Information Models and Information Exchange in Plant-wide Monitoring and Control of Industrial Processes

David Hästbacka[1], Petri Kannisto[2] and Matti Vilkko[2]
[1]*Laboratory of Pervasive Computing, Tampere University of Technology, Korkeakoulunkatu 1, FI-33720 Tampere, Finland*
[2]*Laboratory of Automation and Hydraulic Engineering, Tampere University of Technology, Korkeakoulunkatu 3, FI-33720 Tampere, Finland*

Keywords: Information Models, Data Exchange, Interoperability, Industrial Processes.

Abstract: The efficiency of industrial processes depends on how well the processes can be controlled and this affects the quality, use of resources as well as the environmental impact. Advanced monitoring and control solutions for large-scale industrial processes require information from different systems. The challenge in integration is diverse messaging structures and lack of common semantics in exchange of information between related information systems as well as their human operators. This paper provides a comparison of some of the existing standards of the domain defining suitable structures. Based on these, a model for data and event message structures is developed. The approach builds on a separation of concerns keeping the messaging semantics independent of the transport layer. The requirement is to enable also asynchronous communication as adapters are often needed in distributed environments with heterogeneous systems and communication protocols. The developed structures have been found suitable for communicating measurements and events in industrial process settings as shown by case examples.

1 INTRODUCTION

The efficiency of industrial processes, including reducing environmental impact and the use of resources, is heavily dependent on optimal control (Lamnabhi-Lagarrigue et al., 2017; Lima et al., 2016). The advanced monitoring and control solutions, however, require integration of data and knowledge from different systems involving also humans as operators.

Development of efficient knowledge and information sharing practices involves humans and processes but requires technology to implement. Interoperability is required for efficient collaboration (Panetto et al., 2016). Information sharing is based on a common understanding of semantics, and thus standards and agreed meanings should be favoured when developing means to facilitate information exchange between people as well as information systems conveying the information.

In the setting of industrial processes the production is often distributed and it may span beyond the premises of one single plant. As a result, control and monitoring of the systems are also distributed and represent a multitude of different systems with heterogeneous interfaces and message structures over which information needs to be exchanged.

This paper deals with information models and development of practices for sharing data and knowledge related to industrial processes in such environments. The focus is also on data exchange, communication and alleviating infrastructure developed to improve system integration. The work is based on research being conducted as part of the H2020 funded COCOP project (Coordinating Optimisation of Complex Industrial Processes). The aim of the project is to increase the competitiveness of the European process and automation industry.

The main contributions of the paper are as follows: 1) Comparison of messaging standards suitable for communicating data and events in distributed industrial processes. 2) Design of a messaging API focusing on message structures that facilitates integration of involved information systems. 3) Practical demonstrations that indicate the benefits and relevance of the messaging API and the developed message structures.

The paper is organised as follows. Section 2 presents related work on information models research in industrial production and industrial process environ-

Hästbacka, D., Kannisto, P. and Vilkko, M.
Information Models and Information Exchange in Plant-wide Monitoring and Control of Industrial Processes.
DOI: 10.5220/0006960602160222
In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2018) - Volume 3: KMIS*, pages 216-222
ISBN: 978-989-758-330-8
Copyright © 2018 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

ments. Requirements for interoperability are introduced in section 3 with a comparison of some of the existing standards defining message and data structures potentially suitable for industrial processes. The designed message structures and their implementation is presented in section 4, and case examples utilising them are demonstrated in section 5. A discussion is provided in section 6 before concluding the paper with future work in section 7.

2 RELATED WORK

Previously the ANSI/ISA-95 standard has been applied to production control with agents and a generic manufacturing ontology was developed that used some concepts from the standard (Georgoudakis et al., 2006). An ontological framework for decision-making on the enterprise level has been developed for plant database information using ANSI/ISA-88 (Muñoz et al., 2012).

A model-driven ontology approach for manufacturing systems has been proposed to improve information sharing (Chungoora et al., 2013). Similarly, LinkedData has been proposed as a solution capable to connect industrial data by using globally shared concepts (Graube et al., 2011). Using ontology semantics to analyse industrial systems in engineering has been studied for example by (Dai et al., 2013) and (Hästbacka and Kuikka, 2013).

OPC UA (OPC UA Part 1, 2008) is used as an integration standard with information modelling capabilities for communication with industrial devices and systems. The utilisation of OPC UA for a consolidated information model in service architectures for industrial devices has been studied by (Hästbacka et al., 2014).

Once in operation, industrial processes are dependent on maintenance, and maintenance strategies are typically proactive rather that reactive (Sharma et al., 2014) (Hästbacka et al., 2016). For integration and asset management in maintenance OPC UA has been studied by (Seilonen et al., 2011) and (Hästbacka et al., 2014) among others.

3 INTEROPERABILITY WITH COMMON CONCEPTS

In order to efficiently integrate data from different systems the data structures need consolidation. There are some standards available for data semantics but no one solution that covers the needs. In practice this means that systems implemented by different vendors have slightly varying practices.

Table 1: Comparison of data structure specifications.

Specification	Structures for raw measurement values	Integration of some legacy systems	Structures for data analysis output
OPC UA **(2015)**	Yes	Yes	No
Observations and Measurements **(2011)**	Yes	No	Yes
ANSI/ISA-95 **(2010)**	No	Yes	Yes
ANSI/ISA-88 **(2010)**	No	Yes	Yes

The implementation of advanced monitoring and control applications for industrial processes requires communication from different unit processes as well as communication to the coordination level. For information exchange it means that information also needs to be shared to the lower level units both from the coordination layer and other unit processes. In order to make use of the data in control algorithms, computational models etc. it needs to be unambiguously understood by all parties.

However, as the information systems vary both in implementation as well in their purpose they are therefore heterogeneous exposing and consuming information in different manners. As a result of differing protocols all necessary information should be conveyable in the message structure not relying on system specific communication protocol features. This includes, for example, timestamps, quality and reliability metadata as communication might need to be mediated using asynchronous communication channels such as message buses.

Table 1 provides a comparison of the standards that specify potential message structure. Each standard is discussed in more detail in the following paragraphs.

OPC UA (OPC UA Part 1, 2008) is a de facto integration technology for industrial processes. It is besides a protocol also an information modelling environment that allows for dynamic discovery of data in a secure fashion in networked environments. As such, however, it only provides the basic concepts and data types, and relies on information models such as companion specifications for interoperability (OPC UA Part 5, 2009). OPC UA provides a client-server communication model but its PubSub specification enables the use of a message bus as well (OPC UA Part 14, 2018). With this the scalability and distribution advantages of a message bus apply even when OPC UA is utilised. However, compared to a generic message bus as the platform, the requirements of OPC UA communication reduce the freedom of design related to messaging and message contents.

The Observations and Measurements (O&M) standard (Observations and Measurements, 2011) does not currently have any wide acceptance among industrial production. Therefore, there are no legacy systems that would readily use it in their interfaces. Still, O&M provides an excellent foundation for presenting measurement values. Although the origin of O&M is in the geospatial domain, the structures and metadata are highly similar to the industrial domain. Some of the concepts have different names, though. For instance, what is called ”feature of interest” in O&M maps to ”position ID” or similar in an industrial plant.

ANSI/ISA-88 (ANSI/ISA-88.00.01, 2010) and especially ANSI/ISA-95 (ANSI/ISA-95.00.01, 2010) define information structures for manufacturing operations management (MOM). While ANSI/ISA-88 focus on individual processes and their equipment structures ANSI/ISA-95 focuses on integration of manufacturing information systems related to manufacturing operations. ANSI/ISA-95 covers scheduling, resourcing, production capabilities and personnel, and there is a B2MML (Business to Manufacturing Markup Language) specification to serialise the data structures into XML for that purpose. The data structure specifications are loose allowing for extension, which requires additional specification about the actual data structures utilised.

Not only message structures are important, but the encoding of measurement is another remarkable topic. For measurement units, there should be a contract about encoding, because an ad hoc approach would inevitably lead to multiple encoding methods (such as the temperature ”Celsius” being either ”C” or ”Cel”). Conflicts are also possible. For instance, consider ”a” that may mean either a year or an are; also, is ”C” Celsius or Coulomb? Although the context may reveal the semantics of the unit, ambiguity is rarely positive. Table 2 presents some specifications for encoding measurement units.

UCUM (The Unified Code for Units of Measure) (Schadow and McDonald, 014s) focuses on unambiguous representation of measurement units. The goal is an extensive coverage of all measurement units currently relevant in various fields. The motivation for the development is the limited coverage and ambiguity of existing standards and specifications. UCUM does not aim to define an explicit specification of all units possibly. Thus, to enable the encoding of any kind of unit, UCUM defines rules.

CML (Chemical Markup Language) (Chemical Markup Language, 2018) is focused on chemicals,

Table 2: Specifications to encode measurement units.

Specification	Suitability
UCUM (The Unified Code for Units of Measure) (Schadow & McDonald, 2014)	The best candidate found; extensive coverage, systematic coding approach
CML (Chemical Markup Language) (Chemical Markup Language, 2018)	Potential candidate; limited coverage
UNECE Codes for Units of Measure Used in International Trade (Recommendation No. 20, 2010)	Potential candidate; lacks intuition
UnitsML / UnitsDB (UnitsML Guidelines Version 1.0, 2011)	Unsuitable; no extensive, publicly available unit definitions provided

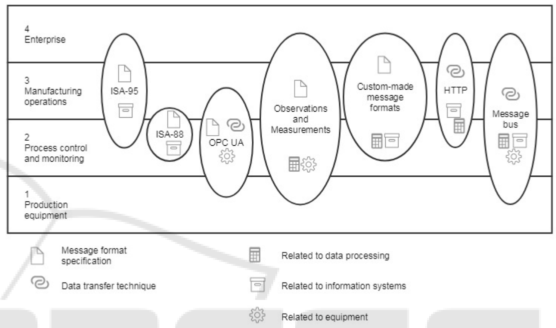

Figure 1: Message formats and communication protocols positioned in the levels of the automation pyramid (the levels are defined in ANSI/ISA-95 (2010)).

and the coverage of the specification is low compared to that of UCUM.

UNECE Codes for Units of Measure Used in International Trade (Recommendation No. 20, 2010) provides good coverage. However, some codes are numeric making some of them difficult to manually interpret.

UnitsML (UnitsML, 2011) is a specification to represent units. The main focus is on the schema rather than an actual specification of units. Related to UnitsML, NIST (National Institute of Standards and Technology) have developed UnitsDB that specifies units, but unfortunately it is not publicly available.

Figure 1 positions the previously mentioned information models and communication channels on the commonly known levels of the ANSI/ISA-95 hierarchy. None of the presented message structures are sufficient as such to cover all the needs of communicating industrial process data measurements and events. They do, however, provide parts that are usable and they can also be used in combination, which will be continued in the next section.

4 IMPLEMENTING MESSAGE STRUCTURES FOR PROCESS MONITORING AND CONTROL

To facilitate messaging, a messaging API has been designed. The current implementation is in C#, but a subset of it exists in Java. Still, considering that standardised, platform-independent message structures are utilised, there are no limitations related to implementation techniques. Instead, the fundamental requirement is to just follow the standard structures. Therefore, there is no requirement to use any particular API, but their utilisation saves effort producing the same output as agreed message structures. When the API is utilised, the application developers do not have to work in the level of the concrete message structures but they can concentrate on the application logic instead.

To implement communication, two aspects must be covered: a communication protocol that provides a delivery medium and the structures of messages. The communication protocol provides part of the interoperability as well as several other characteristics related to reliability, throughput capacity etc. The detailed comparison and selection of communication protocols is out of the scope of this paper. In this work AMQP has been chosen as the main communication protocol. The message formats are based on multiple standards. In the design of the API, it is an intentional choice to completely separate message structures and any bindings to the communication protocol. Then, it is straightforward to utilise other communication protocols if needed, such as other message buses (e.g., ZeroMQ) or HTTP. The utilisation of separate protocol and message libraries slightly adds to complexity (due to separate interfaces), but the flexibility of the approach is considered more important. Figure 2 illustrates the approach.

The high-level typing of the API message structures is given in Table 3. Each type is explained in the following paragraphs. Although the table associates most message types with only one standard, the standards actually consist of other general-purpose standards, such as Geography Markup Language (Geography Markup Language, 2007). The Sensor Observation Service (Sensor Observation Service, 2012) standard directly refers to Observations and Measurements (Observations and Measurements, 2011) as the actual payload of response messages. In standards specification, such modularity introduces dependencies as well as complexity between the standards, but it also enables the reuse of specification effort and added value on previous work. The classes of the API being developed are illustrated in Figure 3 but the implementation details of the API libraries are out of the scope of this paper.

Figure 2: In the API, the utilisation of message formats has been separated from the communication protocol.

Table 3: The most important types utilised to build messages.

Type	Source standard	Purpose
Observation	Observations and Measurements (2011)	Enclose observation-related metadata
Item	(Multiple standards)	The actual measurement value; this abstract base type has multiple sub-types for various measurement types
TimeSeries	TimeSeriesML (2016)	Deliver time series data
GetObservationRequest	Sensor Observation Service (2012)	Used to request for observation

The types have the following uses. Three basic types are utilised to deliver the fundamental measurement data. First, the Observation type encloses the metadata of measurements. It specifies fields that enable the identification of data sources, measurement methods, data quality and so forth. Second, for the actual measurements, there are multiple sub-types of Item. For instance, a single measurement value is enclosed in an Item_Measurement that holds a measu-

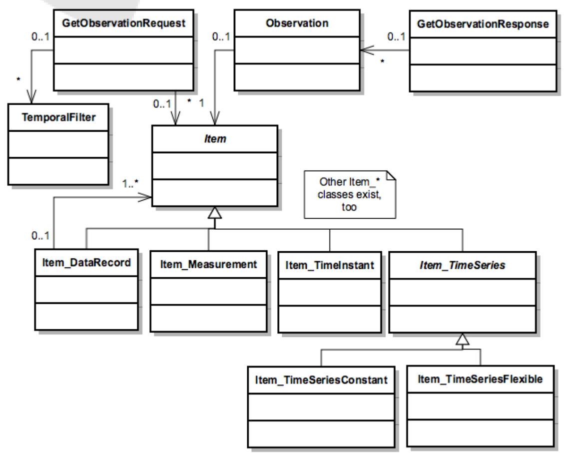

Figure 3: Message type classes provided by the API libraries being developed. API libraries facilitate producing and consuming compatible message structures but are not required as long as agreed message structures are used.

rement unit and the related value. For complex measurements with multiple fields, there is a type called Item_DataRecord. Third, time series data is delivered with the TimeSeries type. However, to implement a request-response scenario, further types are necessary. The GetObservationRequest type specifies multiple fields to identify what actual measurement data is being requested, including temporal filters, measurement procedures and measurement points. Respectively, GetObservationResponse encloses the returned measurement data.

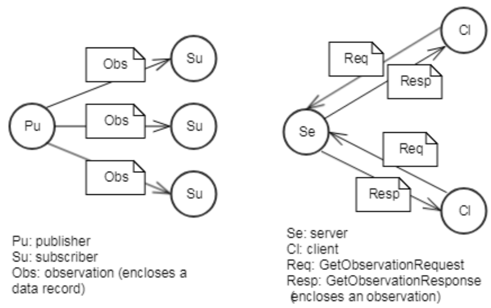

Figure 4: Message exchange in the two demos.

5 CASE EXAMPLES

Two practical demonstration examples have been implemented with the messaging API. The first demo implements a scenario where one unit process submits information to a second unit process so that the second can use that information to optimise its operation. The practical context is copper refinement, and the involved unit processes are a Flash Smelt Furnace (FSF) and a Peirce-Smith Converter (PSC) (for reference about copper refinement, see (Schlesinger et al., 2011)). The FSF provides batches of material to PSC for further refinement. To operate efficiently, the controllers of PSC need an actual estimate of the composition of the batches that come from FSF.

To actually implement the communication, the first demo uses the publish-subscribe pattern for communication. Whenever a batch leaves FSF, a related composition estimate is published. This estimate is then delivered to the PSC, which has a subscription for it. The required communication is straightforward as message structures are concerned; the messaging is one-way, so the composition estimate is only published as such with the related metadata. That is, no request structures or similar are needed but only the observation that contains composition information.

The second demo presents a request-response scenario, where a client requests for a temperature value from a server. The scenario demonstrates a typical need in process plants. Considering messages, the scenario is more complex than the publish-subscribe, because the communication pattern is bi-directional. That is, there must be a structure to enclose the conditions that communicate what is actually being requested. To request the current value of a sensor, simply a position ID is sufficient. However, in more complex scenarios, there could be temporal filters as well (e.g., ”provide me the measurement values from last four hours”).

The message exchange of the demos is illustrated in Figure 4. The first demo (left) only delivers observation from publisher to subscribers, whereas the request-response pattern of the second demo requires bi-directional communication.

In both of the demos, the messaging API has shown its power. The creation, serialisation, deserialization and utilisation of messages is straightforward and does not require many lines of code. Without the API, the manual work on data structures would require a large amount of careful work but is interoperable thanks to the standard structures utilised in the messaging.

6 DISCUSSION

The use case examples presented were simple but proved the point of having a common unified format engineered to improve semantic interoperability on exchanged information. They could easily be extended to real-life production-related information systems.

The API approach is powerful. However, it is also important that no client is forced to use the API, as long as they read or generate the message structures as agreed. However, the XML schemata that are utilised are complex and large. If the API is not utilised in application development, the developer has to know the schemata in detail.

As the serialisation syntax, JSON would generate less overhead compared to the current XML implementation. However, there is a limitation that neither Observations and Measurements nor ANSI/ISA-95 currently offer a standardised JSON schema specification. Fortunately, OPC UA specifies JSON serialisation, but OPC UA was not chosen for the primary implementation technology due to missing data structures.

7 CONCLUSIONS AND FUTURE WORK

Interoperability of information is an essential prerequisite for efficient integration of data. This also applies to the complex domain of industrial processes that typically are distributed and of large scale. This paper presented message structures developed that are needed to implement advanced plant-wide monitoring and control solutions.

First, a comparison of existing standards was presented with some structures applicable for data exchange. Based on this, and utilising constructs from these standards, message structures were proposed for communicating data and events in industrial processes. The developed concepts were demonstrated with use case examples. Although the examples were limited by scope the concept can be scaled to larger real-world industry settings.

In the future, the message structures could be experimented in the integration of actual production systems. In addition, new message structures will likely be added. For instance, for schedules, the structures of ANSI/ISA-95 will likely be utilised. The API libraries are work in progress that will facilitate taking into use the proposed message structures.

ACKNOWLEDGEMENTS

This 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 the project partners in the COCOP project (Coordinating Optimisation of Complex Industrial Processes, https://www.cocop-spire.eu/).

In addition, the authors are grateful to the Academy of Finland for their funding (grant 310098).

REFERENCES

ANSI/ISA-88.00.01 (2010). *ANSI/ISA-88.00.01-2010 Batch Control Part 1: Models and Terminology*. ANSI/ISA.

ANSI/ISA-95.00.01 (2010). *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.

Chemical Markup Language (2018). Chemical markup language. http://www.xml-cml.org (accessed 7 Jun 2018).

Chungoora, N., Young, R. I., Gunendran, G., Palmer, C., Usman, Z., Anjum, N. A., Cutting-Decelle, A.-F., Harding, J. A., and Case, K. (2013). A model-driven ontology approach for manufacturing system interoperability and knowledge sharing. *Computers in Industry*, 64(4):392 – 401.

Dai, W., Dubinin, V., and Vyatkin, V. (2013). Automatically generated layered ontological models for semantic analysis of component-based control systems. *IEEE Transactions on Industrial Informatics*, 9(4):2124–2136.

Geography Markup Language (2007). Geography markup language. http://www.opengeospatial.org/standards/gml (accessed 7 Jun 2018).

Georgoudakis, M., Alexakos, C., Kalogeras, A., Gialelis, J., and Koubias, S. (2006). Decentralized production control through ansi / isa-95 based ontology and agents. In *Factory Communication Systems, 2006 IEEE International Workshop on*, pages 374–379.

Graube, M., Pfeffer, J., Ziegler, J., and Urbas, L. (2011). Linked data as integrating technology for industrial data. In *Network-Based Information Systems, 14th International Conference on*, pages 162–167.

Hästbacka, D., Barna, L., Karaila, M., Liang, Y., Tuominen, P., and Kuikka, S. (2014). Device status information service architecture for condition monitoring using opc ua. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–7.

Hästbacka, D. and Kuikka, S. (2013). Semantics enhanced engineering and model reasoning for control application development. *Multimedia Tools and Applications*, 65(1):47–62.

Hästbacka, D., Jantunen, E., Karaila, M., and Barna, L. (2016). Service-based condition monitoring for cloud-enabled maintenance operations. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 5289–5295.

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

Lima, F., Li, S., Mirlekar, G., Sridhar, L., and Ruiz-Mercado, G. (2016). Chapter five - modeling and advanced control for sustainable process systems. In Ruiz-Mercado, G., , and Cabezas, H., editors, *Sustainability in the Design, Synthesis and Analysis of Chemical Engineering Processes*, pages 115 – 139. Butterworth-Heinemann, Oxford.

Muñoz, E., Capón-García, E., Espuña, A., and Puigjaner, L. (2012). Ontological framework for enterprise-wide integrated decision-making at operational level. *Computers & Chemical Engineering*, 42:217 – 234.

Observations and Measurements (2011). Observations and measurements. Version 2.0. http://

www.opengeospatial.org/standards/om (accessed 7 Jun 2018).

OPC UA Part 1 (2008). *OPC Unified Architecture Specification Part 1: Overview and Concepts v.1.01*. OPC Foundation.

OPC UA Part 14 (2018). *OPC Unified Architecture Specification Part 14: PubSub v.1.04*. OPC Foundation.

OPC UA Part 5 (2009). *OPC Unified Architecture Specification Part 5: Information Model v.1.01*. OPC Foundation.

Panetto, H., Zdravkovic, M., Jardim-Goncalves, R., Romero, D., Cecil, J., and MezgÃ¡r, I. (2016). New perspectives for the future interoperable enterprise systems. *Computers in Industry*, 79:47 – 63. Special Issue on Future Perspectives On Next Generation Enterprise Information Systems.

Recommendation No. 20 (2010). Recommendation no. 20. Codes for units of measure used in international trade. http://tfig.unece.org/contents/recommendation-20.htm (accessed 7 Jun 2018).

Schadow, G. and McDonald, C. (2014s). The unified code for units of measure. http://unitsofmeasure.org/ucum.html (accessed 7 Jun 2018).

Schlesinger, M. E., King, M. J., Sole, K. C., and Davenport, W. G. (2011). *Extractive Metallurgy of Copper*. Elsevier, Oxford, England, 5th edition.

Seilonen, I., Tuomi, A., Olli, J., and Koskinen, K. (2011). Service-oriented application integration for condition-based maintenance with OPC unified architecture. In *9th IEEE International Conference on Industrial Informatics (INDIN)*, pages 45–50.

Sensor Observation Service (2012). Sensor observation service. http://www.opengeospatial.org/standards/sos (accessed 7 Jun 2018).

Sharma, S., CHUGH, R., and Mather, N. (2014). Sirfrt method: Assessing and improving maintenance reliability.

UnitsML (2011). Unitsml guidelines version 1.0. https://unitsml.nist.gov/Presentations/oasis_flyer.pdf (accessed 7 Jun 2018).

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS