Early lessons learned from the development of a local OPC UA-based robotic testbed for research

Quang-Duy Nguyen, Fadwa Tmar, Yining Huang, Saadia Dhouib

Early Lessons Learned from the Development of a Local OPC UA-based Robotic Testbed for Research

Quang-Duy NGUYEN, Fadwa REKIK, Yining HUANG, and Saadia DHOUIB
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France.
Email:{quang-duy.nguyen, fadwa.tmar, yining.huang, saadia.dhouib}@cea.fr

***Abstract*—OPC UA has been emerging as a widely-adopted standard in the industry. An OPC UA system represents its resources using an information model. This information model relies on the base OPC UA information model proposed by OPC Foundation, and is extensible with information models from other companion specifications and user-defined information models. When the system specification and design evolve, system developers need to update the information model from the system design. It is tricky for testbeds that frequently are updated with new devices, protocols, and development tools demanded by different use cases. This paper shares our long-term OPC UA-based robotic testbed for Industry 4.0 research and the strategy to develop it. The testbed's development follows the SysML standard for system specification and design and the OPC UA standard for system implementation. Its information model relies on the information models: OPC UA for Devices and VDMA OPC UA for Robotics.**

***Index Terms*—Industry 4.0, Testbed, OPC UA, Devices, VDMA, Robotics, Mini-waterfall, Trigger**

I. Introduction

A testbed is essential for research and education since it provides an environment for system development and testing. Developing a testbed is different from developing a use case in two following points. First, a testbed can provide the platform for several use cases; thus, developers should maximize its potential instead of focusing on the goals of a specific use case. Second, a testbed for long-term use can grow progressively. For example, it can integrate new devices and protocols. The development should consider improving and reconfiguring the testbed when there is a trigger to change.

This paper presents LocalSEA, a local robotic testbed for multiple purpose experimentations to develop and evaluate research contributions for Industry 4.0. It is under the development and management of LSEA (Embedded and Autonomous Systems Design Laboratory), CEA List. In a long-term strategy, the testbed will progressively extend with different robots and devices. However, the testbed's development relies unchangeably on the SysML standard in the specification and design phase and the OPC UA standards in the implementation phase. SysML, standing for Systems Modeling Language, is an architecture modeling language based on UML and dedicated to System Engineering applications [1]. SysML is developed and maintained by Object Management Group (OMG). Its newest version, v1.6, provides nine diagrams to describe a system from different points of view, thus, enabling participants of system development to understand, communicate, and exchange between them.

OPC UA, standing for Open Platform Communication Unified Architecture, is a well-known and widely-adopted standard in the industry. It is developed and maintained by the OPC foundation. To date, OPC UA's newest version, v1.05, consists of more than twenty specifications for deploying industrial systems with interoperability, security, and robustness [2]. One of the essential advantages of OPC UA is the concept of the OPC UA address space and information model. First, the mechanism of OPC UA address space allows representing all the resources of an industrial system in the form of OPC UA nodes [3]. Second, the OPC UA information model provides a basic vocabulary to define OPC UA nodes and to build a semantical schema for them [4]. This vocabulary is extensible: the companions of the OPC Foundation can extend it by providing a new information model with the vocabulary for a specific domain. As a result, OPC UA address space and information model improve semantic interoperability: external systems can browse and access the resources of an OPC UA system through the OPC UA nodes using OPC UA services.

Technically speaking, the LocalSEA testbed includes an OPC UA server managing an information model representing all internal resources, such as robots, peripherals, and networks. This information model relies mainly on three information models: the base OPC UA information model part 5, the OPC UA for Devices information model part 100 [5], and VDMA OPC UA for Robotics information model [6]. All devices inside the testbed connect respecting the OPC UA communication and security standards.

The motivation of this research is to share our early lessons learned in developing the LocalSEA testbed. First, we introduce a strategy for a long-term testbed development: to use the mini-waterfall with triggers methodology. This new methodology is dedicated to testbeds since they can be updated frequently with new devices, protocols, and development tools demanded by new use cases. Second, this research also presents some obstacles specifying OPC UA-based testbed development that can be helpful experiences for the industrial research community.

The rest of this paper is organized as follows. The second section presents some other research about testbeds in Industry. The third section introduces mini-waterfall with triggers, the methodology used in LocalSEA development. The fourth section is a sample of using the mini-waterfall with triggers methodology in developing the testbed. The fifth section discusses our encountered difficulties and future works. Finally, a brief conclusion sums up this paper.

II. Related Work

Many industrial testbeds and platforms to develop testbeds and case studies address Industrie 4.0 [7]. They are well-maintained and developed by credible organizations. This paper focuses on sharing the experience and strategy of developing a long-term testbed rather than presenting another. The study of Frank et al. has some commons with this research [8]. First is the distinguishing between the development of a testbed and a use case. Second, it presents a new methodology to design a testbed. However, they are different in the domain, the development strategy, and the testbed's core technology—the OPC UA standard.

The testbed presented by Okuda et al. uses the OPC UA communication standard for the system network and an OPC UA information model to represent the resources needed for its use cases [9]. However, this work has no clear strategy to develop its testbed for new changes.

III. Mini-Waterfall with Triggers

The methodology to develop the LocalSEA testbed is mini-waterfall with triggers. Mini-waterfall inspired by the well-known waterfall methodology also contains fives phases: (1) specification, (2) design, (3) implementation, (4) testing, and (5) maintenance [10]. However, the system's developers with the mini-waterfall methodology divide their system's work into small jobs. Each job should have a clear goal. They repeat the sequence of the five phases for each job to progressively develop the system. In this paper, a repeat of mini-waterfall is called sprint, as in the scrum/agile methodology [11]. System developers have two solutions when they find an error in one of the previous phases. The first solution is to note the error, temporarily ignore it and finish the current sprint, then fix it in the next sprint. The second solution is to jump back to the phase with that error, fix it, and refollow the sprint. The first solution is encouraged. The mini-waterfall with triggers use triggers as a point to start a sprint. Figure 1 illustrates the methodology mini-waterfall with triggers.

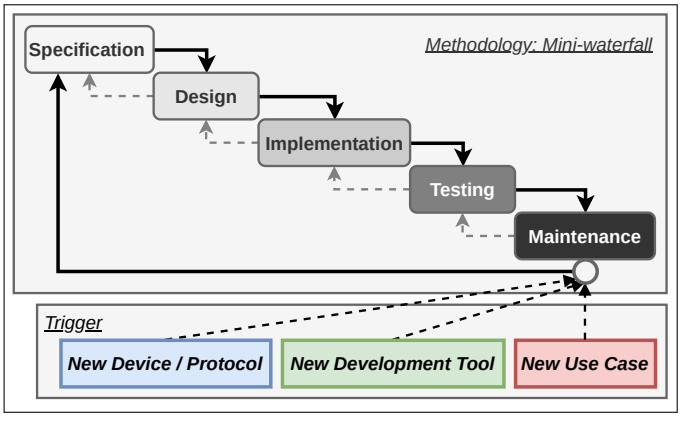

Fig. 1. Testbed development methodology: Mini-waterfall with triggers

Three following triggers is specified for the development of a robotic testbed.

- First, the trigger activates a sprint when deploying a new device or a protocol/standard in a testbed. It is necessary to improve the information model to represent new resources from the device or the protocol. Also, system developers must apply new programs or configurations into the testbed.
- Second, the trigger activates a sprint when using a new development tool. The tool must change one or several development activities and become part of the development toolset.
- Third, the trigger activates a sprint when implementing a new use case. The use case may require adding, adjusting, and modifying the testbed, such as improving the information model with new view nodes corresponding to the required data of the use case.

The result of each sprint should be a testbed's new version and a new report that presents the procedure to develop and the guide to use the testbed.

IV. Development Sample: The First Sprint

The first sprint is also the start point of the testbed development. This sprint has three main jobs: (1) install an OPC UA server on a Raspberry Pi[1] 3 Model B+, (2) install Niryo Ned[2], a robotic arm, and (3) deploy an OPC UA transport profile for the communication between the OPC UA server and the robot. The trigger, in this case, is a new robot and a new protocol. In this first sprint, we only considered the OPC UA standard for the specification, design and implementation of the system. SysML standard will be used in the second sprint for the specification and design phases.

In the specification phase, our developers' team studied the following documents: the technical specification of Niryo Ned, OPC UA specifications, the OPC UA for robotics companion specification, and information about current conditions and future orientations of LocalSEA. The result is a specification document synthesizing essential information for the first sprint.

In the design phase, our developers' team first listed possible solutions for the communication inside LocalSEA. The selected solution is OPC UA PubSub in the brokerless mode. Since the publish-subscribe communication pattern fits the Internet-of-Things (IoT) requirements, the testbed is also ready for Industrial IoT use cases in the future. Figure 2 illustrates the architecture of LocalSEA, in which an OPC UA server ran on the Raspberry Pi manages an address space representing the resources of Niryo Ned. Niryo Ned plays as a publisher, and the OPC UA server plays as a subscriber. The users outside LocalSEA can interact with this testbed through the server. Figure 3 shows essential nodes of the information model of LocalSEA. Most of the types in this information model are basically from the base OPC UA information model part 5, the OPC UA for Devices information model part 100, and the VDMA OPC UA for Robotics information model. Only instances and some new specific types are newly created. The tool used for design is UaModeler[3].

[1]https://www.raspberrypi.com/

[2]https://niryo.com/product/ned/

[3]UaModeler is an OPC UA information model designer developed by Unified Automation GmbH. The tool provides a user-friendly design to design information models and generate them into *NodeSet2.XML* format. However, it requires purchasing a license for the full-featured of the latter function.

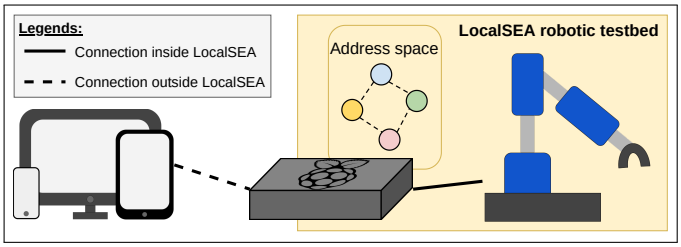

Fig. 2. The architecture of the LocalSEA testbed after the first sprint

Our developers implemented the OPC UA server using the Open62541[4] library. The OPC UA PubSub UDP UADP transport profile[5] and the None security profile[6] realized in a C-codes program allow Niryo Ned to exchange with the server. The manually-made information model in the *NodeSet2.xml* format was converted into C codes using the NodeSet Compiler tool. This tool is also proposed by Open62541.

Since this is the first sprint, it lacks data and scenarios to verify if the testbed can work stably in different conditions. Thus, the verification phase is only about testing whether the other users outside the system can connect and retrieve data from Niryo Ned. One tool to test the information model of the OPC UA server is UaExpert[7].

In the maintenance phase, our developers' team records every feedback from the testbed's users and resolves their issues. Note that the testbed's users of this sprint are only colleagues in our research team.

V. Discussion and Future Works

Figure 4 shows the development plan for LocalSEA, with five sprints. The first sprint, presented in Chapter IV, terminated. It has two main jobs: (1) installing a new Niryo Ned robot and (2) deploying the OPC UA PubSub brokerless network profile. Some remarks learned from this sprint are as follows. First, the Robotics information model is still a developing specification. Indeed, its current vocabulary can cover only robotic arms but no other types of robots, such as mobile robots. Also, the concepts of some robotic parts, for example, the end effectors of a robotic arm, are missing. Thus, the Robotics information model users must wait for a new version or extend it with some new custom-defined vocabulary. Second, the concept of OPC UA views helps distinguish the use cases deployed in the same testbed simultaneously. Since the information model of an OPC UA server represents all resources available in the testbed, an OPC UA view enable users to focus on a subset of OPC UA nodes.

The second sprint is currently in progress. In detail, a new extension for the Eclipse Papyrus[8] enables to specify and design a system using SysML, then to add OPC UA specific concepts on top of the SysML models in order to automate the generation of the OPC UA information model. Consequently, this extension proposes a UML profile for the design of the Robotics information model. Since the manual serialization of an information model is quite a painful job, Papyrus generates automatically the *NodeSet2.xml* file from the models. The generated information models are compatible with the Robotics information model. This tool is essential and helpful for our testbed's development.

After finishing the second sprint, there will be the third sprint. In detail, TurtleBot3 Waffle Pi[9], a mobile robot, will join the testbed system. Until the beginning of this sprint, if there is no update in the Robotics information model, our developers' team will design a custom-defined vocabulary for this robot. The plugin of Eclipse Papyrus of the second sprint will generate a new information model for the testbed.

The fourth sprint concerns the design of simple robotic cell scenario orchestration, including the Niryo Ned, the TurtleBot3 Waffe Pi, and a human operator. The orchestration process will be described in Node-RED[10], this allows the whole system deployment in a single click. At this stage, the relevant algorithms will be set up on the robots to complete the task required by the scenario.

Then the establishment of a digital twin system will gradually be achieved in the final sprint. The Eclipse Papyrus provides an AAS (Asset Administration Shell) modeling environment for manufacturing, where one can model the production assets and communicate with the physical device via the OPC UA server. The capability-based engineering process will be integrated into Papyrus in the future, which provides the management of reconfiguration and automated operation of flexible production lines. The AAS models, physical assets, and their bijective communications form a digital twin system that ensures the monitoring, analysis, simulation, and deployment of the Cyber-Physical system. During this sprint, more complex use cases will be designed and tested to demonstrate the controllability of the digital twin system in different scenarios.

VI. Conclusion

This work-in-progress paper presents the development strategy and early lessons learned from developing LocalSEA, an OPC UA-based robotic testbed for multipurpose research and education. One strategic principle is to use our new-defined testbed development methodology called mini-waterfall with triggers, in which a trigger can be a new device/protocol, a new development tool, or a new use case. The testbed uses an information model based on the base OPC UA information model part 5, the OPC UA for Devices information model part 100, and the VDMA OPC UA for Robotics information model. Since the Robotics information model is still a developing specification, it shows some omissions needed to improve: (1) it cannot cover other types of robots except robotics arms, and (2) the missing vocabulary for some robotic parts, such as end effectors. Also, this paper outlines some of our future works promising to be helpful to the industrial research community. First, we will present a new extension for Eclipse Papyrus to

[4]https://open62541.org/
[5]http://opcfoundation.org/UA-Profile/Transport/pubsub-udp-uadp
[6]http://opcfoundation.org/UA/SecurityPolicy#None
[7]UaExpert is another tool of Unified Automation GmbH. It is a full-featured OPC UA Client for general purpose tests.
[8]https://www.eclipse.org/papyrus/
[9]https://www.robotis.us/turtlebot-3-waffle-pi/
[10]https://nodered.org/

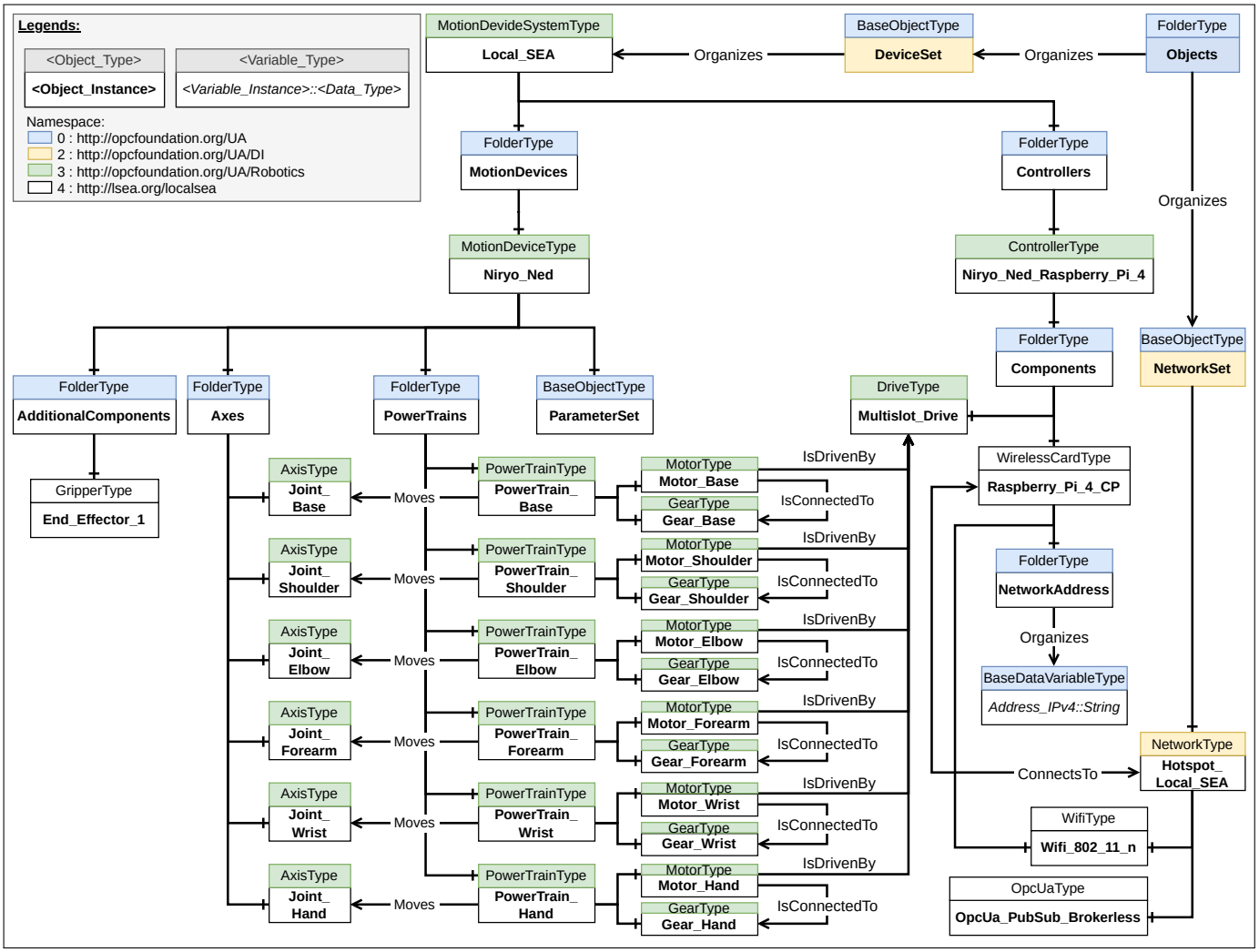

Fig. 3. OPC UA Information Model after integrating Niryo Ned and PubSub UDP UADP protocol into the LocalSEA testbed

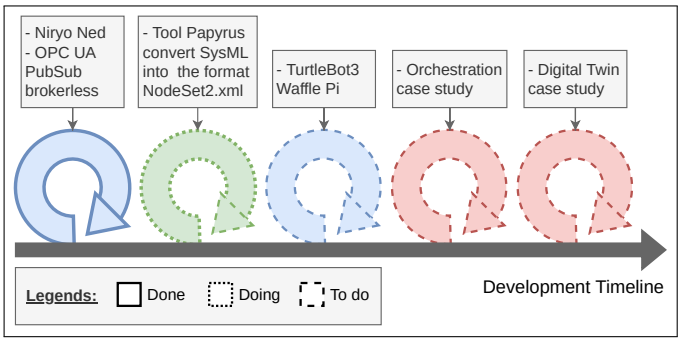

Fig. 4. Development plan for the LocalSEA testbed

design and generate OPC UA information models based on SysML. Second, two planned use-cases address the robots' orchestration and the digital twin.

References

[1] M. Hause, "The sysml modelling language," in *Fifteenth European Systems Engineering Conference*, vol. 9, September 2006, pp. 1–12.

[2] OPC Foundation, "OPC Unified Architecture - Part 1: Overview and Concepts," Industry Standard Specification OPC 10000-1, 2017.

[3] OPC Foundation, "OPC Unified Architecture - Part 3: Address Space Model," Industry Standard Specification OPC 10000-3, 2017.

[4] OPC Foundation, "OPC Unified Architecture - Part 5: Information Model," Industry Standard Specification OPC 10000-5, 2017.

[5] OPC Foundation, "OPC Unified Architecture - Part 100: Devices," OPC Foundation, Industry Standard Specification OPC 10000-100, 2021.

[6] Mechanical Engineering Industry Association (VDMA), "OPC UA for Robotics - Part 1: Vertical Integration," VDMA, Industry Standard Specification VDMA 40010-1, 2019.

[7] C. Koch and K. Blind, "Towards Agile Standardization: Testbeds in Support of Standardization for the IIoT," *IEEE Transactions on Engineering Management*, vol. 68, no. 1, pp. 59–74, February 2021, conference Name: IEEE Transactions on Engineering Management.

[8] M. Frank, M. Leitner, and T. Pahi, "Design Considerations for Cyber Security Testbeds: A Case Study on a Cyber Security Testbed for Education," in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*. Orlando, USA: IEEE, November 2017, pp. 38–46.

[9] M. Okuda, T. Mizuya, and T. Nagao, "Development of IoT testbed using OPC UA and database on cloud," in *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, September 2017, pp. 607–610.

[10] P.-A. Muller and N. Gaertner, *Modélisation objet avec UML*. Paris: Eyrolles, 2004.

[11] K. Schwaber and M. Beedle, *Agile software development with Scrum*, ser. Series in agile software development. Upper Saddle River, NJ: Prentice Hall, 2002.