A Middleware Architecture for Vertical Integration

Ahmed Ismail, Wolfgang Kastner
Vienna University of Technology
Email: {aismail, k}@auto.tuwien.ac.at

***Abstract*—With the advent of the Industrial Internet of Things (IIoT), there is an increasing emphasis on the importance of vertical integration in industrial enterprises. In this paper, we address the challenge of distributed execution and system interoperability by discussing the modelling and networking aspects involved. Based on these technologies, an architecture for a distributed gateway service bus (GSB) operating using a service oriented architecture (SOA) is proposed.**

I. Introduction

Within the industrial domain, the IoT is expected to bring about a fourth industrial revolution (Industrie 4.0) under the flagship of the IIoT. Decidedly, the main approach of this revolution takes form as a push for industry-wide adoption of cyber-physical production systems, which are hybrid systems of integrated computational and physical processes. A well-known framework for the development of these cutting-edge systems is the Reference Architecture Model Industrie 4.0 (RAMI 4.0) [1]. The pillars of this architecture are based on the use of horizontal, vertical and life cycle integration to address the intricacies involved in the conversion process.

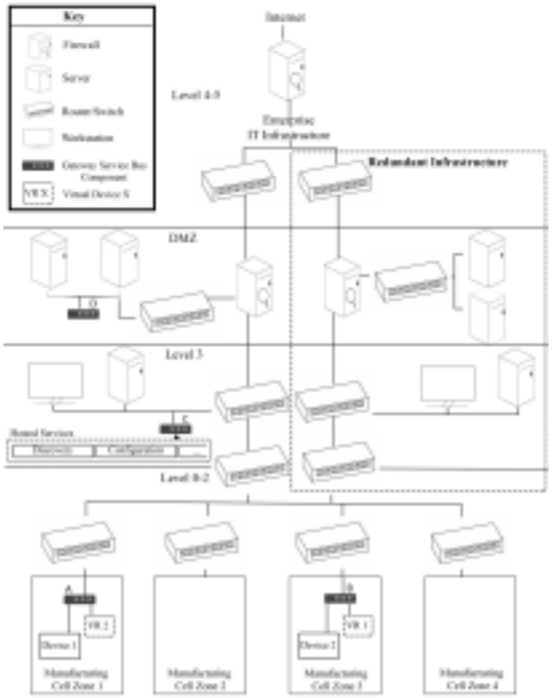
In this paper, we focus specifically on the aspects of vertical integration; a field dedicated to enhancing data exchange connections between all of the layers of an industrial enterprise. The importance of this domain is derived from the fact that each layer of the enterprise typically operates using a mixture of layer-specific protocols that are not necessarily designed for interoperability. Furthermore, the long life cycles of automation systems typically ensure the presence of legacy devices and protocols that lack the means to adopt new measures for interoperability. This results in a hindrance to the adoption of enhancing technologies from the IT domain that are increasingly dependent on the easy accessibility of devices and data. Consequently, it is the task of vertical integration to address system heterogeneity and facilitate the adoption of advanced business-enhancing solutions.

A typical approach for the management of system heterogeneity is the use of gateways. Such gateways are typically tasked with carrying out the protocol translations or mappings necessary for the intra- and inter-layer communication of data. However, due to the very nature and role of gateways, we hypothesise that it is appropriate for their role in industrial plants to evolve with the IIoT from being stand-alone single-purpose devices towards being member of a distributed GSB governed by a SOA. In so doing, gateways may provide a distributed and resilient plant-wide platform for the execution of services and the presentation of semantically-enriched plant data for enterprise-layer applications.

As an example of the potential of such a GSB, consider the industrial Ethernet plant shown in Fig. 1. Assume that each manufacturing cell operates using a different Ethernet protocol and that all cell components are configurable using the IEC 61499 standard. As the IEC 61499 standard requires clear definitions of component interfaces, the GSB nodes may exploit this property to instantiate virtual device representations for the distributed execution of processes across heterogeneous cells. As an example, consider devices 1 and 2 of cells 1 and 3, each of which runs using a different communication protocol. To cooperatively execute a single process, GSB nodes in each of these cells may create a virtual representation of device 1 and 2 in cells 3 and 1, respectively. The GSB may then tunnel or perform wire-speed translations of data on behalf of each of these devices to allow for the distributed execution of the plant process. However, the GSB may also simultaneously map the data to a set of models that semantically-enriches and standardises it. Further devices, such as Device ‘D’ located in the DMZ in Fig. 1, may then act as a portal that allows business-enhancing applications from the enterprise layer to access a standardised form of the data acquired from the myriad of heterogeneous devices operating in the industrial plant. In so doing, the GSB has both allowed for the simplified reconfiguration of heterogeneous manufacturing cells and achieved the primary goal of enhanced vertical integration.

To realise the aforementioned scenario, we theorise that the envisioned GSB will require abilities from the modelling and communication domains. To elaborate, observe Fig. 2 which presents a more detailed view of the use case described above. As has been previously established, the IEC 61499 standard is used to define interfaces that allow cells to access foreign devices through GSB-hosted virtual representations. Effectively, this informs devices of the data types expected and provided by each device. However, to ensure interoperability, and to allow for true measures of reconfigurability, both information and information exchange modelling become necessary requirements for the GSB. The former would provide contents for the latter, which, in turn, would use this information to create representations of real physical devices as well as the data exchange relationships between them. Based on this representation, the various constraints, requirements, and capabilities of each device become clearly discernible and accessible for further use by applications and devices alike. Nonetheless, in order to make certain that these applications and devices are capable of communicating in heterogeneous environments, communication profiles and mechanisms would

Fig. 1. The Deployed Distributed GSB [2]



also be necessary. All together, these features provide an approach that ensures interoperability and vertical integration in industrial enterprises. Based on this concept, the coming section, Section II, will delve further into each of these factors discerning the appropriate considerations and technologies involved in having them form the basis of an architecture for a SOA-governed, distributed GSB. This section is then followed by Section III which concludes the paper along with a description of the current state of the GSB.

II. The Proposed GSB

To clearly communicate the concepts of the GSB from the viewpoint of the design process, this section describes the various components of the GSB. These components are concerned with the modelling of control logic, information, and information exchange, respectively, and the definition of suitable communication profiles and mechanisms.

A. Control Logic Modelling

The importance of the IEC 61499 standard for the operations of the GSB has already been established in Section I; it is therefore the concern of this subsection to instead provide a concise description of the standard. This will serve to ensure that appropriate technologies are selected for the remainder of the GSB's components.

Briefly, the IEC 61499 specification provides a distributed architecture for control based on the standardization of interfaces and components in automation systems. This architecture allows for the portability, reconfigurability, interoperability, reconfiguration, and encapsulation of automation software. These attributes are achieved by requiring that control logic be implemented using modular functions blocks (FB), of which there are three types; the basic, composite, and service interface FB. These FBs encapsulate the logic algorithms and define the event and input data required by said algorithms, as well as the event and output data that they produce for consumption by other FBs. For communication, the IEC 61499 standard defines the communication interface FB (CIFB). The CIFB allows for both TCP client/server and UDP publish/subscribe modes of communication; however, the standard does so without interfering with the networking layer's functions [3, 4] .

Aside from FBs, the distributed model of the IEC 61499 standard also provides definitions for applications, device models, and resource models. Starting with the IEC 61499 device model, as shown by Devices 1 and 2 in Fig. 2, this model is typically used to represent an independent physical component, its interfaces and constituent resources. On the other hand, a resource model, such as Fig. 2's Resource α, represents a functional unit capable of independently controlling its operations and of providing services to applications. In order to provide the functions of a resource with inputs and to share their outputs, resources are given access to a process interface and/or a communication interface. The former provides an I/O interface between the physical process and the resources, while the latter is used for information exchange via communication channels, such as shared memory, domain sockets, and networks. It is important to note at this point that applications are not limited to single resources, but may use one or more resources on a single or on multiple devices [4].

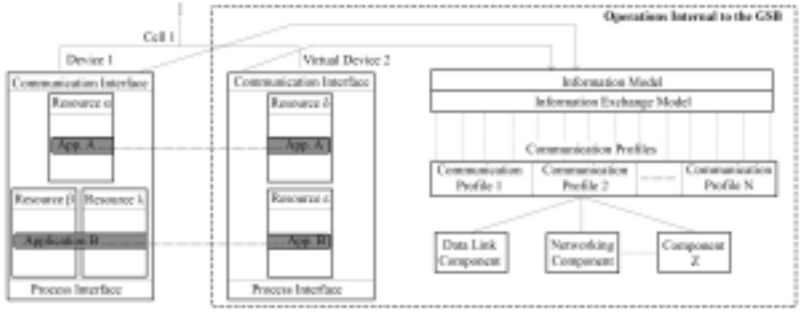
Based on this description of the IEC 61499 specification, the coming section will discuss how this distributed model may be used in conjunction with another standard to allow for information and information exchange modelling.

B. Information & Information Exchange Modelling

The information modelling aspect of the GSB is, at a minimum, expected to provide a standard structure for the presentation of acquired plant data. This is to ensure that information exchange models may then be able to construct an accurate representation of the plant system. Since it is already established that all control logic is modelled using the IEC 61499 specification, it follows that the OPC Unified Architecture (UA) standard is currently the most suitable for the task at hand. This is because the OPC UA standard is the successor to OPC Classic, and, as such, has definitions for the IEC 61131-3 data types. Since the IEC 61499 specification inherits its own data types from IEC 61131-3, the combination of OPC UA and IEC 61499 therefore vastly simplifies the process of achieving a standardised information model for the presentation of data derived from IEC 61499 FBs [5].

In addition to using OPC UA for information modelling, this standard may also be used for information exchange modelling. This is mainly due to the highly configurable address space of OPC UA that is designed to allow for the creation of complex networks of data. Briefly, this address space is composed of *Nodes* and *References*. *Nodes* are the basic units of information in OPC UA. They are used to represent a variety

Fig. 2. A Conceptual Representation of the Proposed GSB



of elements, referred to as *NodeClasses*, which are predefined by the standard and cannot be extended. Each *NodeClass* has its own attributes associated with it. *References*, on the other hand, are pointers connecting *Nodes* in UA, with each *Reference* providing a description on the relation between two *Nodes*. Each *Reference* must therefore have a source *Node* and a destination *Node*. However, *References* must also consist of a direction and a *ReferenceType*, which is used to describe the reference and is used mostly for filtration and organizational purposes [6].

The *NodeClasses* available to OPC UA include the elements *Object*, *ObjectType*, *Variable*, *VariableType*, *DataType*, *ReferenceType*, *Method*, and *View*. An *Object* is a representation of a physical or abstract component of the system. The *Variable* Node contains the value to be held. The *Method* specifies a software function that is a component of the *Object* and that may be called. The *View* Node is a subset of the *AddressSpace* that is of importance to an OPC UA *client*. Finally, the *ObjectType*, *VariableType*, *ReferenceType*, and *DataType* specify the type of the *Object*, *Variable*, *Reference*, and the data of a variable, respectively [6].

The description above is by no means comprehensive, however, the full details of the OPC UA's modelling concepts would be superfluous, and the ones presented above suffice for the establishment of IEC 61499 to OPC UA mappings. To illustrate, it appears that an IEC 61499 resource may be represented as an OPC UA Object. Mappings for the DataType, as has been stated earlier, have already been accomplished by OPC UA and IEC 61499's predecessors. As for the UA Method, this may provide the plant with descriptive representations of resource functions and services. By hierarchically clustering the various Objects and Methods based on the locality of their respective resources, and then using References to establish directional representations of information exchange between them, virtual representations of real devices may be created. This would thereby assist with device interoperability and provide the plant with a pragmatic and representative view of the value and capabilities of each component and the interactions that take place between them.

It is important to note at this point that although the OPC UA standard includes many features other than its information model, such as the definition of a number of services and communication profiles, the GSB's dependence on OPC UA should only be limited to the information model. This is because of two drawbacks to the standard. The first of these is the fact that OPC UA services, such as the discovery service, are not decentralized by design [6]. Embracing the full OPC UA specification would therefore introduce vulnerabilities in the GSB that are typical of centralized architecture, and undesirable in SOAs, such as single points of failure (SPoF) and performance bottlenecks.

The second drawback of the OPC UA standard relates to its communication protocols. Although the OPC UA stack provides several desirable features, such as secure access to web services, its major drawback is in its lack of real-time (RT) capabilities. The OPC UA foundation is currently looking into using the IEEE 802.1 TSN standard as a basis for integrating RT features into its stack; however, as the distributed GSB requires the ability to provide guarantees for real-time environments, this limitation ensures that, in the immediate sense, the GSB must look elsewhere for its communication stack. Consequently, the coming subsection will look into the definition of an outline for the development of an appropriate communication stack for our GSB.

C. Communication Profiles

The general approach outlined throughout this paper for ensuring device interoperability has been predominantly oriented towards the use of protocol translation. Unfortunately, translation only works if the message formats of participating protocols are close, with equivalent information fields; if this is not the case, then resulting translations are incomplete. A second limitation of protocol translation is that it operates within the limits of the protocol's existing standards; standards which were not designed with the IIoT in mind, and, consequently, may lack attributes critical for the safe integration of devices with the IIoT. A case in point is the example of Fieldbus technologies which implement either no or limited security measures [7]. This therefore forces the GSB to consider alternative approaches that are capable of addressing these limitations while achieving device interoperability.

A viable solution is that of tunnelling. Tunnelling applies application level modifications to messages, and then treats the channel as a transparent communication medium. Doing so affords the system the flexibility to include features beyond what exists in currently deployed protocols. For example, this may afford the system features such as secure communication, IPv6 compatibility, and the provision of protocols for service discovery, service description, and group communication for both constrained and resourceful devices. However, tunnelling is not able to completely divorce from the constraints of the plant's protocols. For example, the system may need to tunnel through a network operating using a protocol that has an extremely limited message size. By reserving portions of each message for services-related data, the amount available for actual payload is reduced, which, in turn, may have an undesirable impact on the channel performance. Thankfully, technologies for highly constrained devices and channels, such as the message encoding EXI specification, have been developed as part of the drive for the IoT [8]. Consequently, specifications such as these will have to form the basis of the GSB's communication profiles for the integration of field-level devices operating in legacy networks [7].

Finally, it is prudent to clearly state that it may be the case that not all of the existing devices in a plant will be accepting of the modifications required for it to consume tunnelled communications. The resulting architecture for the GSB would therefore have to be able to provide both tunnelling and translation capabilities.

D. Inter-GSB Coordination

The final aspect involved in the GSB's architecture, is the definition of robust coordination mechanisms for the GSB itself. This is to allow the nodes of the GSB to interact correctly when executing functions and services for the industrial plant. As the GSB network constructs itself over existing physical infrastructure it is, by definition, an overlay network. The most suitable subdomain of overlay networks for the GSB is that of peer-to-peer (P2P) networks. This is mainly due to both the resilience and the services-oriented approach typical of this type of networks. Specifically, P2P networks as cooperative systems may provide the GSB with the ability to create expanded systems of networks, and to perform inter-system traffic engineering, and inter-system content-sharing. This would allow the GSB nodes to safely communicate in a manner that is compatible with the enterprise architecture and its constraints, while also being able to share their resources and cooperatively offer and execute services [2].

However, while discerning the concepts required to successfully apply mechanisms from the field of P2P networks as cooperative systems to a GSB for industrial enterprises, several incompatibilities were found. Preliminary results have shown that the requirements of the GSB differ extensively from those of typical P2P networks to the degree that it has warranted fundamental changes to the conventional network management mechanisms of the domain. Specifically, in [2] it was found that one of the typically used protocols for discovery, which involves the cross-communication of large payloads of routing information on nodes, is detrimental to the performance of the GSB. The current stage of development of the GSB is therefore aimed at improving the speed and efficiency of the P2P-based GSB discovery mechanisms; most likely through the use of zero configuration networking technologies, which, it is hypothesised, would both resolve the aforementioned performance issues while also simplifying the integrability of the GSB with the industrial enterprise [2].

III. Conclusion

This paper has centred on discussing the main factors involved in the design of an architecture for a distributed GSB for vertical integration in industrial environments. The derived requirements influenced the selection of a comprehensive and logically sequenced set of interdependent specifications and standards that would ensure device interoperability and semantically standardized plant data for the simplified accessibility of information by enterprise applications. Furthermore, the limitations of suitable communication solutions were discussed in order to discern that a hybrid solution of tunnelling and translation, and profiles based on compact messaging standards, are required to ensure the integrability of devices and applications with the distributed GSB. Finally, the domain of P2P networks as cooperative systems was highlighted as a suitable solution for inter-GSB coordination. The results of current research in the domain was discussed, highlighting the limitations of core mechanisms from the domain, and suggesting future work to improve their performance.

Acknowledgment

This paper is supported by TU Wien research funds.

References

[1] Peter Adolphs, Heinz Bedenbender, et al. *Reference Architecture Model Industrie 4.0 (RAMI4.0)*. VDI/VDE Society Measurement and Automatic Control (GMA). July 2015.

[2] Ahmed Ismail and Wolfgang Kastner. *Co-Operative Peer-to-Peer Systems for Industrial Middleware*. IEEE WFCS. May 2016. To be published.

[3] Kleanthis Thramboulidis. "IEC 61499 vs. 61131: A Comparison Based on Misperceptions". In: *J. Softw. Eng. Appl.* (July 2013), pp. 405–415.

[4] Valeriy Vyatkin. *IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design*. ISA, 2007.

[5] M. Melik-Merkumians, T. Baier, et al. "Towards OPC UA as portable SOA middleware between control software and external added value applications". In: *IEEE ETFA*. 2012, pp. 1–8.

[6] Wolfgang Mahnke, Stefan-Helmut Leitner, et al. *OPC Unified Architecture*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[7] B.M. Wilamowski and J.D. Irwin. *Industrial Communication Systems*. Taylor & Francis, 2011.

[8] Wolfgang Kastner, Mario Kofler, et al. "Building automation systems integration into the Internet of Things the IoT6 approach, its realization and validation". In: *IEEE ETFA*. 2014, pp. 1–9.