Assessment of industrial middleware technologies for the PERFoRM project

Frederik Gosewehr, Jeffrey Wermann and Armando Walter Colombo
Department of Electrical Engineering and Informatics
University of Applied Sciences Emden/Leer, Germany,
Email: {jeffrey.wermann, frederik.gosewehr}@hs-emden-leer.de, awcolombo@ieee.org

***Abstract*—To meet the increasing requirements which are set for modern production systems regarding flexibility, re-configurability and collaborative behavior, a common information, Communication & Control (ICT) platform for guaranteeing connectivity has to be put in place. Industrial Middleware, i.e. SW/HW solutions, which help connecting and mediating different industrial cyber-physical components of a distributed system (such as a production system) are an essential component of such a platform.**

The paper describes the assessment of existing industrial Middleware solutions which are targeting re-configurability and connectivity of modern industrial environments. It states a set of important functional requirements which need to be considered to select the right solution for individual use cases. A big emphasis is put into finding industrially applicable solutions.

***Index Terms*—Middleware, Assessment, Industry 4.0, ICT**

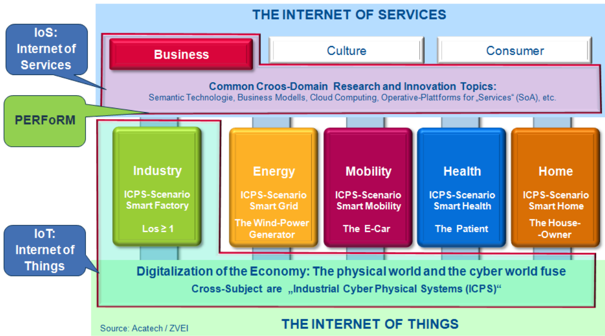

Fig. 1. The Industrie 4.0 Platform concept and the position of PERFoRM objectives for guaranteeing Re-Configurability and Connectivity using Industrial Cyber-Physical Systems.

I. Introduction

Companies of the manufacturing domain are currently facing various demand, structural and functional requirements, for their production systems. Clients are demanding new products with increased quality and functionality within shorter production times and for decreasing prices. Furthermore, a increasingly important factor is the request for customization, leading to production setups which have to be designed for lot sizes down to one. Meeting these requirements is a challenging task itself, but companies also need to take into account economic aspects, which can be heavily influenced by down-times of the production and necessary re-configuration.

As an innovation project, one of the major objectives of the HORIZON2020 FoF Project PERFoRM (Production harmonized Reconfiguration of Flexible Robots and Machinery) is the consolidation of existing research and development results. It's addressing the major requirements to introduce the next generation of agile manufacturing systems into the industrial world. To achieve this, it is necessary to address a considerable set of obstacles, which are delaying the breakthrough of new technologies like IoT, SoA, MAS, IoS, etc. This can for example be done by developing ICT-solutions to support connectivity and migration strategies for an industrial shop floor to become re-configurable.

An important role to approach the requirements addressed above and to solve scientific and technological gaps poses the increasing digitalization of industrial systems. Through interconnecting the different components of an industrial system and realizing the bilateral communication between them, a lot of the aforementioned problems can be addressed. When machines are able to send messages about their current status, possible errors or needs for maintenance to production planning systems, this information can be used to schedule the production in a way to reduce down-times. Among various other examples, it is also possible to use these ways of communication to easily reconfigure production systems to increase the flexibility, e.g. allowing for more customized products. This requires the production to be set up in a structured way, following an architectural design which meets all the different requirements that such a reconfigurable production system can have.

The digitalization of the shop floor, as proposed by PERFoRM, is an approach that follows completely the major characteristics of the "Industrie 4.0 platform" [1]. The physical (HW / mechatronics) and cyber (SW) views of the shop floor components, which are functional modularized for guaranteeing, among others, re-configurability, are melded into one entity. These entities can be recognized as Industrial Cyber-Physical Systems (ICPS) [2]. ICPS are the cross-scientific and technological subjects, that communicate and inter-operate using e.g. Internet-of-Things technology, by exposing and/or consuming Services in an operative Internet-of-Services platform, as depicted in Figure 1.

Another important issue to be taken into account during the specification and development of the ICT-Platform is the use of communication and information technology stan-

978-1-5090-3474-1/16/$31.00

dards. This is especially the case regarding communication, where syntactic and semantic aspects, such as a common language for all cyber-physical components, is unavoidable. These components are always customized and developed for very specific purposes or restricted to certain domains. This often leads to specialized communication protocols, making an interconnection to other components outside of their domain more difficult, if not impossible. A typical example for this is the connection of automation devices, such as Programmable Logic Controller (PLC), to Manufacturing Execution- (MES) and/or Enterprise Resource Planning Systems (ERP). The former already have various bus system solutions, allowing the exchange of simple field data in a way, where even hard real-time requirements can be met. MES and ERP on the other hand have less demand regarding the hard real-time behavior, but have to deal with a much bigger set of more complex, already enriched data and information. The goal is to find a standardized way for connecting every component to be used via a common syntax and semantics.

In the context of the project PERFoRM, a crucial part of the innovation work is the definition, specification and prototype implementation of a reference architecture, including a standard interface for connectivity. Moreover, a benchmarking/evaluation of current used industrial middleware solutions has also been started and and is reported in this paper. Note: as an innovation action, it is especially important to find existing solutions, which are or will be accepted by the industry. Solutions, which only have been tested at laboratory level and haven't been proven to be mature enough to sustain in real industrial use, won't be considered.

As depicted in Fig. 1, the PERFoRM middleware is responsible for facilitating the data and information exchange, i.e. transfer between cyber-physical components located on the shop floor and those located on the business level. Furthermore, it is able to translate the information in a way that the communicating nodes understand. This means, a syntactic translation, e.g. from one protocol to another, but especially also a semantic one.

After the introduction in Section I, Section II is going to briefly present the PERFoRM approach and overview the state-of-the-art concerning HW/SW middlewares. Section III presents the major results of an evaluation of a small set of known middleware solutions. Describing the basic requirements and properties, that are used to evaluate the different solutions, this section will show the pre-evaluation that has been carried out as a first step of the whole assessment process. The three most promising candidates are being selected as a result of this pre-evaluation. Section IV concludes the results of the assessment and outlooks future extensions of the performed work.

II. Overview of the State of the Art

A. Industrial Middleware

A middleware in general is a software component, mainly embedded in adequate hardware components, targeted at connecting different applications or systems in a way where a communication between the involved actors is possible without them having to know about their inner structure and specific interfaces [3]. Therefore, the middleware is essential for the connectivity, acting as a common interface between these systems, able to receive and translate data from each component and forward it to other components.

A middleware can be classified in different ways [4]:

- Application oriented: the middleware is focused on supporting specific, usually distributed applications.
- Communication oriented: the middleware is focused on supporting specific communication technologies, e.g. RPC, Web Services, etc.

Middleware solutions are used in various different applications, such as in mobile devices [5], home entertainment and automation [6], car-to-car communication [7] and Internet-of-Things (IoT) [8].

The capability of being able to interconnect different systems with each other is making Middleware solutions increasingly interesting for industrial applications. In various research projects, different aspects of a Middleware have been evaluated regarding their usability in industrial applications [9]. The different solutions include the development of dedicated software, which is acting as the Middleware in the classic way, as well as existing Manufacturing Execution- and SCADA systems, which functionally already act as intermediary systems between field devices and management systems, implementing standardized interfaces to both worlds to create an integrated data flow [2].

Another solution is the design of a so called Enterprise Service Bus, where each component is providing standardized interfaces, allowing to access data from each component using a service-oriented approach. This way, typical hierarchies are broken up, so that all components are able to access the services of each other, having the specific syntax and semantics of the service as common language and not needing additional software to translate.

Existing middleware solutions which are also interesting to be used in industrial applications or which are even especially designed for this domain, are already purchasable. A set of these solutions are listed and discussed in Chapter III.

B. PERFoRM approach

One of the innovation goals of the work done in the project PERFoRM is to evaluate existing industrial middleware technologies and solutions to be able to derive specifications, which will help industrial companies to introduce fitting middleware solutions into their production systems.

As depending on the domain and other properties of a manufacturing system the requirements for such a middleware can change. The work is not focusing on a specific solution, but instead specifying a middleware component/system architecture which is able to use different solutions as its core and enhancing these solutions with additional functionalities as add-ons (see Figure 2).

Some of these functionalities are: (i) to be able to send and receive data, using the standard interfaces; (ii) to implement

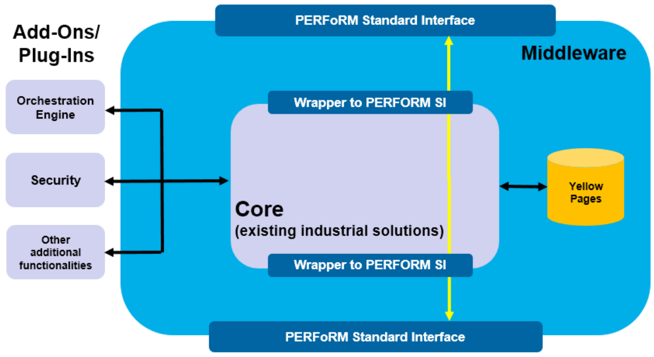

Fig. 2. structure of the PERFoRM middleware

services to add additional middleware functionalities in a modular way, as well as (iii) the ability to connect to and interact with other middleware implementations.

III. Middleware Solution Evaluation

The following chapter is describing the pre-evaluation of middleware solutions which has been carried out as an initial step for the work in PERFoRM's "Middleware & Standard Interfaces" work package. First, the requirements and properties, which have been used to compare the different solutions, will be discussed. A table is showing the assessed results of the evaluation, which are further evaluated to select a sub-set of the most promising solutions.

A. Requirements & functionalities definition

Every PERFoRM middleware candidate has to fulfil a number of major requirements and functions to be even considered in the first place. These requirements are crucial to connect with the outer layer of the PERFoRM middleware architecture in a standardized way. They are as follows:

Requirements:

- Transfer Data & Information: data routing and transformation of different protocols.
- Translate Data & Information: transformation of the same ISO layer protocol but with different payload semantics, e.g. XML files.
- Ensure time constraints: prioritization of messages in terms of delivery.
- Extendibility / Adaptability: open architecture and/or plugin support through a defined interface / API.
- Use of industrial standards: support for standardized formats like e.g. ISA'95 and/or AutomationML.
- Scalability: the system is able to scale to the users need in terms of hardware requirements and distribution.
- Security: data access can be restricted and information manipulation is protected against anonymous access.
- (Data) reliability:
  - incoming data validation against a given schema / template.
  - no important data gets lost (historian, reliable messaging).
- Best industrial practices:
  - industrial support
  - no dependency to a specific kind of hardware which, is not industrial proven.

Functionalities:

- Receiving/sending data: network connected input and output of data to other systems.
- Mapping of data formats: different data formats can be transformed into both directions.
- Modularity: system is built up through different modules, which ensures the tailored functionality for each use case.
- Yellow/white pages: system publishes its public functions and data for others to discover.

Additionally, the chosen middleware needs to satisfy the following PERFoRM architectural and technical requirements:

- Transformation, validation and routing capabilities of the PERFoRM language (major criteria).
- OPC-UA as major data format / transport protocol (base of the PERFoRM markup language though native OPC server/client not a must, could use a plugin or external server, i.e. Softing OPC-UA Stack).
- at least XML parsing and validating capabilities (native or via plugin/library).

These are the major requirements which the middleware must fulfil. Besides that, the minor requirements are mostly important for the evaluation and implementation for the projects use cases, i.e. the custom 3rd party plugin support (although the extendibility is crucial, it is not a must have that the customer himself can extend it), which is needed to create the connection to the outer shell of the middleware, the so called PERFoRM layer (see Figure 2). For the end customer, i.e. the industrial partners, the commercial support is a mandatory criteria, which is not explicitly highlighted.

B. Middleware Solution Assesment

1) Introductional commentary: The market for middleware solutions is a vast space and ranges from specialized applications like IoT to those, which try to be the jack of all trades. One has also to differentiate between a true middleware and full blown integration suit like, for example IBM's *Integration Bus*. These total integrated software architectures tend to be problematic because of the huge amount of different tools (if the software parts are not all developed by the same team or have been bought into the portfolio), that are needed to manage them.

Because of limited time and scope during the evaluation, only the most promising candidates could be pre-evaluated, which are being shown in Table I. For other well established middleware vendors like Oracle or Software AG there exists another analysis ([20]), although with a different focus (ERP and A2A) and target audience. Some vendors could also not be taken into account, as they missed to supply the needed data within the given time frame. This is the reason why e.g. GE Fanuc's *Proficy SOA* middleware is missing in this preliminary evaluation. For some questions the responsible

TABLE I
MAYOR CHARACTERISTICS AND FUNCTIONALITIES OF MIDDLEWARE SOLUTIONS

Product name	WinCC OA [10]	mBS [11]	Octo-/ Meshblu [12]	dataFEED [13]	IB Manuf. [14]	PCo [15]	ESB / GW [16]	Fuse ESB [17]	Mule ESB [18]	Node-RED [19]
Developer	Siemens	Prosyst	Citrix	Softing	IBM	SAP	WSO2	Red Hat	Mulesoft	IBM
License	Comm.	Comm.	Comm., OS	Comm.	Comm.	Comm.	OS Apache 2.0	Comm. Apache 2.0	Comm., CPAL for CE	OS
typical use	Ind.	Home-autom.	IoT	Ind.	Ind.	MES	ERP/IoT	ERP	ERP	IoT
OPC DA	cl./srv.	not yet[a]	no	cl./srv.	client	client	no	no	no	-
OPC UA[b]	cl./srv.	not yet[a]	no	cl./srv.	client	client	no	no	no	cl./srv.
Modular system	yes	yes	yes	no	-	yes	yes	yes	yes	yes
Message prioritization	no[c]	yes[d]	-	no	-	-	yes	yes[e]	yes[f]	-
Open API / Plugin Support	yes	yes	yes	no	yes	yes	yes	yes	yes	yes
Plugin dev. self/company	yes	yes	yes/-	no/yes	yes/-	yes/-	yes	yes	yes/-	yes/-
Message routing /transformation	yes	yes	yes	yes/no[g]	yes	yes	yes	yes	yes	yes
Scalability	high	high	high	high	high	high	high	high	high	-
Hardware req.	med to high	very low	low	med	high	-	low	low to med	med	very low
Comm. Support	yes	yes	yes	yes	yes	yes	yes	yes	yes	-
Message validation[h]	yes	yes	yes	no	yes	yes	yes	yes	yes	-
Security features[i]	yes	yes	yes	yes	yes	yes	yes	yes	yes	-
Message reliability[j]	yes	yes[f]	-	yes[k]	yes[l]	yes	yes[m]	yes[n]	yes[o]	-
Programming Language / Framework	C++, runtime scripting	Java (OSGi)	coffee-/ javascript/	visual	C++/.net	C#	Java, embed. scripting	Java	Groovy, Java, JS, (Python)	JS/node.js
PLC Interface[p]	yes	no	no	yes	no	no	no	no	no	no
Operating Systems[q]	L, W, S	JVM	node.js	Win	L, W, more[r]	W[s]	JVM	JVM	L, W, more[t]	node.js

[a] estimated 2017 [b] important feature as OPC UA will be the core transport technolgy of the PERFoRM middleware
[c] not yet but its scheduled to be implemented [d] indirectly via JMS [e] Apache Camel [f] JMS
[g] not built in but could use Node-Red or other tools interfacing via MQTT. [h] validation against given XML scheme
[i] built-in security measures [j] data is buffered / will not get lost (store and forward) or is stored till successful receivable (i.e. WS-ReliableMessaging of the SOAP protocol) [k] store and forward service but no assurance other than TCP that message has been received
[l] *Resequencing* is possible to guarantee the order of the message flow [m] message buffering, WS-Reliable Messaging (WS-RM; see SOAP specification, currently only supported by the .net OPC-UA implementation) or via JMS message redelivery [n] same as WSO2 ESB [o] via reliability patterns
[p] use case important direct PLCs interfaces [q] only x86 compatibility is listed, JVM means everywhere the JVM is runnable though this might not be the case for some middleware solutions. [r] AIX, Solaris, HP-Itanium, z/OS [s] .NET >= 3.5 [t] AIX, Solaris, HP-UX, Mac OS x

developers missed to supply the requested informations in time or didn't answer at all, which is indicated with a hyphen (-) within Table I.

2) Evaluation criteria explanation: The table shows the major requirements (marked red) and the the capabilities regarding those requirements of each middleware solution. The most promising ones are marked green. Where some are directly capable to fulfil the specific requirement, others need additional software, like for example middlewares, which do not natively support OPC-UA and/or are missing the OPC-UA Stack implementation (client/server). The most weight though went to the major requirements, which explains why some contenders didn't make it into the group of the most promising solutions. The hardware requirement for example, which at first seem to be no major requirement, is a subset of the scalability and therefore important to consider. The Node-Red flow editor, though not a middleware in its own right, has been added to show its capability compared to commercially supported solutions. Another reason why it is listed is the capability to enrich other products with new features like for example message transformation and runtime reconfiguration.

C. Most Promising solutions

Out of the eleven start candidates three were chosen to be further examined with regards to performance and use case applicability. The current best fitting solutions seem to be the following three:

1) Siemens WinCC OA: The currently best fitting solution to all requirements is the *WinCC Open Architecture* MW from Siemens/ETM. Besides the requirement fulfilment and the overall flexible architecture, it's the direct PLC interface, which stands out and makes the use of adapters obsolete. Currently missing features like the message prioritization could be temporarily compensated through external technology like for example *Apache Camel*. Although *WinCC OA* consists of a huge amount of components (full blown SCADA interface), a very lean installation is also possible, which even fits on a Raspberry PI's limited Hardware[1]. With its focus on message security and reliability, it's best suited for information critical processes, where no data must get lost due to network failures or performance issues.

2) Prosyst mBS: Although focusing on a different field of automation, the Prosyst (now subsidiary of Bosch) mBS looks promising as the right choice for an embedded middleware, as it can even run on VxWorks and QNX[2]. This makes it an interesting solution for hardware gateway/mw solutions as the needed processing power and RAM is really low compared to solutions from e.g. IBM and SAP (150-400 MHz CPU depending on runtime and up to 40 MB of RAM). Although its small footprint, it's still very much extensible through the open OSGi Interface. The message prioritization is no direct feature, but can be achieved via the underlying JMS technology, at least for the prioritized data exchange between middleware solutions. As OPC-UA and DA support is also already scheduled for implementation, this solution is currently, besides the one from Siemens, the most promising one.

3) Red Hat Fuse ESB: Besides the fulfilment of all criteria and the use of the JVM, which makes it applicable on almost every OS supported by the JVM, Fuse ESB has been chosen because of its development kit and the used technologies like for example *Apache Camel*, which gives it an edge over the other possible candidates. Through *Apache Camel* OPC UA's WS-RM (reliable messaging technology) isn't needed, which is currently only available for the .net framework and therefore Microsoft Windows products. The missing OPC-UA functionality can be bridged via e.g. Softing's OPC Stack.

D. Additional solutions to consider

Besides the previously mentioned solutions, the ones from IBM, SAP and maybe even GE Fanuc should also be considered to be at least looked at in more detail, as their strong connection to industrial automation and their tightly integrated product portfolio might be the right solution for some partners. Therefore the preliminary action before or concurrent to the actual evaluation of the proposed solutions will be another and more in depth analysis of said Middleware products (marked yellow in the table).

E. Migration of legacy systems

Although the Softing *DataFeed* "middleware" was no direct contender for the *PERFoRM* solution (missing the crucial data transformation and validation capabilities, see Table I), it still fills an important role in the evaluation process. The software is capable to connect major PLC vendor interfaces (Siemens, Schneider, Rockwell and more) and provide the data directly via OPC-UA and DA. Combined with the store and forward functionality, it's the fitting solution for every system without a direct OPC interface or where OPC-DA would be to slow in terms of data throughput and latency. Together with Node-Red or a similar solution, which supports one of the *DataFeed's* transport protocols (i.e. MQTT), it's also possible to create a full middleware. On the other hand, the *DataFeed* suite can also act as an OPC and PLC connection enabler for middleware solutions, that do not provide a OPC server natively. Because of the low system footprint (installation takes roughly 300 MB) the *DataFeed* solution can be installed on older control systems, like for example the use case important Sinumerik 840D Powerline. Besides the ease of integration, the import function for PLC progamming projects makes it even more useful, as the setup time for complicated controls (thousands of exported variables) decreases significantly.

[1] It's strongly advised by the developer to not use the ARM version for or in productive systems. [2] Both are true real-time operating systems

IV. Conclusion & Outlook

As mentioned in Chapter I, an important goal of the PERFoRM project is the definition, specification and prototype implementation of Middleware solutions. As a first step, this paper has described the methodology of the evaluation of such solutions by defining important criteria and scanning the market for promising solutions. The outcome of this work is shown in Table I, which is visualizing the results of this first assessment.

The results of the innovation work performed till now, i.e., the initial / preliminary examination of known industrial middleware solutions, will be completed with future works dealing with the choices of adequate middlewares for different application solutions (industrial use cases). The chosen ones need to be examined and tested in greater detail in regards to system performance, throughput, data reliability under heavy load, stability, latency, etc. This in-depth testing will result in the final recommendation for the PERFoRM middleware solution vendor/s from whereon the actual implementation in regards to the use cases will start.

Acknowledgment

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 680435.

References

[1] Plattform Industrie 4.0. [Online]. Available: http://www.plattform-i40.de/

[2] A. W. Colombo, S. Karnouskos, Y. Shi, S. Yin, and O. Kaynak, "Industrial cyber–physical systems [scanning the issue]," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 899–903, 2016.

[3] P. A. Bernstein, "Middleware: A model for distributed system services," *Communications of the ACM*, vol. 39, no. 2, pp. 86–98, 1996.

[4] Q. H. Mahmoud, Ed., *Middleware for Communications*. Chichester, UK: John Wiley & Sons, Ltd, 2004.

[5] S. Ou, K. Yang, and J. Zhang, "An effective offloading middleware for pervasive services on mobile devices," *Pervasive and Mobile Computing*, vol. 3, no. 4, pp. 362–385, 2007.

[6] K.-D. Moon, Y.-H. Lee, Y.-S. Son, and C.-K. Kim, "Universal home network middleware guaranteeing seamless interoperability among the heterogeneous home network middleware," *Consumer Electronics, IEEE Transactions on*, vol. 49, no. 3, pp. 546–553, 2003.

[7] *A context-aware middleware for applications in mobile ad hoc environments*. ACM, 2004.

[8] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Y. Terziyan, "Smart semantic middleware for the internet of things," *ICINCO-ICSO*, vol. 8, pp. 169–178, 2008.

[9] A. Colombo, *Industrial Cloud-Based Cyber-Physical Systems: The IMC-AESOP Approach*. s.l.: Springer International Publishing, 2014. [Online]. Available: http://lib.myilibrary.com/detail.asp?id=635192

[10] "Siemens wincc oa." [Online]. Available: https://www.automation.siemens.com/salesmaterial-as/brochure/en/brochuresimatic-winccoaen.pdf,

[11] "Prosyst mbs." [Online]. Available: https://dz.prosyst.com/pdoc/feat/mbs-sdk/8.1/mBSSDK81Features.pdf,https://dz.prosyst.com/pdoc/mBSSDK8.1/maintopics/main.html

[12] "Citrix octoblu." [Online]. Available: https://www.octoblu.com

[13] "Softing datafeed." [Online]. Available: http://industrial.softing.com/en/products/software/opc-suite-servers-middleware/the-all-in-one-solution-for-opc-communication/datafeed-opc-suite.html

[14] "Ibm integration bus." [Online]. Available: http://www-03.ibm.com/software/products/en/integration-bus-standard

[15] "Sap plant connectivity." [Online]. Available: https://help.sap.com/saphelppco22/helpdata/en/46/a00344d44852b7e10000000a155369/content.htm

[16] "Esb gw." [Online]. Available: http://wso2.com/products/enterprise-service-bus/

[17] "Fuse ESB." [Online]. Available: http://www.jboss.org/products/fuse/overview/

[18] "Mule esb." [Online]. Available: https://www.mulesoft.com/sites/default/files/mule-esb.pdf

[19] "Node-red." [Online]. Available: http://nodered.org/

[20] "Vendor landscape: Application integration middleware," Info Tech Research Group. [Online]. Available: https://www.infotech.com/research/ss/it-vendor-landscape-application-integration-middleware