A Reconfigurable Communication Gateway for Distributed Embedded Control Systems

Filip Andrén, Thomas Strasser
Electric Energy Systems – Energy Department
AIT Austrian Institute of Technology
Vienna, Austria
Email: {filip.andren,thomas.strasser}@ait.ac.at

Alois Zoitl, Ingo Hegny
Automation and Control Institute (ACIN)
Vienna University of Technology
Vienna, Austria
Email: {zoitl,hegny}@acin.tuwien.ac.at

***Abstract*—Modern automation and control systems used in industrial large-scale applications are characterized by its distributed nature in the areas of power and energy systems, manufacturing, building automation, and logistics. In order to cope with different communication needs and changing requirements in such large-scale scenarios over its life-time, the main aim of this paper is to propose a programmable and (re-)configurable communication gateway device compliant to domain standards. This approach is therefore based on the IEC 61499 reference model for distributed and reconfigurable automation and uses the service interface function block concept and high-level communication patterns to achieve a hardware-independent access to communication services. Moreover, the utilization of the IEC 61499 management model supports the possibility of the online-reconfiguration of communication links and therefore provides a very flexible structure and implementation for an industrial automation gateway device.**

I. INTRODUCTION

The areas of power and energy systems, manufacturing, building automation, and logistics are typically characterized by the usage of modern automation systems controlling large-scale applications which have very often a distributed nature. Typically, a large number of control, measurement, and field devices (typically accomplished as modular I/O devices) are exchanging information over communication networks and field bus systems. Today, a trend towards the usage of Ethernet-based approaches can be observed providing a common hardware communication standard in the aforementioned application areas. However, a high variety of communication protocols (i.e., standardized and vendor specific) are usually in use since they are provided by the control and I/O devices, as well as measurement units applied in industrial control applications. Moreover, also legacy (communication) systems with field bus protocols have to be integrated in larger applications. Summarizing, the vision of only using one hardware and software standard covering all communication needs and requirements in industrial-process measurement, control, and automation systems is still a high-level goal and a nearly unreachable challenge in real-world applications so far.

To overcome the problem of using different hardware and software standards so called "glue logic" in automation and control programs is needed. With this glue logic conversion between different protocols and standards can be realized. Depending on the communication network different solutions are possible in order to provide such a glue logic. Conversion gateways are one possibility, which often have the possibility to convert between two or more communication protocols (e.g., CAN to Modbus, PROFIBUS to CAN, [1]).

The IEC 61499 reference model—introduced by the IEC Technical Committee 65 (TC65)—for distributed and reconfigurable automation was developed as methodology for modeling open and distributed Industrial-Process Measurement and Control Systems (IPMCS) [2]. The goal of this standard is to obtain a vendor-independent application and hardware configuration description in order to manage the increased complexity of next generation automation systems. This can on a higher level be summarized as the portability of automation projects as well as the configurability and interoperability of intelligent field devices in IPMCS. The core element of IEC 61499 are Function Blocks (FB) for the encapsulation of modular control software. These FBs can be later on deployed to intelligent field devices. The IEC 61499 FB model is based on its predecessor IEC 61131-3 [3] but uses an *event-driven execution model* for execution of FB networks. This standard uses an *application-centered modeling methodology* and the possibility of *distributing control applications* to different control devices is also supported. Thus, IEC 61499 provides the basis to realize a platform independent description and realization of automation applications [4], [5]. A more detailed discussion of the IEC 61499 basic functions and services for a reconfigurable communication gateway is provided in Section II and III.

The main aim of this paper is to propose a programmable and (re-)configurable communication gateway, which can be realized in a standard-compliant way. A standard-compliant solution is important in order to integrate the gateway solution with common automation standards used in industry, thus harmonizing engineering of control and communication applications. This is essential in distributed control applications where typically a huge number of actors have to execute joint tasks in order to automate industrial processes. In such environments integration of actors with different and changing communication capabilities is a priority task, which is the focus of this work.

The remaining part of the paper is organized as follows: A short overview and summary of related work is provided in

Section II. Section III provides an analysis about the communication patters provided by IEC 61499. In order to overcome the shortcomings of actual solutions a reconfigurable gateway concept based on the IEC 61499 specification is introduced in the following Section IV. Moreover, in Section V a potential realization and two selected examples are discussed. The main findings and conclusions are presented in Section VI and an outlook about further activities is given.

II. Related Work

A. Automation and Control Systems

In general, centralized and distributed concepts are being used for control systems in industrial applications [4], [5]. PLC-based[1] systems are mainly based on a centralized approach. During the last decades the IEC 61131 standard and especially the corresponding programming languages defined in part 3 [3] has been established as the main domain standard. In contrast to the centralized approaches a trend towards the usage of distributed and embedded control devices in IPMCS can be observed. In order to have an interoperable and portable solution for distributed controllers, the IEC has introduced a reference model for distributed and reconfigurable systems, the IEC 61499 standard [2]. It defines, differently to the IEC 61131 approach, an application and system model (with devices, resources, communication segments and links, and the technical process), and a Function Block (FB) based modeling approach as mentioned in the introduction. A further fundamental difference to the cyclic execution of control applications in PLC systems according to IEC 61131 is the event-based execution in IEC 61499 compliant controllers.

B. Communication Services in PLCs and Distributed Systems

In the domain of PLCs, which are widely used in industrial control applications, the IEC 61131-3 [3] definition for the harmonization of programming languages plays an important role. In this standard no concepts and methods are described for a harmonization of communication concepts and patterns which are seen to be very important in a network of heterogeneous controllers and field devices. Only part 5 of the IEC 61131 standard [6] provide special functions blocks which covers the communication in PLC systems, but up to know they are rarely used in industrial applications. Currently, there is a much more sophisticated development ongoing which has a high potential to harmonize the information and data exchange in PLC-based systems. The international user organization PLCopen for enhancing the IEC 61131 concept is working together with the OPC Foundation to specify an "OPC UA Information Model for IEC 61131-3" [7].

Similar to the IEC 61131-5 definitions the IEC 61499 model provides high-level communication patterns and functions for the information exchange between embedded control devices. These are described in more detail below in Section III. However, the usage of a special communication protocols (e.g., Ethernet, Industrial Ethernet, field bus) is not in the focus of this high-level specification. Some mappings to domain protocols are documented in the literature [8]–[11].

Summarizing, some attempts in the PLC domain has been undertaken to harmonize communication aspects from the modeling and execution point of view but the dynamic change and adaptation of communication links is not really addressed up to know. The OPC UA specification could overcome shortcomings of actual PLC systems. In contrast, the IEC 61499 already covers this important topic and provides some high-level communication patterns and services which can be mapped to well known lower-level domain protocols.

C. Reconfigurable Automation and Control Systems

Reconfigurable concepts and architectures have been used in the research domain for industrial automation and control systems since several years. For example, Kramer et al. [12] have introduced this important topic for distributed systems. In general, the main aim of reconfiguration is related to the altering of functions and algorithms in software models. Especially in the automation domain reconfiguration means the adaptation and adjustment of control functions and algorithms in control, communication, measurement and field devices due to changed requirements. Usually, there are many ways to describe reconfigurability of software elements. This topic can be solved in different ways: reconfigurability can happen on a higher level (e.g., implemented by Multi-Agent Systems) as well as also on a lower level (e.g., real-time control with IEC 61131 or IEC 61499). Moreover, one has to distinguish between static and dynamic reconfiguration [13]–[15].

Especially on the real-time control level, which is in the scope of the approach introduced in this paper, there are some concepts mentioned in the literature. Most of the actual PLC systems based on IEC 61131 support some kind of reconfiguration. This means that control programs can be adapted in PLCs but this is carried out in a relative simple way. Normally, the modified PLC program is downloaded to the PLC hardware and the corresponding software system decide the point in time where the "old" program is replaced by the "new" one. Usually there is no way to control the exchange procedure in modern PLC systems.

In contrast the IEC 61499 reference model [2] for distributed and reconfigurable IPMCS provide a management model and the corresponding Application Programming Interface (API). Therefore, it provides the possibility to control the life-cycle of software components (i.e., FBs) in distributed control devices. Moreover, IEC 61499 defines eight different configuration commands (i.e., `START`, `STOP`, `KILL`, `READ`, `WRITE`, `CREATE`, `DELETE`, `QUERY`) which are often used in the literature to adapt FB instances and therefore control applications [16]–[19]. The main difference to the approach in IEC 61131-3 is that the reconfiguration is carried out on a more detailed level without the need to adapt a whole control program. In addition, a standardized management interface is missing in IEC 61131-3 systems and also the event-driven architecture of IEC 61499 allows to better synchronize adaptation steps. According to the literature, the IEC 61499

[1]Programmable Logic Controller

management interface is mainly used for adapting control algorithms [16]–[19]. The topic of using the management interface as basis for the reconfiguration of communication links is only briefly discussed in a few papers [20].

Summarizing, the IEC 61499 approach provides a very good baseline for reconfigurable control systems in a distributed environment but a more formalized approach for defining reconfigurable gateway functions in order to dynamically adapt communication links is currently missing.

D. Model-Driven Architecture and Engineering

The Model-Driven Architecture (MDA) and its related engineering approach (MDE[2]) has been developed by the Object Management Group (OMG) in the computer science domain for software systems to improve the whole design, development and implementation process [21]. One of the main goals of the MDA approach is the focus on the development of domain specific software models. Following the MDA/MDE definitions, normally a Platform Independent Model (PIM) of application programs, the specification of the execution environment, i.e., a Platform Definition Model (PDM), and the mapping of the PIM to a PDM—resulting in a Platform Specific Model (PSM)—is carried out. The PSM is normally used for the (automatic) generation of code for the execution environment, i.e., Implementation Specific Model (ISM) [22].

While the MDA/MDE approach is well known and widely used in computer science it is not a very common practice in the industrial automation sector. During the last few years some research groups provided first solutions and adaptations of the MDA approach for automation and control systems. Very interesting concepts are for example proposed by Thramboulidis (integrated mechatronic engineering using elements from IEC 61499) [23], the European project MEDEIA (model-driven automation based on IEC 61499 definitions) [24] as well as by Hästbacka et al. (usage of UML Automation Profiles for MDA and code generation based on IEC 61131-3) [25], in order to mention a few of them.

Concluding, the major advantages of using MDA/MDE in automation and control systems are related to the simplification of the design and implementation process. Normally, this would lead to higher software quality with reduced errors and faults as well as in a higher design productivity.

The discussion above shows that the current automation standards provide some concepts for modelling communication services. However the generalized view in IEC 61499 together with its built in support for reconfiguration shows some of the advantages of this standard compared to IEC 61131-3. Another benefit of IEC 61499 is the possibility to use its concept in an MDA/MDE approach.

III. Analysing IEC 61499 Communication Patterns

Since the IEC 61499 standard has been developed for distributed control systems, the communication is an important topic. This can be seen at first in the system model with the interacting devices connected with network segments and the distribution (mapping) model where applications are distributed to these devices and interact via the modeled network segments. As IEC 61499 is a generic standard it cannot provide specific communication means for this interaction. Nonetheless it defines in Annex E two generic communication models and a suggestion for encoding and decoding IEC 61499 events and data according to the ASN.1 specification. The two generic communication models represent a bi-directional transaction with the Client/Server model and a uni-directional transaction with the Publisher/Subscriber model as introduced in Section II-B. These two models fulfill most requirements of distributed IEC 61499 applications. Since the two communication models are generic it is also possible to map other communication protocols that support either the Client/Server or the Publisher/Subscriber approach. This mapping can be seen as the transformation of the PIM to the PDM.

A. Bi-directional Transactions

For bi-directional transactions IEC 61499 defines a pair of generic SIFBs named `CLIENT`/`SERVER` (see Fig. 1 for the interface definition). The data exchange is hereby from the `CLIENT` to the `SERVER` and back from the `SERVER` to the `CLIENT`. Before the data exchange can be performed both SIFBs need to be initialized. IEC 61499 defines that the `SERVER` has to be initialized first before the `CLIENT` can within its initialization phase establish a connection to the `SERVER`. This is one of the main drawbacks of this model as one has to ensure that the `SERVER` is ready before the `CLIENT` starts to initialize, which can be hard to ensure in a distributed control system. The typical usage scenarios for this model are master/slave interactions between application parts or a remote service invocation. There the `CLIENT` triggers a service provided by the `SERVER`, which returns the result of the invoked service to the `CLIENT`.

B. Uni-directional Transactions

The second method is described in IEC 61499 with the SIFB pair `PUBLISH`/`SUBSCRIBE` (see Fig. 2 for the interface definition). The transaction is directed from the `PUBLISH` to the `SUBSCRIBE`. In contrast to the bi-directional model there is no initialization order in the unidirectional model, which is a great advantage. Furthermore, several `SUBSCRIBE` SIFBs may listen to the same `PUBLISH` SIFB. The main drawback of this transaction model is that the `PUBLISH` SIFB gets no feedback if the data has been correctly received and the `SUBSCRIBE` SIFB has no information if a corresponding

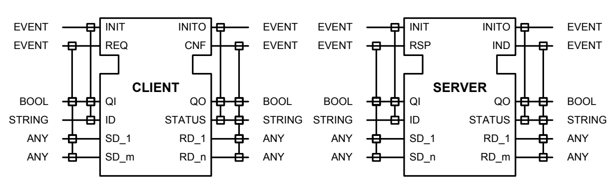

Fig. 1. Generic `CLIENT`/`SERVER` SIFBs for bi-directional transactions according to IEC 61499-1 Annex E [2]

[2]Model-Driven Engineering

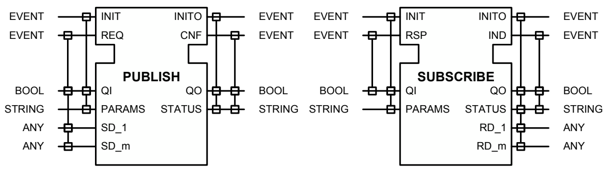

Fig. 2. Generic `PUBLISH`/`SUBSCRIBE` SIFBs for uni-directional transactions according to IEC 61499-1 Annex E [2]

`PUBLISH` exists. For critical communication additional means (e.g., timeout monitoring or handshake mechanisms) may need to be added on the application level. Summarizing, the uni-directional transaction model is suited for representing control and data flow in IEC 61499 application, where the originator needs no knowledge on the receiver and vice versa, as it is the case for IEC 61499 event and data connections.

IV. Reconfigurable Communication Gateway Concept in General

A. Engineering Approach

For modern distributed automation systems, especially in large scale industrial applications, the communication system is of a major importance. Thus decisions about protocols, hardware, software, etc. are often taken at the beginning of the design process. Changing system requirements, updates in hardware, etc. often lead to adaptations of the communication system. This, as well as other competing system requirements (e.g., metrological, control, or safety demands) often causes the communication system to be "glued" together in order to include all components as it evolves over its life-time.

1) Engineering using MDA and IEC 61499: The use of glue-logic requires detailed knowledge about each part from the engineers using the system. To ease the engineering effort, approaches using a MDA/MDE have been introduced in the automation domain as briefly discussed in section II-D. Using such an approach a PIM can be created for the communication system independent from its actual implementation (i.e., protocols, medium, latencies, etc.). Once the design is complete the model can be transformed into a PSM using the specifications defined by the PDM.

An MDA/MDE approach is in general possible using IEC 61499 elements when modeling the communication system [26]. The application model in IEC 61499 can be seen as a PIM, and the PSM is created once the application is mapped to devices. This principle is shown in Fig. 3 where the platform independent communication model is mapped to different platforms using platform specific parameters. By using the generalized communication patterns described in Section III generalized models of the communication services can be created.

2) Process Interface: In order to communicate with the process a corresponding interface is used. This is modeled with the *Sensor* and *Actuator* SIFBs in Fig. 3. These SIFBs uses the communication patterns described in Section III and act as interfaces to the process, used by for example a controller or a SCADA system. The actual mapping from the *Platform Independent Communication Model* to the *Platform Specific Application Part* in *Device 1* and *Device 2* is achieved by providing *Platform Specific Parameters*. The *Platform Specific Parameters* defines the specifics needed for the *Process Interface*, i.e., it acts as a PSM. This means that if the process interface of the application changes a new PSM can be created by only changing the platform specific parameters. In Fig. 3 the SIFBs for the sensors and actuators are mapped to different hardware devices, which is practical if they are spatially separated. If needed a mapping to the same device is however also possible. The main point is that the platform independent communication model remains unchanged even if the platform specifics change, resulting in a mere remapping in order to fit it to the new platform specifications.

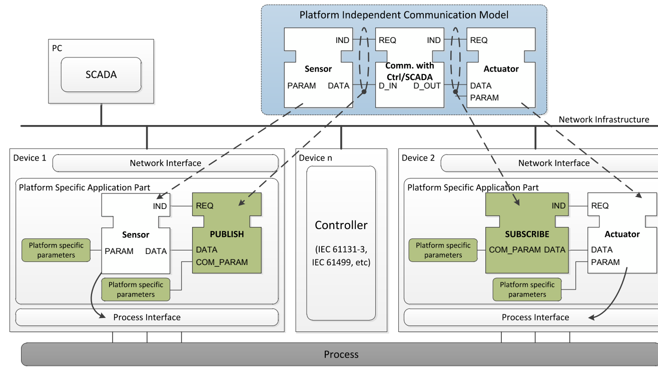

Fig. 3. MDA mapped to the IEC 61499 reference model [22], [26], [27]

3) Network/Communication Interface: The example in Fig. 3 also shows an interface for communication between the devices and with a possible SCADA system connected to the *Network Infrastructure*. The `PUBLISH` and `SUBSCRIBE` SIFBs define the communication between different IEC 61499 devices. The *Platform Specific Parameters* are once again used to define the communication protocol used by the *Network Interface*. This can for example be Modbus/TCP or OPC for communication with the SCADA system or a proprietary protocol for communication with a controller. The controller and the SCADA system are depicted as one FB in the platform independent communication application. However, they can be split up into two FBs in order to better indicate the multiplicity of the communication connection.

B. Management and Reconfiguration Interface

During the life-time of an industrial process the requirements and components/devices are usually changing. Therefore, an adaptation of the main control functions as well as the communication services is in general necessary at some point. For the communication services such an adaptation can be anything from a complete redesign to a simple reconfiguration. Using the MDA approach a reconfiguration is possible since an update (e.g., change of the communication protocol or parameters, device addresses) is only a matter of a remapping to a new PSM. The condition is of course that a PSM of this protocol mapping is already defined. The functionality and availability of a critical system has to be guaranteed requiring

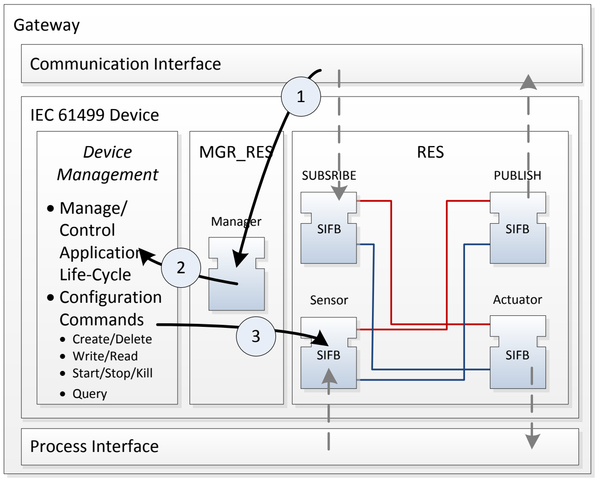

Fig. 4. IEC 61499 standard-compliant reconfiguration approach

adaptations with a minimum influence as possible. The on-line reconfiguration of communication services can be a very promising approach in order to make the necessary adaptation and to maintain functionality, availability, and quality of the system. On-line reconfiguration can be achieved using the management model in IEC 61499. Following the management model, a device has to implement a device management and a management resource (`MGR_RES`) according to IEC 61499.

As mentioned in Section II-C there are eight configuration commands according to the IEC 61499 management model, which should be supported by the device management. An example on how to use these commands is provided in Fig. 4. Through the usage of a management resource (`MGR_RES`) and the communication interface, the device management executes (re-)configuration requests. In a first step (re-)configuration requests are sent over the communication interface to the management resource. The corresponding device management SIFB interprets the (re-)configuration request and informs the device management which is responsible for its execution.

V. Prototypical Realization and Example

A. Framework and Environment

In order to implement a prototype for the IEC 61499-compliant reconfigurable gateway concept, as introduced above, the open source solution “Framework for Distributed Industrial Automation and Control (4DIAC)” [28] was used. 4DIAC provides an open, IEC 61499-compliant environment that comprises a runtime system for small and resource constraint embedded controllers, called “FORTE” and the integrated engineering tool “4DIAC-IDE”. This open source solution is available for different PC-based and embedded hardware platforms. The current version of the 4DIAC environment supports various communication protocols like TCP/IP (ASN.1), UDP/IP (ASN.1), Modbus/TCP as well as OPC-DA. These protocols are useful for communication on a normal Ethernet network, however for field bus communication special protocols are often needed. To extend the possibilities of the IEC 61499 gateway using 4DIAC, two

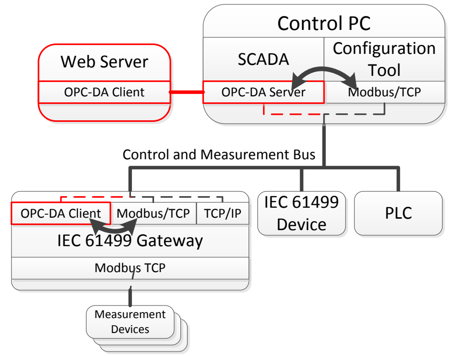

Fig. 5. Reconfiguration of SCADA communication

industrial Ethernet protocols were integrated into 4DIAC: openPOWERLINK and EtherNet/IP [8], [9], [29].

B. Use Case: SCADA Communication

This use case is intended to show the usability of the IEC 61499 gateway as well as to demonstrate its opportunities and advantages. An overview of the communication system for this use case is shown in Fig. 5. The aim is to show how FB reconfiguration can be made using 4DIAC (as described in Section IV-B) and to show how the communication SIFB (see Section III) are used in practice.

In this example the catalyst is the integration of a web server component in order to allow remote supervision of an industrial control process, which consists of some controllers (i.e., IEC 61499 devices and other embedded devices) and measurement devices. Before the addition of the web server the SCADA system communicated with the measurement devices over Modbus/TCP. The control devices which needed measurements did not have support for Modbus/TCP and thus they needed to communicate with the measurement devices over the IEC 61499 gateway which translated from TCP/IP to Modbus/TCP. This configuration represents the network in Fig. 5 without the parts marked red.

To allow remote supervision the web server needs the measurements. One possibility would be to establish a communication between the web server and the measurement devices over the control and measurement bus. This option was however discarded since this would result in even more traffic on the already heavily loaded control and measurement bus. To overcome this problem the communication between the SCADA system and the IEC 61499 gateway was reconfigured from Modbus/TCP to OPC-DA using an OPC-DA server in the SCADA system. Thus the web server can access the data through the OPC-DA server. Since OPC-DA can use reporting (i.e., an update is sent when the data is changed) it also has the advantage that polling is not needed any more, reducing the load on the control and measurement bus. The parts that were reconfigured are outlined red in Fig. 5.

1) IEC 61499 Gateway Implementation: The IEC 61499 gateway shown in Fig. 5 must be configured to support protocol translations between two Ethernet networks. This is made possible by the FB network seen in Fig. 6. The three

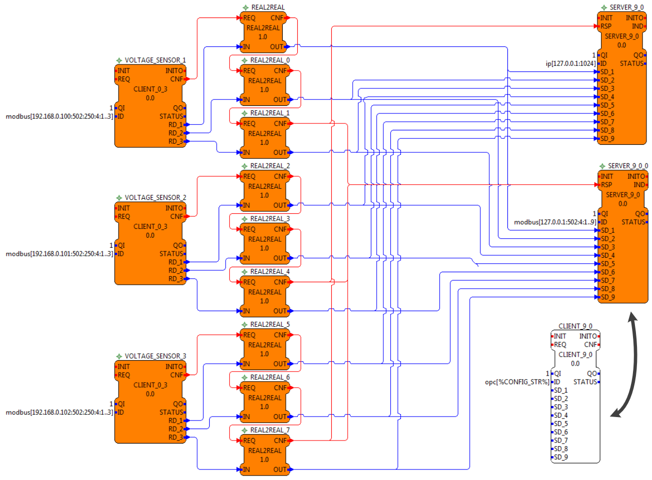

Fig. 6. IEC 61499 FB network used for the gateway

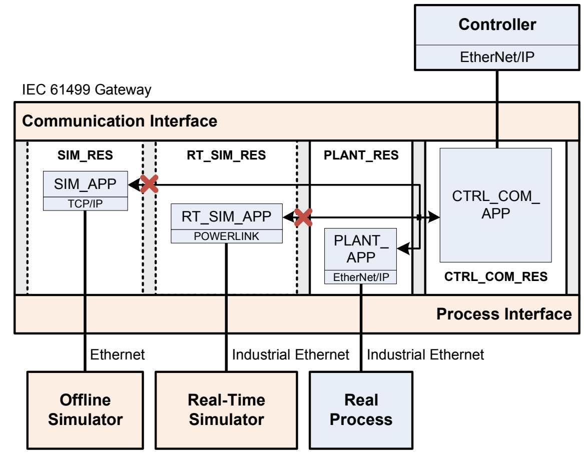

Fig. 7. IEC 61499 gateway used for controller development, currently connected to the real process

`CLIENT`s using Modbus/TCP on the left side of the figure collect the measured data from the measurement devices. In this figure three measurement devices are used. The collected data is published to the "Control and Measurement Bus" by the two `SERVER`s on the right side of Fig. 6. Before the gateway is reconfigured one of the `SERVER`s uses Modbus/TCP for communication with the SCADA system and the other `SERVER` uses plain TCP/IP for communication with other IEC 61499 devices and PLCs. When the gateway is reconfigured the `SERVER` using Modbus/TCP is exchanged with a `CLIENT` using OPC-DA. The new OPC-DA `CLIENT` adopts the same events and data signals used by the Modbus/TCP `SERVER`. More detailed information about the `CLIENT`/`SERVER` SIFBs can be found in [30]. The reconfiguration is done using the management commands described in Section II-C. This use case uses a simple communication model which is mapped only to one device. It can be compared to the description in Section IV. The `REAL2REAL` FBs are needed for data type definition. The `SERVER` and `CLIENT` FBs are generic and can be used with any data type, however when instantiated the used data type needs to be defined.

C. Use Case: Controller Development

The development process of a controller is usually divided into different stages and nowadays it is common to use various kinds of simulations as validations methods before deployment of the controller to the real process. Such simulations may be located on different simulation platforms allowing offline and real-time experiments. Each simulation environment has its own interfaces for communication with the outside world. This makes the controller development challenging since new interfaces must be regarded when the controller is validated against a new simulation tool. In such an environment a real device may at any time be replaced by a simulated one if a controller update must be verified, or the simulation platform of a device may change. The crucial point here is, that these changes in the periphery should not impact the core control application. Ideally this application can remain unchanged even if external devices may change [31]. The concept of the reconfigurable gateway is one way to facilitate controller development, by providing a platform independent communication interface between the controller and the process (real or simulated). Such a solution is shown in Fig. 7.

1) Generic Communication Interfaces: The aim of the specification in Fig. 7 is to make the communication for the controller transparent and independent from the interfaces to the platform below. Depending on the current validation stage (i.e., testing against a simulation or against the real process) the gateway can be configured with the needed interface resource establishing a connection between the `CTRL_COM_APP` and the interface resource. In Fig. 7 the controller is currently tested against the real process, i.e., only the `PLANT_RES` is loaded on the gateway device.

From the `CTRL_COM_APP`s point of view the direct communication to auxiliary devices is replaced with communication to local IEC 61499 interface resources using generic publishers (i.e., `PUBL`) and subscribers (i.e., `SUBL`) according to the "IEC 61499 Compliance Profile for Feasibility Demonstration" [5]. This functionality is provided by communication FBs which are special IEC 61499 SIFBs using shared memory. The `PUBL/SUBL` communication can be compared to UDP/IP connections but it does not generate any network load nor has it problems with network latency. This means that the direct communication to the external devices is implemented in the dedicated interface resources.

In the use case three different interface resources are used for communication with the validation environments:

- `SIM_RES` has one application (`SIM_APP`) which provides the TCP/IP interface to a simulation environment for offline simulations.
- `RT_SIM_RES` has one application (`RT_SIM_APP`), which provides an Ethernet POWERLINK interface to the real-time simulation of the the process [8].
- `PLANT_RES` has one application (`PLANT_APP`) which provides the interface to the real process. In this use case EtherNet/IP is used [9]. Since this interface is the same

as between the controller and the `CTRL_COM_APP` this interface resource achieves the same as would a direct connection between the controller and the process.

The applications running on the interface resources do not only have the purpose to provide the specific interface implementation to the different platforms, but also to provide a generic interface toward the `CTRL_COM_APP`.

VI. Conclusion and Outlook

This paper describes an approach for an adaptable and reconfigurable communication gateway using IEC 61499. Using a MDA/MDE approach with IEC 61499 capabilities the communication system can be modeled as a PIM. This enables excellent opportunities for fast reconfiguration since this is reduced to a matter of a remapping to a new PSM. Moreover, using the generic communication patterns provided by IEC 61499, communication links can be described in a generalized way, i.e., modeled as `CLIENT`/`SERVER` or `PUBLISH`/`SUBSCRIBE` connections. Two use cases have been described showing two possible application scenarios where reconfiguration as well as a generalized communication model for the gateway function are needed.

The future work is related to the integration of additional protocols for more versatile gateway functionality. The proposed concept will also be used in the power and energy laboratory of AIT providing a flexible validation platform for the control application development in Smart Grids.

Acknowledgment

This work is partly supported by the Austrian Climate and Energy Fund and by the Austrian Research Promotion Agency (FFG) under the project *"DG-EV-HIL"*.

References

[1] L. Guohuan, Z. Hao, and Z. Wei, "Research on designing method of CAN bus and Modbus protocol conversion interface," in *BioMedical Information Engineering, 2009. FBIE 2009. International Conference on Future*, Dec. 2009, pp. 180–182.

[2] *IEC 61499: Function blocks*, Part 1-4, International Electrotechnical Commission Std. IEC 61 499, 2005. [Online]. Available: www.iec.ch

[3] *IEC 61131: Programmable controllers*, Part 3: Programming languages, International Electrotechnical Commission Std. IEC 61 131, 2003. [Online]. Available: www.iec.ch

[4] R. W. Lewis, *Modeling control systems using IEC 61499*. IEE Publishing, 2001, no. ISBN: 0 85296 796 9.

[5] J. H. Christensen. (Access Date: May 2012) HOLOBLOC.com - Function Block-Based, Holonic Systems Technology. [Online]. Available: www.holobloc.com

[6] *IEC 61131: Programmable controllers*, Part 5: Communications, International Electrotechnical Commission Std. IEC 61 131, 2000. [Online]. Available: www.iec.ch

[7] *OPC UA Information Model for IEC 61131-3*, PLCopen & OPC Foundation Std. 1.00, 2010. [Online]. Available: www.plcopen.org

[8] F. Andren and T. Strasser, "Distributed open source control with Industrial Ethernet I/O devices," in *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*, Sept. 2011.

[9] W. Leonhardsberger and A. Zoitl, "Using EtherNet/IP with IEC 61499 Communication Function Blocks," in *HoloMAS*, ser. Lecture Notes in Computer Science, V. Maŕık, P. Vrba, and P. Leitão, Eds., vol. 6867. Springer, 2011, pp. 39–49.

[10] J. J. Scarlett and R. W. Brennan, "Evaluating a new communication protocol for real-time distributed control," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 3, pp. 627–635, 2011.

[11] C. Schwab, M. Tangermann, A. Luder, A. Kalogeras, and L. Ferrarini, "Mapping of IEC 61499 function blocks to automation protocols within the TORERO approach," in *Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE International Conference on*, June 2004, pp. 149–154.

[12] J. Kramer and J. Magee, "Dynamic Configuration for Distributed Systems," *Software Engineering, IEEE Transactions on*, vol. SE-11, no. 4, pp. 424–436, Apr. 1985.

[13] R. Brennan, M. Fletcher, and D. Norrie, "An agent-based approach to reconfiguration of real-time distributed control systems," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 4, pp. 444–451, Aug. 2002.

[14] M. Khalgui, O. Mosbahi, Z. Li, and H.-M. Hanisch, "Reconfiguration of Distributed Embedded-Control Systems," *Mechatronics, IEEE/ASME Transactions on*, vol. 16, no. 4, pp. 684–694, Aug. 2011.

[15] M. Vallée, M. Merdan, W. Lepuschitz, and G. Koppensteiner, "Decentralized Reconfiguration of a Flexible Transportation System," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 3, Aug. 2011.

[16] M. Fletcher and D. Norrie, "Realtime reconfiguration using an IEC 61499 operating system," in *Parallel and Distributed Processing Symposium., Proceedings 15th International*, Apr. 2001, pp. 985–991.

[17] A. Schimmel and A. Zoitl, "Distributed online change for IEC 61499," in *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*, Sept. 2011.

[18] G. Stambolov and I. Batchkova, "Reconfiguration processes in manufacturing systems on the base of IEC 61499 standard," in *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on*, vol. 1, Sept. 2011, pp. 161–166.

[19] A. Zoitl, C. Sunder, and I. Terzic, "Dynamic Reconfiguration of Distributed Control Applications with Reconfiguration Services based on IEC 61499," in *Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on*, June 2006, pp. 109–114.

[20] T. Strasser, C. Sunder, A. Zoitl, M. Rooker, and J. Brunnenkreef, "Enhanced IEC 61499 Device Management Execution and Usage for Downtimeless Reconfiguration," in *Industrial Informatics, 2007 5th IEEE International Conference on*, vol. 2, June 2007, pp. 1163–1168.

[21] S. J. Mellor, S. Kendall, A. Uhl, and D. Weise, *MDA Distilled*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.

[22] M. Wenger, M. Melik-Merkumians, I. Hegny, R. Hametner, and A. Zoitl, "Utilizing IEC 61499 in an MDA control application development approach," in *Automation Science and Engineering (CASE), 2011 IEEE Conference on*, Aug. 2011, pp. 495–500.

[23] K. Thramboulidis, "Model-integrated mechatronics - toward a new paradigm in the development of manufacturing systems," *Industrial Informatics, IEEE Transactions on*, vol. 1, no. 1, pp. 54–61, Feb. 2005.

[24] T. Strasser, C. Sünder, and A. Valentini, "Model-driven embedded systems design environment for the industrial automation sector," in *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on*, July 2008, pp. 1120–1125.

[25] D. Hästbacka, T. Vepsäläinen, and S. Kuikka, "Model-driven development of industrial process control applications," *Journal of Systems and Software*, vol. 84, no. 7, pp. 1100–1113, 2011.

[26] M. Wenger, A. Zoitl, R. Froschauer, M. Rooker, G. Ebenhofer, and T. Strasser, "Model-driven engineering of networked industrial automation systems," in *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, July 2010, pp. 902–907.

[27] A. Zoitl and V. Vyatkin, "IEC 61499 Architecture for Distributed Automation: the "Glass Half Full" View," *IEEE Industrial Electronics Magazine*, vol. 3, no. 4, pp. 7–23, 2009.

[28] 4DIAC Consortium, "Framework for Distributed Industrial Automation and Control (4DIAC)," PROFACTOR Research, Tech. Rep., Access Date: May 2012. [Online]. Available: http://www.fordiac.org

[29] F. Weehuizen and A. Zoitl, "Using the CIP Protocol with IEC 61499 Communication Function Blocks," in *Industrial Informatics, 2007 5th IEEE International Conference on*, vol. 1, June 2007, pp. 261–265.

[30] M. Hofmann, M. Rooker, and A. Zoitl, "Improved communication model for an iec 61499 runtime environment," in *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*, sept. 2011, pp. 1 –7.

[31] W. Harrison, D. Tilbury, and C. Yuan, "From Hardware-in-the-Loop to Hybrid Process Simulation: An Ontology for the Implementation Phase of a Manufacturing System," *Automation Science and Engineering, IEEE Transactions on*, vol. 9, no. 1, pp. 96–109, Jan. 2012.