Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 232 (2024) 1503–1512

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on Industry 4.0 and Smart Manufacturing

Enhancing Flexibility in Industry 4.0 Workflows: A Context-Aware Component for Dynamic Service Orchestration

William Ochoa[a,*], Felix Larrinaga[a], Alain Perez[a], Javier Cuenca[a]

[a]*Mondragon Unibertsitatea, Electronics and Computer Engineering department, Loramendi Kalea 4, Mondragon, 20500 Gipuzkoa, Spain*

Abstract

Manufacturing processes of the future will rely on standards for asset interoperability and service orchestration. The Asset Administration Shell (AAS) facilitates information exchange among Industry 4.0 assets, while standardized Business Processes enable workflow execution in manufacturing systems. Combining these technologies provides agility and scalability to manufacturing systems by incorporating asset services within business processes. Service orchestration involves coordinating multiple services, which must be dynamic during runtime to manage unforeseen situations that may arise during the manufacturing process. Context information plays a crucial role in identifying such scenarios and selecting the most suitable devices/services in response, and the Semantic Web accurately represents this information. This paper proposes a context-aware approach for service orchestration using industrial asset services. Our contributions include (1) a component for Context-Aware Service Re-Selection. (2) a domain-specific ontology (DeviceServiceOnt) for Semantic Web-based context representation. And, (3) validation of our proposal in a manufacturing setting where robots are responsible for dispatching and distributing materials within a warehouse. Opportunities for future work are also highlighted, with a primary focus on enhancing workflow dynamicity with context-aware capabilities.

Keywords: Context-Awareness; Workflow Management; Smart Manufacturing; Service Re-selection; Semantic Web

1. Introduction

Industry 4.0 has increased the need for Microservice-oriented architectures that facilitate asset interoperability and service orchestration [27]. In recent years, an emerging trend for the standardization of digital assets has garnered attention in the scientific community. The Asset Administration Shell (AAS) encourages the digital representation of I4.0 components (assets) by adhering to the principles of the Reference Architectural Model for Industry 4.0 (RAMI 4.0) [3]. AAS simplifies the interoperation and identification of assets within the network by other systems through the provision of technical and operational data. Furthermore, the orchestration of these machine services needs to adhere

* Corresponding author: William Ochoa
E-mail address: wsochoa@mondragon.edu

1877-0509 © 2024 The Authors. Published by Elsevier B.V.

10.1016/j.procs.2024.01.148

to standards [26, 11]. Service orchestration involves coordinating multiple services wrapped as a single composite service (workflow). Workflows provide agility and scalability to systems [13], and the Business Process Management (BPM) discipline includes tools and techniques for workflow management [24]. Workflows are formalized using the Business Process Modeling Notation (BPMN) language, which is the de-facto notation language for designing business processes[1].

Smart Manufacturing Systems (SMS) are adaptable systems that can reconfigure themselves and make automatic or semi-automatic modifications to manufacturing processes in response to changing situations [15]. Likewise, service orchestration must be dynamic to handle unpredictable scenarios in the manufacturing process [20]. Context information is crucial for correctly identifying these unforeseen scenarios and selecting the most appropriate devices/services in response. The Semantic Web describes context information and supports machines and applications that compose manufacturing systems. By incorporating semantic web technologies, a system can become context-aware. These systems are capable of self-configuration and exhibit cognitive behaviour when encountering new or altered situations. [22]. Context-awareness support enables dynamic adaptation of workflows during runtime, ensuring the Service Level Agreements (SLA) and increasing service availability and trustworthiness to meet customer needs [5].

This paper presents a context-aware component as part of an architecture for the management of Asset Administration Shell-based business processes. This new component aims to enhance the reactivity of manufacturing workflows by performing service re-selection of workflows during runtime. To the best of our knowledge, there is currently no approach that effectively integrates the Semantic Web for accurate context representation, standardized Business Processes for workflow execution, and the Asset Administration Shell (AAS) as a cornerstone of Industry 4.0 interoperability. This research aims to bridge this gap by presenting a novel solution that combines these three essential elements to enhance the reactivity of manufacturing workflows at runtime. Possible application fields of this proposal could range in various fields such as automotive, energy management, supply chain, and agriculture, where sensors play pivotal roles in monitoring diverse processes. Thus, this proposal should be capable of adapting workflows using the context information provided by sensors.

The structure of this paper is organized as follows: Section 2 reviews the state-of-the-art with respect to service orchestration using AAS and context recognition using semantic web technologies. In Section 3 the architecture for managing AAS-based business processes is presented focusing on the novel context-aware component. Section 4, provides an experimental phase on a resource collection scenario where the context-aware component is implemented and performance metrics are measured. Finally, Section 5 presents the concluding remarks, along with a suggestion for potential future research.

2. Related Work

A revision of the state-of-the-art on context-aware workflow management systems for Industry 4.0 is provided in this section, specifically emphasizing the dynamic reconfiguration of workflows during runtime.

Commonly, workflow reconfigurations are carried out during workflow design. Integrating context-awareness into Industry 4.0 workflow management systems effectively retrieves relevant knowledge about tasks and processes based on contextual information [22]. For instance, context knowledge can be used as a source for service recommendations [6]. Researchers have been increasingly interested in the semantic web-based service recommendation technique for workflow management systems, as services wrapped in task nodes can be dynamically changed when the quality conditions are met. This technique also enables service discovery (SD), service re-selection (RS), and workflow composition (WC) mechanisms.

To enable effective service recommendation, Semantic web-based components and systems rely on the utilization of 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 the system delivers its functionality [16]. QoS properties can be classified into two types: sensor-derived, obtained from IoT devices and sensors, and calculated parameters obtained during the execution of a process [22]. Table 1 provides a selection of significant QoS properties for each category.

[1] https://www.omg.org/spec/BPMN/2.0/About-BPMN/

Table 1. Classification of QoS properties.

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.

Several approaches in the context of dynamic service orchestration have been proposed. Table 2 summarizes relevant approaches and techniques closely related to our research. For instance, Lyu et al. [18] 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. [4] 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 [19]. 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. [9] 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 [1]. 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 [2]. They employed a variant of the hierarchical task network (HTN) planner, which encodes OWL-S processes.

Table 2. Comparison of context-aware workflow management approaches.

REF	Feature			Workflow Format		Architecture		QoS Properties	
	SD	RS	WC	BPMN	BPEL	Cloud	Edge	Sensor-Derived	Calculated-Derived
[18]	-	✓	-	-	-	✓	✓	✓	✓
[4]	-	-	✓	-	-	✓	-	-	✓
[19]	✓	✓	✓	✓	-	✓	-	-	✓
[9]	-	✓	-	✓	-	✓	-	-	✓
[1]	-	✓	✓	✓	✓	✓	-	-	✓
[2]	✓	✓	✓	-	-	✓	-	-	✓

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.

In contrast, this paper presents a novel component for context-aware workflow management that addresses these limitations with advantages. The proposed component plays a pivotal role in the architecture presented in [23], going beyond cloud-centric models and introducing a context-aware workflow management system that operates on edge devices. This cloud-to-edge paradigm enables the gathering and analysis of real-time data, including Calculated and Sensor-Derived QoS properties, thereby enhancing context awareness and responsiveness in manufacturing systems. Furthermore, the integration of the Asset Administration Shell (AAS) as the Industry 4.0 standard allows for the comprehensive description and interoperation of machines (assets) and their services. Additionally, the adoption of

BPMN as a standardized workflow format ensures compatibility and seamless integration with existing manufacturing systems.

3. Architecture for Context-Aware Workflow Management

This section presents a detailed description of our proposed architecture, with a particular focus on the innovative context-aware component, called Context Analyzer. The architecture, as illustrated in Fig. 1 and described briefly in Table 3, 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 proposed architecture has been introduced in a previous publication [23], in this work, we extended that version. The architecture and tools are licensed under the Apache-2.0 license, which allows for community use[2].

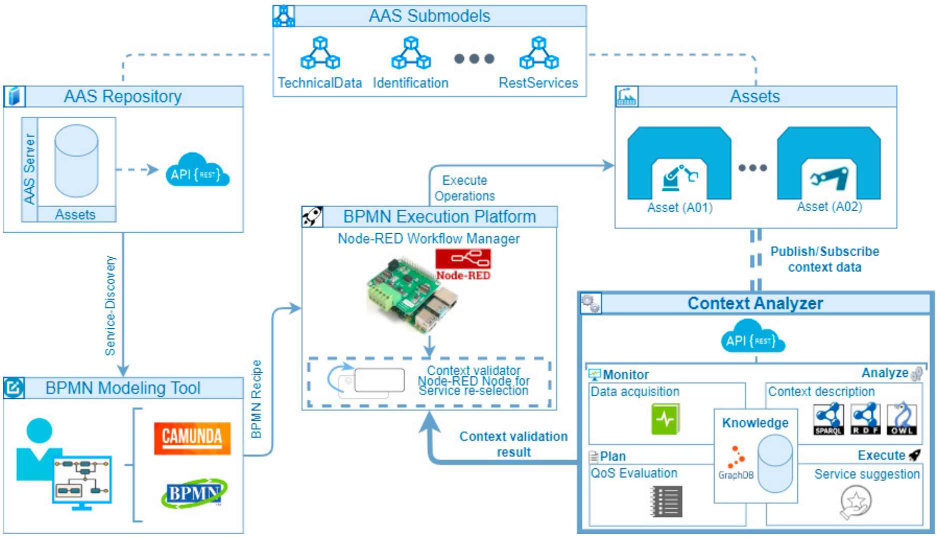

Fig. 1. Architecture for Context-Aware Workflow Management: An Asset Administration Shell-based approach. Extended from [23].

Table 3. Components of the Architecture for Context-Aware Workflow Management.

Component	Description
Assets	AAS is utilized to digitize I4.0 physical machines/devices at plant level and represent them as I4.0 digital assets.
AAS Repository	Contains the Basyx AAS Server that stores administration shell data. The data can be queried and/or maintained using the AAS Server API (Application Programming Interface).
AAS Submodels	Describes technical and operational data of assets [8]. The RestServices AAS submodel characterizes attributes of REST services, including URL, name, method, IsAsync, RequestBody, and Response.
BPMN Modeling Tool	Includes Camunda Modeler and our AAS Web Service Discoverer plugin, which enables Camunda Modeler to discover services from a chosen AAS Repository. With this tool, users can design manufacturing business processes out of asset services in BPMN format.
BPMN Execution Platform	Comprehends our Node-RED Workflow Manager (Node-RED WM) [17], a workflow management system that interprets and runs workflow recipes written in BPMN. It can be installed in embedded systems with low resource requirements.
Context Analyzer	Employs semantic web technologies for context mapping and the MAPE-K (Monitor, Analyze, Plan, Execute, and Knowledge) reference model for autonomous systems [21]. Its goal is to enhance workflow dynamism during runtime.

This architecture orchestrates asset services by first describing the technical and operational data of assets, which are represented as .aasx files. These digital descriptions are stored in the AAS Repository. The next step is to design

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

the workflow recipe using the BPMN Modeling Tool, which is an extended version of Camunda Modeler that lists the services offered by those assets. Users can include them in the workflow by performing drag and drop into the canvas. Once the workflow recipe is complete, the XML should be uploaded to a BPMN Execution Platform. This can be any workflow manager that can read BPMN. In this case, we propose Node-RED WM, an edge and lightweight workflow manager. Once a process is started, Node-RED WM queries the Context Analyzer each time a Service-Task is to be executed. Context Analyzer evaluates whether the service should be replaced or not on-the-fly by analyzing context variables.

The next subsection describes in detail the Context Analyzer component, including the semantic web technologies employed for its development and its role in enabling service replacement during workflow execution.

3.1. Context Analyzer

Context recognition is crucial for the correct reaction of manufacturing systems [14] and semantic web technologies have shown advantages in context description [22]. The main goal of the Context Analyzer is to provide service recommendations based on context and quality of service conditions. Context Analyzer is built using a combination of semantic web technologies and the MAPE-K model for autonomous systems. By utilizing semantic web technologies, the Context Analyzer is able to effectively describe administration shells, devices, sensors, and services (FPs). The MAPE-K model is employed as the architectural reference as it provides the framework for capturing context data (NFPs), analyzing and storing it in a knowledge base, composing a plan to overcome problems, and applying adaptations at runtime [21]. Fig. 2 illustrates the technology stack and sub-components included in the Context Analyzer.

Internally, the Context Analyzer builds SPARQL queries considering a service name and quality conditions in the form of evaluation expressions, which are sent to GraphDB. The resultsets are returned to the Context Analyzer, where the best service is determined using two evaluation strategies. (1) “the lower the quality value, the better” and (2) “the higher the quality value, the better”. These evaluation strategies are subject to the conditional symbol (>, >=, <, <=) provided in the evaluation expression. For instance, $HUMIDITY <= 52$ indicates less humidity is better, while $SuccessRate > 90$ means a higher success rate is better. The following subsections briefly explain these sub-components and their interactions with other components of the proposed architecture.

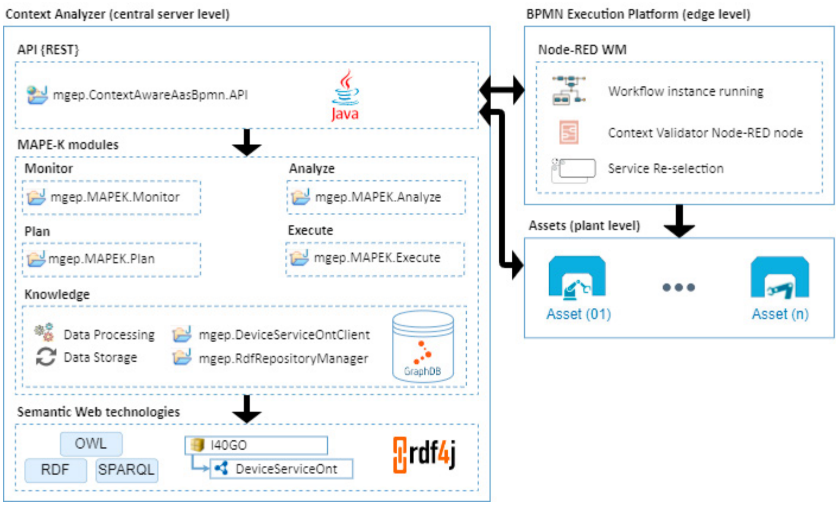

Fig. 2. Context Analyzer technology stack and interaction with other components in this architecture

3.1.1. API {REST}

The API REST sub-component acts as an interface between the MAPE-K modules of the Context Analyzer and external components within our overall architecture. By embracing the principles of Representational State Transfer (REST), our architecture achieves a flexible and standardized approach to data exchange. This flexibility not only simplifies integration efforts but also future-proofs our architecture, ensuring compatibility with emerging technologies and facilitating seamless collaboration with external systems.

3.1.2. MAPE-K modules

Includes modules that gather data, analyze it, and deliver a recommendation object for service replacement. These modules make use of semantic web technologies such as (1) RDF (Resource Description Framework)—to represent data, (2) OWL (Web Ontology Language)—for creating ontologies, and (3) SPARQL (Simple Protocol and RDF Query Language)—to query data [7].

- Monitor: This module constantly gathers real-time data from plant-level. The data is stored in a GraphDB repository, considering the schemas of the DeviceServiceOnt ontology, which describes administration shells, devices, services, and quality parameters.
- Analyze: Details about administration shells, assets, services, and quality conditions are received for the computation phase. SPARQL queries are built dynamically considering the DeviceServiceOnt ontology and the received quality conditions. The queries are then executed in GraphDB.
- Plan: The resultsets returned by GraphDB are used to prepare a final response, a JSON object representing a recommendation for service replacement.
- Execute: The recommendation object is delivered to the API consumer.
- 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 established semantic web standards, such as RDF and OWL, to provide a formal and expressive representation.

3.1.3. Node-RED WM

Node-RED WM is a unique edge-embedded workflow manager designed specifically for this architecture. Leveraging the power of Node-RED, it executes BPMN-written workflow recipes directly at the edge. By running on edge devices, Node-RED WM brings the benefits of localized workflow management, reducing the need for data transmission to centralized servers and enabling faster decision-making. The incorporation of a new Node-RED node - Context Validator further enhances its capabilities, enabling dynamic service replacement based on real-time information from the Context Analyzer. This edge-embedded workflow management approach empowers our architecture with agility, efficiency, and adaptability, making it well-suited for edge-computing environments where low latency and distributed processing are crucial.

3.2. Semantic web integration

DeviceServiceOnt is an application-specific ontology that includes classes giving rich and meaningful descriptions of entities within our architecture, such as administration shells, devices, services, inputs, outputs, and quality parameters. By utilizing the DeviceServiceOnt ontology, the Knowledge module ensures a consistent and structured representation of the data, enabling efficient querying and inference capabilities. Fig. 3 provides an overview of the OWL representation of the DeviceServiceOnt ontology. The ontology design incorporates relevant classes and properties from I40GO[3], a global ontology for Industry 4.0 applications. This reuse ensures interoperability and alignment with established standards, facilitating integration with existing systems and enabling semantic interoperability across domains.

4. Context Analyzer Evaluation

The effectiveness and efficiency of our Context Analyzer for service re-selection are assessed in this section. The evaluation considers the QoS properties of both Sensor-Derived and Calculated-Derived (identified in Section 2). By incorporating these QoS properties into the evaluation, we gather valuable metrics regarding the adaptability and performance of the service re-selection process under varying quality levels.

To conduct the experiments, we utilized a 2.11-GHz Intel Core i5-10210U PC with 16 GB of memory running Windows 10. The AAS Server and GraphDB were installed and configured in a Docker container, while the BPMN

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

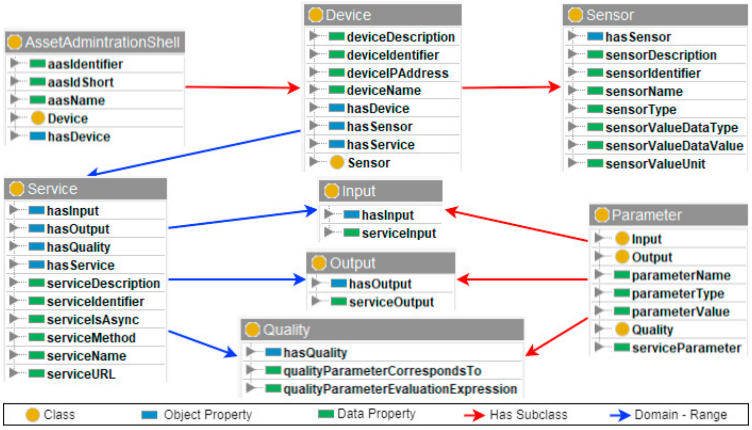

Fig. 3. DeviceServiceOnt ontology classes and relations

Modeling tool was installed on the computer. For these tests, we generated a variety of data using the scripts Context Analyzer offers, this way simulating multiple robots with different quality levels.

4.1. Simulated Test case scenario: Collect work order resources

This test is conducted in a controlled environment, using generated data for the purpose of testing and analysis. A BPMN diagram is designed for a representative manufacturing scenario involving 300 robots. The distribution of these robots is as follows: 50 robots for materials classification, 50 robots for packing materials, and 200 robots for package delivery. As depicted in Fig. 4, the user is required to input the work order materials, providing a list specifying the color and quantity of materials to be dispatched. A conveyor system facilitates the transportation of the materials. For instance, red-colored pieces are conveyed and dispatched into a container on the right side, while pieces of other colors undergo a similar process and are placed into a separate container on the left side. After dispatching the required number of pieces, a second robot undertakes the packing task. It selects a container, places the materials inside, measures the total weight, and applies a label to the package. Upon completion of the packing task, a quality inspection is performed by the user. If the user marks the package as accepted, a third robot is then assigned to pick up the package and deliver it to the designated warehouse, marking the end of the process.

Fig. 4. Collect Work Order Resources - BPMN diagram. This is a simplified version, the full version can be found here.

In this approach, the quality conditions are defined during the design phase. For this experiment, we established the quality conditions for the Delivery task as follows: $BATTERY >= 20$ && $PROXIMITY <= 200$ && $EnergyConsumption < 3$ && $PayloadCapacity > 2$ && $Latency < 100$. Thus 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 (battery percentage consumed per meter traveled), random PayloadCapacity values ranging from 1kg to 10kg (the weight the robot can carry), and random Latency values ranging from 10ms to 500ms (representing 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 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 the completion of this task. In contrast, the AAS128 robot is well-equipped for the task, with a Proximity value of 15m, a Battery level of 92%, a PayloadCapacity of 3kg, meeting the conditions established for the completion of this task.

Table 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

After deploying the BPMN recipe in Node-RED WM, the manufacturing process can be initiated. During execution, Node-RED WM interacts with the Context Analyzer component to validate context and receive recommendations for service-task replacements. Fig. 5 compares the task completion rate between two scenarios: Without Context Analyzer (Random selection) and With Context Analyzer (Best selection). In this experiment, we conducted 9 epochs to analyze the impact of the number of robots employed on the task completion rate. The first epoch utilized 5 robots, and subsequent epochs gradually increased the number of robots employed until epoch 9, which involved the use of 200 robots. To ensure comprehensive analysis, the data was deleted and reinserted during each epoch. Additionally, for each epoch, the best service selection was executed 100 times, and task completion was averaged.

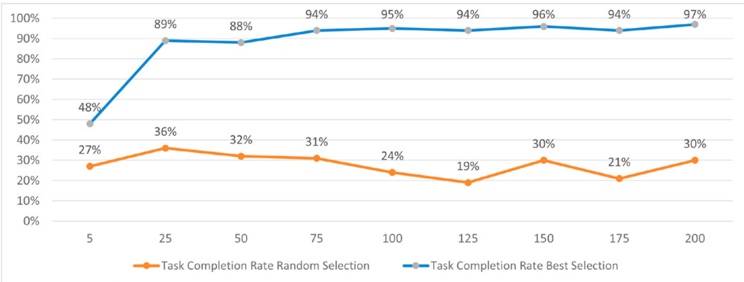

Fig. 5. Task Completion Rate

This test demonstrates the effectiveness of the Context Analyzer in service re-selection while ensuring task completion. However, it is important to note that while the Context Analyzer plays a crucial role in service re-selection and improving task completion rates, it cannot solely guarantee task completion in all scenarios. Task completion is ultimately influenced by the conditions set during the design phase of the manufacturing process. Those conditions should align with the specific goals for the completion of the task.

4.2. Stress test

The second test measures the response time of the Context Analyzer API using stress tests, as in [10]. 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. 6 shows the average response time versus the amount of data in the semantic repository during each epoch. This stress test demonstrates that (1) When the amount of data in the semantic repository is short (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 big (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 [25]:

- Indexing: GraphDB uses various indexing techniques to speed up queries.

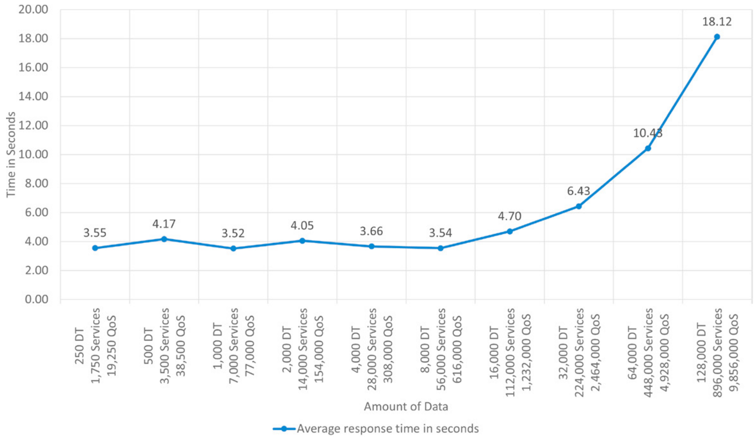

Fig. 6. Stress test - response time

- 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.
- 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 [12].

5. Conclusions

This paper presented a component for context-aware service re-selection as part of a workflow management architecture for cloud-to-edge environments. The goal of this approach is to improve the flexibility of workflows during execution by utilizing semantic web technologies for context mapping and the MAPE-K model as an architectural framework. The source code of all components discussed in this paper is available for community use.

An experimental phase was conducted to test the effectiveness of the Context Analyzer in the architecture for manufacturing workflow management. The results showed that the component is capable of recognizing context and making recommendations for service re-selection based on quality of service conditions. The service re-selection mechanism was achieved through the interaction between the Node-RED WM and Context Analyzer components. The use of semantic web technologies and the MAPE-K model was crucial in this proposal due to their advantage of interpreting context, which enabled successful service replacement during workflow runtime, thus adding context-awareness capabilities to the architecture. Overall, this solution improves the flexibility of workflows in the orchestration of manufacturing services.

For future work, the Context Analyzer component will be enhanced by integrating a notification mechanism for cases where tasks cannot be completed due to a lack of robots meeting the required quality conditions. Additionally, an automatic mechanism to identify and suggest appropriate quality conditions during design time will improve the practical applicability of this proposal. Finally, real-world industrial tests will be conducted to showcase the effectiveness of this proposal. These tests will use simulation software and ROS (Robotic Operating System) robots.

Acknowledgements

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

References

- [1] Alférez, G., Pelechano, V., 2017. Achieving autonomic web service compositions with models at runtime. Computers & Electrical Engineering 63. doi:10.1016/j.compeleceng.2017.08.004.
- [2] Arul, U., Prakash, S., 2020. Toward automatic web service composition based on multilevel workflow orchestration and semantic web service discovery. International Journal of Business Information Systems 34, 128–156.
- [3] Bader, S., Barnstedt, E., Bedenbender, H., Berres, B., Billmann, M., Ristin, M., 2022. 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) .
- [4] Bekkouche, A., Benslimane, S.M., Huchard, M., Tibermacine, C., Hadjila, F., Merzoug, M., 2017. Qos-aware optimal and automated semantic web service composition with user's constraints. Service Oriented Computing and Applications 11, 183–201.
- [5] Boloor, K., Chirkova, R., Viniotis, Y., Salo, T., 2010. 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. pp. 464–472.
- [6] Carrera-Rivera, A., Larrinaga, F., Lasa, G., 2022. Context-awareness for the design of smart-product service systems: Literature review. Computers in Industry 142, 103730.
- [7] DuCharme, B., 2013. Learning SPARQL: querying and updating with SPARQL 1.1. " O'Reilly Media, Inc.".
- [8] Elektroindutrie, Z.D., 2018. The reference architectural model rami 4.0 and the industrie 4.0 component.
- [9] Fahad, M., Moalla, N., Ouzrout, Y., 2015. Dynamic execution of a business process via web service selection and orchestration. Procedia Computer Science 51. doi:10.1016/j.procs.2015.05.299.
- [10] Gao, H., Huang, W., Yang, X., Duan, Y., Yin, Y., 2018. Toward service selection for workflow reconfiguration: An interface-based computing solution. Future Generation Computer Systems 87, 298–311.
- [11] Giret, A., Garcia, E., Botti, V., 2016. An engineering framework for service-oriented intelligent manufacturing systems. Computers in Industry 81, 116–127. doi:https://doi.org/10.1016/j.compind.2016.02.002. emerging ICT concepts for smart, safe and sustainable industrial systems.
- [12] Groppe, J., Groppe, S., 2011. Parallelizing join computations of sparql queries for large semantic web databases, in: Proceedings of the 2011 ACM Symposium on Applied Computing, pp. 1681–1686.
- [13] Hollingsworth, D., Hampshire, U., 1995. Workflow management coalition: The workflow reference model. Document Number TC00-1003 19, 224. Publisher: Citeseer.
- [14] Hribernik, K., Cabri, G., Mandreoli, F., Mentzas, G., 2021. Autonomous, context-aware, adaptive digital twins—state of the art and roadmap. Computers in Industry 133, 103508.
- [15] Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., Eschert, T., 2017. Industrial Internet of Things and Cyber Manufacturing Systems. Springer International Publishing, Cham. chapter 3. pp. 3–19. doi:10.1007/978-3-319-42559-7_1.
- [16] Junghans, M., Agarwal, S., 2010. Web service discovery based on unified view on functional and non-functional properties, in: 2010 IEEE Fourth International Conference on Semantic Computing, IEEE. pp. 224–227.
- [17] Larrinaga, F., Ochoa, W., Perez, A., Cuenca, J., Legaristi, J., Illarramendi, M., 2022. Node-red workflow manager for edge service orchestration, in: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, IEEE. pp. 1–6.
- [18] Lyu, M., Benfenatki, H., Biennier, F., Ghodous, P., 2019. Control as a service architecture to support context-aware control application development. IFAC-PapersOnLine 52, 1085–1090. doi:10.1016/j.ifacol.2019.11.340.
- [19] Mazzola, L., Waibel, P., Kaphanke, P., Klusch, M., 2018. Smart process optimization and adaptive execution with semantic services in cloud manufacturing. Information 9, 279.
- [20] Moghaddam, M., Cadavid, M.N., Kenley, C.R., Deshmukh, A.V., 2018. Reference architectures for smart manufacturing: A critical review. Journal of Manufacturing Systems 49, 215–225. doi:https://doi.org/10.1016/j.jmsy.2018.10.006.
- [21] Nguyen, T.A., Aiello, M., Yonezawa, T., Tei, K., 2015. A self-healing framework for online sensor data, in: 2015 IEEE International Conference on Autonomic Computing, p. 1. doi:10.1109/ICAC.2015.61.
- [22] Ochoa, W., Larrinaga, F., Pérez, A., 2023a. Context-aware workflow management for smart manufacturing: A literature review of semantic web-based approaches. Future Generation Computer Systems .
- [23] Ochoa, W., Larrinaga, F., Pérez, A., 2023b. Architecture for managing aas-based business processes. Procedia Computer Science 217, 217–226. doi:https://doi.org/10.1016/j.procs.2022.12.217. 4th International Conference on Industry 4.0 and Smart Manufacturing.
- [24] OMG, 2022. Business Process Management (BPM). URL: https://www.omg.org/bpm/.
- [25] Ontotext, 2023. Graphdb storage documentation. URL: https://graphdb.ontotext.com/documentation/10.0/storage.html.
- [26] Pauker, F., Mangler, J., Rinderle-Ma, S., Pollak, C., 2018. Centurio. work-modular secure manufacturing orchestration. 16th International Conference on Business Process Management 2018 .
- [27] Thramboulidis, K., Vachtsevanou, D.C., Solanos, A., 2018. Cyber-physical microservices: An iot-based framework for manufacturing systems, in: 2018 IEEE Industrial Cyber-Physical Systems (ICPS), pp. 232–239. doi:10.1109/ICPHYS.2018.8387665.