically delving into the components involved in both workflow design and execution phases. The proposal represents a holistic approach to addressing the challenges identified in the literature review, leveraging both Industry 4.0 and workflow standards. Key contributions of this chapter include:

- **Architecture for context-aware workflow management:** As detailed in Section 4.2, the architecture aims to tackle the challenge of "Utilization of industry-oriented standards and standardized workflow formats" by combining AAS as the Industry 4.0 standard and BPMN as the workflow notation standard. Furthermore, the adoption of a microservice-oriented architecture and a modular design addresses the challenge of "Development of decoupled components and systems, providing more versatility and integration with existing systems". Thus, ensuring scalability and modularity within the architecture.

- **Enhanced BPMN Modelling Tool:** Integral to the workflow design phase, this facet of the architecture focuses on reinforcing standardization in manufacturing process design.

As outlined in Section 4.3, key characteristics of this component include the integration of assets, establishment of a central AAS repository, integration of the Camunda Modeler, and incorporation of the AAS Service Discoverer plugin. This integrated approach facilitates the creation of standardized digital representations through AAS and streamlines the design process using BPMN as the standard notation language. Beyond design, the architecture enables the execution and monitoring of workflows, offering diverse options for manufacturing process execution in BPMN format. This approach is also characterized by being vendor-neutral. Thus, addressing the challenge of "Adoption of user-centric design approaches, transitioning from abstract workflows to executable workflows".

- **Node-RED Workflow Manager:** While the orchestration of machine and device services is efficiently achieved using any BPMN execution platform, Section 4.4 introduced Node-RED WM as a compelling alternative for workflow management. It offers versatility in deployment, seamlessly adapting to edge and cloud environments. One notable advantage of Node-RED WM is its compatibility with the BPMN workflow format, complemented by intuitive user interfaces for workflow execution and monitoring.

- **Demonstrative scenario using classifier robots:** The piece-classifier scenario showcased the capabilities of the architecture in enhancing manufacturing workflow design and execution. In Section 4.3.3, the design phase outlined the process of designing the BPMN workflow recipe, leveraging the components of the architecture for asset representation, service discovery, and standardized notation. Following the design phase, Section 4.4.3 delves into the execution phase, where the BPMN workflow is deployed and executed using Node-RED WM. This experiment demonstrates the practical application and benefits of the proposal.

The amalgamation of these components culminates in a robust architecture that optimizes both the design and execution phases of manufacturing workflows. Consequently, presenting this architecture and its components aligns with research objective O3, "Propose and develop an architecture for context-aware workflow management that takes all the benefits from current approaches and addresses active challenges". Concurrently, it addresses RQ4, "Which components and features should be integrated into a context-aware workflow management solution to address the identified challenges while leveraging advantages from existing approaches?" and delineates the stage for the subsequent chapter which introduces the core component of this architecture, enabling context-awareness capabilities.

5

Context-Analyzer: A Context-Aware Component for Dynamic Service Orchestration

In line with the proposed architecture, this chapter delves into the Context Analyzer component briefly mentioned in the preceding chapter. The Context Analyzer component plays a central role in enhancing flexibility in service orchestration, ensuring dynamic changes to the workflow during runtime by providing context-awareness capabilities. Its design responds to the challenges identified during the state-of-the-art revision, addressing specifically the following concerns: a) "Need to interpret contextual information to perform dynamic adaptations" and b) "consideration of a broader range of context variables, including both calculated and sensor-derived data".

Furthermore, this chapter aligns with the research objective "O3: Propose and develop an architecture for context-aware workflow management that takes all the benefits from current approaches and addresses active challenges". Simultaneously addressing the research question "RQ4: Which components and features should be integrated into a context-aware workflow management solution to address the identified challenges while leveraging advantages from existing approaches?".

It is worth mentioning that the development of the Context Analyzer component contributed to a scientific article in [35], which introduces the component as a key enabler to context reaction within the mentioned architecture. Furthermore, the publication provides a domain-specific ontology called DeviceServiceOnt, which facilitates the interpretation of context for assets, their services, and the quality of services.

5.1 Introduction

As stated in Chapter 3, workflow orchestration must be dynamic to handle unpredictable scenarios during the manufacturing process. Reconfigurable Systems (RS) represent a promising approach to enhancing workflow dynamicity, as they are capable of self-reconfiguration and can automatically adjust manufacturing processes in response to changing conditions [15]. Moreover, the discussion of the literature review in Section 3.3.3, underscored the significance of addressing the challenge “Need to interpret contextual information to perform dynamic adaptations”.

One promising avenue to tackling this challenge involves leveraging context information. Context information can help in determining which device or service is the most suitable for performing a particular task within a workflow [206]. Factors such as such as availability, performance, location, or other relevant variables can influence the successful completion of tasks [207]. The Semantic Web offers a framework for describing context information and providing support to machines and applications within manufacturing systems [208]. By incorporating semantic web technologies, a system can become context-aware, enabling self-configuration and cognitive behaviour in response to new or altered situations [39]. Context-awareness facilitates dynamic adaptation of workflows during runtime, ensuring compliance with Service Level Agreements (SLA) and increasing service availability and trustworthiness to meet customer demands [209].

However, enhancing the reactiveness of manufacturing workflows also requires overcoming the challenge of “Consideration of a broader range of context variables, including both calculated and sensor-derived data”. Considering a wide range of context variables is essential for making more informed decisions about the manufacturing process [194]. In modern manufacturing facilities, where various heterogeneous devices coexist, workflow systems must take into account the current state of machines and products being produced to adapt the process in real-time [195]. Hence, a dynamic context-aware workflow system that considers a diverse set of context variables can optimize resource utilization more effectively, avoiding possible bottlenecks or over-utilization [196].

To address these challenges, this chapter introduces the Context Analyzer, a context-aware component designed to enhance the reactivity of manufacturing workflows within the initially proposed architecture. This component dynamically re-selects services and devices capable of executing tasks within workflows during runtime, harnessing context information and semantic web technologies for enhanced adaptability. The structure of this chapter is organized as follows: Section 5.2 reviews the state-of-the-art with respect to service orchestration and context recognition within smart manufacturing systems. Section 5.3 conveys details on the Context Analyzer component and its role within the architecture for context-aware workflow management. Section 5.4, provides an experimental phase on a resource collection scenario where the Context Analyzer component is implemented and performance metrics are gathered and discussed. Finally, Section 5.5 presents the concluding remarks of this chapter.

5.2 Related Work

A revision of the state-of-the-art workflow management systems with context-awareness capabilities was conducted in Section 3.4. This section focuses on comparing approaches from the literature that specifically feature dynamic workflow reconfiguration through service re-selection techniques during runtime.

Integrating context-awareness into workflow management systems is essential for enhancing adaptability and flexibility within the dynamics of Industry 4.0. It involves leveraging relevant knowledge about current state of machines, products being produced, and environmental factors [210]. Context knowledge serves as a source for the service recommendation technique, which recommends services to users or systems based on context and preferences [14]. In a similar vein, workflows managers can also use context knowledge and the service recommendation technique for service or device re-selection during task execution [211]. Furthermore, combining semantic web and the service recommendation technique into workflow systems, offers an approach to wrap services or devices in task nodes that can dynamically be changed when the quality conditions are met [212].

To facilitate service recommendation, semantic web-based components and systems rely on the utilization of Functional Properties (FPs) and Non-Functional Properties (NFPs) [213]. FPs delineate the functionality of a system and its components, while NFPs encompass Quality of Service (QoS) properties that dictate how the system delivers its functionality [176]. QoS properties can be categorized into two types: sensor-derived properties, obtained from IoT devices and sensors, and calculated parameters acquired during the execution of a process [39]. Table 5.1 outlines a selection of significant QoS properties for each category.

Table 5.1: QoS Properties Classification.

Sensor-Derived	Calculated-Derived
• Environmental: Temperature, Humidity, Air quality, Noise level, Light intensity. • Location-based: Altitude, Latitude, Longitude, Indoor positioning coordinates, Accuracy of the location coordinates. • Motion and physical: Inclination, Acceleration, Angular velocity, Vibration, Pressure, Proximity.	• Performance: Availability, Response time, Latency, Throughput, Error rate, Processing time. • Security-related: Encryption strength, Authentication time, Access control measures. • Usability and user experience: User satisfaction, Ease of use, Intuitiveness of the interface, Accessibility standards. • Energy-related: Energy consumption, Battery voltage, Battery level, Battery capacity, Battery charging status.

A comparison of relevant approaches that feature dynamic service re-selection is conducted in Table 5.2. For instance, Lyu et al. [148] proposed a context manager module as part of an architecture for service orchestration. Their context-aware component continuously analyzes the system environment and decides whether to keep, adjust, or replace a service on the fly. Devices, sensors, and services are semantically described using FPs and NFPs, including inputs, outputs, and communication methods. Similarly, Bekkouche et al. [136] 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.

An automatic workflow composition approach was presented by Mazzola et al. in [128]. Their approach performs re-selection of services within BPMN recipes using optimization algorithms, which are based on QoS parameters. The approach encodes semantic annotations of business process models into the BPMN files. Similarly, Fahad et al. [156] developed a framework that combines user requirements and SWRL rules to re-select services from the semantic repository. The workflow recipe is written in BPMN format.

Alférez and Pelechano created a framework that is context-aware and autonomously adjusts service compositions at runtime in [152]. Their framework performs automatic adjustments to service compositions at runtime. Models are first created at design time using BPMN and WS-BPEL formats and ontologies. Prior to execution, a verification phase is performed to verify the mutability of the models and configurations, ensuring that service re-composition is safe. During runtime, when a contextual event arises, the previously established models are queried to reconfigure service compositions. In a similar vein, Arul et al. introduced a framework that incorporates semantics into web services and caters to the changing demands of users through automatic and dynamic service composition, as described in [155]. They employed a variant of the hierarchical task network (HTN) planner, which encodes OWL-S processes.

Table 5.2: Comparison of approaches featuring dynamic service re-selection.

REF	Feature			Workflow Format		Architecture		QoS Properties	
	SD	RS	WC	BPMN	BPEL	Cloud	Edge	Sensor-Derived	Calculated-Derived
[148]	-	✓	-	-	-	✓	✓	✓	✓
[136]	-	-	✓	-	-	✓	-	-	✓
[128]	✓	✓	✓	✓	-	✓	-	-	✓
[156]	-	✓	-	✓	-	✓	-	-	✓
[152]	-	✓	✓	✓	✓	✓	-	-	✓
[155]	✓	✓	✓	-	-	✓	-	-	✓

In summary, the literature indicates that context awareness enables systems to adapt to unexpected scenarios by leveraging context information. However, these proposals predominantly consider Calculated QoS properties and often lack integration with industrial standards, standardized workflow formats, and real asset services. Furthermore, the experiments conducted are predominantly in laboratory settings. Therefore, the next section introduces a context-aware component for dynamic service re-selection within workflows that overcomes these limitations.

5.3 The Context Analyzer component

A context-aware component for dynamic service re-selection is presented in this section, providing a detailed description of its integration within the proposed context-aware workflow management architecture. The Context Analyzer component, as illustrated in Fig. 5.1, is designed to facilitate workflow adaptations, providing a dynamic and flexible workflow at runtime.

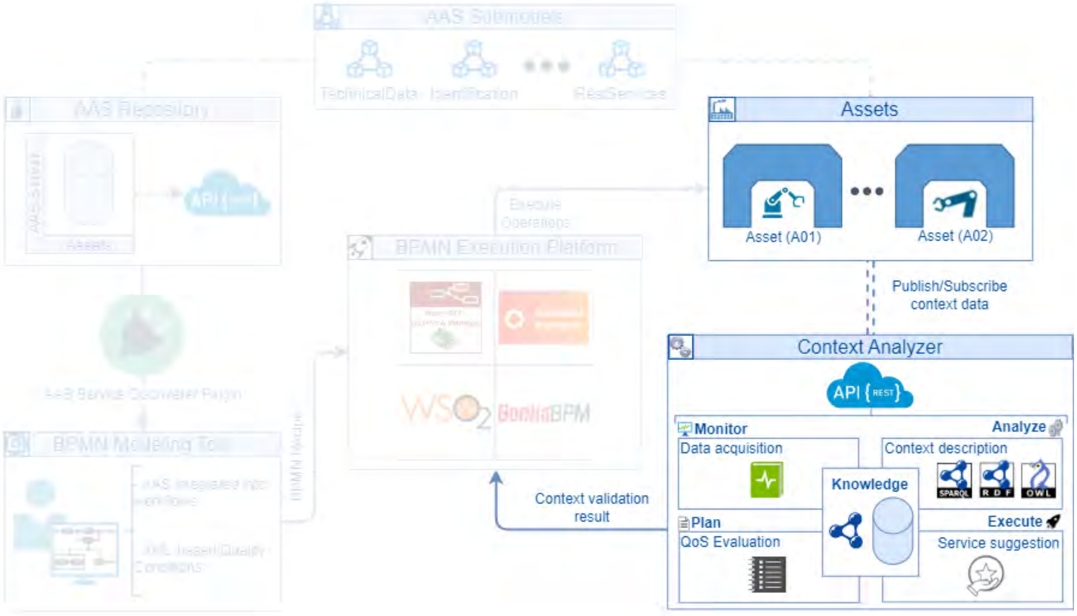


Figure 5.1: The Context Analyzer component for enabling context-aware workflow management

The Context Analyzer evaluates whether a service or device within a workflow should be replaced or not on-the-fly by analyzing context variables. The next subsections describe in detail this component, including the semantic web technologies employed for its development and its role in enabling service replacement during workflow execution.

5.3.1 Context Recognition and the MAPE-K model

Context recognition is crucial for manufacturing systems to react correctly and enable dynamic changes to the workflow during runtime based on context data [214]. Semantic Web technologies have demonstrated advantages in effectively describing and inferring context data [39]. 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 [215]. Service and device re-selection is crucial to perform workflow adaptations in response to context changes [216]. 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 [177]. MAPE-K-based systems are particularly suited to address exceptional situations that may arise during workflow execution [81]. Fig. 5.2 provides an overview of the technology stack and sub-components that comprise the Context Analyzer. This technology stack empowers the architecture to effectively interpret and leverage contextual information, enabling intelligent recommendations, decision-making, and adaptive workflow management.

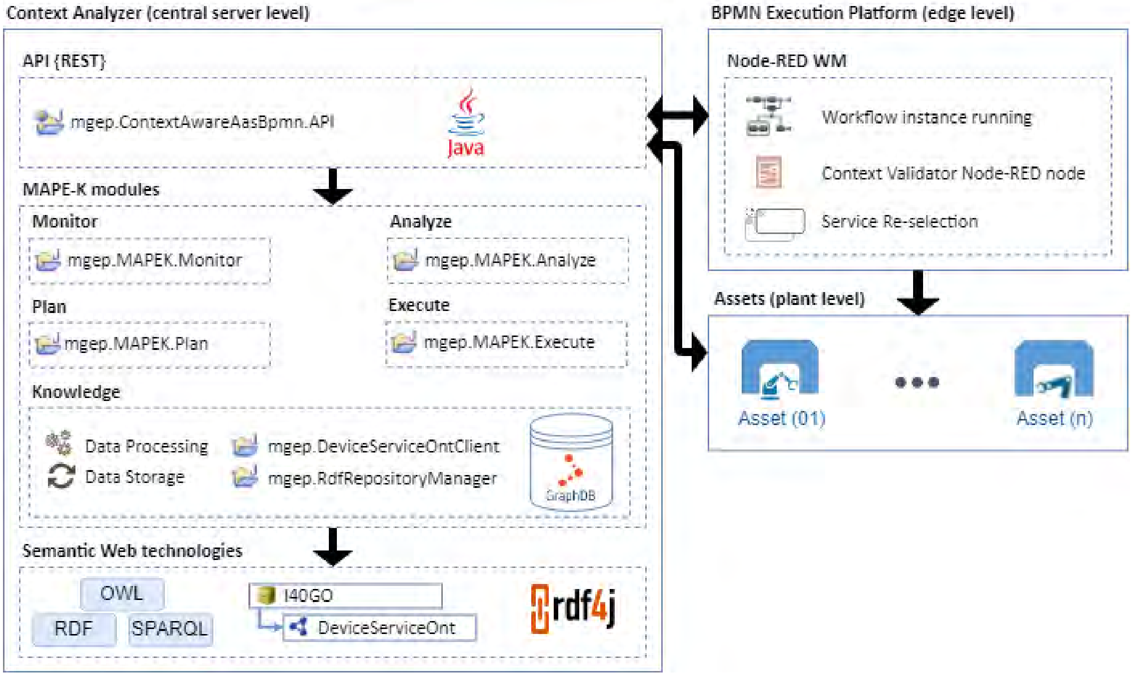


Figure 5.2: Context Analyzer technology stack and interaction with other components in this architecture

The following sections delves into the building blocks of the Context Analyzer. Each building block comprises sub-components that interact with each other to achieve service re-selection.

Building Block 1: API {REST}

The "API {REST}" building block acts as an interface linking the MAPE-K modules of the Context Analyzer with other components of the architecture. By embracing the principles of Representational State Transfer (REST), this block enables the Context Analyzer to adopt a flexible and standardized approach to data exchange. This simplifies integration efforts and ensures compatibility with emerging technologies, thereby facilitating collaboration with external systems.

Building Block 2: MAPE-K modules

The "MAPE-K modules" building block is a framework that encompasses five key subcomponents: Monitor, Analyze, Plan, Execute, and Knowledge (MAPE-K). These modules work in conjunction to enable autonomous and adaptive behaviour within the Context Analyzer. The following itemization briefly explains the MAPE-K modules of the Context Analyzer and their interactions with other components of the proposed architecture:

- **Monitor:** Comprehends a module known as the Context Monitor, which continuously gathers real-time data from devices and sensors at the plant level. Leveraging AAS, the Context Monitor retrieves relevant information regarding connectivity options for accessing real-time data from assets. It offers a variety of connectivity options, including OPCUA, HTTP, ROS Topics, MQTT, and more.

  The collected data is stored in a semantic repository, which can be any semantic web repository supporting RDF. In this proposal, GraphDB is utilized to insert the data. Furthermore, the Context Monitor adheres to the schemas defined in the DeviceServiceOnt ontology (explained in Section 5.3.2), facilitating the dynamic manipulation of the data in RDF format.

  The Context Monitor provides customization possibilities through a configuration file that allows to change its default settings. For example, users can adjust the frequency of data collection, which is set to 1000ms by default. Other configuration options are available as well, such as the IP address of the semantic server, the identifier of the semantic repository, the IRI of the semantic ontology, and the IP address of the AAS server can be modified. These configuration options collaborate in adjusting the behaviour of the Context Monitor module.
- **Analyze:** During the computation phase, this module receives information regarding administration shells, assets, services, and quality conditions. SPARQL queries are built dynamically based on the schemas of the DeviceServiceOnt ontology and the received quality conditions. Subsequently, these queries are executed within the semantic repository to retrieve relevant data. The algorithm for the analysis phase is further detailed in Section 5.3.3.
- **Plan:** In the planning phase, this component utilizes the resultset obtained from executing the SPARQL queries within the semantic repository. It then processes this data to generate the final response, which is a JSON object representing a recommendation for service replacement.
- **Execute:** The recommendation object is delivered to the API consumer, which in this proposal can be any BPMN Execution Platform of those analyzed in Section 4.4. The workflow executor will decide whether to accept the suggestion or not. However, Node-RED WM is prepared to accept the suggestions proposed by the MAPE-K module, 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.

Building Block 3: Semantic Web technologies

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 SPARQL Protocol and RDF Query Language (SPARQL), which collectively empower the architecture with semantic capabilities [217].

OWL serves as the language for ontology design, facilitating the interpretability of information by machines [218]. 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, meanwhile, acts as a data model for expressing information about resources in a graph-like format using subject-predicate-object triples [219]. 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, as the query language for RDF [220], enables the Context Analyzer to extract semantically meaningful information in real-time from the semantic repository.

With these building blocks comprising the Context Analyzer component, the subsequent section explores the DeviceServiceOnt ontology and its utilization of knowledge from I40GO, a global ontology for Industry 4.0 applications.

5.3.2 DeviceServiceOnt Development Process

The DeviceServiceOnt[1] ontology represents knowledge as triplets, encompassing information about devices, sensors, their services, and the quality of which they deliver the services. Furthermore, reusing ontologies ensures interoperability and alignment with established standards, facilitating integration with existing systems and enabling semantic interoperability across domains [221]. Thereby, the design of the DeviceServiceOnt ontology, as illustrated in Fig. 5.3, incorporates relevant classes and properties from I40GO[2], a global ontology for Industry 4.0 applications [37].

The development of DeviceServiceOnt involved collaboration between ontology engineers and web engineers. This ontology was created leveraging modules and classes from I40GO and implemented using Protégé. The development process is outlined in the following key phases:

---

[1] `https://github.com/MUFacultyOfEngineering/ContextAnalyzer/blob/main/DeviceServiceOnt.owl`

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

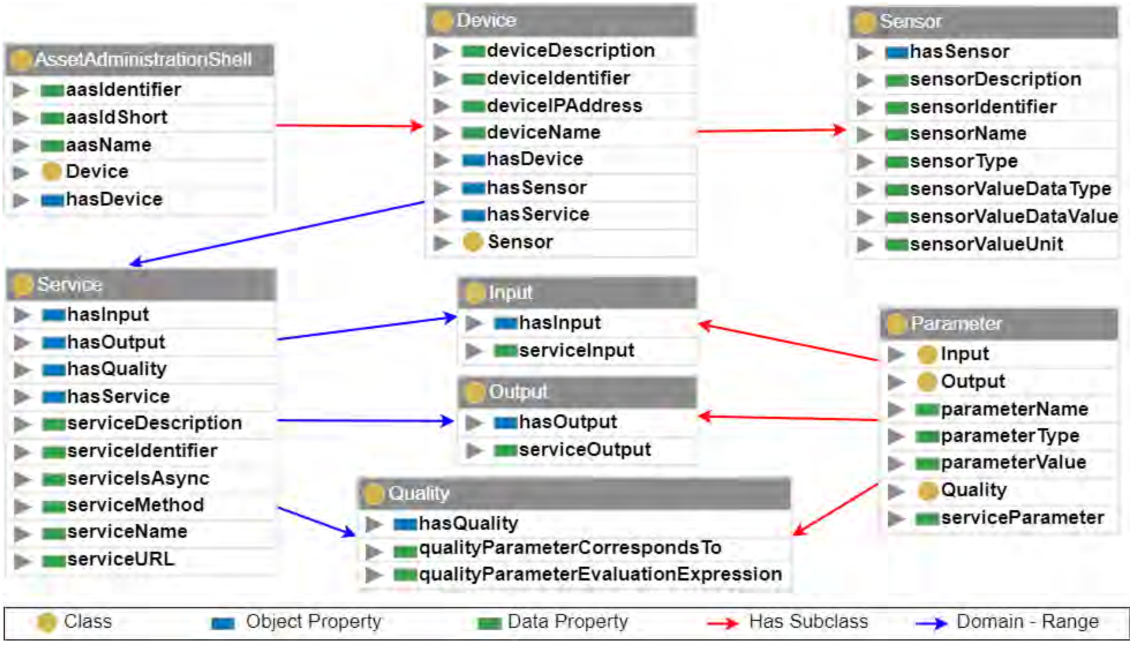


Figure 5.3: DeviceServiceOnt ontology classes and relations

Phase 1. Ontology Requirement Definition: The initial step involves defining functional ontology requirements. These requirements refer to the specific criteria or characteristics that an ontology must fulfill in order to effectively represent the concepts and relationships within a particular domain [222]. For the development of DeviceServiceOnt, the ontology requirements include:

(a) Administration Shells: This involves defining classes and properties associated with the Asset Administration Shell associated to manufacturing assets.

(b) Devices: Specify classes to represent various types of devices utilized in manufacturing processes, including their sensors, network information, and interrelationships.

(c) Services: Define classes for the various services offered by manufacturing devices or machines, including properties such as input and output parameters, synchronous or asynchronous nature, and endpoint.

(d) Quality: This aspect entails describing classes and properties related to the quality aspects of how devices or machines deliver their services, including metrics such as reliability, performance, availability, and others.

These ontology requirements aim to capture the essential concepts and relationships relevant to manufacturing processes, enabling effective representation and interpretation of data within the context of the DeviceServiceOnt ontology.

Phase 2. Ontology Reuse and Re-engineering: In this phase, suitable classes from I40GO are identified for reuse. To accomplish this, ontology engineers analyzed modules from the *Variant-domain*, *Common-domain*, and *Domain-task* layers. Consequently, Table 5.3

summarizes the selected classes and modules from the sensor ontology[3], manufacture ontology[4], equipment ontology[5], and soa ontology[6] for integration into the DeviceServiceOnt ontology. The integration process was carried in Protégé by importing the modules and referencing the classes using the *owl:imports* statement.

Table 5.3: I40GO modules and classes reused by DeviceServiceOnt

Layer	Module	Class	Description
Variant-domain	Sensor	Sensor	Constitutes sensors that belong to machines/devices.
Common-domain	Manufacture Equipment	Administration Shell Device	Represents machines/devices and their administration shells.
Domain-task	SOA	Service	Describes web services of machines/devices.

Furthermore, the reused I40GO modules and classes are incorporated and adapted to align with the functional requirements of DeviceServiceOnt. Domain-specific classes, including Parameter, Input, Output, and Quality, are incorporated to match the needs of this application case. However, existing classes such as AssetAdministrationShell, Device, Sensor, and Service remain unchanged. Fig. 5.4 illustrates the newly added classes and their properties. Additionally, any unnecessary knowledge from the reused modules is pruned.

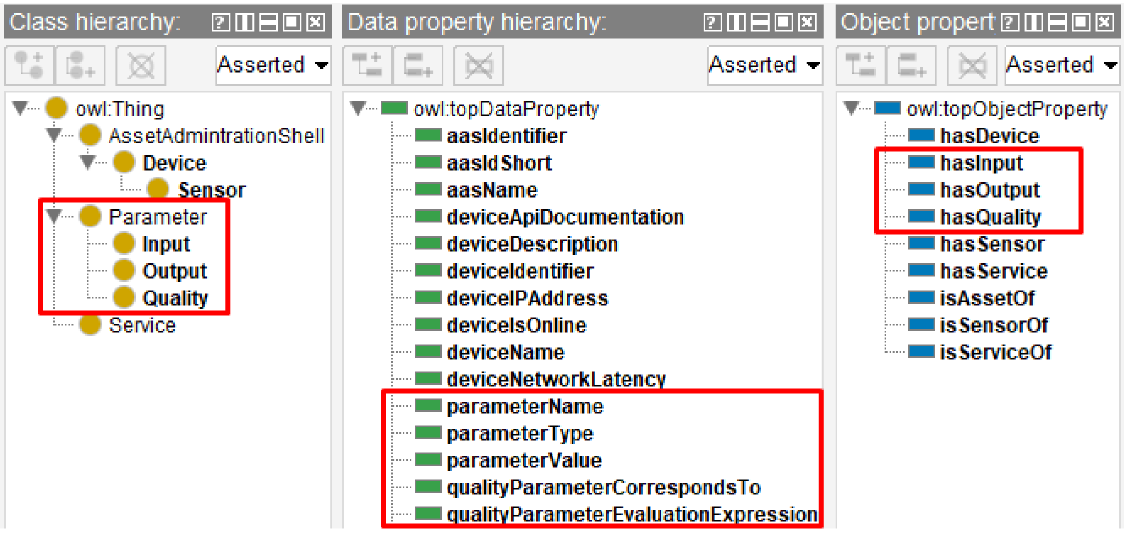


Figure 5.4: Class and property structure of DeviceServiceOnt ontology

Before diving into the testing phase of the DeviceServiceOnt, it is imperative to elucidate the algorithm for service/device re-selection, outlining the possible scenarios that may arise during the selection process. By examining these scenarios the groundwork for testing ontology is established.

---

[3] `http://www.purl.org/i4go/variant-domain/sensor`
[4] `http://www.purl.org/i4go/common-domain/manufacture`
[5] `http://www.purl.org/i4go/common-domain/equipment`
[6] `http://www.purl.org/i4go/domain-task/soa`

5.3.3 Best Service/Device Selection Algorithm

The Context Analyzer component executes the best device or service selection process according to the steps outlined in Algorithm 1. This algorithm relies on quality conditions, which specify the desired characteristics of a service or device. Quality conditions can encompass various aspects such as performance metrics, availability, and other relevant factors that influence the suitability of a service or device for a given task. For instance, a quality condition for a task might specify "$HUMIDITY <= 52$", indicating that a device with lower humidity level is preferable, or "$SuccessRate > 90$", suggesting that a device or service with a higher success rate is desired.

In the Context Analyzer algorithm, quality conditions serve as minimal requirements that a service or device must satisfy to be considered for selection. These conditions are also utilized later to identify the best service or device from the available candidates. To accomplish this, the algorithm generates SPARQL queries based on the service name and the quality conditions. These queries are then sent to the semantic repository, and the resulting dataset is returned to the Context Analyzer for further analysis.

The algorithm in the Context Analyzer assigns weights to all candidates in the semantic repository to determine which of them fully or partially satisfy the quality conditions. For instance, if there are five quality conditions and one instance meets all five, the "Conditions Met Rate" is 100%. Conversely, if an instance meets only two conditions, 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 [215, 223] 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 favourable 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 (>, >=, <, <=) provided in the quality condition. For instance, “$NETWORK\_LATENCY <= 100$” indicates that a device or service with less network latency is better, while “$BATTERY\_LEVEL > 90$” means a device with higher battery level 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.

Algorithm 1 Context Analyzer Best Service/Device Selection

```
Require: Service Name and List of Quality Conditions
  function SelectBestService(ServiceName, QualityConditions)
      $resultSet \leftarrow \emptyset$
      while $resultSet = \emptyset$ and $length(QualityConditions) \geq 1$ do
          $query \leftarrow buildSparqlQuery(ServiceName, QualityConditions)$
          $sortingSubQuery \leftarrow empty$
          for $condition \in QualityConditions$ do
              $symbol \leftarrow extractConditionalSymbol(condition)$
              $propertyName \leftarrow extractPropertyName(condition)$
              $sortingSubQuery \leftarrow sortingSubQuery + buildSortingSubQuery(symbol, propertyName)$
          end for
          $query \leftarrow query + sortingSubQuery$
          $resultSet \leftarrow executeQuerySemanticRepository(query)$
          if $resultSet = \emptyset$ then
              $QualityConditions \leftarrow removeLatestCondition(QualityConditions)$
          end if
      end while
      return $getFirst(resultSet)$
  end function
```

After elucidating the algorithm for selecting the best service or device within the Context Analyzer, the subsequent step involves testing the DeviceServiceOnt through the execution of SPARQL queries. These queries are designed to validate the functionality and effectiveness of the ontology considering the previously mentioned case scenarios.

5.3.4 Testing DeviceServiceOnt through SPARQL Queries

The DeviceServiceOnt ontology was tested using SPARQL queries for the scenarios mentioned in section 5.3.3. Each of these scenarios provides insights into different decision-making situations. It is important to note that these queries are not hardcoded but built dynamically by the Context Analyzer component as explained in Algorithm 1. These queries serve to test the DeviceServiceOnt correctness and flexibility under diverse case scenarios.

Fig. 5.5 provides the SPARQL query and outcomes for the first scenario. The query results confirm that the selected device/service fulfils all specified quality conditions. As the chosen instance complies with all quality criteria, the workflow orchestrator can confidently employ it for executing the task, ensuring a high likelihood of task completion. The task completion probability is directly tied to the number of quality conditions satisfied. For example, if four conditions are successfully met, the task completion probability stands at 100%.

```
PREFIX ↔
select ?deviceName ?serviceUrl ?qosValAvgResponseTime ?qosValHUMIDITY where {
 ?device dsOnt:aasIdentifier ?aasIdentifier .    ?device dsOnt:deviceName ?deviceName .
 ?device dsOnt:hasService ?service .             ?service dsOnt:serviceName ?serviceName .
 ?service dsOnt:serviceURL ?serviceUrl .         ?service dsOnt:hasQuality ?qAvgResponseTime .
 ?qAvgResponseTime dsOnt:parameterName "AvgResponseTime" .
 ?qAvgResponseTime dsOnt:parameterValue ?qosValAvgResponseTime .
 ?service dsOnt:hasQuality ?qHUMIDITY .
 ?qHUMIDITY dsOnt:parameterName "HUMIDITY" .
 ?qHUMIDITY dsOnt:parameterValue ?qosValHUMIDITY .
 filter (xsd:integer(?qosValAvgResponseTime)  < 8000) .
 filter (xsd:decimal(?qosValHUMIDITY)  < 50) .
 filter (?aasIdentifier = "AssetAdministrationShell---2") .
 filter (?serviceName = "Service_MoveLeft") .
```

Showing results from 1 to 1 of 1. Query took 0.1s, minutes ago.

	deviceName	serviceUrl	qosValAvgResponseTime	qosValHUMIDITY
1	"Device_ColorSorter02"	"http://192.168.56.102:80/robot/move_left"	"6619"	"47"

Figure 5.5: SPARQL query and results for case 1 "When a single device/service satisfies all quality conditions"

In the second scenario, depicted in Fig. 5.6, the SPARQL query reflects the outcome when numerous instances (devices/services) meet all quality conditions, resulting in a 100% task completion probability for all of them. To facilitate decision-making, the query incorporates a sorting statement ("ORDER BY"), yielding an ordered list of instances that considers both battery and temperature quality values. Ultimately, the first instance in this list is deemed the most suitable device/service, which the workflow orchestrator can utilize for executing the task.

Moving to the third scenario, Fig. 5.7 portrays the query when none of the devices/services fulfils all quality conditions. Consequently, the query must be iteratively adjusted to eliminate quality conditions one by one, starting with the lowest priority. This iterative process continues until at least one instance appears in the resultset. In this context, the task completion probability becomes variable. For instance, if only two out of four conditions are met, the task completion probability is 50%, prompting the workflow orchestrator to proceed with task execution while being cautioned about potential uncertainties regarding completion.

For the final scenario, Fig. 5.8 depicts the query when none of the devices/services align with the specified quality conditions, and the successive removal of quality conditions fails to yield any single instance. In such cases, all quality conditions are discarded, and an "*ORDER*

```
PREFIX ↔
select ?deviceName ?serviceUrl ?qosValBATTERY ?qosValTEMPERATURE where {
 ?device dsOnt:aasIdentifier ?aasIdentifier .   ?device dsOnt:deviceName ?deviceName .
 ?device dsOnt:hasService ?service .            ?service dsOnt:serviceName ?serviceName .
 ?service dsOnt:serviceURL ?serviceUrl .        ?service dsOnt:hasQuality ?qBATTERY .
 ?qBATTERY dsOnt:parameterName "BATTERY" .      ?qBATTERY dsOnt:parameterValue ?qosValBATTERY .
 ?service dsOnt:hasQuality ?qTEMPERATURE .      ?qTEMPERATURE dsOnt:parameterName "TEMPERATURE" .
 ?qTEMPERATURE dsOnt:parameterValue ?qosValTEMPERATURE .
 filter (xsd:decimal(?qosValBATTERY)  >= 25) .
 filter (xsd:decimal(?qosValTEMPERATURE)  < 400) .
 filter (?serviceName = "Service_ThrowPiece") .
} ORDER BY DESC(?qosValBATTERY && ?qosValTEMPERATURE)
```

Showing results from 1 to 2 of 2. Query took 0.1s, moments ago.

	deviceName	serviceUrl	qosValBATTERY	qosValTEMPERATURE
1	"Device_ColorSorter01"	"http://192.168.56.101:80/robot/throw_piece"	"96"	"47"
2	"Device_ColorSorter03"	"http://192.168.56.103:80/robot/throw_piece"	"44"	"85"

Figure 5.6: SPARQL query and results for case 2 “When multiple devices/services meet all quality conditions”

```
PREFIX ↔
select ?deviceName ?serviceUrl ?qosValBATTERY ?qosValTEMPERATURE where {
 ?device dsOnt:aasIdentifier ?aasIdentifier .   ?device dsOnt:deviceName ?deviceName .
 ?device dsOnt:hasService ?service .            ?service dsOnt:serviceName ?serviceName .
 ?service dsOnt:serviceURL ?serviceUrl .        ?service dsOnt:hasQuality ?qBATTERY .
 ?qBATTERY dsOnt:parameterName "BATTERY" .      ?qBATTERY dsOnt:parameterValue ?qosValBATTERY .
 ?service dsOnt:hasQuality ?qTEMPERATURE .      ?qTEMPERATURE dsOnt:parameterName "TEMPERATURE" .
 ?qTEMPERATURE dsOnt:parameterValue ?qosValTEMPERATURE .
 filter (xsd:decimal(?qosValBATTERY)  >= 25) .
 ~~filter (xsd:decimal(?qosValTEMPERATURE)  < 45) .~~
 filter (?serviceName = "Service_ThrowPiece") .
} ORDER BY DESC(?qosValBATTERY && ?qosValTEMPERATURE)
```

Showing results from 1 to 2 of 2. Query took 0.1s, moments ago.

	deviceName	serviceUrl	qosValBATTERY	qosValTEMPERATURE
1	"Device_ColorSorter01"	"http://192.168.56.101:80/robot/throw_piece"	"96"	"47"
2	"Device_ColorSorter03"	"http://192.168.56.103:80/robot/throw_piece"	"44"	"85"

Figure 5.7: SPARQL query and results for case 3 “When none of the devices/services satisfies all quality conditions but some partially meet them”

BY” statement is added, resulting in an ordered list of instances. Given that these instances do not satisfy any quality conditions, the workflow orchestrator must be aware that utilizing them for task execution does not guarantee its completion, rather it implies a high likelihood of non-completion.

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 performance of the context-aware component using stress tests and growing volumes of data.

```
PREFIX ↔
select ?deviceName ?serviceUrl ?qosValBATTERY ?qosValTEMPERATURE where {
 ?device dsOnt:aasIdentifier ?aasIdentifier .   ?device dsOnt:deviceName ?deviceName .
 ?device dsOnt:hasService ?service .            ?service dsOnt:serviceName ?serviceName .
 ?service dsOnt:serviceURL ?serviceUrl .        ?service dsOnt:hasQuality ?qBATTERY .
 ?qBATTERY dsOnt:parameterName "BATTERY" .      ?qBATTERY dsOnt:parameterValue ?qosValBATTERY .
 ?service dsOnt:hasQuality ?qTEMPERATURE .      ?qTEMPERATURE dsOnt:parameterName "TEMPERATURE" .
 ?qTEMPERATURE dsOnt:parameterValue ?qosValTEMPERATURE .
~~filter (xsd:decimal(?qosValBATTERY)  >= 25) .~~
~~filter (xsd:decimal(?qosValTEMPERATURE)  < 45) .~~
 filter (?serviceName = "Service_ThrowPiece") .
} ORDER BY DESC(?qosValBATTERY && ?qosValTEMPERATURE)
```

Showing results from 1 to 2 of 2. Query took 0.1s, moments ago.

	deviceName	serviceUrl	qosValBATTERY	qosValTEMPERATURE
1	"Device_ColorSorter01"	"http://192.168.56.101:80/robot/throw_piece"	"19"	"47"
2	"Device_ColorSorter03"	"http://192.168.56.103:80/robot/throw_piece"	"14"	"85"

Figure 5.8: SPARQL query and results for case 4 “When none of the devices/services aligns with any quality condition”

5.4 Stress Testing the Context Analyzer Component with Growing Data Volumes

The performance of the Context Analyzer for service re-selection is assessed in this section. This evaluation considers both Sensor-Derived and Calculated-Derived Quality of Service (QoS) properties, as outlined in Section 5.2. The experiment was conducted using a 2.11-GHz Intel Core i5-10210U PC with 16 GB of memory running Windows 10. Installation and configuration of the AAS Server and GraphDB were carried out within a Docker container, alongside the Context Analyzer component on the same computer. To simulate various scenarios, diverse datasets were generated using the “DataGeneration” script available in the Context Analyzer repository[7], simulating multiple instances of robots with varying quality levels.

In this approach, quality conditions are defined during the design phase. For this experiment, the quality conditions for a “Delivery” task are set as follows:

$$Battery >= 20 \ \&\& \ Proximity <= 200 \ \&\& \ EnergyConsumption < 3 \ \&\& \ PayloadCapacity > 2 \ \&\& \ Latency < 100$$

Subsequently, 200 instances of delivery robots were generated, each with random $Battery$ values ranging from 0% to 100%, random $Proximity$ values ranging from 0m to 1000m (where 0m represents the robot being at the pickup location and 1000m indicates a significant distance from the pickup location), random $EnergyConsumption$ values ranging from 1%/m to 5%/m (indicating the percentage of battery consumed per meter travelled), random

---

[7] https://github.com/MUFacultyOfEngineering/ContextAnalyzer/blob/main/mgep.ContextAwareAasBpmn.Test/src/mgep/ContextAwareAasBpmn/Test/Main/DataGenerationLegoColourSorter.java

PayloadCapacity values ranging from 1kg to 10kg (denoting the weight the robot can carry), and random *Latency* values ranging from 10ms to 500ms (reflecting the average network latency experienced by the robot within the network).

To provide a glimpse of the generated data and illustrate which robots could be capable of completing the designated delivery task, a subset of the instances is presented in Table 5.4. For instance, the AAS003 robot would not be able to complete the delivery task due to its *Proximity* value of 259m (relative to the pickup location), its *Battery* level of 22%, and its *PayloadCapacity* of 1kg. These values do not meet the conditions established for completing this task. In contrast, the AAS128 robot is well-equipped for the task, with a *Proximity* value of 15m, a *Battery* level of 92%, and a *PayloadCapacity* of 3kg, meeting the conditions established for completing this task.

Table 5.4: Dataset subset - QoS evaluation examples for Deliver Package Task.

Asset	Battery	Proximity	EnergyConsumption	Payload Capacity	Latency	Task Completion
AAS003	22%	259m	2%/m	1kg	388ms	NO
AAS128	92%	15m	1%/m	3kg	97ms	YES
AAS112	99%	12m	2%/m	5kg	73ms	YES
AAS020	96%	12m	2%/m	3kg	64ms	YES
AAS187	14%	275m	4%/m	1kg	461ms	NO
AAS165	76%	3m	2%/m	6kg	89ms	YES

The stress test measures the response time of the Context Analyzer API using an increasing amount of data, as in [216]. A digital representation of the robot was created in order to conduct this evaluation because of the advantage of creating multiple Digital Twins (DT) from a physical one. Thereby an increasing number of DT were considered. The Response time is evaluated by increasing the amount of data in the semantic repository in each epoch. During the first run, the data consists of 250 Digital Twins (DT), 1,750 Services, and 19,250 QoS instances. In the next epoch, the data is doubled until epoch number 10, in which the data consists of 128,000 DT, 896,000 Services, and 9,856,000 QoS instances. During each epoch, 10 individual tests are conducted, and the average response time is considered.

Fig. 5.9 shows the average response time of the Context Analyzer API versus the amount of data in the semantic repository during each epoch. This stress test demonstrates that (1) When the semantic repository contains a small amount of data (nearly 1,000,000 QoS instances at max), the response time is fast and constant (between 3.52 and 4.70 seconds). (2) When the amount of data in the semantic repository is large (nearly 5,000,000 QoS instances and above), the response time increases significantly. The constant and fast response time for short data might be possible due to various factors [224]:

- Indexing: GraphDB uses various indexing techniques to speed up queries.
- Query Optimization: GraphDB is designed to optimize queries automatically by choosing the best execution plan based on the query structure, data distribution, and available resources.

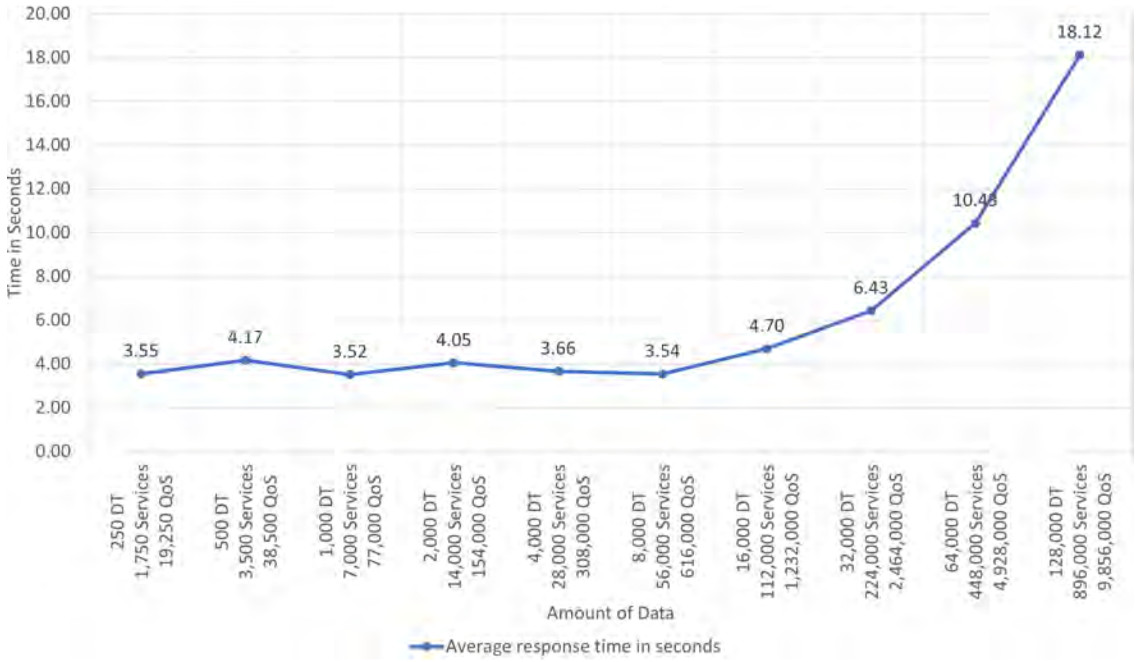


Figure 5.9: Response time of Context Analyzer's API using stress test and growing data volumes

- Data Volume: Although the stress test until epoch 7 used a relatively large number of instances (i.e epoch 7 with 16,000 DT, 112,000 Services, and 1,232,000 QoS), the volume of data may not be large enough to significantly impact the query response times.

It is likely that a combination of these factors contributes to the fast query response times. However, it is important to note that query performance can vary depending on the complexity of the query, the data distribution, and the available hardware resources. In the case of large amounts of data, the long response time could be improved by parallelizing queries [225].

5.5 Conclusions

This chapter introduced a component for context-aware service re-selection, playing a pivotal role within the architecture proposed in Chapter 4. The Context Analyzer component provides a holistic approach to address the challenges identified during the state-of-the-art revision, specifically tackling the following concerns: a) "Need to interpret contextual information to perform dynamic adaptations" and b) "consideration of a broader range of context variables, including both calculated and sensor-derived data". Key contributions of this chapter include:

- **Context Analyzer component:** Integration of the Context Analyzer into the proposed architecture enabled dynamic service re-selection within workflows at runtime. Leveraging semantic web technologies and the MAPE-K model, this component provided context-awareness capabilities to the architecture.

- **MAPE-K modules:** The Context Analyzer follows the MAPE-K architectural model for autonomous systems, incorporating modules for continuous monitoring, analysis, planning, execution, and knowledge management of manufacturing assets. These modules collectively enabled real-time reactiveness within the architecture.

- **DeviceServiceOnt ontology:** The DeviceServiceOnt ontology provides a structured data representation for storing and interpreting data about diverse manufacturing assets, services, and quality properties. Integrated into the Context Analyzer, it empowers semantic interoperability among manufacturing assets and accommodates their quality of service in a semantic repository.

- **Best Service/Device Selection algorithm:** The integration of the best service/device selection algorithm within the Context Analyzer supports the DeviceServiceOnt ontology, allowing inference based on minimal constraints and current quality values of devices and services. This algorithm identifies the most suitable service or device for task execution, providing recommendations for service replacement to the workflow manager.

The amalgamation of these contributions yields a context-aware component that aims to enhance workflow flexibility and adaptability. Consequently, presenting the Context Analyzer component aligns with research objective O3, “Propose and develop an architecture for context-aware workflow management that takes all the benefits from current approaches and addresses active challenges”, and addresses RQ4, “Which components and features should be integrated into a context-aware workflow management solution to address the identified challenges while leveraging advantages from existing approaches?”. This sets the stage for the subsequent chapter, which presents a case study using ROS-based robots in a simulated warehouse scenario to validate the entire proposal.

6

Realistic Implementation of a Warehouse Scenario using ROS-Based Robots

Building upon the comprehensive exploration of the proposed architecture and its individual components described previously, this chapter delves into practical implementation and evaluation. This evaluation conveys a realistic implementation of the proposed architecture for context-aware workflow management through simulation in a warehouse scenario. The scenario revolves around a "collect work order resources" workflow process, wherein robots are assigned roles such as classifying, picking, packing, and delivering materials. Furthermore, this realistic implementation tackles the challenge of "Need for real-world implementations", identified during the literature review.

Furthermore, this chapter aligns with the research objectives "O4: Evaluate the proposal using several case studies on diverse manufacturing scenarios" and "O5: Analyze the results and findings from the evaluation phase". Simultaneously addressing the research question "RQ5: How does the performance of a context-aware workflow management solution that incorporates the features found in RQ4, manifest in a manufacturing scenario, and what are the key findings drawn from its evaluation, including limitations and potential improvements?".

It is worth mentioning that the development of this integral evaluation contributed to a scientific article in [36], which emphasizes the key metrics drawn from evaluating the context-awareness capabilities of the architecture in enhancing flexibility and adaptability of manufacturing workflows. Furthermore, the publication provides key indicators that corroborate the benefits of this proposal, including Task Completion Rate, Task Completion Time, and Resource Utilization.

6.1 Introduction

The literature review conducted in Chapter 3 underscored the significance of context-awareness in enhancing the flexibility and adaptability of workflow systems. As discussed in Section 3.4.2, existing context-aware workflow management systems corroborated their capability to leverage real-time environmental and situational data for dynamic workflow adaptation. Despite these advancements, an imperative challenge remains on the "Need for real-world implementations" to validate and refine these theoretical advancements.

In response to this challenge, this chapter delves into the practical implementation and evaluation of the context-aware workflow management architecture proposed in Chapters 4 and 5. This practical implementation involves a realistic warehouse scenario and robots to demonstrate the capabilities of the proposed architecture. Realistic scenarios are commonly used to identify potential bottlenecks that may arise during actual deployment [226].

In the field of robotics and simulation, various tools are available for developing realistic scenarios. Among the commonly used simulation tools are:

- **Gazebo[1]:** An open-source robot simulation tool often used for academic purposes. It allows for the rapid creation of 3D environments to test algorithms, design robots, and train AI systems. Gazebo offers a wide range of features including physics simulation, sensor simulation, and asynchronous communication support.

- **Unreal Engine[2]:** A photo-real game engine known for its advanced graphics capabilities and a robust development environment. It provides tools such as physics simulation and behaviour scripting for creating realistic worlds, offering immersive experiences to end users.

- **NVIDIA Isaac Sim[3]:** A simulation platform specifically designed for robotics applications. NVIDIA Isaac Sim is widely adopted by industry leaders such as Amazon Warehouse, Fraunhofer, and Festo. It provides photo-realistic and physically accurate virtual environments, facilitating realistic evaluation of robotic systems.

In the context of the warehouse simulation environment presented in this chapter, realism is of high relevance to tackle the "Need for real-world implementations" challenge. Therefore, NVIDIA Isaac Sim was chosen for its ability to provide advanced results in mirroring real-world conditions. This includes factors such as lighting, object physics, and sensor interactions, essential for effectively testing the performance of the proposed context-aware workflow management architecture. Additionally, NVIDIA Isaac Sim seamlessly integrates seamlessly with robotic application technologies, including the Robot Operating System (ROS).

ROS is an open-source framework widely used in robotics research and development that provides a set of tools, libraries, and conventions for building both academic and industrial robotic systems [227]. At its core, ROS offers a distributed computing architecture that enables communication through a publish-subscribe messaging system, facilitating collaboration and code reuse across diverse systems [228].

---

[1] `https://gazebosim.org`

[2] `https://www.unrealengine.com`

[3] `https://developer.nvidia.com/isaac-sim`

The warehouse simulation conducted in this chapter aims to demonstrate that the proposed architecture, with its inherent context-awareness capabilities, can optimize key metrics such as task completion rate, task completion time, and resource utilization. Thus, improving efficiency in the overall manufacturing process. The structure of this chapter is organized as follows: Section 6.2 assesses the effectiveness of the proposal for improving operational efficiency in a real-world manufacturing scenario of collecting work order resources where multiple robots are employed. Section 6.3 discusses the results obtained from this evaluation. Finally, Section 6.4 presents the concluding remarks of this chapter.

6.2 Evaluation

This section conducts an evaluation of the proposed context-aware workflow management architecture through a case study involving a warehouse scenario with ROS-based robots. The evaluation aims to assess the performance of the architecture and its context-awareness capabilities, which rely on the quality properties of robots obtained by the Context Monitor through subscriptions to ROS topics. Key metrics such as task completion rate, resource utilization, and task completion time serve as benchmarks for this evaluation.

6.2.1 Experiment Setup

The experiment was conducted in a simulated scenario using Nvidia Isaac Sim, providing a photo-realistic and physically accurate environment. For robot navigation control, ROS Noetic was employed as the controller software. The simulation was executed on a Google Cloud Virtual Machine with specifications including 12 vCPUs, 28 GB of memory, an NVIDIA Tesla T4 GPU, and Ubuntu 20.04 operating system.

The warehouse scenario includes 10 Carter v1 robots[4], each equipped with differential drive technology, lidar sensors, and cameras for environmental perception. Fig. 6.1 displays the front and back views of the Carter v1 robot utilized in the experiment.

Visualization was enabled through RViz, a complementary ROS tool that enables robots to perceive the world, while providing users with a monitor interface to manage their state. Fig. 6.2 illustrates the simulation environment in both Isaac Sim and RViz, depicting the 10 Carter v1 robots positioned at their respective start positions. Additionally, two transit cones and a wet floor sign were placed as obstacles to difficult the navigation of the robots.

Alongside simulation setup, the components of the architecture outlined in Chapter 4 were properly deployed and configured. This deployment unfolded as follows:

- **Google Cloud Virtual Machine:** The AAS Server (Basyx) and the semantic repository (GraphDB) were installed in this server employing Docker containers. The utilization of

---

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

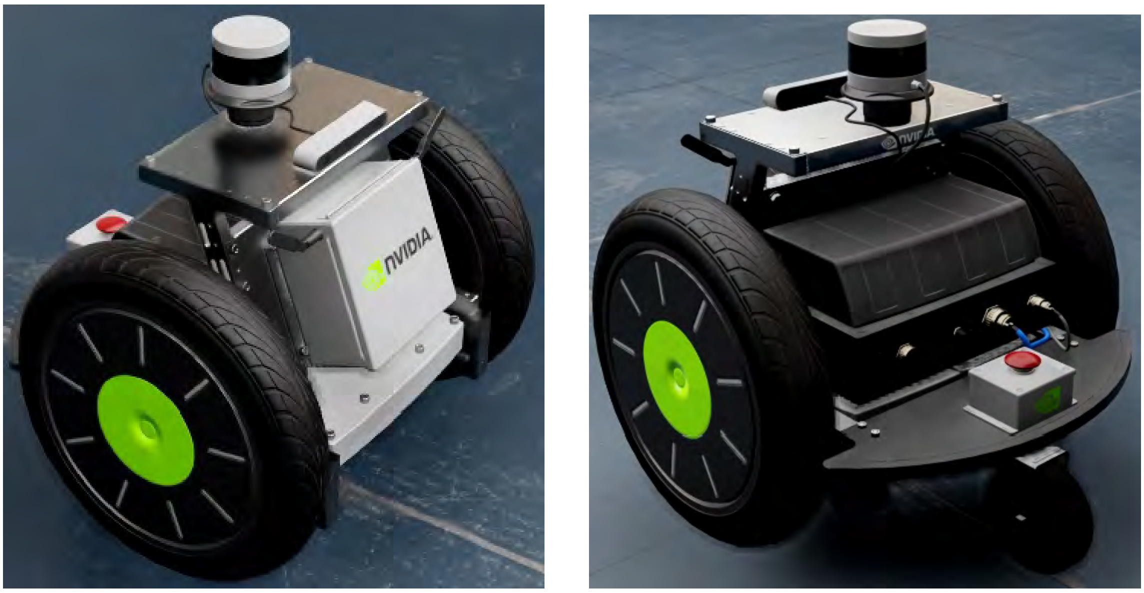

(a) Carter V1 Front (b) Carter V1 Back

Figure 6.1: Front and back side of the Nvidia Carter V1 robot

Docker containers ensured seamless deployment, while the computational capacity of the Google machine allowed smooth management of the semantic repository.

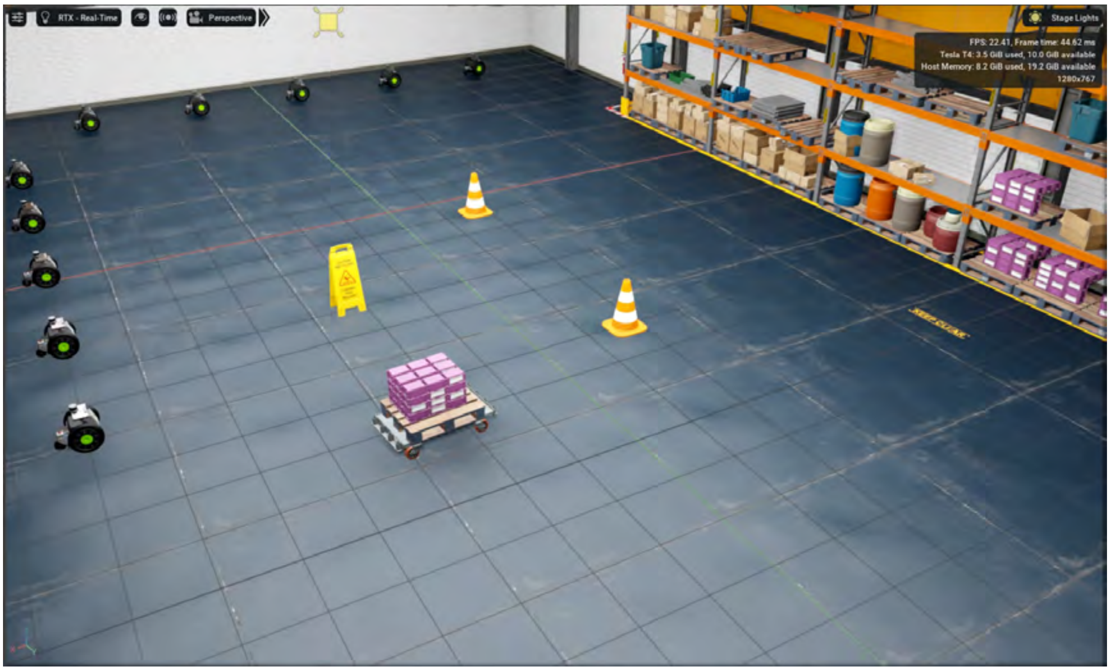
The Context Analyzer component, responsible for real-time monitoring and context interpretation, was also deployed on the Google machine. This close proximity to the simulation allowed stable subscription to ROS topics and low-latency data processing.

- **End User Computer:** Both the BPMN Modeling tool and Node-RED Workflow Manager were installed on this resource. The provision of these tools enables the user to create and execute AAS-based manufacturing workflows, as well as to define quality conditions.

With the experimental setup fully configured and the simulated environment primed, the subsequent section delves into the heart of the evaluation: a detailed case study involving the orchestration of robots within a warehouse scenario. This case study revolves around a workflow process aimed to collecting work order materials.

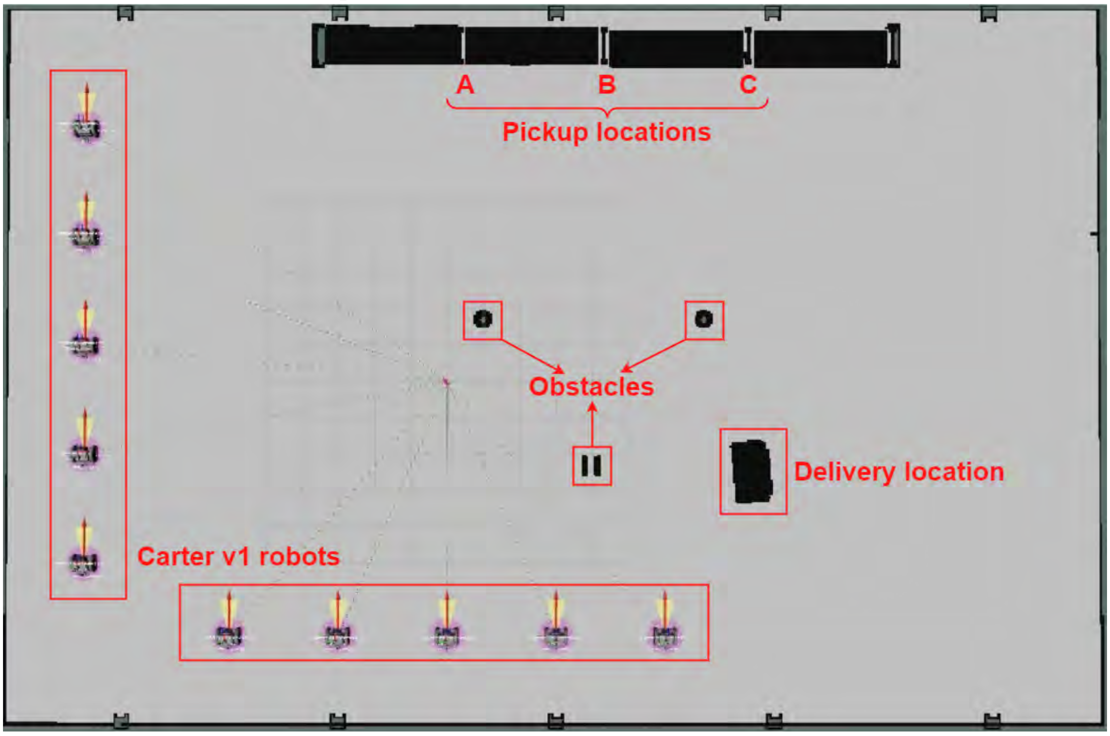
6.2.2 Case Study: Collect Work Order Resources

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. 6.3, the user is required to input the work order materials, providing a list specifying the colour 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 colours (red, yellow, and blue). The Context Analyzer component is tasked with selecting one robot from the pool of 10 to



```
FPS: 22.41, Frame time: 44.62 ms
Tesla T4: 3.5 GiB used, 10.0 GiB available
Host Memory: 8.2 GiB used, 19.2 GiB available
1280x767
```

(a) Isaac Sim: Robots at initial positions



(b) RViz: Robots at initial positions

Figure 6.2: Simulation environment

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 *PROX-*

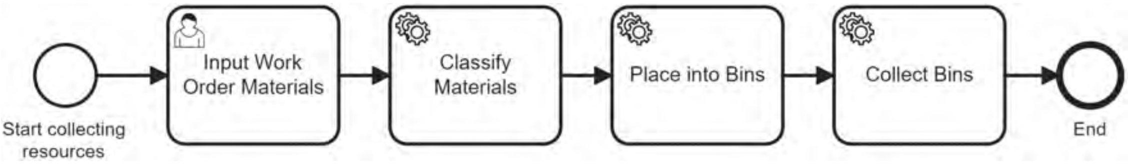


Figure 6.3: BPMN diagram for the "Collect Work Order Resources" process.

IMITY_PICKUP with values ranging from 0m to 30m, where 0m represents the robot being at the pickup location and 30m 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. The *PAYLOAD_CAPACITY* quality property can have values ranging from 0.3kg to 2.0kg, 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. 6.4. 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 <= 20.0 && BATTERY >= 25 && POSITIONAL_UNCERTAINTY <= 0.90 && PAYLOAD_CAPACITY >= 0.60

PROXIMITY_PICKUP <= 20.0 &&
BATTERY >= 25 &&
POSITIONAL_UNCERTAINTY <= 0.90 &&
PAYLOAD_CAPACITY >= 0.60

Figure 6.4: Defining quality conditions using the "AAS Web Service Discoverer" plugin for Camunda Modeler during the workflow design phase.

To provide a glimpse of the scenario and illustrate the execution of the "Collect Bins" task, Fig. 6.5 provides a step-by-step demonstration. Initially, in Fig 6.5a, the 10 robots are located at their start positions, while in Fig. 6.5b, they are placed at random positions. Subsequently, in Fig. 6.5c, 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 Fig. 6.5d, 6.5e, and 6.5f the selected robot executes the task.

To explain which robots could be capable of completing the designated "Collect Bins" task,

(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

Figure 6.5: Step-by-step case study simulation in RViz

a set of instances is presented in Table 6.1. For instance, the Carter10 robot would not be able to complete the designated task due to its Proximity of 26m (relative to the pickup location), its Battery level of 13%, its Positional Uncertainty of 2.37, and its Payload Capacity of 0.5kg. 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 5m, a Battery level of 96%, a Positional Uncertainty of 0.73, and a Payload Capacity of 1.9kg, meeting the conditions established for the completion of this task.

Table 6.1: Quality evaluation examples for the Collect Bins task.

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

6.2.3 Experiment Results

This simulation was iterated 100 times to evaluate the performance and effectiveness of the Context Analyzer, comparing scenarios with and without its support. 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 3.4.2. 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.

6.2.3.1 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. 6.6 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.

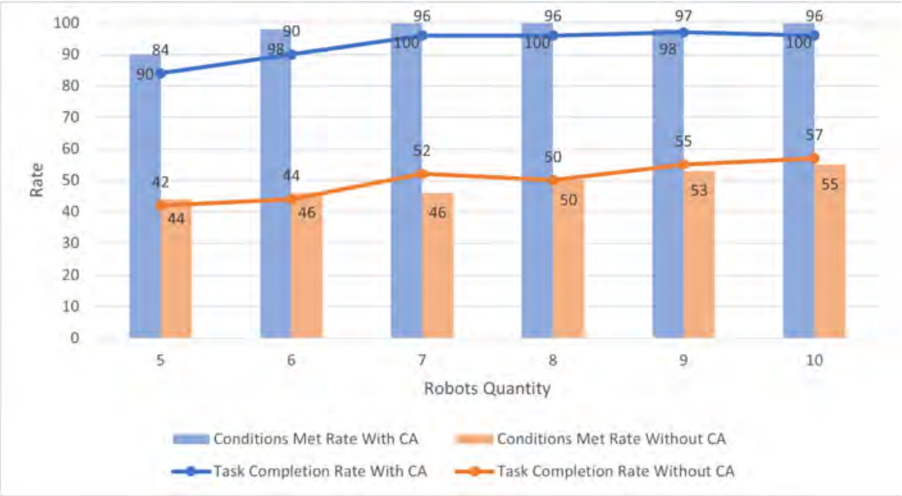


Figure 6.6: Task Completion Rate Comparison

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

6.2.3.2 Resource Utilization

This metric evaluates how efficiently the system utilizes resources, particularly in terms of battery consumption. Fig. 6.7 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. 6.7a, 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% of battery 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. 6.7b), 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 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. 6.7c depicts the median battery consumption in both cases, showing 20% of battery consumption for the robots selected With CA and 28% Without CA. Signifying a median on energy saving of 8%.

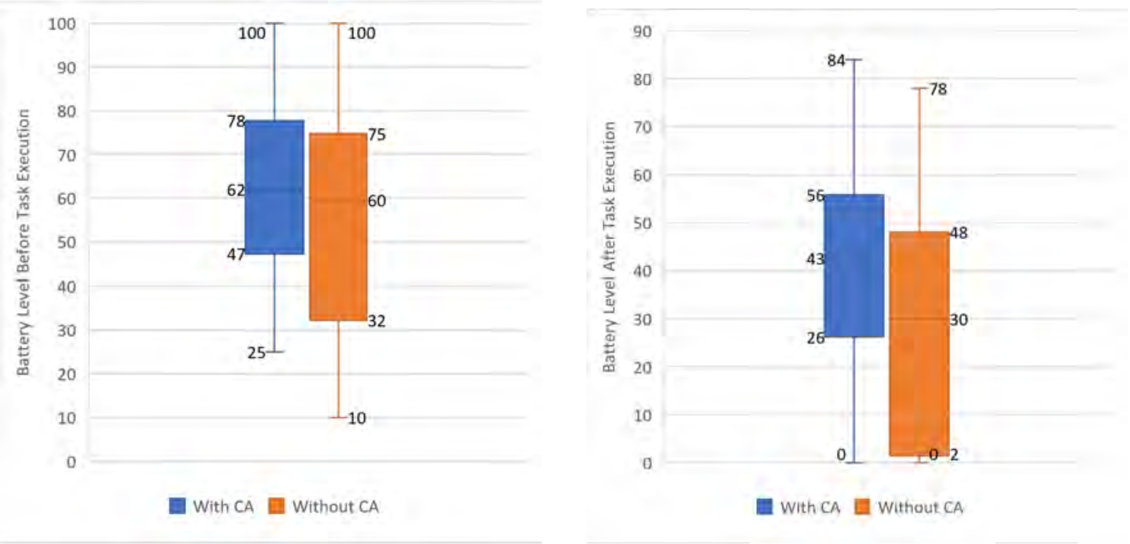
In summary, this metric demonstrates the capability of the Context Analyzer in efficiently utilizing battery power. The Context Analyzer contributes to improving the system performance by intelligently selecting well-charged and nearby robots. This way, optimizing energy consumption and opening up opportunities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

6.2.3.3 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. 6.8 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 best device selection) and Without CA (employing random device selection).

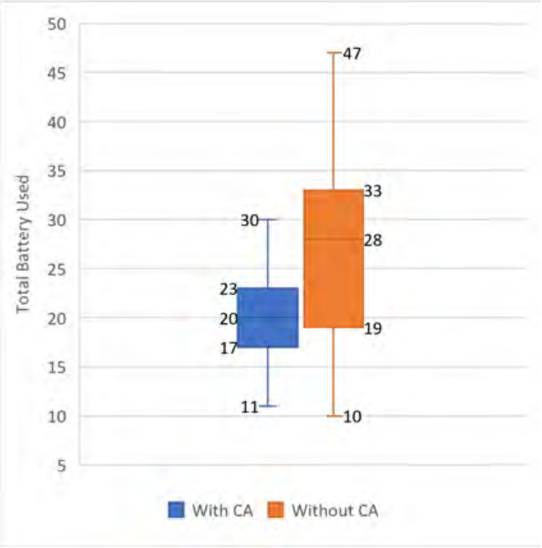
Fig. 6.8a depicts a boxplot comparison of task completion time in seconds. As seen in the figure, the devices selected by the Context Analyzer exhibit shorter task completion times. The median task completion time With CA is 199 seconds, a notable improvement compared to the median time of 285 seconds 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 With CA, representing the maximum completion time, is equivalent to the median completion time Without CA, with 298 and 285 seconds, respectively. This observation indicates a notable improvement in the worst-case scenario.

Additionally, the dispersion graph in Fig. 6.8b illustrates a reduction in task completion



(a) Battery level before task execution

(b) Battery level after task execution



(c) Total battery used

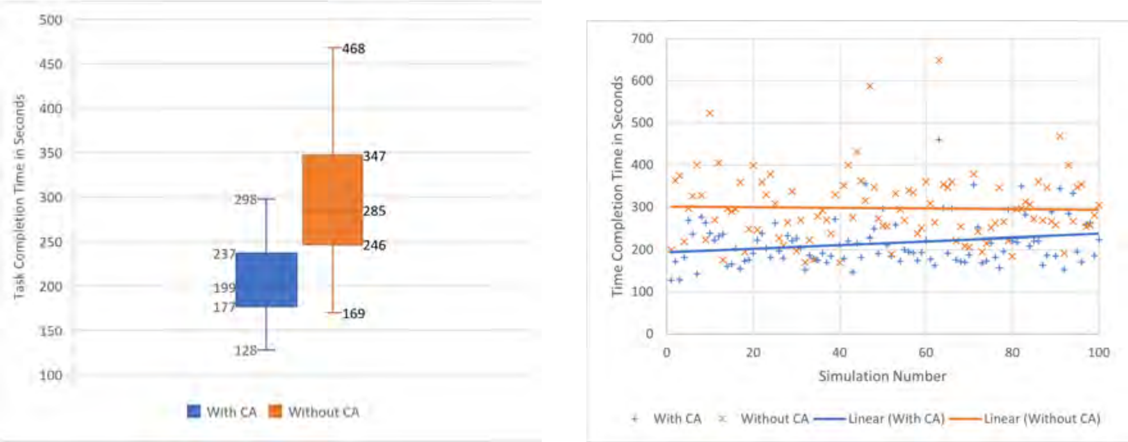
Figure 6.7: Energy Consumption Comparison

time variability when employing the Context Analyzer compared to scenarios without its support. 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 in the GitHub repository[5]. 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 the positions of the robots, quality properties of all

---

[5] `https://github.com/MUFacultyOfEngineering/ContextAnalyzer/tree/main/SimulationProofs`



(a) Task Completion Time (without outliers)

(b) Dispersion Task Completion Time

Figure 6.8: Task Completion Time Comparison

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.

6.3 Discussion and Findings

This section serves as an opportunity to discuss the results of the testing phase and explain the advantages and features of the proposed architecture. Furthermore, the discussion encompasses how the presented architecture addresses the current challenges and limitations identified in Section 3.4.2.

Answer to RQ5 (Part 1) "How does the performance of a context-aware workflow management solution that incorporates the features found in RQ4, manifest in a manufacturing scenario, and what are the key findings drawn from its evaluation, including limitations and potential improvements?"

The first part of this question encompasses how the challenges identified in Section 3.4.2 are addressed by this proposal. The challenges addressed by this proposal 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 as in [229]: 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. 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.

Furthermore, the Context Analyzer component 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.

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. **Adoption of user-centric design approaches, transitioning 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 as those mentioned in Section 5.2.

5. **Development of decoupled components and systems, providing more versatility and integration with existing 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 RQ5 (Part 2) "How does the performance of a context-aware workflow management solution that incorporates the features found in RQ4, manifest in a manufacturing scenario, and what are the key findings drawn from its evaluation, including limitations and potential improvements?"

The second part of this question encompasses identifying the limitations of the proposal and mentioning potential improvements before moving to real-world implementations.

- 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.

- During the experimental phase, it became evident that task completion could not be guaranteed across all scenarios. This limitation stems from the absence of services or robots that align with the minimal constraints. To address this, a requirement emerges for notification mechanisms in instances where tasks cannot be completed due to the unavailability of assets meeting the required quality conditions.

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 favourable evaluation results collectively contribute to its effectiveness and viability.

6.4 Conclusions

This chapter presented a comprehensive evaluation of the context-aware workflow management architecture proposed in Chapter 4. Built upon state-of-the-art advancements and addressing challenges identified in Chapter 3, this evaluation aimed to tackle the "Need for real-world implementations" challenge by implementing the architecture using a realistic warehouse scenario and ROS-based robots. To gather accurate metrics, the realistic simulation was executed 100 times while increasing the number of robots (from 5 to 10) and varying their quality values. The results obtained from this evaluation confirm hypothesis H1, which states:

H1: A context-aware workflow management architecture and components incorporating Industry 4.0 standards and semantic web for context interpretation demonstrate improved manufacturing efficiency in terms of task completion time, task completion rate, and energy utilization.

The results indicate improvements in manufacturing efficiency across the three key metrics:

- **Task completion rate**: With the context-aware capabilities enabled, the task completion rate increased, reaching 84% with 5 robots and 96% with 10 robots, compared to 42% and 57% respectively without it.

- **Task completion time:** With the context-aware capabilities enabled, the median task completion time notably reduced to 199 seconds, compared to 285 seconds respectively without it, demonstrating a remarkable 30% time reduction.

- **Energy utilization:** With the context-aware capabilities enabled, the devices selected exhibited lower energy consumption, compared to those chosen randomly, indicating a median energy saving of 8%.

The amalgamation of these results contributed to validate the hypothesis H1, demonstrating that the context-aware workflow management architecture effectively enhances manufacturing efficiency. 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.

Consequently, presenting this realistic evaluation aligns with research objective O4, "Evaluate the proposal using several case studies on diverse manufacturing scenarios", and O5 "Analyze the results and findings from the evaluation phase". Simultaneously, addressing research question RQ5, "How does the performance of a context-aware workflow management solution that incorporates the features found in RQ4, manifest in a manufacturing scenario, and what are the key findings drawn from its evaluation, including limitations and potential improvements?".

7

Final Remarks

This chapter offers an overview of the main contributions, validates the hypotheses, summarizes the limitations of the proposal, outlines the conclusions of this doctoral work, and suggests potential avenues for future research.

This chapter is organized as follows: Section 7.1 summarises the contributions, aligning them with the research objectives. Section 7.2 discusses the validation of the hypotheses. Section 7.3 highlights the scope and limitations of the proposed solution. A general conclusion of this thesis is conducted in Section 7.4, followed by short and midterm future directions proposed in Section 7.5.

7.1 Summary of the Contributions

A PhD thesis in the domain of context-aware workflow management has been presented throughout this document. Along with this thesis, the developed software, datasets, and scientific publications support its validity. Thus, the main contributions of this thesis are as follows:

1. **State-of-the-art revision:** The state-of-the-art revision followed a systematic approach for identifying current advances in the field of semantic web-based workflow management approaches with context-awareness capabilities. This literature revision is conducted in Chapter 3 and the contributions during its elaboration include:
    (a) **Review on Microservice-Based Workflow Management Solutions for Industrial Automation:** A scientific publication that surveys microservice-based workflow management approaches for Industry 4.0. The publication, accessible in [39], identifies the main challenges faced by industries when implementing workflow management in Industry 4.0. Consequently, the most noteworthy limitations impeding workflow management adoption were incorporated in this thesis, specifically in Section 3.2. These challenges encompass various facets, including dealing

with workflow modelling, heterogeneity, collaboration, parallel and asynchronous task execution, and microservice architectures. Later, these industry-related challenges aided in the identification of approaches in the literature that tackle them.

   (b) **Methodology for conducting Systematic Literature Reviews:** Due to the wide range of papers available for the domain of Industry 4.0 manufacturing workflows, we encountered the need to create a new methodology for computer science research. Consequently, the methodology was used to conduct the literature revision of this thesis and is presented in Section 3.3.1, and is publicly available in [38].

   (c) **Systematic Literature Review on Semantic Web-Based Context-Aware Workflow Management:** A scientific publication that meticulously analyzes the state-of-the-art in semantic web-based context-aware workflow management approaches, identifying advances and challenges by categorizing the approaches and comparing them. This revision is conducted in Section 3.3.2 and was also published in [39].

2. **Architecture for Context-Aware Workflow Management:** During the literature revision, several challenges were identified including (1) Consideration of a broader range of context variables, including both calculated and sensor-derived data, (2) Utilization of industry-oriented standards and standardized workflow formats, (3) Adoption of user-centric design approaches, transitioning from abstract workflows to executable workflows, and (4) Development of decoupled components and systems, providing more versatility and integration with existing systems. In response, an architecture was proposed in Chapter 4 to overcome these limitations. The architecture and its components were delivered periodically, these components are:

   (a) **AAS Enhanced Business Process Modeler:** The literature review underscored the significance of addressing the challenges posed by Industry 4.0, particularly the need to tackle the “Utilization of industry-oriented standards and standardized workflow formats”. Thus, this contribution encompasses an enhanced workflow modelling tool[1] that integrates Asset Administration Shell as the Industry 4.0 standard and BPMN as the workflow modelling standard. The tool features a service-discovery mechanism that facilitates the design of manufacturing processes by providing users with the available services from the AAS repository. Furthermore, quality conditions can also be established during the design phase to constrain and optimize the execution of a given task within the workflow by selecting the best available service or device at runtime.

   These characteristics combined underscore the significance of the enhanced workflow modelling tool while addressing the challenge of “Adoption of user-centric design approaches, transitioning from abstract workflows to executable workflows”.

---

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

Additionally, this component was first presented in a scientific publication in [34]. It is also described in this thesis in Section 4.3.

(b) **Node-RED Workflow Manager for Edge Service Orchestration:** Through the correlation analysis conducted during the literature revision emerged "Need for an edge-compliant process manager software". Thus, this contribution embodies a BPMN-based workflow manager called Node-RED Workflow Manager[2] that can operate at both edge and central environments. This software was first delivered within a scientific publication in [33]. It is also described in this thesis in Section 4.4.

(c) **Context Analyzer:** The Context Analyzer component[3] serves as the core element within this architecture by allowing service re-selection within workflows at run-time. As explained in Chapter 5, these adaptations are performed by leveraging the MAPE-K reference model for building autonomous systems and semantic web technologies for enhanced context interpretation. The design of the Context Analyzer responds to the challenges identified during the state-of-the-art revision, addressing specifically: a) "Need to interpret contextual information to perform dynamic adaptations" and b) "consideration of a broader range of context variables, including both calculated and sensor-derived data".

This component was presented during the ISM conference in Lisbon, which was held on 22-24 November 2023. The article was awarded the **"Best Service Innovation Paper Award"** due to its remarkable contribution to the service innovation field. It is currently under publication process, however, a pre-print is available in [35].

For reproducibility purposes, additional contributions include the source code for the realistic simulation, datasets, and a video showcasing the step-by-step implementation of this proposal:

3. **Realistic Simulation on a Warehouse Scenario:** A realistic warehouse scenario involving ROS-based robots was conducted to evaluate the overall performance of the architecture. The results showcase the actual benefits of this proposal in terms of task completion time, task completion rate, and resource utilization. These results were delivered in a scientific article available in [36] and are also included in this thesis in Section 6.2. The source code for the customized warehouse simulation scenario is provided in the GitHub Repository[4].

4. **Datasets:** Along with the simulation phase, datasets are also provided for reproducibility purposes. For this, data concerning robot positions, uncertainty rate, battery level, payload capacity, and proximity were collected during 100 simulations and over an

---

[2] `https://github.com/MUFacultyOfEngineering/NodeRedWorkflowManager`

[3] `https://github.com/MUFacultyOfEngineering/ContextAnalyzer`

[4] `https://github.com/MUFacultyOfEngineering/RosIsaacSimWarehouseCarterx10`

increasing number of robots (from 5 to 10). This dataset is available in the GitHub repository[5] together with a video showcasing the steps to implement and test the proposal.

7.2 Hypotheses Validation

Hypotheses regarding this thesis were stated in Section 1.5. This section aims to confirm whether the hypotheses have been proven.

H1: "A context-aware workflow management architecture and components incorporating Industry 4.0 standards and semantic web for context interpretation demonstrate improved manufacturing efficiency in terms of task completion time, task completion rate, and energy utilization."

This hypothesis is in line with the research objective "O5: Analyze the results and findings from the evaluation phase". In Chapter 6, a realistic implementation of a warehouse scenario was employed to gather and analyze insights into the performance of the proposed solution. The evaluation phase involved professional tools such as Nvidia Isaac Sim, RViz, and ROS for building a realistic simulation environment for the case study "Collect Work Order Resources". The experiment considered the dynamic nature of the manufacturing environment, where Context-Awareness plays a pivotal role in enhancing the adaptability of manufacturing operations. ROS-based robots were employed in a pick-and-deliver task, and metrics about task completion time, task completion rate, and resource utilization were measured. The experiment was conducted 100 times and iteratively by increasing the number of robots from 5 to 10 under varying quality values.

The results showed that the proposal effectively demonstrates improved manufacturing efficiency in the realistic warehouse scenario. For instance, 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 a similar vein, the median task completion time with the Context Analyzer is 199 seconds, a notable improvement compared to the median time of 285 seconds without it. Demonstrating a 30% reduction in task completion time. With respect to energy utilization, the devices selected by the Context Analyzer exhibited lower energy consumption compared to those chosen randomly. This metric indicates a median energy saving of 8%. As a result, we can conclude that the presented solution validates the first hypothesis.

---

[5] `https://github.com/MUFacultyOfEngineering/ContextAnalyzer/tree/main/SimulationProofs`

H2: "A context-aware workflow management architecture and components that leverage the Microservice Architecture philosophy can be implemented in diverse manufacturing scenarios without necessitating alterations to the individual components or changing the semantic repository structure, demonstrating the flexibility of the proposal."

This hypothesis is in line with the research objectives "O3: Propose and develop an architecture for context-aware workflow management that takes all the benefits from current approaches and addresses active challenges" and "O4: Evaluate the proposal using several case studies on diverse manufacturing scenarios". In Chapter 4, an architecture for context-aware workflow management and components is proposed to automatically analyze context data and dynamically adapt workflows during runtime. The proposal leverages advancements of current approaches and addresses the identified challenges with respect to 1) Need for real-world implementations, 2) Consideration of a broader range of context variables, including both calculated and sensor-derived data, 3) Utilization of industry-oriented standards and standardized workflow formats, 4) Adoption of user-centric design approaches, transitioning from abstract workflows to executable workflows, and 5) Development of decoupled components and systems, providing more versatility and integration with existing systems.

Evaluation of the proposed approach was conducted in line with Research Objective 4 through several prototype experiments for each component and a formal case study. This comprised 3 prototype experiments, for each independent component of the architecture, including 1) "Bridging Functions of Classifier and Arm Robots" for testing the enhanced BPMM modeling tool with the AAS Service Discovery plugin. 2) "Executing and Monitoring Workflows: Demonstration using a Classifier Robot" for testing the Node-RED Workflow Manager. 3) "Stress Testing the Context Analyzer Component with Growing Data Volumes" for testing the Context Analyzer component. In addition, a realistic simulation in a warehouse environment for the case study "Collect Work Order Resources" served as a unifying test that involved all the components of the architecture.

The results demonstrated that the architecture and its components effectively enable the analysis of context data within manufacturing workflows during runtime. This benefit is provided by the combination of Asset Administration Shell, Semantic Web technologies, and the MAPE-K model for autonomous systems. In addition, the Context Monitor module was able to effectively gather context data in real-time from devices and sensors and translate this raw data into semantically enriched data. As a result, we can conclude that the presented solution validates the second hypothesis.

In addition, this hypothesis posits that the proposed architecture can seamlessly adapt to various manufacturing scenarios without necessitating alterations to the individual components. To validate this claim, we conducted various experiments for each component of our architecture. The results consistently demonstrated the adaptability and flexibility of each of them. For instance, experiments concerning the components for achieving enhanced workflow modeling were delivered in Section 4.3, experiments with respect to workflow executors were delivered in Section 4.4, and experiments regarding the context analyzer component were delivered in Section 5.4. One last experiment that notably demonstrated that the architecture effectively accommodates all the components was delivered in Section 6.2. All these experiments together confirm the capability of our proposal to function across diverse scenarios without the need for adjustments to individual components or performing modifications to the semantic repository. This robust flexibility positions the proposed architecture as a versatile and scalable solution for the dynamic landscape of modern manufacturing.

7.3 Limitations of the Proposed Solution and Potential Improvements

While the proposed context-aware workflow management architecture presents a significant advancement in addressing industry challenges, it is essential to recognize the limitations within the scope of this research. These limitations are crucial for a comprehensive understanding of the proposal's scope and future development.

1. **Establishment of Quality Conditions:** The conditions set during the design phase of the manufacturing process are crucial 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. For instance, consider a scenario where the quality conditions for a particular task are defined based on historical data that does not fully represent the current state of manufacturing assets or environmental conditions. In such cases, the predefined conditions may lead to suboptimal task completion rates or, in extreme cases, failure to complete tasks. However, addressing the challenge of automatically identifying and suggesting appropriate quality conditions during the workflow design phase goes beyond the immediate scope of this research. This involves the development of an advanced mechanism incorporating real-time data analytics and machine learning algorithms, a subject that merits dedicated future exploration.

2. **Task-by-Task Optimization Focus:** While the proposed approach has demonstrated improved efficiency for individual tasks, it may face challenges in workflows with intricate inter-task dependencies or the need for global optimization. For example, consider a manufacturing scenario where the completion of a specific task depends on the output of multiple preceding tasks, and the overall workflow efficiency relies on optimizing the entire sequence rather than individual tasks. In such cases, the current design may not fully capture the complex interactions among a large number of assets and tasks, limiting its adaptability. The decision to focus on task-by-task optimization within this research was deliberate, acknowledging the complexity of addressing global workflow dynamics. Future iterations should explore the integration of optimization algorithms, recognizing that this represents a potential avenue for expanding the scope of the proposed solution.

3. **Safety Concerns in Dynamic Manufacturing Environments:** While the proposed architecture has shown improvements in task completion time, task completion rate, and resource utilization, real-world implementations face challenges related to safety in dynamic manufacturing environments. Consider a scenario where the architecture orchestrates tasks involving robots and humans working in close proximity or collaboratively. Ensuring the safety of human workers becomes crucial, and any oversight in task coordination may lead to safety hazards. However, ensuring safety through advanced measures and real-time monitoring involves considerations beyond the immediate scope of this research. The nature of safety protocols and adaptive mechanisms requires dedicated attention and collaboration with safety experts, representing an avenue for future research and development efforts.

4. **Reliance on Asset Administration Shell:** One limitation of the proposed architecture lies in its reliance on AAS as a fundamental component for standardized asset representation and communication. While AAS is an emerging and robust trend aiming to become a standard in future manufacturing, its widespread adoption across diverse industrial settings may still be in the early stages. The current industry landscape may not universally support AAS, potentially limiting the immediate applicability of the proposed architecture in environments where AAS adoption is not prevalent. This multifaceted challenge goes beyond the immediate scope of this research.

5. **Simulated Manufacturing Scenarios:** The experiments conducted to validate the proposed architecture were confined to use cases and a simulated but realistic manufacturing scenario. While the use cases and the simulated environment provide metrics about the performance of this proposal, the real-world implementation of the proposal was intentionally excluded from the scope of this research. Additionally, the limited number of use cases and simulated scenarios represent another limitation. Future efforts should extend the validation to more diverse simulated scenarios, addressing potential safety concerns and refining the proposal before considering real-world implementation.

7.4 Conclusions

This doctoral thesis proposed an Industry 4.0 architecture for context-aware workflow management, addressing critical challenges in the current manufacturing landscape. The solution enabled real-time adaptability of manufacturing workflows through the integration of Industry 4.0 and workflow notation standards, a microservice-oriented architecture, and context-awareness capabilities.

The systematic literature review, presented in Chapter 3, provided valuable discernment into the state-of-the-art in workflow management, emphasizing the significance of semantic web-based approaches. The discussion of the literature highlighted context-awareness as a crucial feature for enhancing manufacturing process flexibility and adaptability. A comparative analysis of existing context-aware workflow management approaches revealed several challenges, including: a) Need for real-world implementations, b) Consideration of a broader range of context variables, including both calculated and sensor-derived data, c) Utilization of industry-oriented standards and standardized workflow formats, d) Adoption of user-centric design approaches, transitioning from abstract workflows to executable workflows, and e) Development of decoupled components and systems, providing more versatility and integration with existing systems.

Inspired by all these challenges, an architecture for context-aware workflow management was proposed in Chapter 4. The architecture leverages both AAS as the Industry 4.0 standard and BPMN as the workflow notation standard, integrating them into components to enhance workflow design, execution, and context-awareness capabilities.

A notable addition to the architecture is the Context Analyzer, a component that enabled the architecture with context-awareness capabilities. As detailed in Chapter 5, this component performs adaptations to workflows during runtime by harnessing semantic web technologies alongside the MAPE-K model. Particularly, the DeviceServiceOnt ontology contributed to achieving context interpretation, and the MAPE-K model provided the framework for gathering, storing, and analyzing context data. The amalgamation of these technologies, along with the best service/device selection algorithm, contributed to enhancing workflow flexibility and adaptability.

To validate the effectiveness of this proposal, Chapter 6 conducted an evaluation using a realistic warehouse scenario and ROS-based robots. The results demonstrated that the proposal improves task completion time, task completion rate, and energy utilization. Furthermore, these positive outcomes underscore the potential impact of the proposed solution on future manufacturing operations and its role in enhancing workflow flexibility and adaptability.

7.5 Future Work

In this doctoral thesis, the objectives were accomplished, and the established hypotheses were validated. However, due to the results obtained during the experimental phase of the proposed context-aware workflow management architecture, several improvements for future work have been identified. Specifically, the limitations identified during the research process present opportunities for improvement:

- **Quality conditions automatic adjustments:** The establishment of quality conditions during the design phase emerged as a crucial factor in determining task completion. Future work involves developing an advanced mechanism incorporating real-time data analytics and machine learning algorithms to automatically identify and suggest suitable quality conditions during workflow design.
- **Safety in collaborative environments:** Manufacturing processes that require collaboration between human operators and robots can raise safety concerns. Ensuring

the safety of human workers is essential, demanding advanced artificial vision, safety protocols, and real-time monitoring. This would allow the system to instantly halt the production line in case of a danger to human safety.

- **Moving from simulated to real-world implementations:** While the proposed architecture demonstrated efficiency in simulated manufacturing scenarios, future efforts can extend validation to real-world implementations. Noting that this expansion should consider first addressing safety concerns, which will pave the way for a more comprehensive transition to real-world applications.
- **Holistic workflow optimization:** Another aspect to improve is the task-by-task optimization approach included in the Context Analyzer component. While the proposed approach demonstrated efficiency for individual tasks, it may face challenges in workflows with intricate inter-task dependencies. Future research efforts can explore the integration of optimization algorithms designed for global workflow dynamics. This strategic enhancement will enable a more holistic approach to process execution, considering dependencies and interactions across the entire workflow.

This thesis lays the groundwork for a transformative shift in manufacturing operations, emphasizing the importance of context-awareness and adaptability. The journey does not end here; continuous research and collaboration will shape the future trajectory of this architecture. By embracing the outlined future work areas, we can further solidify the relevance of the proposed solution, ensuring its continued impact on the dynamic landscape of Industry 4.0.

Bibliography

[1] ZVEI, “The reference architectural model rami 4.0 and the industrie 4.0 component,” 2015, accessed: 2023-01-04. [Online]. Available: https://www.zvei.org/en/press-media/publications/the-reference-architectural-model-industrie-40-rami-40

[2] R. Pansare, G. Yadav, and M. R. Nagare, “Reconfigurable manufacturing system: a systematic review, meta-analysis and future research directions,” *Journal of Engineering, Design and Technology*, vol. 21, no. 1, pp. 228–265, 2023.

[3] L. S. Dalenogare, G. B. Benitez, N. F. Ayala, and A. G. Frank, “The expected contribution of industry 4.0 technologies for industrial performance,” *International Journal of Production Economics*, vol. 204, pp. 383–394, 2018.

[4] J. Davis, T. Edgar, R. Graybill, P. Korambath, B. Schott, D. Swink, J. Wang, and J. Wetzel, “Smart manufacturing,” *Annual review of chemical and biomolecular engineering*, vol. 6, pp. 141–160, 2015.

[5] M. Hankel and B. Rexroth, “The reference architectural model industrie 4.0 (rami 4.0),” *ZVEI*, vol. 2, no. 2, pp. 4–9, 2015.

[6] P. I. 4.0 and ZVEI, “Reference architectural model industrie 4.0 (rami 4.0),” accessed: 2023-01-04. [Online]. Available: https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf

[7] K. Thramboulidis, D. C. Vachtsevanou, and A. Solanos, “Cyber-physical microservices: An iot-based framework for manufacturing systems,” in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2018, pp. 232–239.

[8] I. Nadareishvili, R. Mitra, M. McLarty, and M. Amundsen, *Microservice architecture: aligning principles, practices, and culture*. " O’Reilly Media, Inc.", 2016.

[9] K. He, "Integration and orchestration of heterogeneous services," in *2009 Joint Conferences on Pervasive Computing (JCPC)*. IEEE, 2009, pp. 467–470.

[10] B. Youakim, "Service-oriented workflow," *Journal of Digital Information Management*, vol. 6, no. 1, p. 119, 2008.

[11] M. Chinosi and A. Trombetta, "Bpmn: An introduction to the standard," *Computer Standards & Interfaces*, vol. 34, no. 1, pp. 124–134, 2012.

[12] OMG, "Business process model and notation (bpmn)," 2023. [Online]. Available: https://www.omg.org/spec/BPMN/2.0/PDF

[13] M. Wieland, O. Kopp, D. Nicklas, and F. Leymann, "Towards context-aware workflows," in *CAiSE07 proc. of the workshops and doctoral consortium*, vol. 2, no. S 25. Citeseer, 2007, p. 78.

[14] A. Carrera-Rivera, F. Larrinaga, and G. Lasa, "Context-awareness for the design of smart-product service systems: Literature review," *Computers in Industry*, vol. 142, p. 103730, 2022.

[15] S. Jeschke, C. Brecher, T. Meisen, D. Özdemir, and T. Eschert, *Industrial Internet of Things and Cyber Manufacturing Systems*. Cham: Springer International Publishing, 2017, ch. 3, pp. 3–19. [Online]. Available: https://doi.org/10.1007/978-3-319-42559-7_1

[16] A. I. Maarala, X. Su, and J. Riekki, "Semantic reasoning for context-aware internet of things applications," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 461–473, 2016.

[17] J. Nilsson and F. Sandin, "Semantic interoperability in industry 4.0: Survey of recent developments and outlook," in *2018 IEEE 16th international conference on industrial informatics (INDIN)*. IEEE, 2018, pp. 127–132.

[18] W. Xian, K. Yu, F. Han, L. Fang, D. He, and Q.-L. Han, "Advanced manufacturing in industry 5.0: A survey of key enabling technologies and future trends," *IEEE Transactions on Industrial Informatics*, 2023.

[19] M. Nehring, E. Topal, and P. Knights, "Dynamic short term production scheduling and machine allocation in underground mining using mathematical programming," *Mining Technology*, vol. 119, no. 4, pp. 212–220, 2010.

[20] M. Moghaddam, M. N. Cadavid, C. R. Kenley, and A. V. Deshmukh, "Reference architectures for smart manufacturing: A critical review," *Journal of Manufacturing Systems*, vol. 49, pp. 215–225, 2018.

[21] C. Richardson, *Microservices patterns: with examples in Java*. Simon and Schuster, 2018.

[22] F. Pauker, J. Mangler, S. Rinderle-Ma, and C. Pollak, "Centurio. work-modular secure manufacturing orchestration," *16th International Conference on Business Process Management 2018*, 2018.

[23] A. Giret, E. Garcia, and V. Botti, "An engineering framework for service-oriented intelligent manufacturing systems," *Computers in Industry*, vol. 81, pp. 116–127, 2016, emerging ICT concepts for smart, safe and sustainable industrial systems.

[24] D. Kozma, G. Soós, D. Ficzere, and P. Varga, "Communication challenges and solutions between heterogeneous industrial iot systems," in *2019 15th International Conference on Network and Service Management (CNSM)*. IEEE, 2019, pp. 1–6.

[25] P. I. 4.0, "Structure of the administration shell," accessed: 2023-01-04. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/hm-2018-trilaterale-coop.html

[26] W. M. Van der Aalst, "Business process management: a comprehensive survey," *International Scholarly Research Notices*, vol. 2013, 2013.

[27] M. Chinosi and A. Trombetta, "Bpmn: An introduction to the standard," *Computer Standards & Interfaces*, vol. 34, no. 1, pp. 124–134, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920548911000766

[28] E. Hozdić, "Smart factory for industry 4.0: A review," *International Journal of Modern Manufacturing Technologies*, vol. 7, no. 1, pp. 28–35, 2015.

[29] P. I. 4.0, "Details of the asset administration shell part 2," accessed: 2023-01-04. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part2_V1.html

[30] Z. D. Elektroindutrie, "The reference architectural model rami 4.0 and the industrie 4.0 component," 2018.

[31] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001, publisher: JSTOR.

[32] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, "Systematic literature reviews," in *Experimentation in software engineering*. Springer, 2012, pp. 45–54.

[33] F. Larrinaga, W. Ochoa, A. Perez, J. Cuenca, J. Legaristi, and 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.

[34] W. Ochoa, F. Larrinaga, and A. Pérez, "Architecture for managing aas-based business processes," *Procedia Computer Science*, vol. 217, pp. 217–226, 2023, 4th International Conference on Industry 4.0 and Smart Manufacturing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922022955

[35] W. Ochoa, F. Larrinaga, A. Perez, and J. Cuenca, “Enhancing flexibility in industry 4.0 workflows: A context-aware component for dynamic service orchestration,” *SSRN*, 2023.

[36] W. Ochoa, J. Legaristi, F. Larrinaga, and A. Pérez, “Dynamic context-aware workflow management architecture for efficient manufacturing: A ros-based case study,” *Future Generation Computer Systems*, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X23004831

[37] W. Ochoa, J. Cuenca, F. Larrinaga, and A. Perez, “I40go: A global ontology for industry 4.0,” *Available at SSRN 4698440*, 2024.

[38] A. Carrera-Rivera, W. Ochoa, F. Larrinaga, and G. Lasa, “How-to conduct a systematic literature review: A quick guide for computer science research,” *MethodsX*, p. 101895, 2022.

[39] W. Ochoa, F. Larrinaga, and A. Pérez, “Context-aware workflow management for smart manufacturing: A literature review of semantic web-based approaches,” *Future Generation Computer Systems*, 2023.

[40] J. G. Represa, F. Larrinaga, P. Varga, W. Ochoa, A. Perez, D. Kozma, and J. Delsing, “Investigation of microservice-based workflow management solutions for industrial automation,” *Applied Sciences*, vol. 13, no. 3, 2023.

[41] L. Wang, M. Törngren, and M. Onori, “Current status and advancement of cyber-physical systems in manufacturing,” *Journal of Manufacturing Systems*, vol. 37, pp. 517–527, 2015.

[42] A. G. Frank, L. S. Dalenogare, and N. F. Ayala, “Industry 4.0 technologies: Implementation patterns in manufacturing companies,” *International Journal of Production Economics*, vol. 210, pp. 15–26, 2019.

[43] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, “Cyber-physical systems in manufacturing,” *CIRP Annals*, vol. 65, no. 2, pp. 621–641, 2016.

[44] J. Fitzgerald, C. Gamble, P. G. Larsen, K. Pierce, and J. Woodcock, “Cyber-physical systems design: formal foundations, methods and integrated tool chains,” in *2015 IEEE/ACM 3rd FME Workshop on Formal Methods in Software Engineering*. IEEE, 2015, pp. 40–46.

[45] T. Cerny, M. J. Donahoo, and M. Trnka, “Contextual understanding of microservice architecture: current and future directions,” *ACM SIGAPP Applied Computing Review*, vol. 17, no. 4, pp. 29–45, 2018, publisher: ACM New York, NY, USA.

[46] ZVEI, “The reference architectural model rami 4.0 and the industrie 4.0 component.” [Online]. Available: https://www.zvei.org/fileadmin/user_upload/Themen/Industrie_4.0/Das_Referenzarchitekturmodell_RAMI_4.0_und_die_Industrie_4.0-Komponente/pdf/5305_Publikation_GMA_Status_Report_ZVEI_Reference_Architecture_Model.pdf

[47] P. I. 4.0, “Details of the asset administration shell part 1,” 2021, accessed: 2023-01-04. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html

[48] ——, “Functional view of the asset administration shell in an industrie 4.0 system environment,” accessed: 2023-01-04. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Functional-View.html

[49] Basyx, “Aas submodels,” accessed: 2023-01-04. [Online]. Available: https://wiki.eclipse.org/BaSyx_/_Documentation_/_AssetAdministrationShell

[50] P. I. 4.0, “The administration shell in practice (flyer),” accessed: 2023-01-04. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/admin-shell-in-practice-flyer.html

[51] ——, “Submodel templates of the asset administration shell - zvei digital nameplate for industrial equipment,” accessed: 2023-01-04. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Submodel_templates-Asset_Administration_Shell-digital_nameplate.html

[52] ——, “Submodel templates of the asset administration shell - generic frame for technical data for industrial equipment in manufacturing,” accessed: 2023-01-04. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Submodel_templates-Asset_Administration_Shell-Technical_Data.html

[53] ——, “Submodel templates of the asset administration shell - submodel for contact information,” accessed: 2023-01-04. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Specification_Submodel-Templates.html

[54] I. https://industrialdigitaltwin.org/, “Repo submodel-templates,” 2022. [Online]. Available: https://github.com/admin-shell-io/submodel-templates

[55] P. I. 4.0, “What is the asset administration shell from a technical perspective?” accessed: 2023-01-04. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/2021_What-is-the-AAS.html

[56] Industrial Digital Twin Association e. V., “Aasx package explorer,” accessed: 2021-08-17. [Online]. Available: https://github.com/admin-shell-io/aasx-package-explorer

[57] Eclipse BaSyx, “Eclipse basyx,” accessed: 2021-10-20. [Online]. Available: https://www.eclipse.org/basyx/

[58] L. für Prozessleittechnik, “Pyi40aas – python industry 4.0 asset administration shell,” accessed: 2021-02-15. [Online]. Available: https://git.rwth-aachen.de/acplt/pyi40aas

[59] S. SE, “i40-aas,” accessed: 2021-02-15. [Online]. Available: https://github.com/SAP/i40-aas

[60] N. S. of Science and Technology, “Novaas,” accessed: 2021-02-15. [Online]. Available: https://gitlab.com/gidouninova/novaas

[61] J.B. van Veelen, “D2.2 – State of the Art for Complex Workflow Generation.” [Online]. Available: https://productive40.eu/publications/

[62] D. Hollingsworth and U. Hampshire, “Workflow management coalition: The workflow reference model,” *Document Number TC00-1003*, vol. 19, no. 16, p. 224, 1995, publisher: Citeseer.

[63] Signavio, “What is workflow management?” [Online]. Available: https://www.signavio.com/post/what-workflow-management-is/

[64] P. Leitner, B. Wetzstein, F. Rosenberg, A. Michlmayr, S. Dustdar, and F. Leymann, “Runtime prediction of service level agreement violations for composite services,” in *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops: International Workshops, ICSOC/ServiceWave 2009, Stockholm, Sweden, November 23-27, 2009, Revised Selected Papers 2*. Springer, 2010, pp. 176–186.

[65] OMG, “Business Process Management (BPM),” 2022, accessed: 2022-08-15. [Online]. Available: https://www.omg.org/bpm/

[66] ——, “Business Process Model and Notation (BPMN).” [Online]. Available: https://www.omg.org/spec/BPMN/2.0.2/PDF

[67] bpmn.org, “Business Process Model and Notation.” [Online]. Available: https://www.bpmn.org/

[68] D. Jordan, J. Evdemon, A. Alves, A. Arkin, S. Askary, C. Barreto, B. Bloch, F. Curbera, M. Ford, Y. Goland, and others, “Web services business process execution language version 2.0,” *OASIS standard*, vol. 11, no. 120, p. 5, 2007.

[69] M. Kloppmann, D. Koenig, F. Leymann, G. Pfau, A. Rickayzen, C. von Riegen, P. Schmidt, and I. Trickovic, “Ws-bpel extension for people–bpel4people,” *Joint white paper, IBM and SAP*, vol. 183, p. 184, 2005, publisher: Citeseer.

[70] W. M. Van Der Aalst and A. H. Ter Hofstede, “YAWL: yet another workflow language,” *Information systems*, vol. 30, no. 4, pp. 245–275, 2005, publisher: Elsevier.

[71] T. Srivastava, A. Jain, and N. Rashid, "Market Guide for Intelligent Business Process Management Suites," 2020, accessed: 2023-01-04. [Online]. Available: https://www.gartner.com/en/documents/3993207-market-guide-for-intelligent-business-process-management

[72] R. Dunie, D. Miers, J. Wong, M. Kerremans, K. Iijima, and P. Vincent, "Gartner Magic Quadrant for Intelligent Business Process Management Suites," 2019, accessed: 2023-01-04. [Online]. Available: https://www.gartner.com/en/documents/3899484

[73] auraquantic, "La información en la gestión por procesos con ibpms," 2022, accessed: 2023-01-04. [Online]. Available: https://www.auraquantic.com/es/la-informacion-en-la-gestion-por-procesos-con-ibpms/

[74] A. Furno and E. Zimeo, "Context-aware composition of semantic web services," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 235–248, 2014.

[75] J. Aguilar, M. Jerez, and T. Rodríguez, "Cameonto: Context awareness meta ontology modeling," *Applied Computing and Informatics*, vol. 14, no. 2, pp. 202–213, 2018.

[76] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *International symposium on handheld and ubiquitous computing*. Springer, 1999, pp. 304–307.

[77] J.-y. Hong, E.-h. Suh, and S.-J. Kim, "Context-aware systems: A literature review and classification," *Expert Systems with applications*, vol. 36, no. 4, pp. 8509–8522, 2009.

[78] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of ad Hoc and ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.

[79] U. Alegre, J. C. Augusto, and T. Clark, "Engineering context-aware systems and applications: A survey," *Journal of Systems and Software*, vol. 117, pp. 55–83, 2016.

[80] P. Pradeep and S. Krishnamoorthy, "The mom of context-aware systems: A survey," *Computer Communications*, vol. 137, pp. 44–69, 2019.

[81] T. A. Nguyen, M. Aiello, T. Yonezawa, and K. Tei, "A self-healing framework for online sensor data," in *2015 IEEE International Conference on Autonomic Computing*, 07 2015, p. 1.

[82] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 414–454, 2013, publisher: Ieee.

[83] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kröller, M. Pagel, M. Hauswirth *et al.*, "Spitfire: toward a semantic web of things," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 40–48, 2011.

[84] I. Szilagyi and P. Wira, “Ontologies and semantic web for the internet of things-a survey,” in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2016, pp. 6949–6954.

[85] P. Hitzler, “A review of the semantic web field,” *Communications of the ACM*, vol. 64, no. 2, pp. 76–83, 2021.

[86] RDF Working Group, “Resource Description Framework (RDF),” 2022, accessed: 2023-01-04. [Online]. Available: https://www.w3.org/RDF/

[87] W3C, “Swrl: A semantic web rule language combining owl and ruleml,” 2022. [Online]. Available: https://www.w3.org/Submission/SWRL/

[88] P. Pauwels, S. Zhang, and Y.-C. Lee, “Semantic web technologies in aec industry: A literature overview,” *Automation in construction*, vol. 73, pp. 145–165, 2017.

[89] T. N. BV and J. van Veelen, “Productive4. 0,” *Productive4.0 Project Consortium, 2017*, 2022.

[90] E. Börger, “Approaches to modeling business processes: a critical analysis of bpmn, workflow patterns and yawl,” *Software & Systems Modeling*, vol. 11, no. 3, pp. 305–318, 2012.

[91] A. Rhayem, M. B. A. Mhiri, and F. Gargouri, “Semantic web technologies for the internet of things: Systematic literature review,” *Internet of Things*, vol. 11, p. 100206, 2020.

[92] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, “Semantics for the internet of things: early progress and back to the future,” *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 8, no. 1, pp. 1–21, 2012.

[93] M. Gheisari, H. E. Najafabadi, J. A. Alzubi, J. Gao, G. Wang, A. A. Abbasi, and A. Castiglione, “Obpp: An ontology-based framework for privacy-preserving in iot-based smart city,” *Future Generation Computer Systems*, vol. 123, pp. 1–13, 2021.

[94] W3C, “Semantic Annotations for WSDL and XML Schema — Usage Guide,” 2022, accessed: 2022-07-18. [Online]. Available: https://www.w3.org/TR/sawsdl-guide/

[95] J. G. Breslin, D. O’Sullivan, A. Passant, and L. Vasiliu, “Semantic web computing in industry,” *Computers in Industry*, vol. 61, no. 8, pp. 729–741, 2010, semantic Web Computing in Industry.

[96] P. Pauwels, S. Zhang, and Y.-C. Lee, “Semantic web technologies in aec industry: A literature overview,” *Automation in Construction*, vol. 73, pp. 145–165, 2017.

[97] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Service composition approaches in iot: A systematic review," *Journal of Network and Computer Applications*, vol. 120, pp. 61–77, 2018.

[98] P. Bazan and E. Estevez, "Industry 4.0 and business process management: state of the art and new challenges," *Business Process Management Journal*, 2021.

[99] L. M. Vaquero, F. Cuadrado, Y. Elkhatib, J. Bernal-Bernabe, S. N. Srirama, and M. F. Zhani, "Research challenges in nextgen service orchestration," *Future Generation Computer Systems*, vol. 90, pp. 20–38, 2019.

[100] S. Ivanov and 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.

[101] Y. Jiang, Z. Huang, and D. H. K. Tsang, "Challenges and solutions in fog computing orchestration," *IEEE Network*, vol. 32, no. 3, pp. 122–129, 2018.

[102] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, "Fog orchestration for internet of things services," *IEEE Internet Computing*, vol. 21, no. 2, pp. 16–24, 2017.

[103] D. Hollingsworth and U. Hampshire, "Workflow management coalition: The workflow reference model," *Document Number TC00-1003*, vol. 19, no. 16, p. 224, 1995.

[104] S. Du, P. Wu, G. Wu, C. Yao, and L. Zhang, "The collaborative system workflow management of industrial design based on hierarchical colored petri-net," *IEEE Access*, vol. 6, pp. 27 383–27 391, 2018.

[105] M. Tanaka and 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.

[106] P. Zhao, L. Di, and G. Yu, "Building asynchronous geospatial processing workflows with web services," *Computers & Geosciences*, vol. 39, pp. 34–41, 2012.

[107] C. Rotsos, D. King, A. Farshad, J. Bird, L. Fawcett, N. Georgalas, M. Gunkel, K. Shiomoto, A. Wang, A. Mauthe, N. Race, and D. Hutchison, "Network service orchestration standardization: A technology survey," *Computer Standards & Interfaces*, vol. 54, pp. 203–215, 2017, sI: Standardization SDN & NFV.

[108] L. Velasco, A. Castro, D. King, O. Gerstel, R. Casellas, and V. Lopez, "In-operation network planning," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 52–60, 2014.

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

[110] M. Meribout, A. Baobaid, M. O. Khaoua, V. K. Tiwari, and J. P. Pena, “State of art iot and edge embedded systems for real-time machine vision applications,” *IEEE Access*, vol. 10, pp. 58 287–58 301, 2022.

[111] M. Bahrami and M. Singhal, “Dccsoa: a dynamic cloud computing service-oriented architecture,” in *2015 IEEE international conference on information reuse and integration*. IEEE, 2015, pp. 158–165.

[112] S. Keele *et al.*, “Guidelines for performing systematic literature reviews in software engineering,” Technical report, Ver. 2.3 EBSE Technical Report. EBSE, Tech. Rep., 2007.

[113] D. Stefanovic, S. Havzi, D. Nikolic, D. Dakic, and T. Lolic, “Analysis of the tools to support systematic literature review in software engineering,” in *IOP Conference Series: Materials Science and Engineering*, vol. 1163. IOP Publishing, 2021, p. 012013.

[114] J. Cruz-Benito, “Systematic literature review & mapping,” -, 2016, accessed: 2021-02-15.

[115] E. Vieira and J. Gomes, “A comparison of scopus and web of science for a typical university,” *Scientometrics*, vol. 81, no. 2, pp. 587–600, 2009.

[116] V. Ramesh, R. L. Glass, and I. Vessey, “Research in computer science: an empirical study,” *Journal of systems and software*, vol. 70, no. 1-2, pp. 165–176, 2004.

[117] Y. Alotaibi, “Business process modelling challenges and solutions: a literature review,” *Journal of Intelligent Manufacturing*, vol. 27, no. 4, pp. 701–723, 2016.

[118] R. S. Aguilar-Saven, “Business process modelling: Review and framework,” *International Journal of production economics*, vol. 90, no. 2, pp. 129–149, 2004.

[119] K. Suri, W. Gaaloul, A. Cuccuru, and S. Gerard, “Semantic framework for internet of things-aware business process development,” in *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2017, pp. 214–219.

[120] M. Barz, P. Poller, M. Schneider, S. Zillner, and D. Sonntag, “Human-in-the-loop control processes in gas turbine maintenance,” in *Industrial Applications of Holonic and Multi-Agent Systems*, V. Mařík, W. Wahlster, T. Strasser, and P. Kadera, Eds. Cham: Springer International Publishing, 2017, pp. 255–268.

[121] C. Kaar, J. Frysak, C. Stary, U. Kannengiesser, and H. Müller, “Resilient ontology support facilitating multi-perspective process integration in industry 4.0,” in *Proceedings of the 10th International Conference on Subject-Oriented Business Process Management*, ser. S-BPM One ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1. [Online]. Available: https://doi.org/10.1145/3178248.3178253

[122] S. Li and J. Liu, "Research on custom-order-oriented semantic process modelling approach," in *2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*, 2018, pp. 1981–1985.

[123] A. Bucchiarone, A. Marconi, M. Pistore, and H. Raik, "A context-aware framework for dynamic composition of process fragments in the internet of services," *Journal of Internet Services and Applications*, vol. 8, no. 1, pp. 1–23, 2017.

[124] A. Marconi, M. Pistore, A. Sirbu, F. Leymann, H. Eberle, and T. Unger, "Enabling adaptation of pervasive flows: Built-in contextual adaptation," *In: Baresi, Luciano (ed.); Chi, Chi-Hung (ed.); Suzuki, Jun (ed.): Service-Oriented Computing, 7th International Joint Conference,ICSOC-ServiceWave 2009, Stockholm, Sweden, November 24-27, 2009, pp. 445-454*, 11 2009.

[125] H. Kir and N. Erdogan, "A knowledge-intensive adaptive business process management framework," *Information Systems*, vol. 95, p. 101639, 2021.

[126] L. Mazzola, P. Kapahnke, and M. Klusch, "Pattern-based semantic composition of optimal process service plans with oderu," in *Proceedings of the 19th International Conference on Information Integration and Web-Based Applications & Services*, ser. iiWAS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 492–501. [Online]. Available: https://doi.org/10.1145/3151759.3151773

[127] ——, "Semantic composition of optimal process service plans in manufacturing with oderu," *International Journal of Web Information Systems*, vol. 14, 11 2018.

[128] L. Mazzola, P. Waibel, P. Kaphanke, and M. Klusch, "Smart process optimization and adaptive execution with semantic services in cloud manufacturing," *Information*, vol. 9, no. 11, p. 279, 2018.

[129] N. Boissel-Dallier, F. Benaben, J.-P. Lorré, and H. Pingaud, "Mediation information system engineering based on hybrid service composition mechanism," *Journal of Systems and Software*, vol. 108, pp. 39–59, 2015.

[130] P. C. Calcina-Ccori, L. C. C. De Biase, C. E. L. De Oliveira, G. Fedrecheski, F. C. da Silva, and M. K. Zuffo, "Location-aware discovery of services in the iot: a swarm approach," in *2019 Global IoT Summit (GIoTS)*, 2019, pp. 1–6.

[131] A. Thuluva, D. Anicic, S. Rudolph, and M. Adikari, "Semantic node-red for rapid development of interoperable industrial iot applications," *Semantic Web*, vol. 11, pp. 949–975, 10 2020.

[132] A. Javed, S. Kubler, A. Malhi, A. Nurminen, J. Robert, and K. Främling, "biotope: Building an iot open innovation ecosystem for smart cities," *IEEE Access*, vol. 8, pp. 224 318–224 342, 2020.

[133] P. Valderas, V. Torres, and E. Serral, “Modelling and executing iot-enhanced business processes through bpmn and microservices,” *Journal of Systems and Software*, vol. 184, p. 111139, 2022.

[134] B. Jiang, Y. Qin, J. Yang, H. Li, L. Wang, and J. Wang, “Web service composition optimization with the improved fireworks algorithm,” *Mobile Information Systems*, vol. 2022, 2022.

[135] T. Baker, M. Asim, H. Tawfik, B. Aldawsari, and R. Buyya, “An energy-aware service composition algorithm for multiple cloud-based iot applications,” *Journal of Network and Computer Applications*, vol. 89, pp. 96–108, 2017.

[136] A. Bekkouche, S. M. Benslimane, M. Huchard, C. Tibermacine, F. Hadjila, and M. Merzoug, “Qos-aware optimal and automated semantic web service composition with user’s constraints,” *Service Oriented Computing and Applications*, vol. 11, no. 2, pp. 183–201, 2017.

[137] A. Abid, M. Rouached, and N. Messai, “Semantic web service composition using semantic similarity measures and formal concept analysis,” *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 6569–6597, 2020.

[138] L. Mazzola, P. Kapahnke, and M. Klusch, “Oderu: Optimisation of semantic service-based processes in manufacturing,” in *KESW17, Proc. 8th International Conference on Knowledge Engineering and Semantic Web*, 11 2017, pp. 337–346.

[139] M. Aksit and Z. Choukair, “Dynamic, adaptive and reconfigurable systems overview and prospective vision,” in *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.* IEEE, 2003, pp. 84–89.

[140] M. V. Ciasullo, G. Fenza, V. Loia, F. Orciuoli, O. Troisi, and E. Herrera-Viedma, “Business process outsourcing enhanced by fuzzy linguistic consensus model,” *Applied Soft Computing*, vol. 64, pp. 436–444, 2018.

[141] F. Herrera, E. Herrera-Viedma, and J. verdegay, “A model of consensus in group decision making under linguistic assessments,” *Fuzzy Sets and Systems*, vol. 78, no. 1, pp. 73–87, 1996.

[142] E. Herrera-Viedma, S. Alonso, F. Chiclana, and F. Herrera, “A consensus model for group decision making with incomplete fuzzy preference relations,” *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 5, pp. 863–877, 2007.

[143] C. De Maio, G. Fenza, V. Loia, F. Orciuoli, and E. Herrera-Viedma, “A framework for context-aware heterogeneous group decision making in business processes,” *Knowledge-Based Systems*, vol. 102, pp. 39–50, 2016.

[144] R. Song, J. Vanthienen, W. Cui, Y. Wang, and 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.

[145] A. Ordóñez, L. Eraso, H. Ordóñez, and L. Merchan, "Comparing drools and ontology reasoning approaches for automated monitoring in telecommunication processes," *Procedia computer science*, vol. 95, pp. 353–360, 2016.

[146] J. Zhang, B. Ahmad, D. Vera, and R. Harrison, "Automatic data representation analysis for reconfigurable systems integration," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, 2018, pp. 1033–1038.

[147] A. M. Gutiérrez Fernández, F. Van Rijswijk, C. Ruhsam, I. Krofak, K. Kogler, A. Shadrina, and G. Zucker, "Applying business architecture principles with domain-specific ontology for acm modelling: A building construction project example," in *Business Process Management Workshops*, C. Di Francescomarino, R. Dijkman, and U. Zdun, Eds. Cham: Springer International Publishing, 2019, pp. 388–399.

[148] M. Lyu, H. Benfenatki, F. Biennier, and P. Ghodous, "Control as a service architecture to support context-aware control application development," *IFAC-PapersOnLine*, vol. 52, pp. 1085–1090, 01 2019.

[149] M. Abramovici and Y. Aidi, "A knowledge-based assistant for real-time planning and execution of pss engineering change processes," *Procedia CIRP*, vol. 30, pp. 445–450, 12 2015.

[150] A. Montarnal, W. Mu, F. Benaben, J. Lamothe, M. Lauras, and N. Salatge, "Automated deduction of cross-organizational collaborative business processes," *Information Sciences*, vol. 453, pp. 30–49, 2018.

[151] M. Santambrogio, "From reconfigurable architectures to self-adaptive autonomic systems," *IEEE*, vol. 4, 01 2009.

[152] G. Alférez and V. Pelechano, "Achieving autonomic web service compositions with models at runtime," *Computers & Electrical Engineering*, vol. 63, 08 2017.

[153] A. N. Lam and Ø. Haugen, "Supporting iot semantic interoperability with autonomic computing," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2018, pp. 761–767.

[154] ——, "Applying semantics into service-oriented iot framework," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1. IEEE, 2019, pp. 206–213.

[155] U. Arul and S. Prakash, “Toward automatic web service composition based on multilevel workflow orchestration and semantic web service discovery,” *International Journal of Business Information Systems*, vol. 34, no. 1, pp. 128–156, 2020.

[156] M. Fahad, N. Moalla, and Y. Ouzrout, “Dynamic execution of a business process via web service selection and orchestration,” *Procedia Computer Science*, vol. 51, 06 2015.

[157] Brian Prasad, “What Distinguishes KBE From Automation,” 2022. [Online]. Available: https://web.archive.org/web/20120324223130/http://legacy.coe.org/newsnet/Jun05/knowledge.cfm

[158] M. Borsato, “An energy efficiency focused semantic information model for manufactured assemblies,” *Journal of Cleaner Production*, vol. 140, pp. 1626–1643, 2017.

[159] M. Elkady, A. Elkorany, and A. Allam, “Acaiot: A framework for adaptable context-aware iot applications,” *International Journal of Intelligent Engineering and Systems*, vol. 13, pp. 271–282, 08 2020.

[160] J.-L. Hippolyte, Y. Rezgui, H. Li, B. Jayan, and S. Howell, “Ontology-driven development of web services to support district energy applications,” *Automation in Construction*, vol. 86, pp. 210–225, 2018.

[161] K. Regulski, D. Wilk-Ko?odziejczyk, G. Rojek, and S. Kluska-Nawarecka, “Austempered ductile iron knowledge components management via ontological model and business processes,” *METAL 2015 - 24th International Conference on Metallurgy and Materials, Conference Proceedings*, pp. 1763–1768, 2015.

[162] D. Wilk-Kołodziejczyk, A. Kluska-Nawarecka, K. Regulski, W. Adrian, and K. Jaskowiec, “Austempered ductile iron manufacturing data acquisition process with the use of semantic techniques,” *Archives of Metallurgy and Materials*, vol. 61, 12 2016.

[163] E. Schäffer, A. Mayr, T. Huber, T. Höflinger, M. Einecke, and J. Franke, “Gradual tool-based optimization of engineering processes aiming at a knowledge-based configuration of robot-based automation solutions,” *Procedia CIRP*, vol. 81, p. 736–741, 06 2019.

[164] Rehwaldt, “An example of how to support custom elements in bpmn-js while ensuring bpmn 2.0 compatibility,” 2022, accessed: 2021-02-15. [Online]. Available: https://github.com/bpmn-io/bpmn-js-example-custom-elements

[165] N. Papulovskaya, I. Izotov, and P. Orekhov, “Implementing iot systems in service-oriented architecture,” in *2019 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*, 2019, pp. 264–267.

[166] C. Paniagua and J. Delsing, “Industrial frameworks for internet of things: A survey,” *IEEE Systems Journal*, vol. 15, no. 1, pp. 1149–1159, 2020.

[167] R. Petrasch and R. Hentschke, “Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method,” in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, 2016, pp. 1–5.

[168] Z. A. Almusaylim and N. Zaman, “A review on smart home present state and challenges: linked to context-awareness internet of things (iot),” *Wireless Networks*, vol. 25, pp. 3193–3204, 2019.

[169] N. J. van Eck and L. Waltman, *Visualizing Bibliometric Networks*. Cham: Springer International Publishing, 2014, ch. 1, pp. 285–320. [Online]. Available: https://doi.org/10.1007/978-3-319-10377-8_13

[170] H. Nguyen Ngoc, G. Lasa, and I. Iriarte, “Human-centred design in industry 4.0: case study review and opportunities for future research,” *Journal of Intelligent Manufacturing*, vol. 33, no. 1, pp. 35–76, 2022.

[171] C. O. Klingenberg, M. A. V. Borges, and J. A. V. Antunes Jr, “Industry 4.0 as a data-driven paradigm: a systematic literature review on technologies,” *Journal of Manufacturing Technology Management*, vol. 32, no. 3, pp. 570–592, 2021.

[172] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny, and K. Wenger, “Pegasus, a workflow management system for science automation,” *Future Generation Computer Systems*, vol. 46, pp. 17–35, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X14002015

[173] G. Adomavicius and A. Tuzhilin, “Context-aware recommender systems,” in *Recommender systems handbook*. Springer, 2010, pp. 217–253.

[174] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context aware computing for the internet of things: A survey,” *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 414–454, 2013.

[175] L. Malburg, M. Hoffmann, and R. Bergmann, “Applying mape-k control loops for adaptive workflow management in smart factories,” *Journal of Intelligent Information Systems*, pp. 1–29, 2023.

[176] M. Junghans and 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.

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

[178] Z. Ahmad, B. Nazir, and A. Umer, "A fault-tolerant workflow management system with quality-of-service-aware scheduling for scientific workflows in cloud computing," *International Journal of Communication Systems*, vol. 34, no. 1, p. e4649, 2021.

[179] F. U. Ambursa, R. Latip, A. Abdullah, and S. Subramaniam, "A particle swarm optimization and min–max-based workflow scheduling algorithm with qos satisfaction for service-oriented grids," *The Journal of Supercomputing*, vol. 73, pp. 2018–2051, 2017.

[180] N. Sharma, P. Garg *et al.*, "Ant colony based optimization model for qos-based task scheduling in cloud computing environment," *Measurement: Sensors*, vol. 24, p. 100531, 2022.

[181] 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.

[182] Plattform Industrie 4.0, "Information model for capabilities, skills & services," 2023. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/CapabilitiesSkillsServices.html

[183] C. E. Pereira and L. Carro, "Distributed real-time embedded systems: Recent advances, future trends and their impact on manufacturing plant control," *Annual Reviews in Control*, vol. 31, no. 1, pp. 81–92, 2007.

[184] L. Sakurada, P. Leitao, and F. De la Prieta, "Agent-based asset administration shell approach for digitizing industrial assets," *IFAC-PapersOnLine*, vol. 55, no. 2, pp. 193–198, 2022.

[185] T. Miny, M. Thies, L. Lukic, S. Käbisch, K. Oladipupo, C. Diedrich, and T. Kleinert, "Overview and comparison of asset information model standards," *IEEE Access*, 2023.

[186] W. M. van der Aalst, "Business process management: a personal view," *Business Process Management Journal*, vol. 10, no. 2, 2004.

[187] D. Garijo and Y. Gil, "A new approach for publishing workflows: abstractions, standards, and linked data," in *Proceedings of the 6th workshop on Workflows in support of large-scale science*, 2011, pp. 47–56.

[188] C. Flechsig, J. Lohmer, and R. Lasch, "Realizing the full potential of robotic process automation through a combination with bpm," in *Logistics Management: Strategies and Instruments for digitalizing and decarbonizing supply chains-Proceedings of the German Academic Association for Business Research, Halle, 2019.* Springer, 2019, pp. 104–119.

[189] J. Dobaj, J. Iber, M. Krisper, and C. Kreiner, "A microservice architecture for the industrial internet-of-things," in *Proceedings of the 23rd European Conference on Pattern Languages of Programs*, 2018, pp. 1–15.

[190] K. Aksakalli, T. Celik, and A. Can, "Deployment and communication patterns in microservice architectures: A systematic literature review," *Journal of Systems and Software*, vol. 180, p. 111014, 2021.

[191] N. Chen, N. Cardozo, and S. Clarke, "Goal-driven service composition in mobile and pervasive computing," *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 49–62, 2016.

[192] M. Oh, J. Baik, S. Kang, and H.-j. Choi, "An efficient approach for qos-aware service selection based on a tree-based algorithm," in *Seventh IEEE/ACIS International Conference on Computer and Information Science (icis 2008)*. IEEE, 2008, pp. 605–610.

[193] J. Bokrantz, A. Skoogh, T. Ylipää, and J. Stahre, "Handling of production disturbances in the manufacturing industry," *Journal of Manufacturing Technology Management*, vol. 27, no. 8, pp. 1054–1075, 2016.

[194] N. G. Shepherd and J. M. Rudd, "The influence of context on the strategic decision-making process: A review of the literature," *International journal of management reviews*, vol. 16, no. 3, pp. 340–364, 2014.

[195] Y. Lu, X. Xu, and L. Wang, "Smart manufacturing process and system automation–a critical review of the standards and envisioned scenarios," *Journal of Manufacturing Systems*, vol. 56, pp. 312–325, 2020.

[196] A. Bozzi, S. Graffione, J. Jiménez, R. Sacile, and E. Zero, "Reliability evaluation of emergent behaviour in a flexible manufacturing problem," in *2023 18th Annual System of Systems Engineering Conference (SoSe)*. IEEE, 2023, pp. 1–7.

[197] Slintel, "Top competitors of workflow automation," 2022. [Online]. Available: https://www.slintel.com/tech/workflow-automation/camunda-market-share

[198] A. Deuter and S. Imort, "Plm/alm integration with the asset administration shell," *Procedia Manufacturing*, vol. 52, pp. 234–240, 2020.

[199] Basyx, "BaSyx," 2022. [Online]. Available: https://wiki.eclipse.org/BaSyx

[200] Q. Lu, X. Li, G. Li, M. Li, J. Zhou, J. Liu, X. Shen, and F. Zhu, "A general asset administration shell platform for production line design," in *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*. IEEE, 2021, pp. 192–195.

[201] S. Schönig, L. Ackermann, S. Jablonski, and A. Ermer, "An integrated architecture for iot-aware business process execution," in *Enterprise, business-process and information systems modeling*. Springer, 2018, pp. 19–34.

[202] I. Graja, S. Kallel, N. Guermouche, and A. H. Kacem, “Bpmn4cps: A bpmn extension for modeling cyber-physical systems,” in *2016 IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2016, pp. 152–157.

[203] R. Martinho, D. Domingos, F. Martins, and C. Cândido, “Internet of things aware ws-bpel business processes context variables and expected exceptions,” *Journal of Universal Computer Science*, vol. 20, pp. 1109–1129, 08 2014.

[204] Camunda, “Camunda - overview components,” 2022. [Online]. Available: https://docs.camunda.io/docs/components/

[205] IDTA, “admin-shell-io aasx server,” 2022. [Online]. Available: https://github.com/admin-shell-io/aasx-server

[206] T. Buchholz, A. Küpper, and M. Schiffers, “Quality of context: What it is and why we need it,” in *Workshop of the HP OpenView University Association*, vol. 2003, 2003.

[207] B. Kim, J. Kim, Y. Kim, C. Min, and J. Choi, “Context-aware planning and environment-aware memory for instruction following embodied agents,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 936–10 946.

[208] E. Negri, L. Fumagalli, M. Garetti, and L. Tanca, “Requirements and languages for the semantic representation of manufacturing systems,” *Computers in Industry*, vol. 81, pp. 55–66, 2016.

[209] K. Boloor, R. Chirkova, Y. Viniotis, and 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.

[210] N. Keddis, G. Kainz, and A. Zoitl, “Capability-based planning and scheduling for adaptable manufacturing systems,” in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE, 2014, pp. 1–8.

[211] X. Wang, J. Zhu, and Y. Shen, “Network-aware qos prediction for service composition using geolocation,” *IEEE Transactions on Services Computing*, vol. 8, no. 4, pp. 630–643, 2014.

[212] Y. Zhong, Y. Fan, W. Tan, and J. Zhang, “Web service recommendation with reconstructed profile from mashup descriptions,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 468–478, 2016.

[213] D. Fang, X. Liu, I. Romdhani, P. Jamshidi, and C. Pahl, “An agility-oriented and fuzziness-embedded semantic model for collaborative cloud service search, retrieval and recommendation,” *Future Generation Computer Systems*, vol. 56, pp. 11–26, 2016.

[214] K. Hribernik, G. Cabri, F. Mandreoli, and G. Mentzas, "Autonomous, context-aware, adaptive digital twins—state of the art and roadmap," *Computers in Industry*, vol. 133, p. 103508, 2021.

[215] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender systems: an introduction.* Cambridge University Press, 2010.

[216] H. Gao, W. Huang, X. Yang, Y. Duan, and Y. Yin, "Toward service selection for workflow reconfiguration: An interface-based computing solution," *Future Generation Computer Systems*, vol. 87, pp. 298–311, 2018.

[217] B. DuCharme, *Learning SPARQL: querying and updating with SPARQL 1.1.* " O'Reilly Media, Inc.", 2013.

[218] D. L. McGuinness, F. Van Harmelen *et al.*, "Owl web ontology language overview," *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.

[219] F. Manola, E. Miller, B. McBride *et al.*, "Rdf primer," *W3C recommendation*, vol. 10, no. 1-107, p. 6, 2004.

[220] J. Pérez, M. Arenas, and C. Gutierrez, "Semantics and complexity of sparql," *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 3, pp. 1–45, 2009.

[221] E. Simperl, "Reusing ontologies on the semantic web: A feasibility study," *Data & Knowledge Engineering*, vol. 68, no. 10, pp. 905–925, 2009.

[222] M. C. Suárez-Figueroa and A. Gómez-Pérez, "Ontology requirements specification," in *Ontology Engineering in a Networked World.* Springer, 2011, pp. 93–106.

[223] P. Wang, K.-M. Chao, and C.-C. Lo, "On optimal decision for qos-aware composite service selection," *Expert Systems with Applications*, vol. 37, no. 1, pp. 440–449, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417409004898

[224] Ontotext, "Graphdb storage documentation," 2023. [Online]. Available: https://graphdb.ontotext.com/documentation/10.0/storage.html

[225] J. Groppe and S. Groppe, "Parallelizing join computations of sparql queries for large semantic web databases," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, 2011, pp. 1681–1686.

[226] M. Kikolski, "Identification of production bottlenecks with the use of plant simulation software." *Economics & Management/Ekonomia i Zarzadzanie*, vol. 8, no. 4, 2016.

[227] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[228] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: a practical introduction to the Robot Operating System.* " O'Reilly Media, Inc.", 2015.

[229] M. E. Jeyd and A. Yari, "Advanced adaptive probabilities and energy aware algorithm for scheduling tasks in mcc," in *2017 3th International Conference on Web Research (ICWR).* IEEE, 2017, pp. 157–162.

[230] Z. Maamar, M. Asim, S. Cheikhrouhou, and A. Qamar, "Orchestration- and choreography-based composition of Internet of Transactional Things," *Service Oriented Computing and Applications*, 2021.

[231] E. Evans and E. J. Evans, *Domain-driven design: tackling complexity in the heart of software.* Addison-Wesley Professional, 2004.

[232] M. von Rosing, S. White, F. Cummins, and H. de Man, "Business process model and notation-bpmn." 2015, accessed: 2022-08-15.

[233] P. Xiong, M. Zhou, C. Pu, and Y. Fan, "A petri net solution to protocol-level mismatches in service composition," *Contemporary Issues in Systems Science and Engineering*, pp. 619–643, 04 2015.

[234] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[235] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, Volume 1*, ser. Monographs in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2013. [Online]. Available: https://books.google.es/books?id=NburCAAAQBAJ

[236] K. Jensen, S. Christensen, L. M. Kristensen, and M. Westergaard, "CPN Tools," 2022, accessed: 2023-01-04. [Online]. Available: https://cpntools.org/

[237] D. Kozma, P. Varga, and F. Larrinaga, "Dynamic multilevel workflow management concept for industrial iot systems," *IEEE Transactions on Automation Science and Engineering*, 2020, publisher: IEEE.

[238] S. Keith, "What is Case Management?" 2022, accessed: 2022-08-15. [Online]. Available: http://www.adaptivecasemanagement.org/AboutACM.html

[239] OMG, "CASE MANAGEMENT MODEL AND NOTATION™ (CMMN™)," 2022, accessed: 2023-01-04. [Online]. Available: https://www.omg.org/cmmn/

[240] ——, "Desicion Model and Notation (DMN)," 2022, accessed: 2023-01-04. [Online]. Available: https://www.omg.org/dmn/

[241] R. Song, J. Vanthienen, W. Cui, Y. Wang, and L. Huang, *A DMN-Based Method for Context-Aware Business Process Modeling Towards Process Variability*. Cham: Springer International Publishing, 05 2019, ch. 1, pp. 176–188.

[242] F. Hasić, J. De Smedt, and J. Vanthienen, "Towards assessing the theoretical complexity of the decision model and notation (dmn)," in *International Working Conference on Business Process Modeling, Development and Support (BPMDS)*, 06 2017.

[243] M. Wiemuth and O. Burgert, "A workflow management system for the or based on the omg standards bpmn, cmmn, and dmn," in *Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling*, vol. 10951. International Society for Optics and Photonics, 2019, p. 1095127.

[244] A. Suchenia, K. Kluza, P. Wiśniewski, K. Jobczyk, and A. Ligeza, "Towards knowledge interoperability between the uml, dmn, bpmn and cmmn models," in *MATEC Web of Conferences*, vol. 252. EDP Sciences, 2019, p. 02011.

[245] A. Delgado and D. Calegari, "Towards integrating bpmn 2.0 with cmmn and dmn standards for flexible business process modeling." in *CIbSE*, 2019, pp. 697–704.

[246] V. Gopal, "An overview of service ontology in semantic service search," in *International Journal of Applied Engineering Research*, 2017, pp. 5905–5909.

[247] W3C, "OWL Web Ontology Language for Services (OWL-S)," 2022, accessed: 2023-01-04. [Online]. Available: https://www.w3.org/Submission/OWL-S/

[248] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean *et al.*, "Swrl: A semantic web rule language combining owl and ruleml," *W3C Member submission*, vol. 21, no. 79, pp. 1–31, 2004.

[249] D. D. Gessler, "SSWAP: A Simple Semantic Web Architecture and Protocol for semantic web services," 2022, accessed: 2023-01-04. [Online]. Available: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-10-309

[250] W3C, "Web Service Modeling Ontology (WSMO)," 2022, accessed: 2023-01-04. [Online]. Available: https://www.w3.org/Submission/WSMO/

[251] D. Fensel and C. Bussler, "The web service modeling framework wsmf," *Electronic Commerce Research and Applications*, vol. 1, no. 2, pp. 113–137, 2002.

[252] TechnoSoft Inc., "Knowledge-Based Engineering(KBE)," 2022. [Online]. Available: http://www.technosoft.com/knowledge-based-engineering/

[253] IBM, "An architectural blueprint for autonomic computing," 2005, accessed: 2023-01-04. [Online]. Available: https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf

[254] N. J. Van Eck and L. Waltman, “Software survey: VOSviewer, a computer program for bibliometric mapping,” *scientometrics*, vol. 84, no. 2, pp. 523–538, 2010, publisher: Springer.

[255] A. Ordonez, V. Alcázar, J. C. Corrales, and P. Falcarin, “Automated context aware composition of advanced telecom services for environmental early warnings,” *Expert Systems with Applications*, vol. 41, no. 13, pp. 5907–5916, 2014.

[256] A. A. Patil, S. A. Oundhakar, A. P. Sheth, and K. Verma, “Meteor-s web service annotation framework,” in *Proceedings of the 13th International Conference on World Wide Web*, ser. WWW ’04. New York, NY, USA: Association for Computing Machinery, 2004, p. 553–562. [Online]. Available: https://doi.org/10.1145/988672.988747

[257] ProcessMaker, “Orquestación de procesos vs. coreografía en microservicios,” 2022, accessed: 2021-02-15. [Online]. Available: https://www.processmaker.com/es/blog/process-orchestration-vs-choreography-microservices/

[258] S. Overflow, “Orchestration vs. choreography,” 2022, accessed: 2021-02-15. [Online]. Available: https://stackoverflow.com/questions/4127241/orchestration-vs-choreography

[259] C. K. Rudrabhatla, “Comparison of event choreography and orchestration techniques in microservice architecture,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 8, pp. 18–22, 2018.

[260] OMG, “Bpmn, cmmn and dmn specifications at omg,” 2022, accessed: 2023-01-04. [Online]. Available: https://www.omg.org/intro/TripleCrown.pdf

[261] ——, “Case management model and notation,” 2022, accessed: 2021-02-15. [Online]. Available: https://www.omg.org/spec/CMMN/1.1/PDF

[262] ——, “Decision model and notation,” 2022, accessed: 2021-02-15. [Online]. Available: https://www.omg.org/spec/DMN/1.3/PDF

[263] ProcessMaker, “Decision model and notation (dmn) tutorial,” 2022, accessed: 2021-02-15. [Online]. Available: https://www.processmaker.com/blog/decision-model-and-notation-dmn-tutorial-examples/

[264] Camunda, “Adding custom elements to the bpmn properties panel,” 2022, accessed: 2021-02-15. [Online]. Available: https://camunda.com/blog/2017/04/adding-custom-elements-to-bpmn-properties-panel/

[265] BonitaSoft, “Custom widget for bpmn visualization,” 2022, accessed: 2021-02-15. [Online]. Available: https://community.bonitasoft.com/blog/bonita-ui-designer-custom-widget-bpmn-visualization

[266] Signavio, "Add a custom attribute for a modeling element," 2022, accessed: 2021-02-15. [Online]. Available: https://documentation.signavio.com/suite/en-us/Content/process-manager/userguide/workspace-admin/custom-attributes.htm

[267] BaSyx, "Basyx / research / capability modeling ontology," 2022, accessed: 2021-02-15. [Online]. Available: https://wiki.eclipse.org/BaSyx_/_Research_/_Capability_Modeling_Ontology

[268] S. Hamdan, M. Ayyash, and S. Almajali, "Edge-computing architectures for internet of things applications: A survey," *Sensors*, vol. 20, no. 22, 2020.

[269] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.

[270] D. Knoll, J. Waldmann, and G. Reinhart, "Developing an internal logistics ontology for process mining," *Procedia CIRP*, vol. 79, pp. 427–432, 2019.

[271] M. A. Iñigo, A. Porto, B. Kremer, A. Perez, F. Larrinaga, and J. Cuenca, "Towards an asset administration shell scenario: a use case for interoperability and standardization in industry 4.0," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, 2020, pp. 1–6.

[272] Ontotext, "What is a nosql graph database," 2022, accessed: 2021-02-15. [Online]. Available: https://www.ontotext.com/knowledgehub/fundamentals/nosql-graph-database/

[273] H. Yang, S. Kumara, S. T. Bukkapatnam, and F. Tsung, "The internet of things for smart manufacturing: A review," *IISE Transactions*, vol. 51, no. 11, pp. 1190–1216, 2019.

[274] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, and M. Lefrançois, "Sosa: A lightweight ontology for sensors, observations, samples, and actuators," *Journal of Web Semantics*, vol. 56, pp. 1–10, 2019.

[275] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szmeja, and K. Wasielewska, "Semantic interoperability in the internet of things: An overview from the inter-iot perspective," *Journal of Network and Computer Applications*, vol. 81, pp. 111–124, 2017.

[276] S. Vaidya, P. Ambad, and S. Bhosle, "Industry 4.0–a glimpse," *Procedia manufacturing*, vol. 20, pp. 233–238, 2018.

[277] L. D. Xu, E. L. Xu, and L. Li, "Industry 4.0: state of the art and future trends," *International Journal of Production Research*, vol. 56, no. 8, pp. 2941–2962, 2018.

[278] J. Z. Reis and R. F. Gonçalves, "The role of internet of services (ios) on industry 4.0 through the service oriented architecture (soa)," in *IFIP International Conference on Advances in Production Management Systems*. Springer, 2018, pp. 20–26.

[279] J. A. Bigheti, M. M. Fernandes, and E. P. Godoy, “Control as a service: a microservice approach to industry 4.0,” in *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4. 0&IoT)*. Ieee, 2019, pp. 438–443.

[280] D. G. Pivoto, L. F. de Almeida, R. da Rosa Righi, J. J. Rodrigues, A. B. Lugli, and A. M. Alberti, “Cyber-physical systems architectures for industrial internet of things applications in industry 4.0: A literature review,” *Journal of Manufacturing Systems*, vol. 58, pp. 176–192, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612520302119

[281] I. C. IBM Cloud Team, “Soa vs. microservices: What’s the difference?” 2021. [Online]. Available: https://www.ibm.com/cloud/blog/soa-vs-microservices

[282] L. Baresi and M. Garriga, “Microservices: the evolution and extinction of web services?” *Microservices*, pp. 3–28, 2020.

[283] J. Delsing, *Iot automation: Arrowhead framework*. Crc Press, 2017.

[284] P. Varga, F. Blomstedt, L. L. Ferreira, J. Eliasson, M. Johansson, J. Delsing, and I. Martínez de Soria, “Making system of systems interoperable – the core components of the arrowhead framework,” *Journal of Network and Computer Applications*, vol. 81, pp. 85–95, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804516301965

[285] X. Larrucea, I. Santamaria, R. Colomo-Palacios, and C. Ebert, “Microservices,” *IEEE Software*, vol. 35, no. 3, pp. 96–100, 2018.

[286] OMG, “Business Process Management (BPM).” [Online]. Available: https://www.omg.org/technology/readingroom/BPM.htm

[287] I. E. T. Services, “Node-RED.” [Online]. Available: https://nodered.org/

[288] A. Perez, F. Larrinaga, and W. Ochoa, “node-red-contrib-arrowhead 0.2.0,” 2022. [Online]. Available: https://flows.nodered.org/node/node-red-contrib-arrowhead

[289] arrowhead, “Nodered users can now easily connect to eclipse arrowhead infrastructure,” 2022. [Online]. Available: https://www.arrowhead.eu/eclipse-arrowhead/news/nodered-users-can-now-easily-connect-to-eclipse-arrowhead-infrastructure/

[290] Node-RED, “Node-red, creating nodes,” 2022. [Online]. Available: https://nodered.org/docs/creating-nodes/

[291] M. Cengarle, S. Bensalem, J. McDermid, R. Passerone, A. Sangiovanni-Vincetelli, and M. Torngren, “Characteristics, capabilities, potential applications of cyber-physical systems: a preliminary analysis,” *Project Deliverable D*, vol. 2, 2013.

[292] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. ., 01 2005.

[293] H. Derhamy, M. Andersson, J. Eliasson, and J. Delsing, “Workflow management for edge driven manufacturing systems,” in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2018, pp. 774–779.

[294] J. Garcia Represa and J. Delsing, “Autonomous production workstation operation, reconfiguration and synchronization,” *Procedia Manufacturing*, vol. 39, pp. 226–234, 2019.

[295] D. Kozma, P. Varga, and F. Larrinaga, “Dynamic multilevel workflow management concept for industrial IoT systems,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–13, 2020. [Online]. Available: https://doi.org/10.1109/tase.2020.3004313

[296] D. Kozma, P. Varga, and K. Szabo, “Achieving flexible digital production with the arrowhead workflow choreographer,” in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, oct 2020. [Online]. Available: https://doi.org/10.1109/iecon43393.2020.9254404

[297] F. Pauker, I. Ayatollahi, and B. Kittl, “Service orchestration for flexible manufacturing systems using sequential functional charts and opc ua,” *Dubrovnik*, vol. 9, pp. 11–09, 2015.

[298] Y. Lu and F. Ju, “Smart manufacturing systems based on cyber-physical manufacturing services (cpms),” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 883–15 889, 2017.

[299] S. Cavalieri and M. G. Salafia, “A model for predictive maintenance based on asset administration shell,” *Sensors*, vol. 20, no. 21, p. 6028, 2020.

[300] G. Di Orio, P. Maló, and J. Barata, “Novaas: A reference implementation of industrie4.0 asset administration shell with best-of- breed practices from it engineering,” in *IECON’19 - IEEE 45th Annual Conference of the IEEE Industrial Electronics SocietyAt: Lisbon, Portugal*, 10 2019, pp. –.

[301] Camunda, “Camunda modeler - plugins documentation,” 2022. [Online]. Available: https://docs.camunda.io/docs/components/modeler/desktop-modeler/plugins/

[302] R. Seiger, S. Huber, P. Heisig, and U. Aßmann, “Toward a framework for self-adaptive workflows in cyber-physical systems,” *Software & Systems Modeling*, vol. 18, no. 2, pp. 1117–1134, 2019.

[303] E. Tantik and R. Anderl, “Potentials of the asset administration shell of industrie 4.0 for service-oriented business models,” *Procedia CIRP*, vol. 64, pp. 363–368, 2017.

[304] M. Achir, A. Abdelli, L. Mokdad, and J. Benothman, “Service discovery and selection in iot: A survey and a taxonomy,” *Journal of Network and Computer Applications*, vol. 200, p. 103331, 2022.

[305] S. Bader, E. Barnstedt, H. Bedenbender, B. Berres, M. Billmann, and 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)," ., 2022.

[306] K. Thramboulidis, D. C. Vachtsevanou, and A. Solanos, "Cyber-physical microservices: An iot-based framework for manufacturing systems," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, 2018, pp. 232–239.

[307] R. F. Da Silva, R. Filgueira, I. Pietri, M. Jiang, R. Sakellariou, and E. Deelman, "A characterization of workflow management systems for extreme-scale applications," *Future Generation Computer Systems*, vol. 75, pp. 228–238, 2017.

[308] G. Di Orio, P. Maló, and 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.

[309] M. Platenius-Mohr, S. Malakuti, S. Grüner, J. Schmitt, and T. Goldschmidt, "File- and api-based interoperability of digital twins by model transformation: An iiot case study using asset administration shell," *Future Generation Computer Systems*, vol. 113, pp. 94–105, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X20302600

[310] Y. Chen, P. Yu, Z. Zheng, J. Shen, and M. Guo, "Modeling feature interactions for context-aware qos prediction of iot services," *Future Generation Computer Systems*, vol. 137, pp. 173–185, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X2200245X

[311] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, *The NeOn Methodology for Ontology Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ch. -, pp. 9–34. [Online]. Available: https://doi.org/10.1007/978-3-642-24794-1_2

[312] J. Cuenca, F. Larrinaga, and E. Curry, "MODDALS methodology for designing layered ontology structures," *Applied Ontology*, vol. 15, no. 2, pp. 185–217, may 2020. [Online]. Available: http://www.edwardcurry.org/publications/MODDALS.pdf

[313] 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," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[314] P. Marcon, C. Diedrich, F. Zezulka, T. Schröder, A. Belyaev, J. Arm, T. Benesl, Z. Bradac, and 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.

[315] S. Lemaignan, A. Siadat, J.-Y. Dantan, and A. Semenenko, "Mason: A proposal for an ontology of manufacturing domain," in *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*, 2006, pp. 195–200.

[316] H. Ehm, N. Ramzy, P. Moder, C. Summerer, S. Fetz, and C. Neau, "Digital reference – a semantic web for semiconductor manufacturing and supply chains containing semiconductors," in *2019 Winter Simulation Conference (WSC)*, 2019, pp. 2409–2418.

[317] F. Wawrzik, K. A. Rafique, F. Rahman, and C. Grimm, "Ontology learning applications of knowledge base construction for microelectronic systems information," *Information*, vol. 14, no. 3, 2023. [Online]. Available: https://www.mdpi.com/2078-2489/14/3/176

[318] I. Grangel-González, L. Halilaj, G. Coskun, S. Auer, D. Collarana, and M. Hoffmeister, "Towards a semantic administrative shell for industry 4.0 components," in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, 2016, pp. 230–237.

[319] S. R. Bader, I. Grangel-Gonzalez, P. Nanjappa, M.-E. Vidal, and M. Maleshkova, "A knowledge graph for industry 4.0," in *European Semantic Web Conference*. Springer, 2020, pp. 465–480.

[320] O. Kovalenko, M. Wimmer, M. Sabou, A. Lüder, F. J. Ekaputra, and S. Biffl, "Modeling automationml: Semantic web technologies vs. model-driven engineering," in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, 2015, pp. 1–4.

[321] J. Morbach, A. Wiesner, and W. Marquardt, "Ontocape—a (re)usable ontology for computer-aided process engineering," *Computers & Chemical Engineering*, vol. 33, no. 10, pp. 1546–1556, 2009, selected Papers from the 18th European Symposium on Computer Aided Process Engineering (ESCAPE-18). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135409000362

[322] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001, publisher: JSTOR.

[323] L. Mazzola, P. Kapahnke, and M. Klusch, "Pattern-based semantic composition of optimal process service plans with oderu," in *Proceedings of the 19th International Conference on Information Integration and Web-Based Applications & Services*, ser. iiWAS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 492–501. [Online]. Available: https://doi.org/10.1145/3151759.3151773

[324] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering–a systematic literature review," *Information and software technology*, vol. 51, no. 1, pp. 7–15, 2009.

[325] S. Hachem, T. Teixeira, and V. Issarny, "Ontologies for the internet of things," in *Proceedings of the 8th middleware doctoral symposium*, 2011, pp. 1–6.

[326] J. Cuenca, F. Larrinaga, and E. Curry, "Experiences on applying spl engineering techniques to design a (re) usable ontology in the energy domain." in *SEKE*, 2019, pp. 606–777.

[327] M. Poveda-Villalón, A. Gómez-Pérez, and M. C. Suárez-Figueroa, "Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 10, no. 2, pp. 7–34, 2014.

[328] M. Ghobaei-Arani and A. Shahidinejad, "A cost-efficient iot service placement approach using whale optimization algorithm in fog computing environment," *Expert Systems with Applications*, vol. 200, p. 117012, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422004304

[329] P. Bazan and E. Estevez, "Industry 4.0 and business process management: state of the art and new challenges," *Business Process Management Journal*, vol. 28, no. 1, pp. 62–80, 2022.