Semantic Micro-Front-End Approach to Enterprise Knowledge Graph Applications Development

Milorad Tosic[1,2][a], Nenad Petrovic[1][b] and Olivera Tosic[2]
[1]*Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, Nis, Serbia*
[2]*Virtuona d.o.o, Nis, Serbia*

Keywords: IoT, Low-Code, Ontologies, Semantic Micro-Front-Ends, Mapping, Enterprise Knowledge Graph.

Abstract: Industry 4.0 has been mainly driven by IoT devices and artificial intelligence developments rising heterogeneity of the data acquired by sensing devices as well as data from existing legacy systems (such as ERP) the crucial for digital transformation. Until recently, migration of enterprise applications to Cloud has been considered the only viable long-term solution. However, after hidden infrastructure costs of the Cloud-only based approach have been discovered, a number of businesses have begun considering hybrid Cloud-Edge architectures where Micro-Services Architectures (MSA) on backend are complemented with Micro-Font-End (MFE) applications. However, the architecture must be very carefully optimized in order to avoid high risks and costs due to increased system's complexity. In this paper, a semantic-driven approach based on Enterprise Knowledge Graph (EKG) and ontologies with their automated mapping is introduced in order to manage the complexity. Ontologies are adopted for automated, low-code approach to composition and deployment of MFE components targeting enterprise productivity applications. MFE applications generated this way are built upon Semantic Micro Services backend that can transparently be distributed between Cloud and Edge. Our approach is illustrated on the case study for semantic annotation of manufacturing area which utilizes a shared marketplace component for IoT-based indoor positioning.

1 INTRODUCTION

During the past decade, Internet of Things (IoT) and artificial intelligence (AI) have emerged becoming key enablers of novel usage scenarios and innovation across various application domains. Industry 4.0 is certainly one of them aiming to accelerate and make more efficient manufacturing as well as other industry-related activities (Babu et al., 2023). While IoT assumes acquisition of huge data amounts thanks to incorporation of large number of smart wearable and sensing devices, artificial intelligence strives to analyse the acquired data in order to extract useful knowledge. The extracted knowledge can be further leveraged for reasoning and decisions which aim to improve not only the manufacturing process but high-level enterprise effectiveness as well. However, development of such applications and services faces several challenges (Petrovic et al., 2019). First, there are many different types of devices involved, each of them possibly using different data formats. Then, in Industry 4.0 applications, it is traditionally needed to integrate these data streams with the data originating from existing industrial platforms, including legacy systems and ERPs (Alsaadi, 2022). Finally, deployment of such services often constraints some parts to be strictly executed on Edge servers located within the enterprise premises (such as collecting/sensing potentially sensitive data), while, the acquired data is often offloaded to Cloud for heavy processing tasks, such as artificial intelligence empowered predictive analytics (Petrovic et al., 2019). In order to overcome such bottlenecks, artificial intelligence techniques have been identified as one of the crucial methods (Romero et al., 2023).

This paper aims to bridge the Cloud-Edge gap with the goal to make development of state-of-the-art industry-oriented applications more effective. The proposed approach makes use of ontologies and semantic technology in synergy with Micro-Font-End

[a] https://orcid.org/0000-0001-8142-5788
[b] https://orcid.org/0000-0003-2264-7369

Tosic, M., Petrovic, N. and Tosic, O.
Semantic Micro-Front-End Approach to Enterprise Knowledge Graph Applications Development.
DOI: 10.5220/0012236200003584
In *Proceedings of the 19th International Conference on Web Information Systems and Technologies (WEBIST 2023)*, pages 488-495
ISBN: 978-989-758-672-9; ISSN: 2184-3252

(MFE) components in order to cope with the target systems' complexity. Semantic technology is adopted for integration of various data sources together with domain conceptualization, application component composition, configuration and customization. In the same time, MFE components are leveraged to make deployment of the no-code platform more flexible and capable of adopting Cloud as well as Edge services. The novelty of this combination is the high degree of automation of software development and deployment steps, making it usable even by professionals without IT background.

The proposed approach is implemented in the ASMOSA (Agile Semantic Model driven development for Smart Applications ecosystem in manufacturing) software platform, which adopts semantic approach to information systems design. ASMOSA enables users to create a no-code/low-code applications tailored to the specific needs of individual enterprise operations in various business domains. In our case study, we make use of WiPos, one of existing SHOP4CF components (Zimniewicz, 2020) aiming indoor localization for implementation of manufacturing area monitoring application.

2 BACKGROUND AND RELATED WORKS

2.1 Ontologies and Semantic Technology

As domain conceptualization, ontologies have been widely used in practice for knowledge representation across many domains – biology, healthcare, public administration, Internet of Things, computing infrastructure management to business and enterprise information systems. In these cases, ontologies make various usage scenarios possible, including interoperability of heterogeneous devices, data integration, reasoning against large amount of semantically represented information to discover hidden knowledge and code generation based on specific parameters (Petrovic, Tosic, 2020).

In this paper, ontologies are leveraged to tackle various aspects that make development of MFE applications development more convenient (semantic repository annotation, application model), while we also make use of them in order to create unified semantic knowledge based using data from various sources (ERP, IoT devices). Furthermore, our work makes use of ontology alignment, also known as ontology matching. In this work, a set of proprietary ontologies (modelling production, planning, and transportation) are used for alignment with domain-specific ontologies. This way, knowledge sharing and reuse is facilitated.

2.2 MSA and MFE Applications Development

Micro services-based architecture (MSA) represents paradigm where application is developed as a collection of services, which can be developed, deployed and maintained independently (Pontarolli et al., 2023). MFEs build upon the underlying concepts of MSA, while their main idea is to think about a web application as a composition of features owned by independent teams (Taibi et al., 2022, Geers, 2023). Apps created using MFE components should be independent, self-contained and do not share state or global variables.

In this work, MFE-based approach is leveraged as enabler for semantic-driven composition of micro services. This way, execution of different components is made possible either in Cloud or on Edge, considering the isolation and independence of runtimes for distinct MFEs. This way, deployment flexibility is improved making possible industry-oriented scenarios where sensor data is collected within the Edge, but sent to Cloud only for demanding processing tasks.

2.3 Shared Component Marketplaces

The approach proposed in this paper builds upon the concept of shared component marketplace as proposed by several EU funded projects to address complexity of enterprise applications development (FIWARE 2023, MARKET4.0 2023, SHOP4CF 2023, Zimniewicz 2020). The case study application, presented in the paper, SHOP4CF components repository is adopted. IoT-based indoor localization system using UWB is implemented by the WiPos component (Wi-POS, 2023). The WiPos hardware consists of a sink node connected via USB to Raspberry Pi, 4 anchor nodes and a tag device which position is calculated. WiPos pushes the calculated relative position data from IoT devices to Orion Context Broker instance. Orion Context Broker (Orion Context Broker, 2023) is middleware aiming convenient data acquisition from IoT devices relying on publish-subscribe mechanism. Indoor positioning is performed periodically and corresponding information is published as NGSI v2 entity (Fonseca et al., 2023) with the following properties: 1) *id* – tag identifier, 2) *observedAt* – timestamp, and 3) *relativePosition – [x, y]* coordinates.

2.4 Related Works

There are several existing works that aim to enable convenient low/no-code development of MSA-based applications by tackling some of the relevant aspects, relying on semantic technology or other underlying approaches as well. Overview of these solutions together with their descriptions and covered aspects are given in Table 1.

Table 1: Overview of similar solutions.

Solution	***Description***	***Aspect***
SMADA-Fog (Petrovic et al., 2019)	Model-driven, semantic-enabled deployment of containerized micro services, supporting both Edge and Cloud.	Deployment
Formaloo (Formaloo, 2023)	Building customer portals, CRMs, and other business apps without any code.	App creation and deployment
AppSheet (AppSheet, 2023)	Low-code development of cross-platform mobile apps, using tabular data sources	App creation and deployment of mobile applications
Ontopic Studio (Ontopic, 2023)	Environment for designing semantic layers as a knowledge graph with no code.	Domain modelling

It can be seen that existing approaches often do not include convenient integration of data from various sources and usage of external service. On the other side, we aim to cover all the mentioned aspect within one solution. Our proposed approach aims to tackle the mentioned challenges in area of Industry 4.0 enabling automated deployment of complex industry-oriented applications, bridging the gaps related to heterogeneity of devices and integration with legacy systems. However, our approach also aims to provide unified interoperability and integration of external services based on their specification, such as in case of OpenAPI. Additionally, our ecosystem provides low-code approach to incorporation of domain modelling and conceptualization using Domain editor. Finally, apps created using our solution based on MFE are mobile device-compatible as well.

3 IMPLEMENTATION

Figure 1 depicts logical software architecture of the proposed ASMOSA's ecosystem.

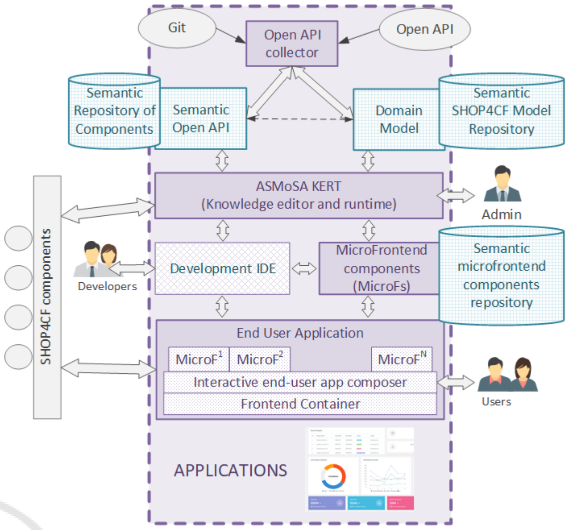

Figure 1: System architecture overview.

The central component of ASMOSA platform is Knowledge editor and runtime (KERT) application which gives ability to domain experts (such as operation managers in different departments - manufacturing, sales, marketing, development, etc.) to intuitively configure different individual applications in a drug-n'-drop manner without any programming-related knowledge. The key enabler for easy-to-use, automated application creation using KERT is semantic technology in synergy with predefined MFE components. MFE components can be imported from semantically annotated central component repository, integrating various sources (such as SVN or Git). When it comes to underlying technology, the core of semantic engine is implemented in Java relying on a triplestore implementation of SPARQL endpoint, while MFEs are implemented in HTML and JavaScript related technologies.

On the other side, adoption of domain ontologies provides convenient way to construct applications by leveraging conceptualization, provided by domain experts. Moreover, with respect to given semantic representations, we provide mechanisms which enable import of tabular data from various external sources, such as ERP. Additionally, ontologies are also used for semantic mapping of API specifications (such as OpenAPI), which provides convenient

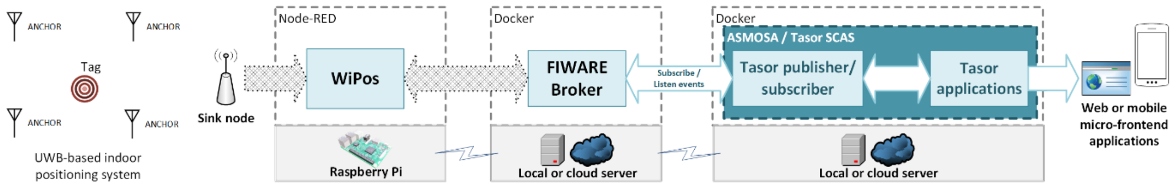

Figure 2: Component interaction within the ASMOSA case study application.

extension with new services. This way, seamless integration of data, services and knowledge available on Edge servers as well as public or private Cloud services is achieved.

When it comes to app creation using KERT, the desired components can be imported from central components repository and added to user's project. Furthermore, it is possible to customize UI elements, such as colors of buttons, controls and labels. Once the app design is finished, build and deployment can be done, so the application becomes immediately available to the end-users. Required back-end functionality is provided by semantic integration of different sources, including enterprise knowledge graphs, simple data tables and interfaces of services. This way, we also provide the means for leveraging the existing SHOP4CF components and their adoption within our MFEs. End-users simply access available applications using either smartphone, tablet or PC devices.

Additionally, we implemented BrokerConnector Service which acts as general adapter between TASOR-based applications and FIWARE Orion Context Broker, relying on TASOR-compliant libraries and service architecture. Its methods can be used to interact with any instance of FIWARE Orion Context Broker server specified by IP address as API call parameter. BrokerConnector service methods are application independent and can be used for data collection in different use-case specific applications. BrokerConnector service provides set of data retrieval and management operations: creation of new entities, creation of new subscriptions, handling of entity updates utilizing publish-subscribe mechanism, updating entities, deleting entities and subscriptions. In this way, interaction of ASMOSA platform and SHOP4CF infrastructure is achieved, as illustrated in diagram within Figure 2 gives more detailed view focused on interaction of the involved components in the presented case study.

4 SEMANTIC FRAMEWORK

In this section, we give overview of the underlying ontologies behind our approach, including both domain-specific ontologies and auxiliary ontologies which are further used for semantic mapping.

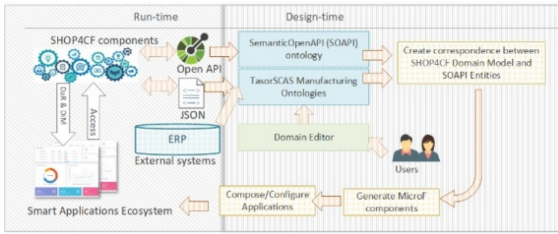

Figure 3: Overview of the proposed semantic-driven approach to low code application development.

Illustration showing the role of ontologies within our proposed approach is given in Figure 3, while the part focused on MFEs is depicted in Figure 5. ASMOSA is based on the TasorSCAS platform (TasorSCAS, 2023) that represents all system metadata semantically using ontologies. Its aim is to adopt a common ontology to encode the shared understanding. The corresponding communication language is built upon this understanding and implemented using the common ontology. Thanks to that, each participant can "understand" the communication language by alignment of the common ontology of interaction to its own local knowledge represented by one or more ontologies. For that purpose, we make use of semantic mapping, which is defined using a mapping language specified by an ontology as well. In our case study, mapping is used to cover the following aspects: 1) integration with functionality of external services leveraging OpenAPI specification – such as SHOP4CF components 2) representation of domain-specific data - such as JSON with relative position or ERP. In order to unify the domain knowledge about various aspects related to manufacturing, proprietary TasorSCAS Manufacturing Ontologies (Production, Planning and Transportation) are used. On the other side, we provide Domain Editor where users can provide additional information useful for integration of external data using GUI.

Finally, once all the knowledge is grounded to common base – semantic knowledge graph grounded

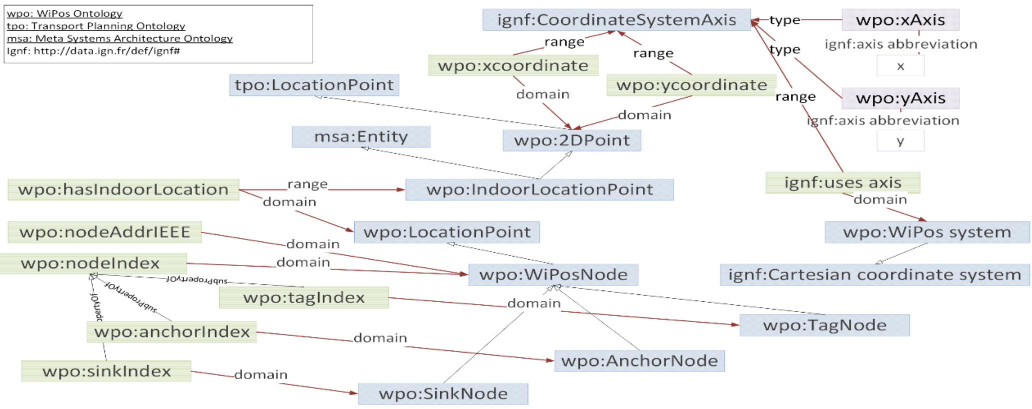

Figure 4: Indoor localization Domain Ontology - an illustrative segment.

on Tasor Ontologies. Knowledge from this graph is further used for generation of micro frontend components that are finally composed into end-user applications, where different parts of semantic knowledge graph are leveraged for rendering various parts of the result.

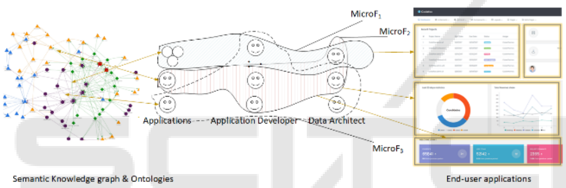

Figure 5: Semantic micro frontend based architecture.

4.1 WiPos Domain Ontology

The main purpose of this ontology is to formalize the understanding of relative positioning data coming from IoT-based WiPos indoor localization system within the presented ASMOSA case study. Sink node calculates tag position, while it is connected to Raspberry Pi sending messages containing the 2D coordinates representing tag device position in JSON format to Orion Context Broker. On the other side, UWB-based anchor nodes are also part of localization system and they participate in position calculation. Each of anchor nodes has both ordinal number and IEEE address in order to be distinguished. In order to semantically interpret the position-related content within ASMOSA, a domain ontology was developed using TasorSCAS tools, as illustrated in Figure 4.

4.2 TasorSCAS Manufacturing Ontologies Framework

Conceptual foundation of the TasorSCAS framework related to manufacturing builds upon of the following three ontologies: *Production Ontology*, *Planning Ontology*, and *Transportation Planning Ontology*. These ontologies enable us to effectively understand and work with specific data in different domains.

The production ontology primarily deals with definitions of concepts related to production resources, encompassing units, products, machines, and two types of actions: durative (actions that occur over a specific duration) and instantaneous (actions that occur at a specific moment). By utilizing the production ontology, our system gains a comprehensive understanding of the production-related entities and their associated actions.

The planning ontology, on the other hand, focuses on data related to product orders, schedules, product IDs, customer IDs, and prioritization. This ontology assists in managing and organizing the planning aspects of our system, allowing for efficient handling of product orders, scheduling operations, and prioritizing tasks based on predefined criteria.

Furthermore, we have developed a transportation planning ontology that works with geolocations and relative positioning, including locations on a factory floor. It incorporates indoor positioning technologies like WiPos. By integrating geolocation and positioning data, our system can optimize transportation planning within the factory environment, enhancing efficiency and logistics.

4.3 Semantic Mapping Approach

ASMOSA is built upon the semantic-oriented TasorSCAS infrastructure and makes use of semantic mapping as defined there. Despite that more than few respectful approaches to model mapping have been proposed and some of them are established as standards, together with more general foundational ontologies (such as BFO, DOLCE, UFO) to the best

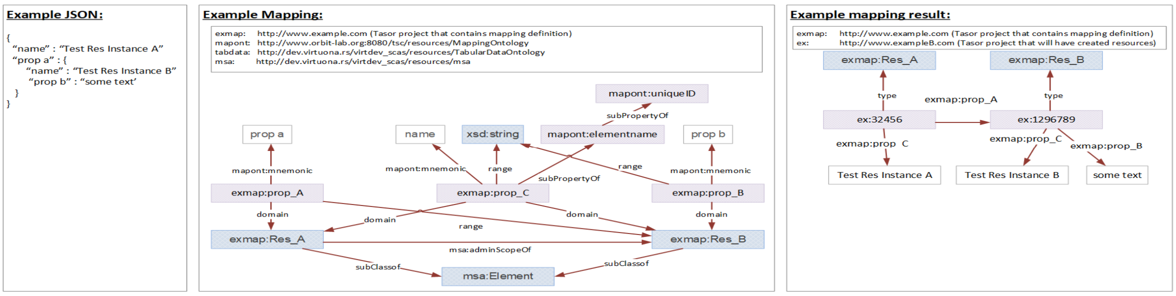

Figure 6: Example semantic mapping of JSON file for data exchange with FIWARE Context Broker.

of our knowledge, none of them is fully ontology based. Additionally, we provide an additional tool - Domain Editor, which enables users to describe their data using simple and intuitive GUI. In this way, user-specific data are grounded on our Production Ontology, so several applications related to manufacturing (planning, logistics, scheduling, etc.) can be adopted by users without any additional complexities. Our aim is to eliminate traditionally complex, costly and timely implementation projects that are main barrier for modern agile digital transformation practices, which represents value add. An illustrative example showing how the mapping ontology is applied within our case study for data exchange with broker, shown in Figure 6 .

4.4 Domain-Specific Mapping Ontology

As a proof of concept, we developed DMV mapping ontology representing set of facts generated by implementation of our Mapping Ontology on data from specific Enterprise Resource Planning (ERP) software used in our testing partner DMV Company. This ontology is specifically tailored for the requirements of the partner's data. The DMV mapping ontology serves as an intermediary, facilitating the import of data from their ERP system and aligning it with the corresponding ontologies within our system, namely the production ontology, planning ontology, and transportation planning ontology. Importing data relies on automated steps, eliminating the need for manual interventions in order to save the time and effort needed. Therefore, such customer-specific ontologies represent the main target for future extensions for various cases.

4.5 OpenAPI Mapping Ontology

In order to implement and add new services within our system, we decided to develop our services in accordance with the OpenAPI scheme. We are using Open API Schema version 3.0 and have developed domain and mapping ontology for this version. The developed ontology adopts and customizes the previous work (Aikaterini et al., 2020) with a goal to better fit the task in hand. Note that the Orion Broker component, that is the central component in the SHOP4CF infrastructure, publishes its API compliant to the OpenAPI standard. Component Orion Broker Connector is used to facilitate data integration between ASMOSA and Orion Broker component. As such, it publishes its API compliant to the OpenAPI as well. In this way, any existing SHOP4CF component can be included into solutions generated using ASMOSA. As a proof-of-concept, we develop applications that use WiPos component that publishes data to the Orion Broker while ASMOSA collects these data automatically using our connector component (see Figure 2). When creating the ontology, we opted for OpenAPI's JSON format. The key element in the ontology itself is represented by the Schema Object class, which represents the root object of JSON itself. From this class we then derive basic Open API objects such as: Info, Services, Paths, and Components. Depiction of corresponding semantic representation for OpenAPI Schema object is given in Figure 7. For each of them, we define the OpenAPI Object domain class using properties of the same name, which represents a single object within OpenAPI. Then, for each class that represents the basic objects in the OpenAPI configuration, we added properties that characterize the given class. An interesting problem concerning the OpenAPI ontology mapping relates to the *Paths* class. Namely, in this case identifier is implicitly defined. We solved the problem by introduction of the *PathItem* class that represents the *adminscopeOf* of the Paths class and represents the domain of the *pathPrefix* and *pathString* properties as well as properties for the methods used to work with the service itself.

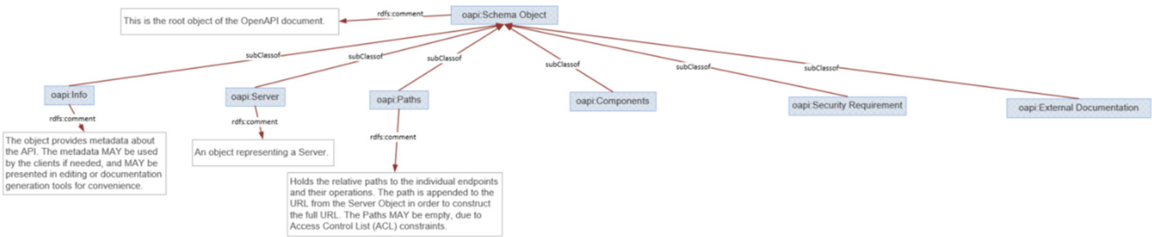

Figure 7: Ontology segment for semantic representation of the OpenAPI Schema object.

5 CASE STUDY

Goal of the presented case study application is to enable efficient workforce and operation monitoring in a manufacturing company. For this purpose, we make use of WiPos indoor localization and semantic annotations. In our experiment, the final application was set up within the manufacturing area of DMV company in Niš, Serbia.

Video of the live demo in action is available on YouTube [3]. *Semantic Annotation* view gives the ability of adding information about the machines involved in industrial process, based on their indoor position within the manufacturing area. Indoor position is collected by means of WiPos tag that publishes its position to the SHOP4CF Orion Context broker component. On the other side, Figure 8 shows the *Operation Monitoring* view, whose goal is to provide insight to the managers about the machines, operations performed, usage and work hours spent by employees assigned to them.

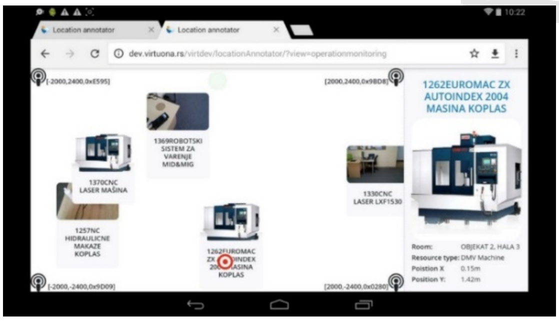

Figure 8: Operation Monitoring view, rendered on Android tablet device.

In order to create such applications, ASMOSA's app studio was used. Moreover, *Domain Editor* as used in order to define crucial concepts – machines, operations and location. After that, managers were able to simply import tabular data about operations and machines from ERP.

6 EVALUATION

For purpose of evaluation for our approach, several relevant aspects are considered based on the presented case study: data sources integrated, time needed for KERT-aided app creation and additional artifacts needed for extension of the approach in order to use it for another case study if required. Overview of experiments and results is given in Table 2.

Table 2: Evaluation aspects and achieved outcomes based on DMV case study.

Aspect	Outcome	Efforts needed
Data sources	WiPos positioning system, DMV's ERP	WiPos ontology DMV ontology
App creation	Semantic Annotator and Operation Monitoring	Domain specification using KERT
Conecptualization of domain	Domain ontologies	Around 9 min for ontology creation
BrokerConnector customization	WiPos broker connector	Around 5 min for API call parameterization

As it can be seen, for custom tailoring of industry-oriented applications, less than 15 minutes overall were needed for additional steps that would make it possible to apply the proposed solution to other case studies, which eliminates the need for developing components from scratch. Just several minutes are enough for domain expert to define crucial concepts of the underlying domain using Domain Editor, making the import of external data from new sources possible. This way, it is possible to significantly

[3] https://youtu.be/ajMcmGThpYw

speed up the overall software development process, while overcoming the barriers of both data source and external service integration, thanks to utilization of ontologies and semantic mapping. Finally, to achieve customization of the presented solutions for new application, programming expertise is not a must, thanks to adoption of no-code approach enabled by combination of ontologies and frontend components.

7 CONCLUSION AND FUTURE WORKS

In this paper, we introduced an approach to semantics-aided development of industry-oriented applications. As a proof of concept, we demonstrated case study evaluated within a realistic usage scenario. According to the results, the proposed solution has proven effective in several aspects of importance for industry-oriented applications: integration of data from heterogeneous sources (IoT devices, ERP) as well as execution environments (IoT devices on the Edge and Cloud services). MFE has been proven to speed up the whole process of application development and deployment, while making the software applications development learning curve much less step, bringing it closer to domain experts without programming experience. Finally, the presented case study could be easily adapted for other IoT data sources given the corresponding domain ontology and broker connector API calls.

In future, we plan to explore the potential of trending ChatGPT conversational agent and Large Language Models (LLM) in order to integrate it within our semantics-enabled framework. Our goal is to automatize some of the ontology-related tasks, such as automated domain ontology construction based on textual or tabular sources; complex SPARQL query generation with minimal or without user intervention; automated service code generation.

ACKNOWLEDGEMENTS

This research work has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 873087 (SHOP4CF) as one of the winners in OC2.

REFERENCES

Alsaadi, N. (2022). Modeling and Analysis of Industry 4.0 Adoption Challenges in the Manufacturing Industry. Processes. 10. 2150 (2022). 10.3390/pr10102150.

AppSheet, https://about.appsheet.com/home/, last accessed: 17/07/2023.

Babu, B., Nallasivam, J. D., Aravinth, S. (2023). Overview of Industry 4.0. 15th International Conference on Science, Technology and Management, pp. 14-16 (2023)

FIWARE, https://www.fiware.org/marketplace/ last accessed: 17/07/2023.

Fonseca, J.; Márquez, F.; Jacobs, T. NGSIv2 API Specification—v2.0n. Available online: https://fiware.github.io/specifications/ngsiv2/stable/ (accessed on 26. September 2023).

Formaloo, https://www.formaloo.com/en/, last accessed: 17/07/2023.

Geers, M., What are Micro Frontends?, https://micro-frontends.org/, last accessed 16/07/2023.

Aikaterini, K., Mainas, N., Petrakis, E. GM. "Ontology for openapi rest services descriptions." In 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), pp. 35-40. IEEE, 2020.

MARKET4.0, Marketplaces, https://market40.eu/project/marketplaces/index.html last accessed: 17/07/2023.

Ontopic Studio, https://ontopic.ai/en/ontopic-studio/, last accessed: 17/07/2023.

Orion Context Broker. https://fiware-orion.readthedocs.io (accessed on 18 July 2023).

Petrovic, N., Tosic, M.: SMADA-Fog: Semantic model driven approach to deployment and adaptivity in fog computing, Simulation Modelling Practice and Theory Volume 101, 102033, 1-25 (2020). https://doi.org/10.1016/j.simpat.2019.102033

Pontarolli, R.P., Bigheti, J.A., de Sá, L.B.R., Godoy, E.P.L. (2023). Microservice-Oriented Architecture for Industry 4.0. Eng 2023, 4, 1179-1197. https://doi.org/10.3390/eng4020069

Romero, J. R., Medina-Bulo, I., Chicano, F.: Optimising the Software Development Process with Artificial Intelligence, Springer Singapore. (2023)

SHOP4CF, https://www.shop4cf.eu/, last accessed 2023/07/12.

Taibi D. and Mezzalira, L. (2022). Micro-Frontends: Principles, Implementations, and Pitfalls. SIGSOFT Softw. Eng.

TasorSCAS, https://www.virtuonasoft.com/scas.html, last accessed 2023/07/24.

Wi-POS Indoor Localization, https://shop4cf.github.io/communication-docs/partners/IMEC/Wi-POS.html, la st accessed 2023/07/12.

Zimniewicz, M. Deliverable 3.2 - SHOP4CF Architecture. 2020. https://elk.adalidda.com/2021/05/SHOP4CF-WP3-D32-DEL-210119-v1.0.pdf (accessed on 18 July 2023).