Seilonen, Ilkka; Vyatkin, Valeriy; Atmojo, Udayanto Dwi

OPC UA information model and a wrapper for IEC 61499 runtimes

Published in:
Proceedings of the 17th IEEE International Conference on Industrial Informatics, INDIN 2019

DOI:
10.1109/INDIN41052.2019.8971964

Published: 01/07/2019

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

OPC UA Information Model and a Wrapper for IEC 61499 Runtimes

Ilkka Seilonen[1], Valeriy Vyatkin[1,2], Udayanto Dwi Atmojo[1]
[1]Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland
[2]Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden

{ilkka.seilonen, udayanto.atmojo}@aalto.fi, vyatkin@ieee.org

Abstract — **OPC UA is considered as an important communication technology for Industry 4.0 components. On the other hand, IEC 61499 is one of the technologies available for developing applications to the same components. In this paper, a proposal for an OPC UA information model for IEC 61499 systems is developed in order to enhance communication with IEC 61499 applications through OPC UA. Already existing related information models, particularly OPC UA for Devices and IEC 61131-3, are utilized as models and parts of the design. An OPC UA wrapper for IEC 61499 runtimes is presented as one possible way to implement the information model. Both the information model and the wrapper are then tested through a case study.**

Keywords — OPC UA, IEC 61499, Information Model, Wrapper

I. INTRODUCTION

OPC UA is a standardized technology for communication primarily between automation and information systems and also between parts of automation systems. The communicated data may be related to the controlled process, control applications and utilized devices. Standardization of the communication technology aims to limit or decrease the cost of developing the communication systems. OPC UA is quite suitable to the role of a communication technology for Industry 4.0. However, when utilizing OPC UA in an application domain it is a good idea to reuse previously designed and widely enough adopted information models as parts of communication.

IEC 61499 is a standardized technology for developing applications for distributed automation with function block type of programming. It is one of the alternative technologies that can be used to develop automation applications of Industry 4.0 components. In this role it is necessary that the IEC 61499 applications can communicate with information systems and other parts of automation systems of different kinds. The communication has to fulfil the requirements of these other parties as well. Thus it cannot be designed purely from the viewpoint of IEC 61499 applications. If using OPC UA for this communication design of an OPC UA information model for IEC 61499 applications is likely to be useful.

The main objective of this paper is to propose a design of a tentative OPC UA information model for IEC 61499 systems. The information model will define those data structures within the OPC UA address space model that are needed for identified use cases. The users in these use cases are mainly information systems of manufacturing enterprises. Another related objective of the work is to present a design of an OPC UA wrapper software that enables access to IEC 61499 runtimes using the designed information model.

The remainder of the paper is organized as follows. Chapter 2 will discuss the related work about IEC 61499 systems and development of OPC UA information models. The proposed designs of the information model and the wrapper software are presented in chapters 3 and 4. A proof of concept implementation and an illustrating case study are described in chapter 5 followed by conclusions in chapter 6.

II. RELATED WORK

Industry 4.0 presents quite challenging objectives of customized and flexible mass production. Self-optimization and self-configuration have been envisioned to be necessary capabilities of future manufacturing systems. In order to achieve these objectives applications creating intelligent behaviour through algorithms and data and capability to communicate with other parts of automation and information systems and humans will be needed. OPC UA with its client-server [11] and publish-subscribe [15] types of communication seems to offer a large part of the required communication capabilities [19]. The IEC 61499 standard is one useful technology for development of distributed automation applications [7].

Although the IEC 61499 specification defines communication with software tools the type of communication is specific to IEC 61499. OPC UA could provide a more general type of communication to IEC 61499 applications. A few different ways to combine IEC 61499 applications and OPC UA communication have been reported. One way is to develop SIFBs that use OPC UA as a communication protocol [9][5]. Another way is to use OPC UA as a communication mechanism to access services provided by IEC 61499 applications [21]. A third possibility is to present entities of IEC 61499 applications in an OPC address space according to its rules of data representation [1]. All these efforts approach the question of combining IEC 61499 with OPC UA from the IEC 61499 viewpoint. However, the viewpoint could also be the opposite, i.e. how a client can use OPC UA to access IEC 61499 applications among other things of his interest.

OPC UA is a communication technology originally aimed for data transfer between automation and information systems [10]. However, during last few years increasing attention has been paid to developing communication within automation using the new PubSub specification [15]. In both cases the communicated data is defined according to the OPC UA address

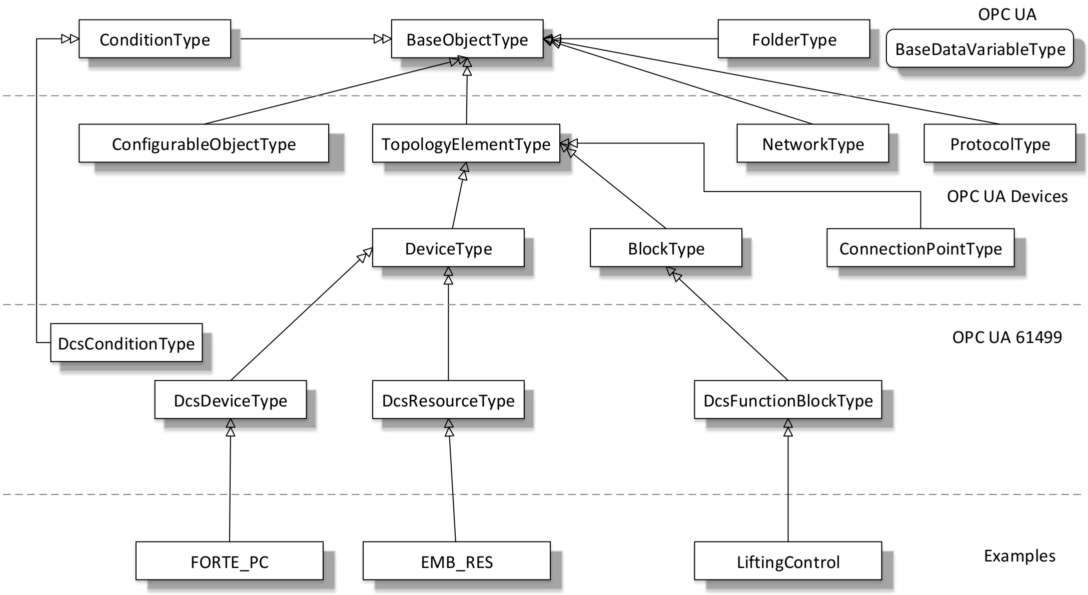

Fig. 1. Overview of the IEC 61499 information model

space model [12]. This model is an object-oriented data representation which allows definition of new types as subtypes of existing ones. This capability has been used for various purposes resulting in an increasing set of different OPC UA information models. The OPC UA standard itself includes a few information models, e.g. a model of alarms and conditions [14]. Several other information models have been published by OPC Foundation as companion specifications. Many, but not all, of these specifications are based on other previously existed standards which then have been modified to the OPC UA address space model.

One important topic of the companion specifications has been devices. Several information models have been developed for modelling data about various aspects of automation devices (and not the controlled process itself). Many, if not all, of these models are sub-models of a companion specification called OPC UA for Devices [16] which enables modelling basic features of automation devices and networks for different purposes. The sub-models of the Devices specification include, e.g. FDI [3][20], FDT [4], AutomationML [6], and IEC 61131-3 [17] companion specifications. These specifications concern about condition monitoring and asset management of devices and engineering automation applications to be deployed to them.

III. OPC UA Information Model for IEC 61499

When developing an OPC UA information model for an application area one needs to define the necessary data modelling constructs within the data modelling capabilities of the OPC UA address space model. Particularly this means definition of different data types, i.e. object, reference, variable and data types. Also general purpose instances of these types can be defined if needed. In the case of IEC 61499 necessary concepts are already defined in the IEC 61499 standard [7]. Now the task is to define the respective types and instances within the OPC UA address space model and ensure that the resulting definition is acceptable to all parties of communication.

The interests of the communicating parties may be identified through describing relevant use cases. The use cases of the IEC 61499 information model are assumed to be similar to the ones of the previously defined IEC 61131-3 information model. The use cases are outlined in the following. The communicating parties include IEC 61499 runtimes, information systems, e.g. SCADA, MES or IDE, and maybe other automation applications.

1. Observation which includes querying and browsing defined sets of object types and instances and references of an address space and reading and subscribing to variables and events. The accessed data is a subset of IEC 61499 data required by the clients. This is part of the use case of a supervising information system.
2. Operation which includes writing values of selected variables and calling chosen methods in an address space. The accessed data is particularly the function blocks and devices of IEC 61499. This is another part of the use case of a supervising information system.
3. Engineering which means creating and modifying object instances and references. The data is the applications and systems of IEC 61499. This is particularly the use case of an engineering information system.
4. Service which is currently somewhat less well defined but includes functions that could be build on the basis of the preceding use cases, e.g. uploading and configuring system software and creating new type information associated with it.

Another relevant viewpoint to the definition of the IEC 61499 information model is how well the model would fit to other relevant models. The parties of communication are likely to use also other information models for their other needs. The new information model should find its natural place among other information models and be easily understood assuming knowledge about the other models. In the case of IEC 61499 relevant other information models are OPC UA for Devices [16] as a base model and OPC UA for IEC 61131-3 [17] as a model with a very similar purpose [2][22]. In addition to these also the other sub-models of the Devices model are associated. It is quite possible that a client would require them to co-exist in the same OPC UA address space together with the planned IEC 61499 information model.

It is proposed in here that the IEC 61499 information model is a sub-model of the Devices information model and similar to the IEC 61131-3 model unless there is a good reason to deviate. An overview of the proposed model is illustrated in Fig. 1. The prefix "Dcs" is used to identify the definitions of the IEC 61499 information model. The super models of the IEC 61499 model are the Base information model of OPC UA [13], model of Alarms & Conditions [14] and the Devices companion specification. The Alarms & Conditions model is intended to be used for modelling events of IEC 61499 and the Devices model for modelling relevant properties of devices, blocks and networks. The IEC 61499 model itself will contain more detailed models of its devices, resources and function blocks. The types and instances defined in the super models will be available in the IEC 61499 information model. The resulting model becomes very similar, but not identical, to the IEC 61131-3 model.

The concepts of devices and resources in the IEC 61499 information model are very similar to the ones in the IEC 61131-3 model. Thus, it is proposed that this part of the model is designed similar to the IEC 61131-3 model unless there is an obvious need to deviate. The models of devices and resources in the IEC 61499 information model are illustrated in Fig. 2 and Fig. 3. Both of these models utilize a design pattern of configurable objects adopted from the Devices model. The pattern allows representation of both the allowed component types and the configured component instances. This is a slightly more complicated model of the concept of wholes and parts than in IEC 61499. However, it is proposed in here because it is assumed to be useful in the identified use cases, particularly engineering. Both the device and resource object types have methods for starting and stopping them in their associated MethodSet objects which are inherited from the TopologyElementType object type of the Devices information model. Instances of the device object type are placed as components of the DeviceSet object instance which is also defined in the Devices model. The DeviceSet object provides a client an easy mechanism to locate devices in the OPC UA address space. In addition, the device and resource object types can inherit several other optional properties from the DeviceType of the Devices information model. The essential difference to the IEC 61131-3 information model in here is that the presented models of devices and resources do not refer to IEC 61131-3 programs.

An important part of the IEC 61499 specification is the model of function blocks which are clearly different from the ones in IEC 61131-3. However, their role in the information model is similar to the IEC 61131-3 information model. The proposed information model of function blocks is illustrated in Fig. 4. The information model defines an abstract object type of function blocks as a subtype of BlockType from the Devices model. Actual function block types that associated runtimes in a system are able to run will then be defined as concreate subtypes of the abstract function block object type. Function blocks of applications will then be instances of the concrete function block types and become components of the resources where they have been deployed to. This arrangement is quite similar to the IEC 61131-3 model.

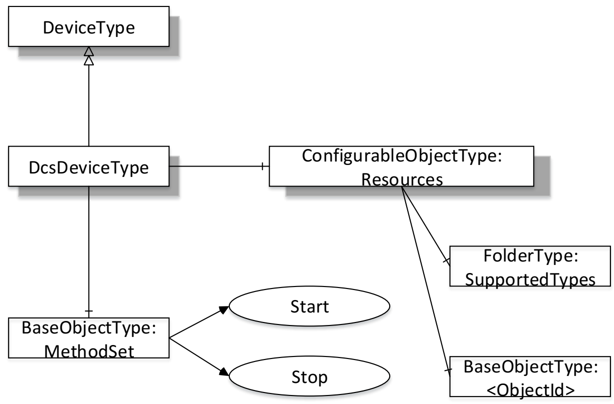

Fig. 2. Model of devices

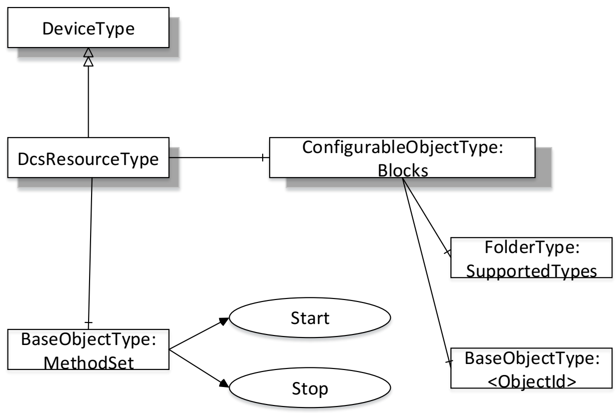

Fig. 3. Model of resources

An important difference in the IEC 61499 information model to the IEC 61131-3 model is in modelling of events and event ports. IEC 61499 event ports are modelled with a particular event object type, DcsConditionType, which is a subtype of ConditionType from the Alarms & Conditions model [12]. In this model ConditionType is the base type of events that may be triggered multiple times. Thus it will match the IEC 61499 model of events. Data ports of function blocks are modelled simply with BaseDataVariableType from the Base information model [13] which is able to represent variables of different data types. The mapping between the elementary data types of IEC 61499 and OPC UA is proposed to be done in a similar way than in the IEC 61131-3 information model [17]. The event objects and variables representing ports are connected to function blocks through subtypes of HasComponent reference which will make them follow whole-part type of semantics. This means that parts can exist only together with the whole. In addition to this, event ports are connected to function blocks also with HasEventSource references which will make event occurrences observable also through the function blocks. Ports can be connected to other ports trough HasConnectionTo references (not visible in the figure). In the model the internal structure of function blocks is not visible. It is assumed not to be available from the runtimes in the identified use cases. This assumption might not be true in a more general case. One particular feature of the IEC 61499 function blocks, called adapters, have not yet been included in the information model.

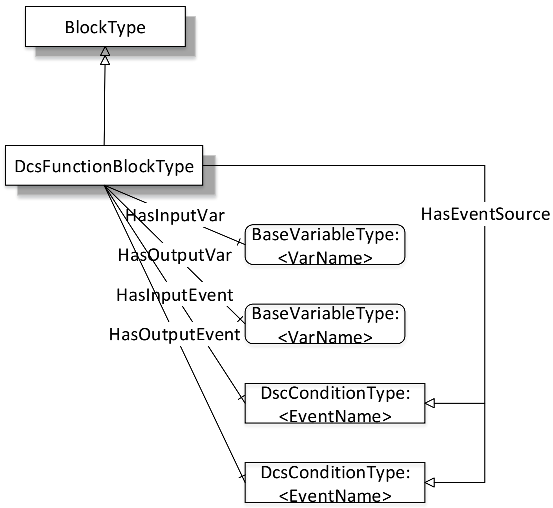

Fig. 4. Model of function blocks

In addition to the mandatory models of devices, resources and function blocks there are a few other concepts whose modelling needs to be considered in the IEC 61499 information model. These are the IEC 61499 concepts of segments, applications and systems. The concept of a segment refers to the network connecting multiple devices of a distributed system. The Devices information model uses object types NetworkType, ProtocolType and ConnectionPointType to model the same concept in a somewhat different way. Despite the slightly different approach to modelling it is proposed that the concept of a segment is modelled with the object types of the Devices model. According to this model devices connect to networks through connection points. The devices can have many connection points. Protocols are properties of networks and addresses of connection points. Networks are accessible through a particular object instance called NetworkSet. The IEC 61499 concepts of a system and an application are proposed to be modelled as OPC UA views if needed. This way of modelling would make these concepts collections of other presented objects. An application view would simply be a collection of its function blocks. A system view would contain device, resource and network objects as well. The views are assumed to be useful when the address space contains many applications or systems or other data that is not relevant to the IEC 61499 viewpoint. If a more comprehensive model of applications is needed an object type DcsApplicationType could be added to the information model with references to the parts of the application and methods for controlling them as a whole (e.g. start and stop).

IV. OPC UA Wrapper for IEC 61499 Runtimes

An OPC UA wrapper is presented here as one possible way to implement the IEC 61499 information model and enable OPC UA communication with IEC 61499 runtimes. This is not the only possible way to utilize OPC UA communication within IEC 61499 applications. An obvious alternative is that similar functionality is included into the runtimes themselves. Another alternative might be to create SIFBs that provide at least part of the same functionality. However, the intended good feature of the wrapper is that it does not require any changes to the IEC 61499 runtimes or function blocks. The design of the wrapper is assumed to be useful particularly in the use cases presented in the previous chapter. This means that the other parties communicating with the runtimes through the wrapper are information systems. They are required to have the capabilities of OPC UA clients.

The OPC UA wrapper is a separate system communicating with one or more IEC 61499 runtimes and OPC UA clients. The interaction between the wrapper and a runtime is illustrated in Fig. 5. The topic of communication is an IEC 61499 application which is first written to and then run and monitored on the runtimes and represented in the address space of the wrapper at the same time. The clients use the wrapper in order to access the IEC 61499 application according to the use cases mentioned in the previous chapter. The wrapper can present the devices and resources of the runtimes and their properties to the clients. A client can deploy an application to the wrapper which will then decompose it and pass its parts to specified runtimes and start them when requested. The clients can then monitor and control the operation of the application through the address space of the wrapper. Monitoring will take place through subscribing to variables or events in the address space. Controlling can be done through calling methods to trigger events or writing values of variables. However, the wrapper is not involved in the communication between the runtimes. It will take place according to IEC 61499 practices regardless the wrapper.

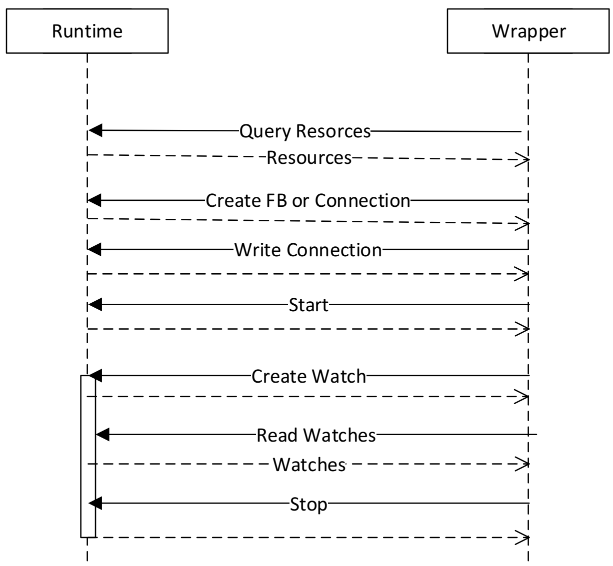

Fig. 5. Interaction between an OPC UA wrapper and an IEC 61499 runtime

The task of the IEC 61499 wrapper is to broker between OPC UA clients and IEC runtimes. The wrapper transforms OPC UA communication with the clients to IEC 61499 communication with the runtimes through a mapping between these two type of communication as described in TABLE I. Actually, the IEC 61499 wrapper is an OPC UA server application implementing the mapping of the OPC UA service calls to IEC 61499 messages exchanged with external software tools [8]. The role of the IEC 61499 information model in the wrapper is to enable the mapping of messages together with representing data needed in the services to the clients.

TABLE I. MAPPING OF IEC 61499 AND OPC UA COMMUNICATION

IEC 61499 action	OPC UA service
Request Query Resources	No service. Used when initializing the wrapper application.
Response Resources	No service. Add resources to the address space.
Request Create FB	AddNodes
Request Create Connection	AddReferences
Request Write Connection	Write (value of a variable)
Request Start / Stop	Call (method)
Request Create Watch	CreateSubscription
Request Delete Watch	DeleteSubscriptions
Request Read Watches	Part of performing the subscription service.
Response Watches	Part of performing the subscription service.

V. IMPLEMENTATION AND CASE STUDY

Implementations of the information model and the wrapper were performed in order to proof the concepts of the IEC 61499 information model and the OPC UA wrapper particularly concerning the use cases of observation, operation and engineering. The less clear use case of services was not considered at this time. The implementation as a whole consisted of three IEC 61499 runtimes of the same kind, the OPC UA wrapper and two different OPC UA clients (see Fig. 6). Communication between the parties concerned about an example IEC 61499 application called Lifting Luggage. The IEC 61499 application was first defined in one of the OPC UA clients, then deployed to the runtimes through the wrapper and finally its running was monitored and controlled from the other OPC UA client. An open-source IEC 61499 runtime environment called Forte [23] was utilized as runtime systems and a general purpose OPC UA client, UaExpert, as the monitoring client. The other parts of the system were programmed by the authors as a part of this case study.

The OPC UA wrapper including its IEC 61499 information model was implemented as a server application with Prosys OPC UA Java SDK [18]. The IEC 61499 information model was first edited with UaModeler editor and saved to a NodeSet2 file. Then Java source code was generated from it with the tools of the SDK and incorporated to the application. The SDK defined large parts of the structure of the application. The application logic for mapping OPC UA services to IEC 61499 defined messages was implemented as event handler functions overriding the default ones provided by the SDK. Event handler functions are a basic mechanism to develop applications with the SDK. Handlers were implemented for each of the service calls mentioned in TABLE I. The complementing part of the application logic, i.e. communicating data updates from the runtimes to the address space of the wrapper, was implemented as a separate thread polling the runtimes and updating events and variable values in the address space.

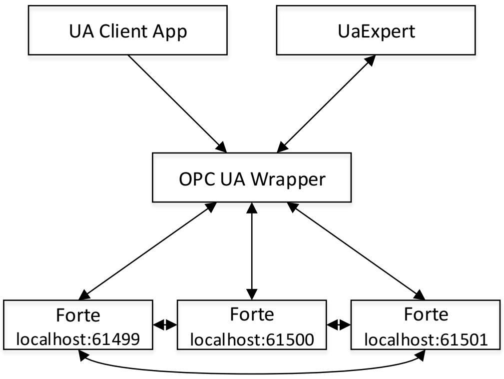

Fig. 6. Arrangement of the case study

A small part of the example IEC 61499 application used in the case study is illustrated in Fig. 7. As a whole the application is distributed consisting of three main function blocks deployed to three different devices. One of the main function blocks (Model_Logic, see Fig. 7) simulated a luggage lifting process whereas the two other main function blocks contained control logic of lifting and thrusting and operated on the simulation function block. The three main function blocks were deployed to different devices which communicated through publish and subscribe types of communication function blocks. Altogether the application consisted of 16 function blocks. A part of the representation of the Model_Logic function block in the address space of the OPC UA Wrapper as seen in the UaExpert client is presented in Fig. 8. Although the IEC 61499 application was relatively simple it enabled testing the operation of the implementation and proofing the underlying concepts of the OPC UA wrapper and its IEC 61499 information model.

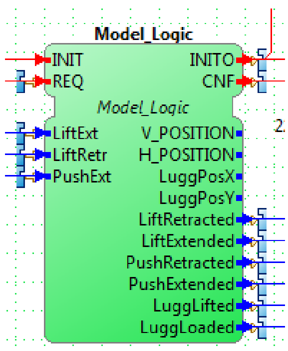

Fig. 7. Function block Model_Logic was a part of the distributed IEC 61499 application used in the case study

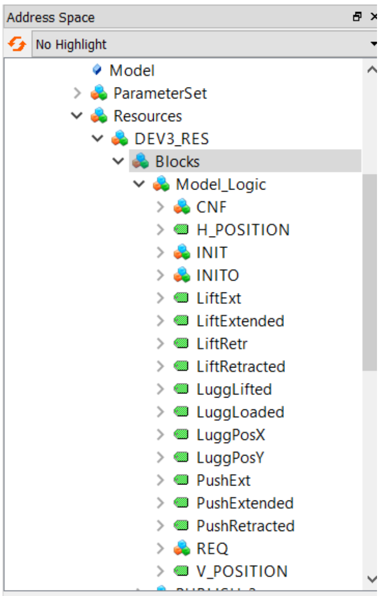

Fig. 8. Representation of the function block Model_Logic in the address space of the OPC UA wrapper in the case study

VI. Conclusion and Future Work

A few conclusions can be drawn from the presented work. Firstly, not only is the OPC UA address space model capable to represent IEC 61499 systems but a separate information model can be built based on the existing ones. Standardised OPC UA information models Base and Alarm & Conditions and the Devices companion specification are a satisfactory basis for an IEC 61499 information model. Several object types and some instances of these specifications are useful also in the IEC 61499 information model, e.g. object types DeviceType, BlockType, ConfigurableObjectType, ConditionType, NetworkType and instances DeviceSet and NetworkSet. The resulting model is quite similar to the IEC 61131-3 information model. Such a model would be easy to understand by a client who is familiar with the Devices and IEC 61131-3 companion specifications. Secondly, an OPC UA wrapper that will allow accessing IEC 61499 runtimes without modifications to them or their applications can be designed and implemented. The capability of the wrapper to present data will then depend on the capabilities of the runtimes.

The presented IEC 61499 information model can be considered as a proposal at this stage. Its design is a subject to discussion and further development. If the information model is to be developed into a companion specification in the future involvement of companies will be needed. The companies should represent both the providers and users of the information model. The capabilities of the presented wrapper would require further development as well. Capability to write new function block types to runtimes through it might be useful. However, support from the runtimes would be needed for this as well. In a longer term view interesting questions are if the communication in distributed IEC 61499 applications could be directed through OPC UA address spaces and the PubSub protocol and what this would mean to the design of applications. Furthermore, a new version 1.02 of the Devices information model and amendments to the OPC UA address space model are currently being developed which is likely to affect the design of the IEC 61499 information model in the future.

References

[1] W. Dai, Y. Song, P. Wang, C. Pang, and V. Vyatkin, “Modelling Industrial Cyber-Physical Systems using IEC 61499 and OPC UA”, 16th IEEE International Conference of Industrial Informatics (INDIN), Porto, Portugal, 2018.

[2] W. Dai, and V. Vyatkin, A Case Study on Migration from 61131 PLC to IEC 61499 Function Block Control, 7th IEEE International Conference of Industrial Informatics (INDIN), Cardiff, Wales, 2009.

[3] FieldComm Group and OPC Foundation, Filed Device Integration (FDI) – FDI Information Model, Edition 1.1.0, 2017.

[4] FDT Group and OPC Foundation, OPC UA Information Model for FDT Technology, Version 1.00, 2016.

[5] M. Garcia, E. Irisarri, F. Perez, E. Estevez, and M. Margos, An Open CPPS Automation Architecture based on IEC 61499 over OPC UA for flexible manufacturing in Oil&Gas Industry, 20th IFAC World Congress, Toulouse, France, 2017.

[6] R. Henssen, and M. Schleipen, Interoperability between OPC UA and AutomationML, Procedia CIRP, No. 25, 2014, pp. 297-304.

[7] IEC, IEC 61499 Function Blocks, Part 1: Architecture, 2nd Edition, 2012.

[8] IEC, IEC 61499 Function Blocks,Part 2: Software tool requirements, 2nd Edition, 2012.

[9] S. Kozar, and P. Kadera, Integration of IEC 61499 with OPC UA, 21st IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, Germany, 2016.

[10] W. Mahnke, S. Leither, and M. Damm, “OPC Unified Architecture”, Springer, 2009.

[11] OPC Foundation, OPC Unified Architecture Specification, Part 1: Overview and Concepts, Release 1.04, 2017.

[12] OPC Foundation, OPC Unified Architecture Specification, Part 3: Address Space Model, Release 1.04, 2017.

[13] OPC Foundation, OPC Unified Architecture Specification, Part 5: Information Model, Release 1.04, 2017.

[14] OPC Foundation, OPC Unified Architecture Specification, Part 9: Alarms & Conditions, Release 1.04, 2017.

[15] OPC Foundation, OPC Unified Architecture Specification, Part 14: PubSub, Release 1.04, 2018.

[16] OPC Foundation, OPC Unified Architecture for Devices Companion Specification, Release 1.01, 2013.

[17] PLCopen and OPC Foundation, OPC UA Information Model for IEC 61131-3, Release 1.00, 2010.

[18] Prosys OPC, Prosys OPC UA Java SDK [Online] Available: https://prosysopc.com/products/opc-ua-java-sdk/

[19] M. Schleipen, S.-S. Gilani, T. Bischoff, and J. Pfrommer, OPC UA & Industrie 4.0 – enabling technology with high diversity and variability, Procedia CIRP, No. 57, 2016, pp. 315-320.

[20] D. Schulz, FDI and the Industrial Internet of Thnings, 20th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Luxembourg, Luxembourg, 2015.

[21] T. Terzimehic, M. Wenger, A. Zoitl, A. Bayha, K. Becker, T. Muller, and H. Sauerte, Towards an Industry 4.0 Compliant Control Software Architecture Using IEC 61499 & OPC UA, 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus, 2017.

[22] M. Wenger, R. Hametner, and A. Zoitl, IEC 61131-3 control applications vs. control applications transformed in IEC 61499, 10th IFAC Workshop on Intelligent Manufacturing Systems (IMS), Lisbon, Portugal, 2010.

[23] 4diac, FORTE – the 4diac runtime enviroment [Online] Available: https://www.eclipse.org/4diac/en_rte.p