Hästbacka, D., Kannisto, P. & Vilkko, M., "Data-driven and Event-driven Integration Architecture for Plant-wide Industrial Process Monitoring and Control", in 44th Annual Conference of the IEEE Industrial Electronics Society, 2018, pp. 2979-2985.
DOI: https://doi.org/10.1109/IECON.2018.8591323

Data-driven and Event-driven Integration Architecture for Plant-wide Industrial Process Monitoring and Control

David Hästbacka
Laboratory of Pervasive Computing
Tampere University of Technology
Korkeakoulunkatu 1, FI-33720 Tampere, Finland
Email: david.hastbacka@tut.fi

Petri Kannisto, Matti Vilkko
Laboratory of Automation and Hydraulic Engineering
Tampere University of Technology
Korkeakoulunkatu 3, FI-33720 Tampere, Finland
Email: petri.kannisto@tut.fi, matti.vilkko@tut.fi

***Abstract*—Efficiency of industrial processes and a high quality of the products can be achieved with advanced monitoring and control solutions. In addition, industrial processes often consume large amounts of energy and through their efficiency and use of resources also have a significant environmental impact. For industrial process optimisation it is necessary to integrate distributed data and functionality into plant-wide coordinating level solutions. From an implementation point of view this is challenging due to different communication protocols and messaging structures. In this paper an integration architecture is proposed that decouples control systems using a message bus mediator approach. The mediator acts as a unified point of access that through adapters facilitates integration of existing control systems to advanced plant-wide data-driven and event-driven control. It is demonstrated with a laboratory case presenting two examples applicable to real-life industrial problems.**

I. Introduction

Industrial processes are essential for production of materials, food and energy as well as in recycling resources at the end of products' life-cycle. Such processes are typically of large scale, consume energy and resources, but they can be hard to control as many goals can be contradicting. Optimal control and small improvements in efficiency can have a significant global effect and, for example, reduce the environmental impact and use of resources, ensure a desired quality of the product, and maintain sustainable profit margins [1], [2], [3].

The challenges in control are manifold. First of all, the phenomena to be controlled can be complex and often express non-linear behavior requiring advanced mathematical models for control. Distributed sub processes might depend on and affect each other, and for example certain processing in an earlier step might affect what can and needs to be done in a following processing step. Shared resources or even logistics might also pose restrictions on how industrial processing can be optimised. The challenges are therefore not only in control but also in monitoring these processes and making process operators aware of the impact of their decisions. This requires integrating local sub process control models as well as coordinating top level models that integrate parameters, constraints and the detailed sub process model results. To meet these requirements, research on models, optimisation algorithms, HMI interaction, integration of automation systems, and systems engineering is required as also pointed out in [4].

Plant-wide monitoring and control is researched in the European Union H2020 funded project COCOP (Coordinating Optimisation of Complex Industrial Processes) with the aim to increase the competitiveness of the European process industry. The objectives are to increase product quality, improve sustainability, reduce operation costs, and to improve working conditions of process operators. As part of this project a new integration approach for industrial process control systems is being developed and this concept is presented in this paper.

Monitoring and control of such complex industrial processes is heavily dependent on integrating both information and functionality that are distributed in different systems. From a data integration perspective the information systems are typically heterogeneous with different information structures and communication protocols. From the point of view of integrating distributed functionality the diverse communication technologies are also a challenge. In addition, functionality that is composed to larger and more meaningful compositions requires new operational models and a slight paradigm shift towards data-driven and event-driven collaboration. In this paper, data-driven and event-driven refers to the reactive model of software applications acting upon new data observed instead of being invoked by some other system. This supports implementing the plant-wide coordinating control approach in the COCOP approach. The aim is to decompose the entire optimisation problem into sub-problems, and their constraints and results can be communicated using a shared message broker without direct dependencies.

Industrial process control systems and information systems in production typically have a long lifespan. It is common that these systems and the sub processes evolve concurrently. This maintains heterogeneous system interfaces and different kinds of data structures communicated between sub processes and even different process plants that affect each other's operation.

The work presented in this paper facilitates integration of industrial process control systems. The contributions are an architecture with complex requirements that has been developed for scalability and to enable plant-wide monitoring and control while taking into account the existing practices and typical information systems used in the domain. The principal ideas of the developed architecture are loosely-coupled systems, asynchronous communication and event-driven execution. The second contribution are use case examples demonstrating the power of the developed architecture in actual real-life scenarios. A third important contribution of this paper is how the developed asynchronous communication architecture can be aligned with request-response based communication patterns.

In section II related work is presented with a review on other integration and communication architectures developed for industrial automation and control applications. Requirements identified for the integration architecture are described in section III, and the developed architecture enabling data-driven and event-driven asynchronous communication is explained in section IV. In section V the implemented communication platform and a laboratory case example is presented. The paper is summarised with a discussion and conclusion in section VI.

II. Related work

Cyber-physical systems (CPS) support human employees in production-related tasks [5]. In contrast to CPS, traditional industrial automation operates on a low abstraction level. For instance, PID controllers provide means for automatic positioning, temperature control, level control in containers and so forth. Instead, CPSs take advantage from the increasing intelligence of computer systems and their constantly improving user interfaces [6], [7]. In various countries, there are CPS-related initiatives to advance industrial production, such as those of Industry 4.0 and Smart Manufacturing.

The application of software systems and advanced computational models enables the optimisation of production tasks in a manner that is not possible with traditional methods, i.e. those that do not communicate between sub processes nor make the impact of decisions beyond local control evident to the human operators. This has also been noted by [8] in research on dynamics and control of chemical process networks. In general, control has evolved from local control loops to real-time optimisation but scalability is still considered a challenge [9]. According to [10] the implementation of cyber-physical systems requires agents to achieve distributed autonomous intelligence and adaptation, services as the enabler for interoperability, and cloud technologies for more efficient use of resources and big data analytics.

Industrial communication frameworks and platforms have been developed in several previous research undertakings. The SOCRADES approach focused on a collaborative manufacturing model using web services in a service-oriented architecture (SOA) [11], [12]. IMC-AESOP studied a SOA approach for monitoring of large-scale process control systems [13], [14]. The Arrowhead framework provides a SOA infrastructure for collaborative IoT automation with focus on development, deployment and operation [15]. It has also been developed to support so called local clouds [16] useful in production settings and other restricted environments. While Arrowhead decouples service providers and consumers it still relies on interoperable interfaces whereas the approach presented in this paper takes it even further, i.e. primarily relying on agreed message structures and messaging patterns using a mediator.

Middleware solutions suitable for industrial environments have been studied by [17] as part of the PERFoRM project aiming to develop a common reference architecture for Agile Manufacturing Control systems. In their reference architecture they propose that protocols such as REST, SOAP and technologies like OPC UA are of importance [18]. In comparison, the concept promoted in this paper also advances event-driven interaction that, for example, instead of direct invocation requires other system components to react upon new data.

III. Identified architectural requirements for scalable plant-wide integration

Plant-wide industrial process control and monitoring applications face challenges from system complexity as well as networked integrations of systems from multiple disciplines. The control systems controlling a process are typically application specific and often also with vendor-specific features. This makes plant-wide integrations difficult to implement especially as production processes may span beyond one single plant and integrations are required in order to optimise production beyond the local processes. Also, a traditional periodic control approach, that scans or queries for values and based on this decides the control actions, may prove very complex and too rigid to adapt new control configurations. From an implementation point of view it can turn out laborious if several point-to-point connections to different systems need to be implemented and maintained.

In addition, integration requires agreed data structures and semantics so that information can be communicated in an way understood by the parties. Information models for the proposed architecture have been developed that facilitate the unambiguous transfer of measurements, events, parameters and set-points as well as results calculated from simulations and control algorithms. The detailed requirements and design of these models and structures are, however, out of the scope of this paper. From the architecture point of view they are considered as message structures containing all necessary information without relying on any specific metadata from the underlying transport protocols.

The following subsections will examine the requirements from decoupled systems point of view and loose coupling, typical features of typical process plants, and integration to existing systems used in the domain. These general requirements presented in this paper have been derived from two pilot use cases examined as part of the COCOP project.

A. *Decoupled systems for scalability and performance*

Large-scale industrial processes need integration solutions that are scalable and enable extensive utilisation of data. Point-to-point integrations should be avoided which might otherwise lead to a high number of direct dependencies between systems. Future maintenance issues can easily appear as one or more systems are changed, updated or replaced in the future. In order to make use of sensor raw data or already refined data it is beneficial to decouple the information producers from the consumers, e.g. using a message bus or message queues as a broker of data and events.

From a performance viewpoint, scalability can be achieved by removing redundant queries to the lower system levels by having a mediator in between the information producer and the consumer. This mediator can then be scaled according to need, and not depend on the capabilities of individual subsystems or devices if more requests to their data is needed.

It is worth mentioning that such a centralised mediator can introduce a single point of failure. This can be mitigated with preventive measures to ensure reliability, e.g. fail-over redundancy or restrictions on how plant-wide monitoring and control applications operate in case of communication failure.

Loose coupling and an environment where different modules interact with each other without knowing their internal details is desirable. Systems should exchange data and request functionality on such interfaces that do not expose the underlying implementation technology. From an integration perspective the modules should be able to operate with minimal direct dependencies to each other, relying mainly on data provided or received.

A trade-off of the mediator approach is that producers of data and events do not know who and how information is utilised. As a result, additional information security measures are needed in the mediator or in the plant-wide applications built on top of the mediator. This is especially the case in environments where devices and systems from competing vendors operate and exchange data. The Internet enabled nature in paradigms such as Industrial Internet and Industry4.0 makes cyber-security an even more important aspect [19].

B. Reactive control applications and development of new distributed control functions

Reactive, event-driven actions are desired especially from a control point of view. For example, when a process step finishes it may trigger the start of another. Advanced calculations may need to be performed to optimise following control actions, i.e. on that particular batch or product. From a systems point of view it is not practical to have direct point-to-point integrations between the different systems, i.e. those controlling the following processing steps. Instead, when data and events are made available using a common platform the dependant systems can choose how and when to react upon such new information.

This shifts the traditional programming from invoking methods or starting remote functionality to a reactive model in which the dependant party observes new data and executes functionality as it sees most appropriate. This kind of a model is desired especially for development of new plant-wide monitoring and control solutions, design of advanced control methods, and in cases where process units are being updated and when systems or sub processes are being added.

From a development perspective the burden in point-to-point integrations is on connecting systems with a multitude of different communication interfaces. Development of adapters is needed for the mediator based approach as it is not intended to replace existing control systems. Rather integrate them into the new control environment. This should also be possible system by system. Also in practice, full replacement of industrial control systems is typically not feasible as it is expensive and creates a significant hurdle towards implementation with new risks with potential production losses.

Distribution is typical to industrial production environments, and a production plant can consist of several factories each having a dedicated task. A relaxing characteristic of industrial process plants is that the physical facilities are typically static and do not include physical reconfiguration with the exception of modernisation and revamping. The information available for control purposes, on the other hand, may vary more frequently. An example of this is estimated calculations versus laboratory measurements which might arrive at different times but can provide differing accuracy. Same applies to complex data analytics processing that, for example, is calculated on an hourly basis.

TABLE I. INFORMATION AND CONTROL SYSTEMS THAT NEED TO BE INTEGRATED.

System	Description of information
DCSs, PLCs	(Soft) real-time measurement values and parameters from production control processes.
HMIs	User interface information displayed to operators, e.g. using OPC UA or HTTP(S)
LIMS, quality control	Measurements, samples, analysis results, restricted resources
MES	Plant-level production coordination; plant-level production-related information, scheduling restrictions, availability of resources
ERP	Enterprise-level production coordination, resources and financial costs
Logistics	Constraints related to material flows, transport capabilities

C. Control systems and other information systems used with industrial processes

Table I lists some of the relevant systems for implementing advanced plant-wide process control functionality that may span beyond the premises of one plant. DCSs and PLCs read measurements from the process and control the process using actuators. These values need to be read to plant-wide monitoring and control applications and, similarly, parameters and set-points for control functions need to be set. Communication to these devices can be implemented in a multitude of different ways, and, depending on the vendor, implemented e.g. using OPC DA, OPC UA, Modbus or vendor-specific application programming interfaces (API).

HMIs in control rooms and mobile devices are essential in monitoring and control as it is the means for operators to follow and steer the processes. When a plant-wide concept is applied it is of grand importance to communicate the coordinating level optimisations and to make the operators aware of the impact of their decisions and actions. Modern HMI solutions support retrieving data using several kinds of protocols and fieldbuses, including OPC DA and OPC UA, as well as receiving it from message buses.

Laboratory Information Management Systems (LIMS) provide information that is related to quality control, increasing the yield and reducing the amount of waste. These systems are diverse and application specific. The produced information is often delayed but may provide relevant and more accurate information for the next processing steps. It is also common that this information is queried direct from databases.

Manufacturing Execution Systems (MES) are scheduling production and handling daily and hourly production related information on the site, e.g. orders, available raw materials and other resources. Enterprise Resource Planning (ERP) systems, on the other hand, deal with enterprise-level information. From a plant-wide process optimisation point of view, they can provide restrictions and parameters for local optimisation, e.g. cost related parameters or long-term production data. Similarly, logistics form an important part of distributed production as both internal and external logistics factors might affect optimisation of production, e.g. delivery of raw material and transport capability of intermediate products. These systems are typically integrated using interfaces and protocols including those such as SOAP or RESTful web services.

To cope with these multiple systems and their heterogeneous interfaces an adaptation layer is inevitable in order to efficiently implement new advanced plant-wide monitoring and control solutions. As many of the aforementioned systems are restricted, and not allowing new functionality to be developed, it is often required to implement adaptation logic in the form of an adapter between those systems and the information mediator on which information is transferred.

IV. COCOP Message Bus Mediator Architecture

The presented architecture focuses on the integration between industrial control systems and other related information systems rather than how these systems should be implemented. At its core is a message bus based approach brokering data and event messages. Thus, the approach does not provide engineering tools, run-time of controllers or similar concrete applications. In its essence it is an intermediary communication architecture for accessing events and data. It is designed applicable and adaptable to real industrial environments making most of already in-use control systems when implementing advanced plant-wide monitoring and control solutions.

A. *Conceptual View*

To meet the requirements presented in section III, the integration architecture provides multiple features related to production tasks. To increase the efficiency of production, software-based optimisation aids everyday plant operation. Software systems enable both data processing in large volumes and the application of complex mathematical algorithms. However, for their operation, the optimisation software also require connections to the actual measurement data that production-related information systems provide. Considering the outcome of optimisation, there must be a connection of the user interfaces of process operators. This requirement stems from the undesirability of full automatic control of complex production processes. Instead, optimisation modules should also provide instructions to aid process operators. The operators can then use their expertise to decide whether to follow optimisation suggestions or not while also keeping them aware of the reasoning behind control actions.

From the ICT point of view, there are further features. The messages structures use openly specified formats to save specification effort and facilitate the integration of information systems. In addition, the communication enables event-based operation. Only then, the optimisation software may react to the actual state changes of production processes. However, as the platform is designed to have a long life cycle, the installation and maintenance of the connected systems is crucially important. Therefore, the platform enables loose coupling in integration to reduce any direct runtime dependencies between systems. With few dependencies, any changes or fresh installations are straightforward.

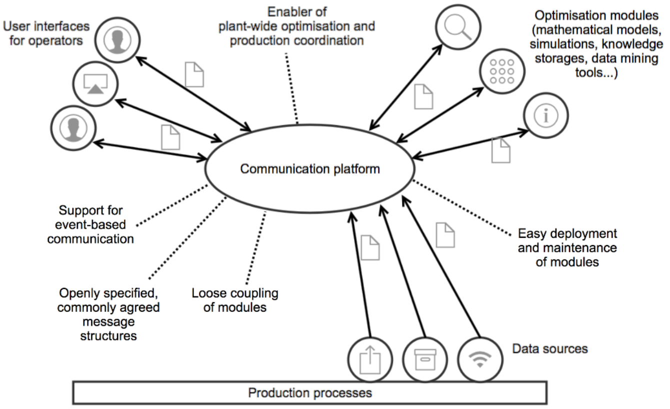
User interfaces for operators
Enabler of plant-wide optimisation and production coordination
Optimisation modules (mathematical models, simulations, knowledge storages, data mining tools...)
Communication platform
Support for event-based communication
Easy deployment and maintenance of modules
Openly specified, commonly agreed message structures
Loose coupling of modules
Data sources
Production processes

Fig. 1. The features of the integration architecture enabling plant-wide monitoring and control of industrial processes.

Figure 1 illustrates the earlier explained features. The platform provides a communication medium that is generic and flexible but still provides a solid foundation to conform to the concrete communication-related requirements.

B. *Adapters and state information considerations*

One of the core principles is to support the use of existing control systems when developing new plant-wide functionality using the approach. Adapters are therefore required to adapt existing systems to the message bus based approach. Simple message bus implementations transmit messages received from producers either to all consumers or only those that have registered for a particular message. More advanced implementations can include complex routing, caching and access management. Some implementations even store messages for a short duration and may, for example, retain the most recent messages for new consumers to receive. This is beneficial for data or events that are rarely updated.

Implementing the event-driven model with traditional systems can be implemented in several ways using adapters that wrap existing systems to new interfaces:

1) all operational data is periodically updated (even if no changes occur),
2) most recent messages are retained (as some events are rare), or
3) there are means to also query recent data and events.

The first method is easy to implement but introduces unnecessary data and requires consumer applications to know when to process the new data. The second option of retaining recent messages is only supported by some message bus implementation standards but the model would reduce consumer and producer adapter application complexity. This can also be implemented as a cache periodically echoing data and events. The third option provides most versatility by enabling request-response messaging but incur additional functional requirements for adapters realising the query capabilities.

C. *Supporting Client-Server Communication*

Many existing industrial control systems and HMIs rely on periodically polling values or on request-response messaging in

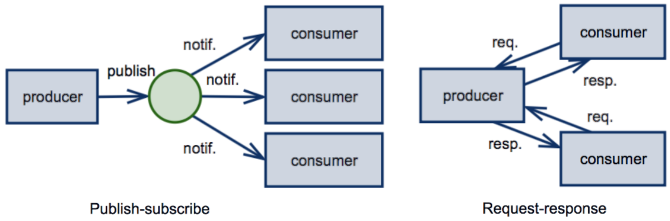

Fig. 2. Publish-subscribe decouples producers and consumers of data.

comparison to a publish-subscribe messaging pattern, as illustrated in Figure 2. To realise client-server communication with a message bus as the core, new functionality for the adapters or systems directly integrated to the COCOP architecture is required. The adapters push new information onto the bus but also need to listen for specific requests from others, and then be capable of returning the requested information onto the bus. This is illustrated in Figure 3 and elaborated in section V.

Regarding long-running tasks that require request-response messaging, the out-of-the-box functionality of typical protocols (such as HTTP) is insufficient. The most common request-response techniques have been designed for occasions where servers provide responses within a few seconds at maximum. However, such a short response time is unreachable in complex optimisations. A response time of hours is typically infeasible even from the human user point of view while a response time of a few minutes is often acceptable but still too long for the requesting client application.

Then, to implement the communication of long-running tasks, a straightforward approach is to use one client request to start a task and then poll for its progress and results from the client. However, frequent polling is inefficient regarding network traffic, and long polling intervals cause unnecessary delays. Instead, a more sophisticated approach is to implement an asynchronous pattern where the client provides a callback interface. The client would then receive notifications from the server. Regarding HTTP-based communication, the WS-Eventing specification [20] provides such functionality, although it would also bring additional complexity. On the other hand, OPC UA also provides a subscription mechanism to provide status information from a server to clients [21].

OPC UA is a de facto protocols used for industrial systems integration, and also on the roadmap of the proposed architecture. It is envisioned a possible wrapper for the message bus as illustrated in Figure 3. Recently, OPC UA has also defined a specification for publish-subscribe communication where a message bus can be used as a broker [22].

V. Implementation and Laboratory Case Example

The implementation supports two messaging patterns, namely publish-subscribe and request-response. Publish-subscribe implements the ”push” communication model where the source of the information makes the information available for any subscribers [23, p. 339]. The subscribers subscribe for any subjects or topics they are interested in. The approach is scalable when there are multiple consumers for the same information and they want the information whenever it changes or becomes available. In contrast, request-response is a pattern where the consumers of information request for it whenever they need it, therefore using the ”pull” communication approach [24]. Figure 2 illustrates this.

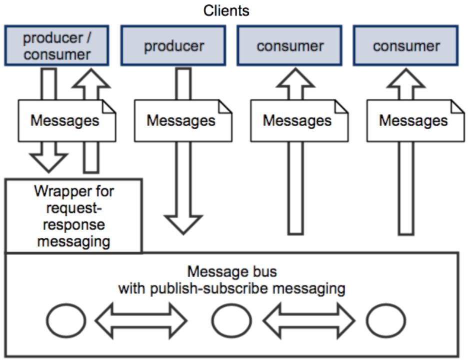

Fig. 3. The COCOP integration approach supports communication either using a message bus or via a request-response wrapper.

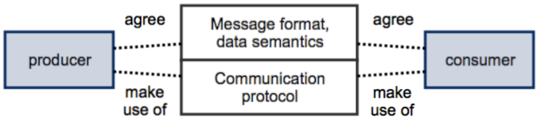

Fig. 4. Message structures and communication protocol are separated by design in the integration architecture.

Figure 4 illustrates the fundamental design approach of separating the utilised message formats from the communication protocol. The approach enforces a ”separation of concerns”, which is advantageous. On one hand, during design, separation helps to concentrate on one aspect only. In message design, that aspect is message structures, and in the design of communication functionality, the aspect is the actual communication mechanism. On the other hand, separation enables both flexibility and adaptability. This advantage realises whenever there is a need to change (or even replace) either the message formats or the communication protocol. For instance, in the future, more advanced message bus protocols likely appear. Moreover, another advantage is to enable multiple communication protocols together, such as AMQP and HTTP. AMQP excels at event-based messaging, whereas HTTP provides more straightforward integration in request-response scenarios. In practice, the separation is realised by implementing a dedicated software library for messaging without introducing any dependencies to communication protocols.

Certain technical specifications have been chosen to form the contract of communication, namely AMQP for the communication protocol and certain messaging standards. AMQP provides native tools for publish-subscribe communication. The currently implemented messaging standards are O&M (Observations & Measurements) [25] and related, such as SOS (Sensor Observation Service) [26]. Although developed in the geospatial domain, O&M and SOS are generic thus providing a suitable basis for industrial production as well.

Although the specifications form the basis of communication, APIs have been implemented to facilitate integration tasks. In principle, it is sufficient for any network node to implement the standards to communicate with other nodes.

However, application development becomes easier if an API or its software development kit (SDK) performs generic communication-related tasks. Then, the application developers may concentrate on the actual application logic instead of communication-related nuances.

Currently, the API SDK has been implemented in C#, and a part of it also exists in Java. The implementation consists of three software components that perform the following tasks:

1) Communicate with the messages bus (native RabbitMQ library)
2) Serialisation of messages
3) Facilitate request-response messaging

To communicate using the API SDK, applications or systems that connect with the message bus use component 1. However, the protocol-level connection is of little use without common message structures. Component 2 implements the creation and interpretation of messages in O&M and SOS. In the future, additional standards, such as ANSI/ISA-95 [27], may be included as needed. Component 3 stems from the need for request-response communication. AMQP has not been designed for the request-response pattern, and so the pattern requires additional design. To save effort of application developers, a software component has been implemented. Clients connect to the message bus, communicate with messages and utilise a separate request-response component when needed. Although the current communication protocol is AMQP, the request-response wrapper could use HTTP in the future, if needed.

Utilising the communication API (i.e., the SDK and message bus), two demos have been implemented. Both demos utilise the message bus and the standard-based message formats. The first demo implements the publish-subscribe pattern. The demo is related to the refinement of copper, and, in particular, the communication between the control rooms of two unit processes. The first unit process provides batches for the second. As the receiving process must further refine the batches, it should have information about the composition of the batches (to reduce the loss of copper and to remove enough waste minerals; see figure 5). In the demo, the first unit process submits batch composition estimates to the second. In contrast, the second demo implements the request-response pattern. In this demo, a server provides temperature values whenever the client requests for it, i.e. for performing advanced control calculations and predictions. Both the demos shown perform typical production-related information exchange. In the first demo, information is submitted to the receiver once it becomes available, while the second demo has an on-demand approach in information delivery.

The demonstrations indicate how the messaging API facilitates communication. The implementation of the API has required careful work, but now that it exists, the creation and interpretation of the messages is realised with only a few lines of code. The demos show the power of the concept, as the producers and consumers of information may be run on any vendor-specific platform, but a single communication contract is provided a channel for information exchange.

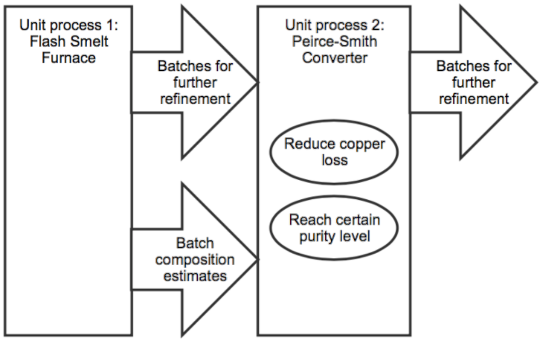

Fig. 5. The scenario of the demo where a batch composition estimate received from one unit process aids refinement in another.

VI. Discussion and Conclusion

General requirements were outlined for an integration architecture with the aim to facilitate development of plant-wide monitoring and control applications for industrial processes. An integration architecture was presented based on a message bus approach separating the message structures from the transportation layer. A laboratory use case with two examples was presented demonstrating the power of the approach. The chosen technological approaches have appeared feasible to implement an adaptable and scalable communication architecture for the communication needs of industrial plants.

The integration architecture decouples information producers from consumers. This is especially beneficial for developing coordinating level control applications as well as local control applications that e.g. benefit from data from other process units. The centralised message bus provides a uniform channel to data and events reducing the number of point-to-point integrations to heterogeneous system interfaces. The new layer also facilitates upgrading of individual systems as consumers do not need to adapt to changes in the source information system.

Adapters are required for connecting existing systems to the message bus architecture. The message bus can be seen as a mediator onto which data and events are pushed but also as a source which needs to be observed by the consumers. This has an impact on the way typical polling or request-response behaving systems need to be integrated. The concept of request-response wrappers was presented that could be applied to RESTful as well as OPC UA based connectivity over the message bus. Request state information may then be managed either directly in the adapters or as a combination of state information in the client application and data on the message bus.

The architecture causes some overhead in implementation. However, the overhead is compensated with improved scalability and easier maintenance of systems integration. The easier maintenance stems from the system-to-system dependencies that only exist on the logical level of agreed messages and messaging patterns instead vendor-specific information structures. These features enabled by a data and event driven approach are desirable for implementation of new control algorithms utilising massive amounts of data. The mediator approach may introduce a minor additional delay compared to direct invocations between system components. This, however, is of minor concern as coordinating level control typically does not incur hard real-time requirements.

In the future, the architecture could be experimented with actual production-related information systems. Currently, only prototypes have been connected, albeit they already demonstrate the core functionality of the architecture concept. OPC UA connectivity will also be aligned to the architecture to further facilitate integration and adoption to existing systems.

Acknowledgment

The work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723661. This study reflects only the authors' views, and the Commission is not responsible for any use that may be made of the information contained therein. The authors want to express their sincere gratitude to the funder and project partners in the COCOP project (Coordinating Optimisation of Complex Industrial Processes, https://www.cocop-spire.eu/). Furthermore, the authors are also grateful to the Academy of Finland for their funding supporting this work (grant 310098).

References

[1] F. Lamnabhi-Lagarrigue, A. Annaswamy, S. Engell, A. Isaksson, P. Khargonekar, R. M. Murray, H. Nijmeijer, T. Samad, D. Tilbury, and P. V. den Hof, "Systems & control for the future of humanity, research agenda: Current and future roles, impact and grand challenges," *Annual Reviews in Control*, vol. 43, pp. 1 – 64, 2017.

[2] F. Lima, S. Li, G. Mirlekar, L. Sridhar, and G. Ruiz-Mercado, "Chapter five - modeling and advanced control for sustainable process systems," in *Sustainability in the Design, Synthesis and Analysis of Chemical Engineering Processes*, G. Ruiz-Mercado, , and H. Cabezas, Eds. Oxford: Butterworth-Heinemann, 2016, pp. 115 – 139.

[3] K. J. Åström and P. Kumar, "Control: A perspective," *Automatica*, vol. 50, no. 1, pp. 3 – 43, 2014.

[4] S. Engell and I. Harjunkoski, "Optimal operation: Scheduling, advanced control and their integration," *Computers & Chemical Engineering*, vol. 47, pp. 121 – 133, 2012, fOCAPO 2012.

[5] P. Fantini, M. Pinzone, and M. Taisch, "Placing the operator at the centre of industry 4.0 design: Modelling and assessing human activities within cyber-physical systems," *Computers & Industrial Engineering*, 2018.

[6] C. Liu, S. Cao, W. Tse, and X. Xu, "Augmented reality-assisted intelligent window for cyber-physical machine tools," *Journal of Manufacturing Systems*, vol. 44, pp. 280 – 286, 2017, special Issue on Latest advancements in manufacturing systems at NAMRC 45.

[7] M. Krugh and L. Mears, "A complementary cyber-human systems framework for industry 4.0 cyber-physical systems," *Manufacturing Letters*, vol. 15, pp. 89 – 92, 2018, industry 4.0 and Smart Manufacturing.

[8] M. Baldea, N. H. El-Farra, and B. E. Ydstie, "Dynamics and control of chemical process networks: Integrating physics, communication and computation," *Computers & Chemical Engineering*, vol. 51, pp. 42 – 54, 2013, cPC VIII.

[9] A. Konstantellos, "A short overview of control in european r&d programmes (1983-2013): From local loop designs, through networked and coordinated control, to stochastic, large scale and real time optimization systems," *European Journal of Control*, vol. 19, no. 5, pp. 351 – 357, 2013, the Path of Control.

[10] P. Leitão, A. W. Colombo, and S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges," *Computers in Industry*, vol. 81, pp. 11 – 25, 2016, emerging ICT concepts for smart, safe and sustainable industrial systems.

[11] A. Colombo and F. Jammes, "Integration of cross-layer web-based service-oriented architecture and collaborative automation technologies: The socrades approach," *IFAC Proceedings Volumes*, vol. 42, no. 4, pp. 2101 – 2106, 2009, 13th IFAC Symposium on Information Control Problems in Manufacturing.

[12] A. Cannata, S. Karnouskos, and M. Taisch, "Evaluating the potential of a service oriented infrastructure for the factory of the future," in *2010 8th IEEE International Conference on Industrial Informatics*, July 2010, pp. 592–597.

[13] A. W. Colombo, T. Bangemann, and S. Karnouskos, "Imc-aesop outcomes: Paving the way to collaborative manufacturing systems," in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, July 2014, pp. 255–260.

[14] D. Mora, M. Taisch, A. W. Colombo, and J. M. Mendes, "Service-oriented architecture approach for industrial s̈ystem of systems:̈ State-of-the-art for energy management," in *IEEE 10th International Conference on Industrial Informatics*, July 2012, pp. 1246–1251.

[15] P. Varga, F. Blomstedt, L. L. Ferreira, J. Eliasson, M. Johansson, J. Delsing, and I. M. de Soria, "Making system of systems interoperable – the core components of the arrowhead framework," *Journal of Network and Computer Applications*, vol. 81, pp. 85 – 95, 2017.

[16] J. Delsing, J. Eliasson, J. van Deventer, H. Derhamy, and P. Varga, "Enabling iot automation using local clouds," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Dec 2016, pp. 502–507.

[17] F. Gosewehr, J. Wermann, and A. W. Colombo, "Assessment of industrial middleware technologies for the perform project," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2016, pp. 5699–5704.

[18] N. Chakravorti, E. Dimanidou, G. Angione, J. Wermann, and F. Gosewehr, "Validation of perform reference architecture demonstrating an automatic robot reconfiguration application," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, July 2017, pp. 1167–1172.

[19] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat, "Cyber-physical systems and their security issues," *Computers in Industry*, vol. 100, pp. 212 – 223, 2018.

[20] D. Davis, A. Malhotra, K. Warr, and W. Chou, "Web services eventing (WS-eventing)," W3C, W3C Recommendation, Dec. 2011.

[21] *OPC Unified Architecture Specification Part 1: Overview and Concepts v.1.01*, OPC Foundation, 2008.

[22] *OPC Unified Architecture Specification Part 14: PubSub v.1.04*, OPC Foundation, 2018.

[23] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-oriented Software Architecture: A System of Patterns*. John Wiley & Sons Ltd., 1996.

[24] "Transport message exchange pattern: Single-request-response," https://www.w3.org/2000/xp/Group/1/10/11/2001-10-11-SRR-Transport_MEP (accessed 7 Jun 2018), 2000.

[25] "Observations and measurements. Version 2.0," http://www.opengeospatial.org/standards/om (accessed 7 Jun 2018), Open Geospatial Consortium, Standard, 2011.

[26] "Sensor observation service," http://www.opengeospatial.org/standards/sos (accessed 7 Jun 2018), Open Geospatial Consortium, Standard, 2012.

[27] "ANSI/ISA-95.00.01-2010 (IEC 62264-1 mod). Enterprise-control system integration – part 1: Models and terminology," ISA, Research Triangle Park, NC, USA, Standard, 2010.