Data aggregation architecture "Smart-Hub" for heterogeneous systems in industrial environment

Ahmed AHMED[1*], Lionel ROUCOULES[1], Rémy GAUDY[2] and Bertrand LARAT[2]

[1] Arts et Métier ParisTech, CNRS, LSIS, 2 cours des Arts et Métiers, 13697 Aix-en-Provence, France

[2] Sogeti High Tech – Groupe Capgemini, 2-10 rue Marceau – CS70400, 92136 Issy Les Moulineaux, France

* Corresponding author. Tel.: +33-6-66-31-33-50; E-mail address: ahmed.ahmed@ensam.eu

Abstract Distributed systems spread widely in industrial environments. One of the key challenges is the exchange and the aggregation of data between these systems. Although standards play an important role to solve data interoperability issues between systems, these standards do not completely address existing industrial problems. In fact, it is not granted to have an industrial environment that complies with a unique standard; therefore, ad-hoc solutions are used to solve this issue. In this article, the authors propose a generic architecture to address the interoperability between systems. This architecture is developed based on model-based techniques and principles. Besides, it reduces the need for human intervention and time by developing once and reusing the building blocks of the architecture. Finally, the architecture is described in its application to a case study.

Keywords: interoperability; data aggregation; smart systems; model-based engineering; supervision;

1 Introduction

In the environments of industrial enterprise, distributed systems widely spread to perform useful operation independently. The networking and integration of these heterogeneous systems to achieve a common goal leads us to a larger complex system mostly defined as System of Systems SOS [1]. These systems include emerging smart systems such as smart grids, smart gas, smart cities, etc. [2, 3]. As shown in figure 1, they are widely found at the control level and enterprise level.

On the one hand, the control level depicts the vertical data exchange. It includes the full data acquisition chain from the control module of devices (sensors and actuators) up to the acquisition servers through Machine-To-Machine (M2M) communication standards. Afterward, the acquisition data are monitored on a human machine interface (HMI). This HMI not only assists in reading the measurements provided by the sensors but also to drive the actuators. This chain for monitoring and driving the physical devices is referenced by an architecture called Supervisory Control and Data Acquisition (SCADA) [4]. Various SCADA systems may co-exist in a given environment, each with its own data format such as OPCUA[1], MQTT[2], or Sigfox[3], etc. and this exhibits the need to use the appropriate standard and mechanism to access the data. In this SCADA architecture, we are interested in the acquisition server for exchanging the data with other systems.

On the other hand, the enterprise level represents all the other existing information systems in the industrial environment, e.g. maintenance systems (Computerized Maintenance Management System - CMMS), Geographic Information System (GIS), decision support systems, forecasting systems, logistic systems, etc. Therefore, the data need to be shared horizontally between the acquisitions servers at the control level and the systems at the enterprise level. Examples on figure 1: a CMMS information system (data producer - DP) produces data in CSV format for the SCADA server (data consumer - DC) that itself uses OPCUA standard; a GIS system (DC) displays alarms originated from the acquisition server OPCUA (DP) and Backend Sigfox (DP) and using data from the CMMS (DP) system as well.

The heterogeneity, due to different data format and data semantic (OPCUA, Sigfox 12 bytes, etc.) of these systems, makes it very challenging to guarantee the interoperable exchange of data between them. Therefore, this paper addresses the problem of syntactic and semantic interoperability to guarantee the interoperable exchange of data between heterogeneous systems. So, the authors propose a generic interoperability architecture for ad-hoc solutions that relies upon model-based engineering principles to deal with this issue.

The remainder of the paper is organized as follows: section 2 discusses the related work; section 3 is devoted to the proposal of the interoperable architecture; section 4 illustrates the technical aspects of implementing the proposed architecture, its application on a case study and finishes with a discussion; finally, section 5 concludes with a summary and some insights on ongoing and future work.
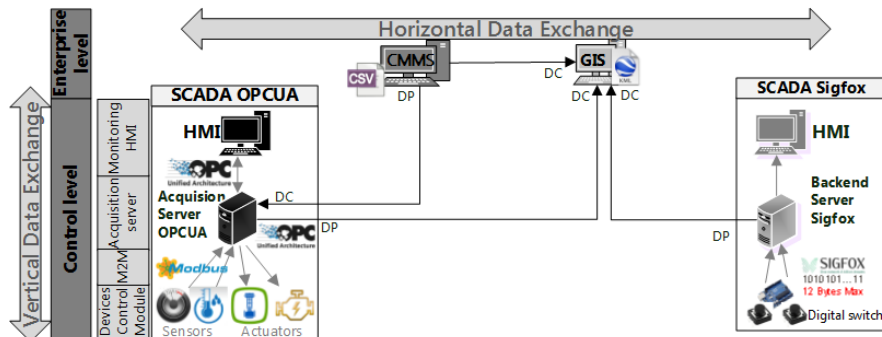
---

[1] https://opcfoundation.org/

[2] http://mqtt.org/

[3] http://www.sigfox.com/

Fig. 1. Data Flow and exchange between heterogeneous systems

2 Related Works

The integration and the automatic exchange of data among heterogeneous systems lead us to the essential requirement of interoperability. Some workgroups and organization steer their effort to decompose the interoperability issues by defining a reference model in architectural approaches, such as Grid Wise Architectural Council [5] and the Smart Grid Architecture Model for electricity domain [3] and Reference Architecture Model for Industrie 4.0 (RAMI 4.0) [6]. However, in all these reference models, the information interoperability in terms of syntax and semantic are defined by domain-specific standards. Several standards are developed for various domains, examples include: the Common Information Model (IEC 61970, 61968 and 62325) [7] in the electricity domain that defines the components of the electrical power systems and their relationships; the standards ISA-95 [8], B2MML (Business to Manufacturing Markup Language), MIMOSA [9], ISO-15926 [10], or PRODML [11] in the oil and gas industry; ISO-10303 [12] in automation systems and product data exchange.

Furthermore, there exist some technical standards such as OPCUA [13] for data integration platforms. In spite of these standards, it is not granted to have a fully end-to-end interoperable environment [14]. For instance, it is impossible to have all the information systems using OPCUA. Therefore, some work had been done for the mapping between standards such as CIM to OPCUA in the electricity domain [15]. Other work proposed the use of a model-based integration framework using PRODML as the central metamodel for the oil and gas industry [16]. The major limitation of these approaches is their application on specific standards and domain only. Similarly, to our best knowledge and based on our industrial partners' feedback, this issue is usually solved by developing ad-hoc mediator solutions. This approach may not be practical and conventional in all situations because it involves numerous human manual interventions that are time-consuming, error-prone, and lack both flexibility and generality. Thus, the previous solutions provide domain-dependent solutions. Therefore in this work, a domain-independent interoperability framework for the aggregation of data among systems is proposed.

3 Conceptual proposal: "Smart-Hub Architecture"

As shown in figure 1, the data are originated from various sources, *i.e. data producers,* and consumed by other systems, i.e. *data consumers*. As a result, the data consumer must interpret the received data. This leads us to decompose the problem as following:

1. *Syntactic interoperability* due to different data syntax (format) between systems (e.g. XML, DB, CSV, etc.)
2. *Semantic interoperability* due to the different interpretation of data between systems. (e.g. data attribute "temperature" in system A is "temp" in system B)
3. *Various Interfacing mechanisms* to read and write data from data producers and to data consumers (e.g. file system, request-reply, publish-subscribe, SQL queries, web services, etc.)

This work introduces a 4-layered generic interoperability architectural solution to address the above-mentioned problems. This architecture dubbed "Smart-Hub" acts as a hub between all heterogeneous systems at all levels of an industrial environment. It aggregates data originating from various data producers, manipulates it in terms of syntax and semantic and finally communicates it for data consumers. The 4-layers architecture is as follows:

- **Communication & services layer:** This layer manages various protocols and mechanisms to establish communication between various data producers and consumer systems. It includes a repository for various data connectors, such as file systems connectors, publish/subscribe connectors, request/reply connectors, etc. It also supports some essential services to read (consume) and write (produce) data from and to the various systems, respectively.
- **Extensible layer:** This layer handles the projection (conversion) of data between DP and DC format and the modeling environment. This includes a repository for the supported data formats and syntax projection rules.
- **Integration layer:** This layer involves the aggregation of the model-based data in a global data model (GDM) repository and the fetching of the specific model-based dataset from the GDM. It relies on a number of operational and transformation rules and deal with the semantic of data.
- **Configuration Layer:** This layer focuses on the internal configuration of the smart-hub. It is shared between all the previous layers to guarantee the integration and the functionality of the smart Hub with the surrounding interconnected systems.

Figure 2 illustrates the smart-hub for the context of the example given previously in figure 1. At this stage, we evaluate the architecture from a macro level and all the inner details and structure of each layer is out of the scope of this paper.
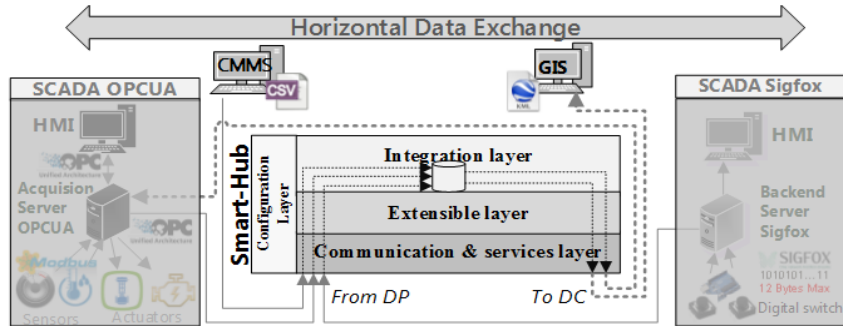
Fig. 2. Smart-hub layered architecture

4 Technical proposals

In this section, the authors propose the use of Model-based engineering (MBE) techniques and operations to implement the smart-hub layers interoperability architecture.

4.1 Modeling fundamentals

MBE [17] helps to handle semantic and syntactic interoperability between standards and languages. In MBE, everything is a simplified representation of certain reality, i.e. a model. The OMG proposes 3-layer architecture and the main notions is **model**, **metamodel** and **metametamodel** as shown in figure 3[18]. Each lower notion **conforms to** the upper notion, i.e. conforms to its modeling language. MBE supports **model transformation**. It is the generation of a model M_b from a model M_a by a transformation Mt. The different heterogeneous systems are not interoperable with MBE environments. This results in the requirement of **projection** phases. Projection is the generation of a model in the chosen modeling environment from structured data in the technical space (TS) of the system and vice versa. This operation is called **injection** and **extraction**, respectively.
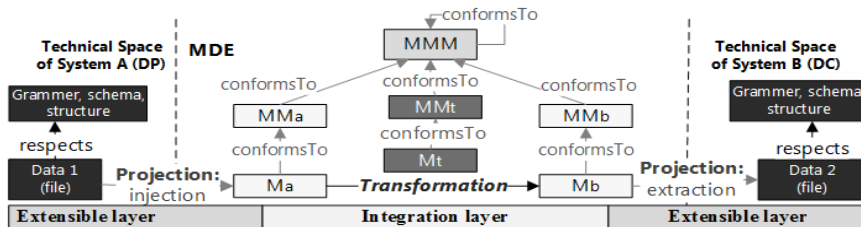

Fig. 3. Model based Engineering and projection main concepts

4.2 Architecture Implementation

Many tools support system modeling such as Papyrus[4], Uml2[5], and Eclipse Modeling Framework (EMF)[6]. According to the laboratory expertise, we have chosen the Eclipse Modeling Framework (EMF) to support the implementation of the smart-hub solution. EMF supports model and data interchange via the XML Metadata Interchange (XMI) format. Metamodels are defined using the ECORE language. The **extensible layer** for projection has been being realized using a combination of the Acceleo tool[7] and EMF's built-in XSD/XML support. The **integration layer,** for aggregation and fetching rules, has been being implemented using a combination of the rule-based ATL[8] tool [19] and direct java manipulation. The **communication and services layer** has been being implemented through extensible eclipse plugins. The evaluation and selection of the various available tools are under investigation with regards to different case studies.

In figure 4, we will illustrate the aggregation of data from the data producers (SCADA server and Backend Sigfox Server) and prepare it for the data consumer (myHMI) that displays the aggregated data. The steps for aggregating the data for the producers to the consumers are as follows:

- As a preliminary step, the smart-hub is configured with all the information required to establish the communication between it and the SCADA server using OPCUA connector and the Backend Sigfox **using the configuration layer**. Furthermore, the repositories in the extensible and integration layer must be configured with the mapping rules for the projection and transformation rules.
- By the use of the **Communication & services layer**, the smart-hub establishes a request for the information from the SCADA server and subscribe to the data of the Sigfox backend server as well.
- Using the projection repository at the **extensible layer**, the data of both the SCADA server (A) and Backend Server (B) are injected in the modeling environment.
- With the **Integration layer** functionalities, the data (A&B) are aggregated and fetched (creation of C=A+B). This includes some operations such as avg, sum, semantic mapping, etc.
- With the **extensible layer**, the data (C) are then formatted according to the consumer component (myHMI).
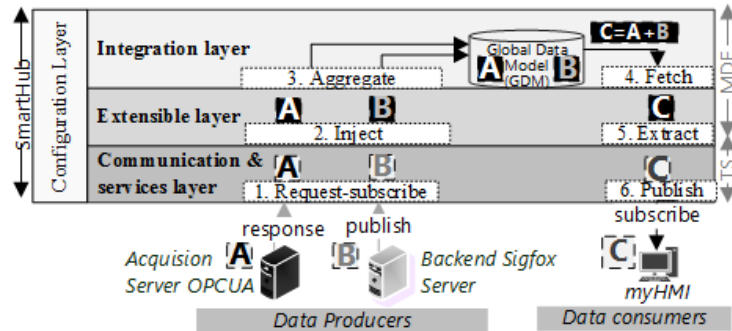- Finally, the data (C) are published to myHMI using the **Communication & services layer**.

---

[4] https://eclipse.org/papyrus/

[5] https://wiki.eclipse.org/MDT-UML2

[6] https://eclipse.org/modeling/emf/

[7] https://eclipse.org/acceleo/

[8] https://eclipse.org/atl/

Fig. 4. Aggregating data from OPCUA server and backend server using smart-hub framework

4.3 Discussion

The solutions existing today (in the related works) require the development of ad-hoc solutions and domain dependent solutions. They are time-consuming, error-prone and lack both of flexibility and generality due to numerous human interventions. Therefore, for all systems involved in information exchange, a separate ad-hoc solution is required. The originality of our solution relies on the fact that the smart-hub is developed as domain-independent and extensible blocks in each layer of the architecture with respect to the related works. These blocks may require more investment in terms of human intervention and time for development, but once it is developed, it is reused for several times. The end user saves time because he just needs to reuse and setup these pre-built blocks to accomplish the required interoperability exchange.

5 Conclusion and future work

In this paper, we have studied interoperability solutions between heterogeneous systems in an industrial environment. We proposed a model-based framework to handle various format and standards. The value added of this proposal is the reduction of human intervention and time by just configuring and reusing the building blocks of the smart-hub. Currently, the work focuses on defining the repository models for the layers of the architecture. In our future work, we intend to fully describe and formalize this modeling architecture. This proposal is still in its early phases of development and several practical and theoretical issues remain to be addressed. This includes hub data persistence and historization, the generality of aggregation/fetching rules, performance, security, etc.

References

1. M. Jamshidi, Systems of Systems Engineering: Principles and Applications, Boca Raton: CRC Press, 2008.
2. K. Su, J. Li and H. Fu, "Smart city and the applications," in International Conference on Electronics, Communications and Control (ICECC), Zhejiang, September 2011.
3. CEN-CENELEC-ETSI and Smart Grid Coordination, "Smart Grid Reference Architecture," European Committee for Standardization: Brussels, Belgium, p. 2437, 2012.
4. P. Zhang, Advanced Industrial Control Technology, Oxford: William Andrew Publishing, 2010.
5. The GridWise Architecture Council, "GridWise Interoperability Context-Setting Framework," 2008.
6. D.-I. P. Adolphs, "RAMI 4.0 An architectural Model for Industrie 4.0," 2015. [Online]. Available: http://www.omg.org/news/meetings/tc/berlin-15/special-events/mfg-presentations/adolphs.pdf. [Accessed 18 03 2016].
7. M. Uslar, M. Specht, S. Rohjans, J. Trefke and M. Specht, The Common Information Model CIM: IEC 61968/61970 and 62325 - A Practical Introduction to the CIM, Springer-Verlag Berlin Heidelberg, 2012.
8. Enterprise-control system integration, IEC 62264 Standard, 2000.
9. "MIMOSA, An Operations and Maintenance Information Open System Alliance," [Online]. Available: http://www.mimosa.org/.
10. Industrial automation systems and integration—Integration of life-cycle data for process plants including oil and gas production facilities, ISO 15926, 2003.
11. Energistics, "PRODML Standards," [Online]. Available: http://www.energistics.org/production/prodml-standards.
12. "Industrial automation systems and integration -- Product data representation," [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=20579.
13. W. Mahnke, S.-H. Leitner and M. Damm, OPC Unified Architecture, Berlin: Springer Science & Business Media, 2009.
14. G. A. Lewis, E. Morris, S. Simanta et L. Wrage, «Why Standards Are Not Enough to Guarantee End-to-End Interoperability,» Seventh International Conference on Composition-Based Software Systems (ICCBSS 2008), pp. 164-173, 2008.
15. S. Rohjans, K. Piech and S. Lehnhoff, "UML-based modeling of OPC UA address spaces for power systems," in Intelligent Energy Systems (IWIES), 2013 IEEE International Workshop on, November 2013.
16. V. Veyber, A. Kudinov and N. Markov, "Model-driven Platform for Oil and Gas Enterprise Data Integration," International Journal of Computer Applications (0975 - 8887), vol. 49, no. 5, 2012.
17. V. G. Daz, Advances and Applications in Model-Driven Engineering, IGI Global, August 2013.
18. OMG. Meta Object Facility (MOF), version 2.4.1, OMG Document.
19. F. Jouault and I. Kurtev, "Transforming models with ATL," in Proceedings of the 2005 international conference on Satellite Events at the MoDELS, Montego Bay, Jamaica, 2005.