Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-10 (2022) 773–778

Streamline 3D simulation model development for virtual commissioning with IEC61499

Diego Rovere*,, Marco Silvestri**,*, Giovanni Dal Maso***, Hilmo Dzafic****, Paolo Pedrazzoli*.**

** Department of Innovative technologies, University of Applied Sciences and Arts of Southern Switzerland, 6900 Lugano, Switzerland, (e-mail: diego.rovere@supsi.ch, marco.silvestri@supsi.ch, paolo.pedrazzoli@supsi.ch)*

*** Department of Engineering and Architecture, University of Parma, 43124 Parma, Italy, (e-mail: marco.silvestri2@unipr.it, diego.rovere@unipr.it)*

**** Technology Transfer System s.r.l., 20131 Milan, Italy, (e-mail: dalmaso@ttsnetwork.com)*

***** nxtControl GmbH, 2544 Leobersdorf, Austria (e-mail: hilmo.dzafic@se.com)*

Abstract: Current discrete manufacturing systems are characterized by an ever-increasing complexity, demanding for innovative solutions, capable to optimize performances while increasing the resilience and the capability to adapt to production modifications. With such a background, changing perspective to deal with distributed modular architectures of Cyber Physical Systems is mandatory, and the IEC 61499 standard, its object oriented, and event-based approaches promote this paradigm shift. The multi-disciplinary nature of the CPS entities and the possibility to exploit their digital counterparts, paves the way for the development of enhanced decision-support systems like the ones dedicated to Virtual Commissioning (VC). VC supports the automation developer in evaluating the impact of different management strategies, increasing the reliability of the final control applications, while reducing the amount of time to carry on physical tests on the real mechatronic system. However, creating a virtual commissioning model is still a complex and potentially expensive process that needs to be carried out by different professionals who must tightly cooperate to generate an effective playground for the automation testing. We propose a new approach to the design and develop virtual commissioning models, that, leveraging the synergies between modular simulation and IEC 61499 automation technologies, aims at improving the efficiency of the overall process of implementing 3D simulation digital twins for complex automated discrete manufacturing systems. The paper describes an open architecture, composed of reference data models and software API, and presents a proof-of-concept implementation of an integrated engineering platform of VC models.

Keywords: Virtual Commissioning, Simulation, IEC61499, CPS, Distributed Automation

1. INTRODUCTION

Current discrete manufacturing systems are characterized by an ever-increasing complexity, mainly due to the quick change of demand and to the request for highly customized multi-components products (Mazzolini et al., 2018). Device, machine tool and plant builders must promptly react to the modifications of the surrounding environments, adapting their products to meet the customer needs, that more than ever move in the direction of using reliable and multi-purpose automated systems. It has been demonstrated that the real breakthrough takes place thanks to the improvement of the automation and, in general, of the digital facets of the industrial products (Cheng, 2017). Industry 4.0 and its whole ecosystem of methodologies and technologies lay their basis on the assumption that no future development is possible without a tight integration between the mechatronic system and its digital counterpart (Osterreider, 2020). Within this context the simulation represents a fundamental component of the new automation pyramid because it exploits at maximum level the digital nature of the CPS, bringing a great number of benefits at design and operation time. The need to implement industrial products with a "first time right" approach stresses the automation development procedures so that if on one side the complexity of the work increases, the time to produce optimized and reliable customized devices decreases. With such a background, changing perspective, from a centralized classical approach to the distributed modular one, is mandatory, in order to exploit tools and methodologies that support the control engineers in taking decisions, reducing the overall burden of the operational planning. This kind of architectural approach is strongly promoted by the IEC 61499

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2022.09.503

standard, which replaces the old concept of monolithic program, written in a language compliant with the standard IEC 61131, with the concept of Application which is composed of hierarchical bricks called Function Blocks (FBs) and that can be dynamically distributed at runtime on multiple Resources (Vyatkin et al., 2009 and 2011). Nevertheless, the problem of obtaining a *zero-defect* behavior in a short implementation time, particularly for large systems, is attenuated but not removed, as the automation solution must be still tested and accurately debugged, to assess the impact of the operating strategies on the production. This evaluation process often generates inefficiencies on the overall development of a manufacturing system because it requires expensive and time-consuming tests on the real devices. To this purpose, the adoption of a Virtual Commissioning (VC) methodology, based on the connection of the automation with a simulation model reproducing the reactions of the real environment, can improve the efficiency of the validation phase (Cevican et al., 2018) (Rosen et al., 2016). VC, in fact, can significantly reduce the amount of time needed to carry on physical tests on the mechatronic system because they provide the means to perform off-line strategy debugging and optimization sessions, anticipating decisions (Um et al., 2017), widening the possibility to consider multiple operating scenarios and parallelizing key activities. If the benefits of the approach are evident, it is important noting that currently creating a VC model is still a complex and potentially expensive activity, where the automation developer and the simulation expert must closely interface to produce an effective model. This cooperation is affected by some inefficiencies that are mainly related to the lack of integration between engineering platforms, which determine long lasting and error prone knowledge transfers from the automation developer to the simulation expert. This impacts on the possibility for a VC model to rapidly align with the real behavior when the underlying mechanical system is evolving quickly. Moreover, with the traditional approach, the communication of the world of automation with the simulation one is realized only at runtime level, so that until execution time there is no relation between the two contexts. In this way, the development process tends to become almost vertical, with the result that the conceptual structure of the control logics can differ significantly from the organization of the digital counterpart. This silo effect of the artifacts building process mirrors and emphasizes the inefficiencies.

Background

At runtime, most of the architectures currently available for VC rely on widely accepted interfacing standards for signals exchange in order to integrate the controlling logics with the simulation models. The automation solution is treated no more than a black box that generates output signals controlling actuators and receives input signals provided by sensors. It is a quite rigid paradigm, whose quasi-unique variant is represented by the type of connection established, that on its turn, normally is limited by the availability of connectors within the PLC. Each platform provides its own set of connectivity solutions that range from purely proprietary (e.g., the Fanuc Focas2 libraries) to completely standard ones (e.g., OPC-UA) but none of them tends to cope with the way the automation solution is structured, they only rely on the I/O maps. However, the possibility to run distributed VC sessions is strongly limited by this approach and only few experiences report encouraging results about the possibility to perform multi-sided communication between nodes of the same control solution and all of them are based on the IEC 61499 standard (Santos et al., 2021). A good example is provided as an extension to the work (Mazzolini et al., 2017), where the validation of the optimized control of a distributed system is interfaced with a simulation model based on Simio platform. An attempt of exploiting the modularity of the IEC61499 standard and transfer it to an FMI (Functional Mockup Interface) based simulation environment described in (Cabral et al., 2018)). More automation-oriented research is present in (Xavier et al., 2021) where the formal verification of the IEC 61499 solution is performed with simulation in the loop (SIL), but the cited simulation is constituted by Function Blocks and doesn’t work with an external 3D simulation environment. One of the most promising works for performing the collaborative VC for PLC validation is represented by the approach of (Lyu et al., 2021). All the reviewed approaches share the focus on the integration at runtime of the deployed control applications, and the same does most of the literature review related to the topic of VC on IEC 61499 platform (Park et al., 2014). From the commercial point of view, instead, there exist solutions that integrate development environment belonging to different knowledge domains to create multi-perspective engineering platforms (an example is the Siemens Tecnomatix) However, all the of them behave like closed ecosystems, dedicated to specific hardware and none of them complies with the IEC 61499 standard. This analysis of the state-of-the-art highlights how the field of integration of software tools for the IEC 61499 needs to be explored, looking for solutions that are vendor independent and open for adaptation to multi-disciplinary scenarios to promote the development of real smart factories.

1.1 Objectives

The main objective of the work described in this paper is the engineering of a new approach to the design and development of VC models for complex automated discrete manufacturing systems, to improve the efficiency of the overall implementation process. The proposed architecture, leveraging the synergies between modular simulation and automation technologies, aims at reducing the required interaction between competences, increasing the level of independence of the automation engineer, and overcoming some of the cited limitations. The foundation of the work is established on the capability of the IEC-61499 standard to orchestrate CPS hierarchies relying on the concepts of object-orientation, so that that the modularity and reconfigurability of the mechatronic product coincides to that of the software governing them. This means that the same organized and scalable methods can be applied in the concurrent design of the digital models mirroring the control system, promoting the development of a new generation of CPS entities whose simulation counterpart not only exist and operate at production time but streamlines the decision making of the automation developer.

The tangible result is the full integration between the behavioral models of CPS devices with their IEC-61499 functional architecture, enabling a complete and seamless connection between the information flows originating through the sensing and acting capabilities of the CPS itself on one hand, and the data structure of the simulation model on the other hand, independently from where they reside and transparently to the engineers' efforts. This objective can be achieved only embracing the problem space from the perspectives of two main domains that compose a VC model: the system engineering and its runtime execution. In fact, both at automation and at simulation sides the concept of separation between model in preparation and model in execution is strongly present and mainly arises from the big difference between the requirements of design time formats, that need to retain all the source information for continuous editing, and the requirements for the runtime artifacts, that usually need to be optimized for performance and resource consumption, i.e. compiled.

The runtime facet of the VC domain is based on the live and quasi real time connection between the automation and the simulation engines and deals with the objectives of ensuring a high-throughput data exchange among the artifacts. This approach mimics the physical exchange of low level I/O signals between the automation solution and the real set of sensors and actuators of the device hardware. Therefore, at this level, the research aimed at developing a high performance and secured I/O channel that leverages the two fundamental features of the IEC 61499 standard: the object-orientation and the event-based paradigm. The main expected benefits are a significant reduction of the configuration burden and the improvement of the support of the communication architecture to distributed cyber physical systems.

The engineering aspect, first in terms of production workflow, represents also the most difficult context to deal with. The nature of the software environments involved in the engineering phase of the cyber-commissioning are completely different and share only a similar, elevated degree of complexity. The objective of this research is therefore study and implement a proof of concept of an integrated engineering platform composed by software tools instrumented to cooperate for the joint production of "virtual-commissioning-ready" digital twins. Achieving such result requires the design of an open architecture made of a reference data model and a set of software API that, once deployed, manages the communication between the applications.

The activities have been carried out in the context of Daedalus European Research Project, and its natural follow up, the H2020 1-SWARM European Research Project. For this reason, the paper presents the design of platform-independent solutions, characterized to be openly accessible and applicable virtually to any third-party commercial application, but each aspect has been implemented in a set of proof-of-concept prototypes based on the software platforms available in the research projects consortium: the nxtStudio platform provided by NxtControl GmBH for the IEC 61499 automation side and DDD Platform provided by Technology Transfer System s.r.l. for the simulation side.

3. PROPOSED ARCHITECTURE

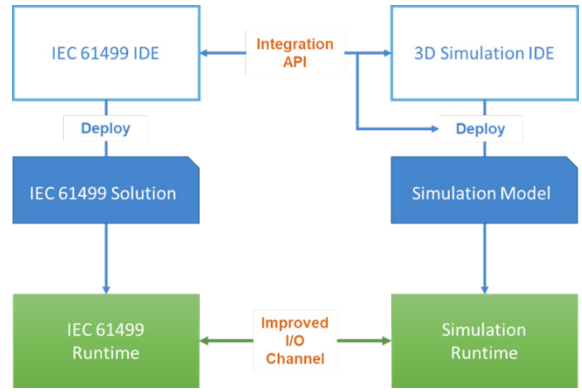

Figure 1 Proposed architecture evolution

Figure 1 shows from high level perspective the points (orange) where the software components developed during the research activities have brought a progress behind the state of the art. The main interventions targeted both levels of the architecture: at design level, filling the gap between the automation and the simulation IDEs, implementing a real integration between the two environments through the definition of an Integration API; at runtime level, improving the I/O runtime communication to comply with the distributed approach of the IEC 61499 and to provide an effective data stream satisfying the requirements to execute reliable virtual commissioning scenarios

4. RUNTIME INTEGRATION

IO Data Model

The IO data model formalizes the data structures that organize the exchange of input and output signals among the automation and the simulation runtimes. The data model has been organized with the aim to preserve the object-oriented approach which is a fundamental aspect of the IEC 61499 standard. In IEC 61499, in fact, each signal is related to events of a well identified Function Blocks, that typically represent logics dedicated to handle a device or a sub assembly of it. Figure 2 provides a UML class diagram of the data model.

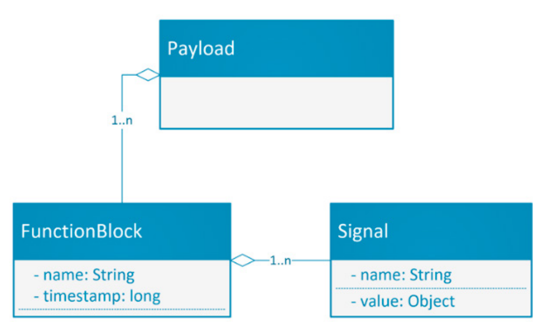

Figure 2 UML Class Diagram of the I/O Data Model

JSON Data Model Implementation

The data model presented from a formal point of view has been transformed into JSON (JavaScript Object Notation) open standard format to serve as a reference implementation of the message content. The choice of JSON is related to its wide acceptance in socket communication for the encoding of the payloads, its low overhead and wide availability of serialization libraries. The data structures defined in the UML have been adapted and optimized to limit as much as possible the size of the messages, reducing the footprint of the communication protocol on the network. The final format of the payload, expressed in JSON is the following:

```
{
	[FunctionBlock_ID]:{
		"ts": [clock_value],
		"Param":{
			[Signal_ID]: [Signal value],
			[Signal_ID]: [Signal value],
			…
		},

	}
}
```

Communication over MQTT

The evolution of the runtime communication has been implemented using MQTT (Message Queue Telemetry Transport ISO standard) as transport infrastructure. MQTT promotes a complete decoupling among the actors of the communication, in fact the publisher doesn't need to know anything about the subscribers, their presence, location or number and vice-versa. This approach significantly simplifies the management of the low-level sockets, which can be turned on and off without affecting the integrity of the other components connected to the same broker. The latter feature, together with the small footprint and the capability to support large bandwidth, makes the MQTT protocol a suitable technology to implement an evolution of the VC runtime communication. The advantages are multiple: the complete transparency of each client to the other ones connected to the same broker, allows each engine, automation or simulation, even in multiple instances to be configured to address only the broker, to produce messages that will be automatically broadcasted to different endpoints and to consume information flowing from different devices, whose location can remain completely unknown (see Figure 3). It is evident how such solution complies with the distributed nature of IEC61499.

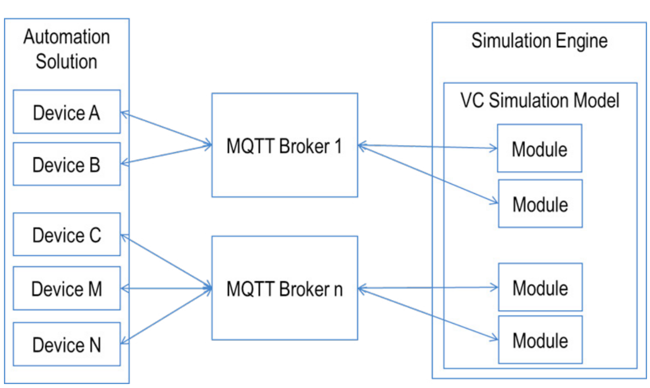

Figure 3 Scalability of the MQTT approach with multiple brokers

From the usability and stability perspectives, the broker is the sole component needed to initiate a VC session, while the automation or simulation engine can connect and start publishing/consuming messages in a completely asynchronous way. Moreover, the implementations of the MQTT brokers are typically characterized by a very high level of resilience, standing continuous connections and disconnections of clients without impacts on the quality of service. This makes the overall architecture not only more fault tolerant but also simpler to control during the execution of the VC tests and easily scalable to different deployment scenarios on both sides of the virtual commissioning actors. Finally, the architecture is extremely interesting also from a load balancing perspective because allows selecting the correct distribution of the workload on the supporting hardware network, making it suitable for large plants, where the amount of function blocks and simulation modules interacting during the same virtual commissioning session is particularly high.

MQTT Topics structure

The JSON implementation of the IO Data Model does not contain any specification of the signal direction: no distinction between input and output signal is done at message payload level, therefore, in order to distinguish the semantics of the signals, the hierarchical structure of the topics within the MQTT broker has been exploited. The topics have been organized to reflect the flow direction of the signal messages and the identifiers of the connected simulated and controlled devices. In particular the identifier of the topic is composed as follows:

VirtualCommissioning/{System_name}/{module_id}

6. DESIGN -TIME INTEGRATION

The interface between the authoring environments is composed of two main complementary components: The Digital Avatar Data Model and the integration API. The proposed solution, relying on cross-platform technologies can be adapted to any IEC 61499 automation IDE and any modular simulation engine

The Digital Avatar Data Model

When configuring the simulation entities to behave like the real devices, it is necessary to wrap a pure digital behavior into a container capable to mimic the interfaces of the physical units. This means defining artifacts capable of consuming the same signals of the physical systems, elaborating a reaction and generate the same responses. The digital avatar data model aims at defining the structure of the digital counterparts of the IEC 61499 Function Block, composing its interface with its internal logics, supporting the parametrization of the whole entity. From an architectural perspective, these objects formalize a set of configuration parameters, a set of I/O endpoints corresponding to the automation I/Os and set of logical tasks that must be executed to react to external or internal events. Figure 4 shows the UML class diagram of the developed model.

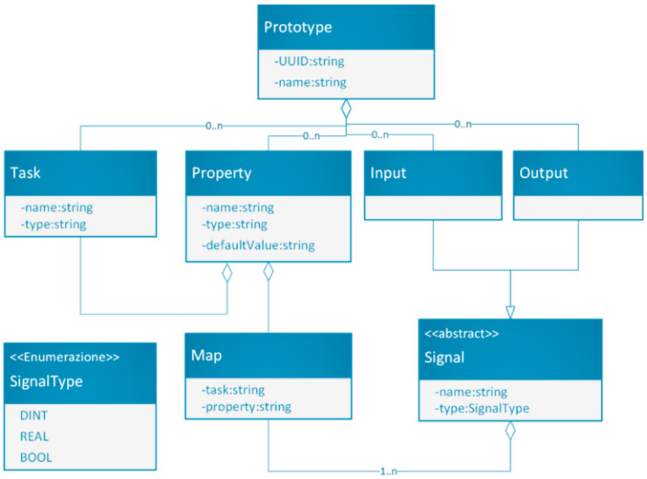

Figure 4 Digital Avatar Data Model UML Class Diagram

The Design Integration API

The Integration API layer represents the means by which the two applications collaborate to co-design the VC solution. Through this layer, the actions performed by the automation expert induce automatic modifications of the simulation model that in this way is kept in synch with the IEC 61499 logics. The Design Integration API can be considered the dual of the Runtime Communication Layer and the natural completion of the upstream development process. Nevertheless, the two interfaces are meant to support two so different phases that the requirements for each one are really different. The design API supports operations happening at slow rate, triggered by the end user actions, therefore it does not demand for the same high speed and low latency required by the runtime communication. On the contrary, it requires the exposure of high-level functionalities, identifiable with actions or methods of a high-level software interface. For this reason, the choice of the communication protocol has been based on criteria that consider the capability to efficiently support complex operations in a cross-platform approach. The best integration technology for this specific use case is the represented by gRPC, which is a high performance, open-source universal remote procedure call framework, based on Protocol Buffers.

The gRPC API

An analysis of the use cases related to the authoring phase led to the formalization of functions and data structures of the gRPC IDL using the Protocol Buffer version 3 syntax. The following table shows a synoptics overview of the main rpc signatures and the corresponding messages that have been defined, grouped according to the target object.

Table 1 Design Integration API methods

Procedure name	Input messages	Output message
CreateProject	CreateProjectRequest	ProjectHandle
OpenProject	OpenProjectRequest	ProjectHandle
CloseProject	ProjectHandle	Result
DeleteProject	ProjectHandle	Result
CreatePrototype	CreatePrototypeRequest	ResourceHandle
DeletePrototype	ResourceHandle	Result
GetPrototypes	ProjectHandle	ResourceList
CreateInstance	ResourceHandle	CreateResult
DeleteInstance	ResourceHandle	Result
GetInstances	ProjectHandle	InstanceList
SetInstanceProperty	SetInstancePropertyRequest	Result
SetSignalInstance	SetSignalInstanceRequest	Result
CompileProject	ProjectHandle	Result
RunProject	RunProjectRequest	Result
StopProject	ProjectHandle	Result
QueryProjectRunning	ProjectHandle	Result
BeginTransaction	ProjectHandle	Result
CommitTransaction	ProjectHandle	Result
RollbackTransaction	ProjectHandle	Result

User workflows

This section described a complete workflow with the integrated system, analyzing the major phases. The sequence diagrams are presented, in form of UML sequence diagrams, from the Automation Engineer perspective using a generic IEC 61499 IDE, to describe which is the envisioned user experience and how it reflects on the data exchange between environments. The messages flowing between lifelines correspond to gRPC calls.

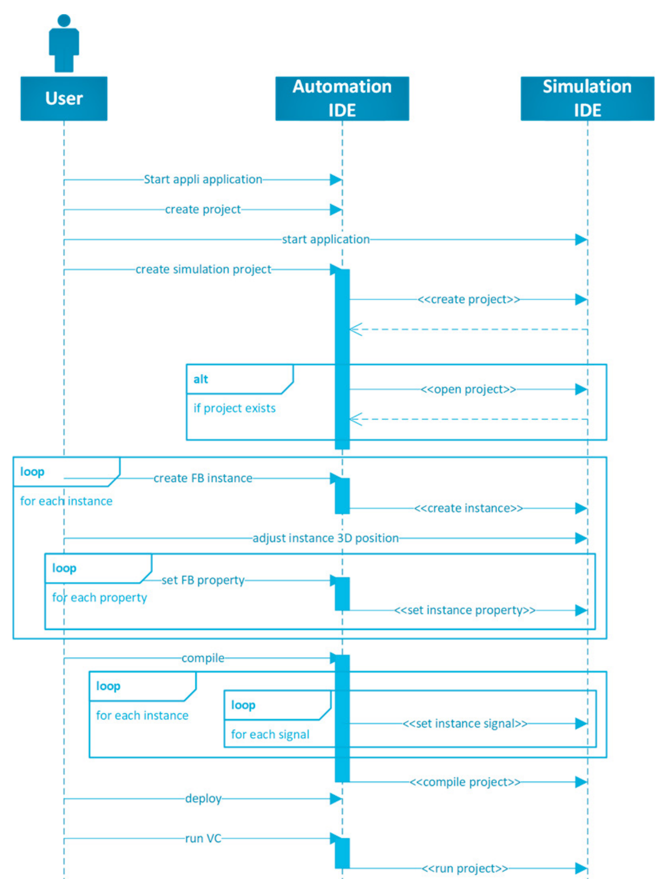

Figure 5 Startup and creation workflow with gRPC API

Start up and creation

In this sequence, (see Figure 5) the user starts the IDE applications and then, following the natural engineering process at automation side, creates a new project, adding as many CPS instances as needed and setting their parameters, and finally compiles, deploys, and runs the project. The Design Integration API intervenes during the compiling and running phases, when the corresponding simulation model is transparently built, deployed, run, and connected to the control application, giving the end user the possibility to immediately execute VC operations.

Stop and close

In this sequence (see Figure 6) the user stops a running project, closes it, and exits the application. This workflow is very simple, and it is meant only to highlight the fact that the interaction between IDEs involves the management of cleanup operations to ensure the correct release of the system resources.

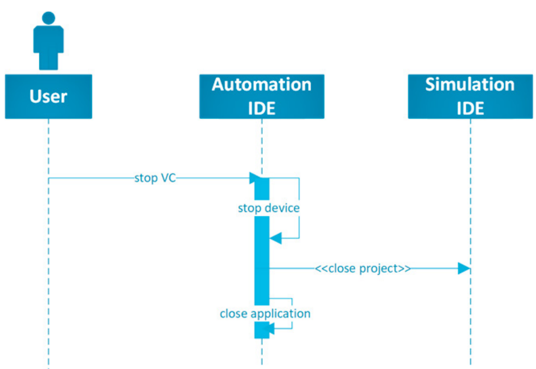

Figure 6 Stop and close workflow

Destroy

In this sequence (see Figure 7) the end user deletes an instance and, if the originating prototype is not used anymore, the automation IDE is in charge of removing the prototype from the simulation project, to allow its garbage collection.

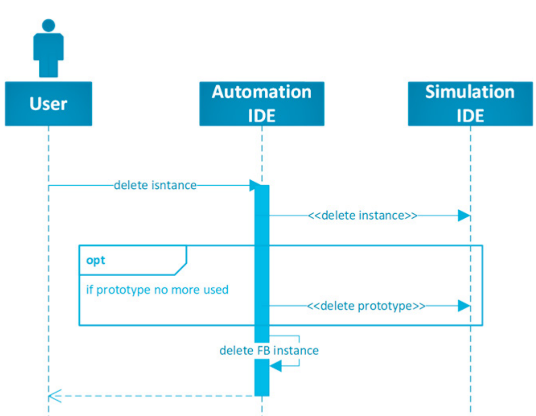

Figure 7 Destroy workflow

REFERENCES

Santos, A., Silva, S. (2021). Simulation and Control of a Cyber-Physical System under IEC 61499 Standard. *Procedia Manufacturing*, volume 55, pp. 72-79.

Cevikcan, E., Ustundag, A. (2018). Industry 4.0: Managing The Digital. *Cham: Springer International Publishing*.

Sheng, F.Q., Cheng, K. (2017). Future Digital Design and Manufacturing: Embracing Industry 4.0 and Beyond. *Chinese J. Mech. Eng.*, volume 30(5), pp. 1047–1049.

Cabral, J., Wenger, M., et al. (2018). Enable co-simulation for industrial automation by an FMU exporter for IEC 61499 models. *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 449-455.

Um, J., Weyer, S., et al. (2017). Plug-and-Simulate within Modular Assembly Line enabled by Digital Twins and the use of AutomationML. *FAC-PapersOnLine*, volume 50(1), p. 15904–15909.

Mazzolini, M., Cavadini, F.A., et a. (2017). Structured Approach to the Design of Automation Systems through IEC 61499 Standard. *Procedia Manufacturing*, volume 11, pp. 905-913.

Mazzolini, M. (2018). Engineering Support System for Sustainable Optimization of Automation Task Supervision, *Università degli studi di Bergamo*

Xavier, M., Patil, S., et al. (2021). Cyber-physical automation systems modelling with IEC 61499 for their formal verification. *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, pp. 1-6.

Osterrieder, P., Budde, L., et al. (2020). The smart factory as a key construct of industry 4.0: A systematic literature review. *International Journal of Production Economics*, volume 221.

Lee, C.G., Park, S.C. (2014). Survey on the virtual commissioning of manufacturing systems. *Journal of Computational Design and Engineering*, volume 1(3), pp. 213-222.

Boschert, S., Rosen, R. (2016). Digital Twin—The Simulation Aspect. Mechatronic Futures, Cham: *Springer International Publishing*, pp. 59–74.

Lyu, T., Atmojo, U.D., et al. (2021). Towards cloud-based virtual commissioning of distributed automation applications with IEC 61499 and containerization technology. *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1-7.

Vyatkin, V. (2009). The IEC 61499 standard and its semantics. *IEEE Industrial Electronics Magazine*, volume 3(4), pp. 40-48.

Vyatkin, V. (2011). IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review. *IEEE Trans. Ind. Informatics*, volume 7(4), pp. 768-781.