Node-RED Workflow Manager for Edge Service Orchestration

Felix Larrinaga
Faculty of Engineering
Mondragon Unibertsitatea
Arrasate-Mondragon, Gipuzkoa, Spain
flarrinaga@mondragon.edu

William Ochoa
Faculty of Engineering
Mondragon Unibertsitatea
Arrasate-Mondragon, Gipuzkoa, Spain
wsochoa@mondragon.edu

Alain Perez
Faculty of Engineering
Mondragon Unibertsitatea
Arrasate-Mondragon, Gipuzkoa, Spain
aperez@mondragon.edu

Javier Cuenca
Faculty of Engineering
Mondragon Unibertsitatea
Arrasate-Mondragon, Gipuzkoa, Spain
jcuenca@mondragon.edu

Jon Legaristi
Faculty of Engineering
Mondragon Unibertsitatea
Arrasate-Mondragon, Gipuzkoa, Spain
jon.legaristi@alumni.mondragon.edu

Miren Illarramendi
Faculty of Engineering
Mondragon Unibertsitatea
Arrasate-Mondragon, Gipuzkoa, Spain
millarramendi@mondragon.edu

***Abstract*—Microservice Architectures have increasingly become popular in Industry 4.0 as they allow heterogeneous systems to interact, reduce the complexity in the management of individual components, and support distributed deployments. The integration of those distributed services into orchestrated production processes is performed by workflow managers. Next generation workflow managers must overcome a number of challenges when operating in microservice architectures and IoT environments. To overcome these challenges (heterogeneity, high dynamism, edge deployment or scalability), we propose a workflow manager alternative built in Node-RED. Node-RED provides instruments for the development of IoT systems and leverages the edge computing paradigm. This solution is deployable in embedded systems, is able to load and execute business processes by means of BPMN recipes and enables the integration of other frameworks and architectures.**

***Index Terms*—Business processes, Workflow Management, Industry4.0, Service Oriented Architecture, Microservices, Node-RED**

I. Introduction

Industry 4.0 is a new era for manufacturing where digitalization and collaboration among devices through digital services is essential. The aim is to integrate the Internet of Things (IoT) with the Industrial Internet enabling the concepts of Smart Manufacturing [1] and Cloud-based Manufacturing [2]. Industry 4.0 relies on the adoption of digital technologies to gather data in real-time, to analyze it and provide useful information to the manufacturing systems [3]. In this context, the concept of Cyber-Physical System (CPS) [4] acquires special relevance. CPSs are constituted by collaborating computational entities that are connected to the surrounding physical industrial assets to provide and consume Internet services [5]. The proliferation of CPSs and the high servitization of industrial assets imply the need to meet the requirements of a Service-Oriented Architecture (SOA) which main concerns is the integration of heterogeneous systems enabling the communication and interaction among the different players of the manufacturing environment [6]. Microservice architectures are considered the evolution of SOA [7], where services are the central element for the implementation of digital applications. In a microservice architecture, services are finer-grained and loosely coupled [8]. Microservices reduces complexity in the management of individual components by supporting distributed deployments. This decreases development and maintenance costs [9]. On the contrary composition of processes that combine heterogeneous services is a challenge.

Business Process Management (BPM) is a discipline in which a set of techniques is used for the continuous and iterative improvement of a company's operations. The objective is to improve the automation of tasks, the reduction of costs and the achievement of error-free processes [10]. In the Industry 4.0 domain, the automation of tasks is conceived as the management of workflows where machines and humans collaborate in the completion of business processes. Many are the alternatives commercial and open source that propose solutions for the correct management of processes or workflows. Eclipse Arrowhead [11] is a framework that addresses the requirements imposed by microservice architectures. In the context of SOA and microservice architectures, the management of workflows addresses a number of challenges (see L. Vaquero et al. [12]). One of the key challenges is to provide solutions at edge level close to the workstation.

In this paper we present a workflow manager for the orchestration of services in the edge using Node-RED. Node-RED [13] is a programming tool specially designed for the integration of APIs, hardware devices, web services, databases, messaging systems, etc. The paper is structured as follows. Section II provides current workflow managers implementations and the motivation for a new alternative. Section III presents the architecture of this workflow manager alternative. Section IV, addresses the implementation of the proposal in a use case. Finally, Section V presents the conclusions.

II. Related Work

Several frameworks have emerged in recent years to meet the challenges introduced by microservice architectures in the context of Industry 4.0. The goal is to achieve high automation and digitisation of services. The proposed frameworks include: FiWare [14], AUTOSAR [15], Eclipse BaSyx [16] and Eclipse Arrowhead [17] [11]. A comparison of recent platforms and state-of-the-art of industrial IoT frameworks can be found in [18]. Among these alternatives the Eclipse Arrowhead framework has positioned itself as an interesting solution. It enables the provision of local cloud services allocated in different devices, gateways, or systems to facilitate and standardize the interaction between within a microservice architecture [17].

A. Workflow Management Systems

In recent years, different 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 suits are often utilized for robust collaboration, automation, and tracking of business processes [19]. 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"* [20]. iBPMS is an evolution of WfMS, since it incorporates a set of technologies to coordinate services, processes, machines, and people [21], improving and transforming workflow management discipline through a collaborative platform. Among the open-source iBPMS mentioned in the Gartner's Magic Quadrant [22], the following are the prefer choices: Camunda [1], Bonita [2], and Process Maker [3]. All these solutions run on a centralised server usually deployed in cloud infrastructure.

The Eclipse Arrowhead framework in addition to provide core systems for service register, discovery and authentication [23] offers systems to improve security, adapters for interoperability and communication management for services located in different ecosystems. These systems are organised in layers. The Workflow Support System layer includes different alternatives for workflow management in SOA and microservice architectures [24] [25] [26] [27].

B. Challenges and Motivation

The main challenges for the orchestration of services in the context of business processes have drawn the interest of the research community and opened the door for further investigation. L. Vaquero et al. [12] identified the orchestration requirements for next-generation workflow management in microservice architectures and IoT environments. Those requirements are taken as the ground for this analysis and as the motivation to build another workflow manager alternative.

- Architecture: Fog computing delivers a growing number of services as more devices connect to the network. This increases the difficult for workflow managers to orchestrate fog services, as most solutions rely on cloud infrastructure or central servers. Thereby, workflow managers are needed at workstation level, close to the final resources to leverage them by the integration of event driven architectures. This also improves scalability. Workflow managers must support scalability by aggregating functionality in sub-workflows.
- Heterogeneity: In a fog computing environment where devices operate together, the information and functionalities become fragmented since data elements reside inside devices. These fragmented/heterogeneous information/functionalities increase the difficulty of orchestration. Moreover, the nature of services and the capacity of the edge device to store information have a high impact on this challenge. Asynchronism, parallelism, capacity of devices orchestrating services are to be considered here.
- Collaboration and Standardization: Workflow managers must be designed to generate and manage workflows in collaboration with other systems and their associated resources. Thus, standardization of inputs and outputs are required to enable communication between systems. The workflow recipes must be based in standard notation languages in order for workflow managers to support scalability. The number of available notation languages is vast, but those that have wider acceptance are EPC, YAWL, BPEL, XPDL, and BPMN [28]. Moreover, the Object Management Group (OMG) ratified BPMN as the standard language for the design of business processes [29].
- Dynamism: In a continuous integration software development, dynamic deployment becomes critical in an edge environment while dealing with device heterogeneity. Moreover, dynamic reconfigurations of edge devices can increase network incidents and temporary malfunctions. A service orchestrator has to provide root cause diagnosis during service disruptions and return fast and accurate decisions. In this way, an orchestrator needs to cope with the loss of connectivity and the increased likelihood of device failure.
- Discovery: Edge devices are intrinsically volatile, they can be disconnected and their functions will become unavailable. As a consequence, the availability of resources and functionalities is unreliable. In order to accomplish the service level agreements (SLA), workflow managers must be provided with an agile service discovery mechanism at the plant level to leverage new resources of processes as they are created.
- Security: IoT and edge networks are vulnerable to attacks. These attacks condition the management of workflows. Workflow managers must support security aspects at the edge such as authentication, authorization, access management, etc.

Considering these challenges, the motivation of this paper strives in provisioning a workflow manager that: 1) Can be easily deployed in an embedded system and consequently

[1] https://camunda.com/
[2] https://www.bonitasoft.com/
[3] https://www.processmaker.com/

placed close to the devices on a workstation, 2) Understands recipes build in BPMN, 3) Enables the integration of other systems and architectures, 4) Can be easily replicable and scalable. Additionally, the Eclipse Arrowhead framework supports workflow managers in the last two challenges so an additional requirement will be to make it Eclipse Arrowhead compliant.

C. Node-RED

Node-RED is an open-source flow-based programming tool that provides instruments for the development of IoT systems. Node-RED allows integrating APIs, hardware devices, and web services as part of the IoT paradigm [30]. In Node-RED, a flow is a system or a portion of a system (subflow) containing functionalities wired together called nodes. Nodes communicate among them by sending and receiving messages in order to collaborate for the completion of the flow. Node-RED is built on Node.js which requires little resources during execution making it ideal to run at the edge on low-cost hardware (Raspberry Pi or similar). In addition Node.js takes full advantage of event driven architectures which makes Node-RED an interesting alternative for IoT based systems. These characteristics have led us to choose Node-RED as the tool to build a workflow manager that can be deployed in an embedded system and integrate the benefits of an IoT-based architecture.

Another interesting feature of Node-RED is the possibility to create custom nodes [31] making it a way to extend the tool with more functionality. Moreover, the custom nodes can be added into Node-RED's palette for reuse and they can be further published for the use of the community on the npm repository and/or the Node-RED Flow Library. This is key for the integration of Node-RED with other frameworks. In addition to the workflow manager presented in this paper we have created Node-RED nodes for the integration between the core systems of the Arrowhead Framework (Authorization, ServiceRegistry, and ServiceDiscovery) and flows in Node-RED [32]. These nodes are available online for the use of the community. A demonstrator can be seen in [33].

III. Node-RED Workflow Manager

The Node-RED Workflow Manager [4] (Node-RED WM) is a new 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 Workflow Manager developed in Node-RED offers the endpoints for communication, holds the code to manage the BPMN recipes and interacts with the external systems by invoking their services or receiving their asynchronous replies.

Node-RED WM uses a relational database to store the information necessary to complete workflow processes (recipes), instances, tasks, and status. This way the Node-RED WM knows what process instances are running, which is the currently active task and goes through the recipe by identifying

[4]https://github.com/wsteven24/NodeRed-Workflow-BPMN

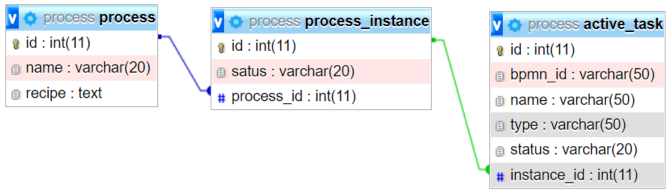

Fig. 1. Database Entity–relationship model.

the next task to run in the process until the flow is completed. The structure of the database is shown in Fig. 1 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*.

As seen in Fig. 2, Node-RED WM is composed of two building blocks:

1) The Process Manager: Manages the functionalities related to the BPMN recipe as a whole. Its flows are:

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

- *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 requests 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.
- *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.

2) The Task Executor: Manages the functionalities for the orchestration of individual tasks. The flows in this block are:

- *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.
- *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.

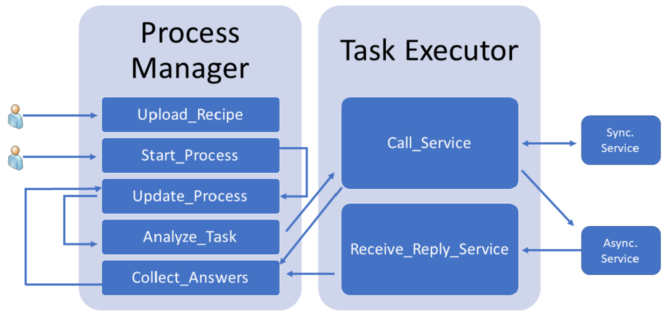

Fig. 2. Node-RED Workflow Manager

IV. Implementation of the Node-RED Workflow Manager

This section presents the implementation of the Node-RED Workflow Manager in a demonstrator. The demonstrator implements a workstation build with two Lego Education Mindstorms machines (see Fig. 3). The first Lego Device (LD1) is a color sorting machine (Lego Educational Mindstorm EV3). The second Lego Device (LD2) is a crane that collect buckets with pieces and moves them to a given position. Each Lego Device brick (Dexter Industries BrickPi) is attached to a Raspberry Pi board. MySQL, Node-RED and OPC-UA servers are installed in those boards. Node-RED and MySQL necessary to execute the Node-RED WM only occupy 16% of the Raspberry Pi 1G memory capacity. 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.

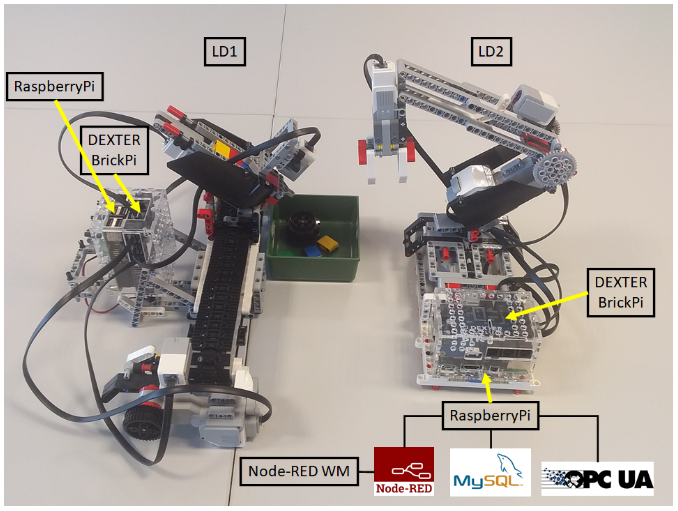

Fig. 3. Node-RED Workflow Manager Demonstrator.

The Node-RED WM orchestrates 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 offers Initialized station (S1), Move Left (S2), Move Right (S3), Through piece (S4) and Select pieces(array with color and number) (S5)
- LD2 offers Initialized station (S1) and Move Bucket(from pos1 to pos2) (S2)

The Node-RED WM as described in Section III is installed 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 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 from asynchronous replies from the Lego devices.

Once the BPMN recipe is uploaded to the Node-RED WM as an 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. By invoking the endpoint with the process recipe name as a parameter, an instance of the process is executed. Node-RED WM creates separated instances of a process recipe if invoke 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. 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. The Node-RED WM offers a web dashboard to monitor the status of the processes.

V. Conclusions

In response to the motivation points identified at the end of section II-B, we have developed and tested in a use case a new workflow manager (Node-RED WM) that runs in systems with little resource requirements. The main advantages are:

- Node-RED WM is installed and runs in a embedded system (Raspberry Pi) consuming little resources. This makes the alternative appropriate to be deployed at the edge close to the services available at workstation level and answers to point 1 in the motivation list.
- Node-RED WM works with business process management standard notation languages. It manages BPMN recipes complying with point 2 in the motivation list.
- Node-RED WM is prepared to manage different architectures and flow conditions. Node-RED is a framework with modules and nodes for the integration of different architectures (i.e. event driven), IoT oriented data repositories (i.e. No-SQL data repositories) and streaming or communication channels. This complies with point 3 in the motivation list.
- Node-RED WM is prepared to work inside local Arrowhead clouds using the core services provided by the Eclipse Arrowhead framework and the Node-RED adapters presented in Section II-C.
- The solution is able to manage multiple process and instances simultaneously. Parallelism and concurrency are managed using correlation for process instance and task identifiers. A process can be launched multiple times and execute multiple instances simultaneously. This depends on the availability and nature of resources.
- Node-RED WM is prepared to accommodate asynchronous services.
- Node-RED WM is offered in Docker technology what makes it highly scalable and easy to replicate. This complies with point 4 in the motivation list.

Although the technology readiness level of this workflow manager alternative is low, Node-RED offers the potential to enrich the solution by 1) accommodating other types of data formats or recipes, 2) integrating other frameworks and systems such message brokers, databases, protocols (OPC-UA), 3) including IoT context data from sensors or 4) collaborating with other alternatives like WSO2.

Many of the challenges presented in Section II-B such as security and discovery are addressed by the core services of the Eclipse Arrowhead framework. Thus, Node-RED WM can rely on the Eclipse Arrowhead framework to inherit these features and confront those challenges. Future work includes testing this edge solution in real scenarios and measure and evaluate the performance in comparison to cloud-centric solutions.

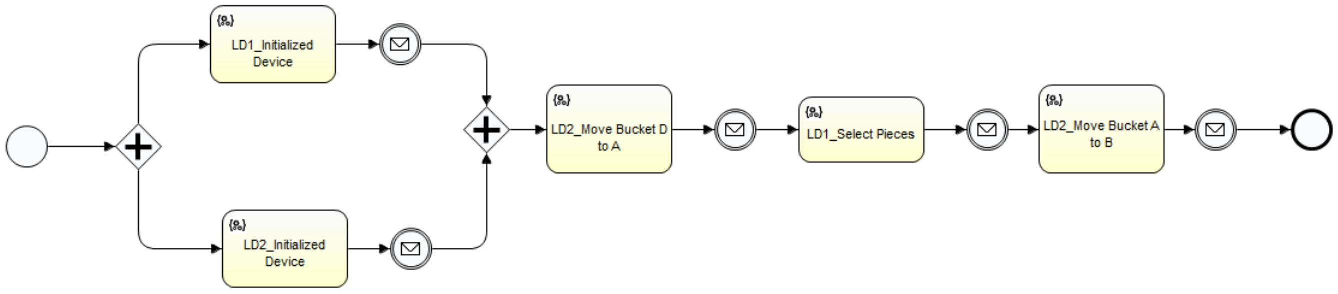

Fig. 4. BPMN Recipe

Acknowledgment

This research work has been funded by the European Commission, through the European H2020 research and innovation program, ECSEL Joint Undertaking, and National Funding Authorities from 18 involved countries under the research project Arrowhead Tools with Grant Agreement no. 826452 and the Department of Education, Universities and Research of the Basque Government under the projects Ikerketa Taldeak (Grupo de Sistemas Embebidos)

References

[1] S. Jeschke, C. Brecher, T. Meisen, D. Özdemir, and T. Eschert, *Industrial Internet of Things and Cyber Manufacturing Systems*. Cham: Springer International Publishing, 2017, pp. 3–19. [Online]. Available: https://doi.org/10.1007/978-3-319-42559-7_1
[2] S. Vaidya, P. Ambad, and S. Bhosle, "Industry 4.0–a glimpse," *Procedia manufacturing*, vol. 20, pp. 233–238, 2018.
[3] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612515000400
[4] 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.
[5] 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.
[6] 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.
[7] L. Baresi and M. Garriga, "Microservices: the evolution and extinction of web services?" *Microservices*, pp. 3–28, 2020.
[8] C. Richardson, *Microservices patterns: with examples in Java*. Simon and Schuster, 2018.
[9] 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.
[10] OMG, "Business Process Management (BPM)." [Online]. Available: https://www.omg.org/technology/readingroom/BPM.htm
[11] [Online]. Available: https://projects.eclipse.org/projects/iot.arrowhead
[12] 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.
[13] I. E. T. Services, "Node-RED." [Online]. Available: https://nodered.org/
[14] Fiware, "Fiware." [Online]. Available: https://www.fiware.org/
[15] Autostar, "Autostar - the standardized software framework for intelligent mobility." [Online]. Available: https://www.autosar.org/
[16] [Online]. Available: https://www.eclipse.org/basyx/
[17] J. Delsing, *Iot automation: Arrowhead framework*. Crc Press, 2017.
[18] C. Paniagua and J. Delsing, "Industrial frameworks for internet of things: A survey," *IEEE Systems Journal*, vol. 15, no. 1, pp. 1149–1159, 2020.
[19] Signavio, "What is workflow management?" [Online]. Available: https://www.signavio.com/post/what-workflow-management-is/
[20] Gartner, "Market Guide for Intelligent Business Process Management Suites." [Online]. Available: https://www.gartner.com/en/documents/3993207-market-guide-for-intelligent-business-process-management
[21] ——, "Gartner Magic Quadrant for Intelligent Business Process Management Suites." [Online]. Available: https://www.gartner.com/en/documents/3899484
[22] auraquantic, "La información en la gestión por procesos con ibpms." [Online]. Available: https://www.auraquantic.com/es/la-informacion-en-la-gestion-por-procesos-con-ibpms/
[23] 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
[24] 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.
[25] J. Garcia Represa and J. Delsing, "Autonomous production workstation operation, reconfiguration and synchronization," *Procedia Manufacturing*, vol. 39, pp. 226–234, 2019.
[26] 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
[27] 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
[28] 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, no. 3, 2015.
[29] OMG, "Business process model and notation (bpmn)." [Online]. Available: https://www.omg.org/spec/BPMN/2.0/About-BPMN/
[30] I. E. T. Services, "Node-red," 2022. [Online]. Available: https://nodered.org/
[31] Node-RED, "Node-red, creating nodes," 2022. [Online]. Available: https://nodered.org/docs/creating-nodes/
[32] 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
[33] 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/