






Information Exchange and Integration Within Industrial Automation Domain

Václav Jirkovský¹ , Marek Obitko² , and Petr Kadera¹ 

¹ Czech Institute of Robotics, Informatics, and Cybernetics,
Czech Technical University in Prague, Jugoslávských partyzánů 1580/3,
160 00 Prague, Czech Republic

{[vaclav.jirkovsky](mailto:vaclav.jirkovsky@cvut.cz),[petr.kadera](mailto:petr.kadera@cvut.cz)}@cvut.cz

² Rockwell Automation R&D Center, Argentinská 1610/4,
170 00 Prague, Czech Republic
mobitko@ra.rockwell.com

Abstract. Information exchange and integration are essential in various systems and their interactions across different domains. In this paper, we discuss the core of the information exchange and integration problems together with possible solutions including a description of various standards for information exchange and a description of several wide-spread formalisms for a definition of information models. Finally, information exchange and integration problems used in our Semantic Big Data Historian is described, to illustrate the approach for solving some of the problems.

Keywords: Information exchange · Information integration ·
Ontology · Industrial automation

1 Introduction

Information exchange and integration are important within and across various areas as well as domains including industrial automation where they are also enablers for Industry 4.0. Their importance is emphasized within the industrial automation domain mainly because of increasing digitization in various systems during various steps as well as levels of manufacturing.

In general, there is no strict boundary between information exchange and integration. However, the difference may be expressed as:

- *Information exchange*—An interoperable system is formed by loosely coupled subsystem, where subsystems are responsible for maintenance of their own information models, and they exchange information for fulfilling demanded or requested goals.
- *Information integration*—An integrated system is composed of tightly coupled components which share a mutual information model, and therefore every component has a full and proper understanding of an information meaning.

This perspective is valid when we limit the scope to a particular platform or system. On the other hand, information exchange and integration would be perceived as parts which complement each other. It is obvious in the situation when we refer to a system which is a part of some heterogeneous platform. Every system has its own integrated information model and also requires to exchange information with surrounding platform parts. In other words, information stored in integrated information models has to be shared with its neighborhood and exchanging information has to be subsequently integrated into local information models.

Researchers have been dealing with these problems for many years, and thus there are several solutions for these two different disciplines. In general, the information exchange problem is mainly solved using various communication formats together with corresponding standards (described in Sect. 2) and the information integration problem is solved by the exploitation of some formalisms with sufficient expressivity (described in Sect. 3).

This paper is structured as follows: first, the information exchange problem is introduced together with formats, which facilitate a solution to the problem. Next, the information integration problem is described followed by descriptions of formalisms for a building of information models. Finally, the proposed and implemented solution for the information exchange and integration problems by Semantic Big Data Historian is presented.

2 Information Exchange Problem

Designers, developers, and operators are tackling with the information exchange problem in every industrial information system. Complex engineering problems are typically tackled by various engineering tools, each dealing with a specific sub-problem. For instance, an eCAD tool is used to design electrical wiring, while another tool is used to define the layout of a control system. In this case, an information exchange strategy is needed to export information about physical connection schema into the schema of communication links among automation components (e.g., Programmable Logical Controllers and Input/Output modules).

2.1 Formats for Information Exchange

In this section, various formats for information exchange facilitation will be described. We may distinguish neutral formats as well as specialized formats, which are derived from the specialized formats and have a particular purpose. Many of formats will be skipped in the following paragraphs because of the limited scope of this paper, and only the most wide-spread or the most promising formats are described.

XML. XML is eXtensible Markup Language that is used for transferring data on the Web and has been accepted as a W3C Recommendation in 1998. XML documents are used to store data and information on the Web, and their content

is structured in nested tags. An opening and a closing tag delimit a particular content (called an element), and each tag can be supplemented with a set of additional name-value pairs, called attributes.

The XML format is widely accepted and used due to its relatively simple structure and easy processing. Based on these characteristics, XML format could be understood as a universal format for data and information exchange and even for their storage.

The XML format is an important technology which has been used for information exchange in many applications and domains thanks to its simple and powerful syntax which is versatile enough for information sharing among multiple sources. However, XML does not address issues of the explicit, intensive semantic structure of XML documents, i.e., XML files may be shared with many systems, but they are meaningless outside the application.

The importance of XML for information exchange and integration is obvious from the fact that a prevalent part of formats exploits XML for their serialization. From another point of view, the XML-based formats try to add (more or less successfully) some domain-specific vocabulary as well as constructs for expressing relations between concepts.

An XML document can be supplied by a document that specifies the allowed tags and their structure—XML Schema¹. In other words, XML Schema defines constraints on XML documents. It provides simple vocabulary and predefined constructs for modeling relations among entities.

AutomationML. AutomationML is an XML-based format with the objective to enable seamless automation engineering of production plants [6]. This standard was developed as neutral data and information exchange format of manufacturing systems by a consortium of leading vendors and users of automation technologies.

The AutomationML format is based on the following standards:

- **CAEX** [5] standard is the cornerstone of the hierarchical structure of plant objects.
- **PLCopen** [12] describes plant behavior and control as a sequence of actions.
- **COLLADA** [3] standard is used for geometry and kinematic modeling.

AutomationML provides relatively universal architecture on how to capture information including for example device concepts such as a sensor or an actuator unit class.

OPC UA. The next very interesting way how to model and even exchange information is by means of OPC Unified Architecture (UA) standard [7]. In general, OPC UA is a secure and open mechanism for exchanging data and information between servers and clients in industrial automation. One of the motivations for the OPC UA standard was to overcome the main obstacle of its predecessor (OPC Data Access together with OPC Historical Data Access and OPC

¹ <https://www.w3.org/standards/xml/schema>.

Alarm&Events)—dependency on COM². Therefore, the OPC UA was designed for the replacement of all existing COM-based specification to be platform independent with extensible modeling capabilities.

OPC UA is built on two main components [11]—transport mechanisms and data modeling. The transport component offers the possibility to communicate via optimized binary TPC protocol for high-performance intranet communication and the next possibility to communicate via Web Services. The data modeling component represents rules and building blocks for the creation and exposing information model. It also defines base types to build a type hierarchy.

In the original OPC standard, only “raw” data is exchanged, i.e., there was not enough information included for understanding the semantics of provided data—a tag name and some information like engineering unit. On the contrary, OPC UA offers more flexible possibility to expose the semantics of the data because of complete object-oriented capabilities including type hierarchies as well as inheritance.

The OPC Foundation has started with standardization of information models of various devices (UA Devices) for the unification of models. Every device vendor may extend these base models with vendor-specific information. This approach is also assumed in other scenarios, e.g., providing data of MES (Manufacturing Execution System) or ERP (Enterprise Resource Planning) systems by exposing the ISA 95 model [4].

There are many interesting features described in OPC UA specification—triggering of methods, variable subscriptions, security, device discovery (local as well as global), etc. Because of these features, OPC UA seems to be one of the most promising frameworks for information modeling and exchange in the automation domain.

ISA-95. ANSI/ISA-95 “Enterprise-Control System Integration” [1], also published as IEC 62264 [2] is an industry standard describing information flows between enterprise and control activities and interfaces between the respective systems. The ISA-95 standard comprises several parts, which contain models focusing on specific integration aspects and terminology to analyze and provide insights into various aspects of manufacturing companies. The three focus areas of the ISA-95 standard are:

- Models of information exchanged between business systems and manufacturing operations systems (parts 1/2/5)
- Models of activities in manufacturing operations systems (part 3)
- Models of information exchanged within manufacturing operations systems (parts 4/6)

The objective of the ISA-95 standard is to reduce costs, errors, and risk associated with implementing interfaces between enterprise and control systems by simplifying their implementation, therefore easing integration. ISA-95 can be utilized as an analysis tool to provide insights into the manufacturing company.

² <https://www.microsoft.com/com/default.aspx>.

The standard can also be used as a basis for developing standardized MES applications that can easily interface with other systems and as a basis for message exchange between ERP and MES systems to achieve vertical integration.

Resource Description Framework. The Resource Description Framework³ (RDF) developed by World Wide Web Consortium (W3C) represents a standard model for data publishing or exchanging on the Web. Data and their corresponding properties are expressed in the form of RDF statements in the form of triples (s - subject, p - property, o - object) denoting that a resource s has a property p with a certain value o.

For denoting resources including subject, predicate, and object, RDF uses Unique Resource Identifiers (URIs) to allow interoperability on the web. The usage of URIs allows RDF data to be mixed, exposed, and shared across different applications. RDF triples may be serialized in various formats including XML, N3, Turtle. The set of RDF triples may be perceived as an RDF graph consisting of linked nodes.

RDF Schema. RDF Schema⁴ (RDFS) provides a data modeling vocabulary for RDF data, and it is used to describe classes and relationships between classes (e.g., inheritance). Next, RDFS specifies also properties and corresponding relationships. Relationships may hold between pairs of properties, or between a class and property. RDFS statements are represented as triples as well, and thus RDFS forms an RDF graph. RDFS triple is called schema triple and other triples data triples.

Web Ontology Language. Web Ontology Language⁵ (OWL) is another W3C recommendation. It is built on RDF and RDFS, i.e., it follows the RDF/RDFS meaning of classes and properties and adds primitives added to support the additional expressiveness. On the other hand, RDF and RDFS have very voluminous modeling concepts such as `rdf:Property` and `rdfs:Class`. Their expressive power causes uncontrollable computational complexity, and for some applications a trade-off is needed for efficient reasoning. For this, the OWL defines different levels or profiles⁶ to be chosen according to the needs.

2.2 Solutions/Frameworks for Information Exchange

There are a lot of proprietary solutions including OPC UA which provide means not only for information model specification but also means for facilitating communication, for example, the client/server architecture.

Such architectures allow to form a system for information exchange—including “centralized” (with a main central control component) or “decentralized” architecture. The centralized architecture may be built on, for example, OPC UA architecture, where the central component communicates using OPC

³ <https://www.w3.org/RDF/>.

⁴ <https://www.w3.org/TR/rdf-schema/>.

⁵ <https://www.w3.org/OWL/>.

⁶ <https://www.w3.org/TR/owl2-profiles/>.

UA (i.e., OPC UA client), and sensors or actuators are connected using OPC UA servers. The example of the “decentralized” architecture could be for example multi-agent systems.

The increasing popularity of Internet of Things emphasized needs for a versatile solution of the information exchange problem. However, a common integration of “things” providing solutions for information exchange relies on ad-hoc solutions and there is no general solution for such integration. These solutions can provide very effective systems. On the other hand, they may bring many drawbacks—difficult system maintenance and malfunction corrections, adding or adjusting components, lower re-usability, etc.

3 Information Integration Problem

When we consider the information integration problem, then we may distinguish two different scenarios—“integration on-the-fly” and “full integration”. The first scenario is tightly coupled with the information exchange problem and has no central data storage maintained by the central component. The second scenario represents a common situation with the central component, which handles an information model together with corresponding data storage. The requirement for both of them is an existence of some information model which handles a “schema” comprising all of the participating components. Furthermore, relations between corresponding elements (e.g., concepts with equivalent meaning in a given context) should be included in the “schema” or stated in some explicitly expressed set of mappings.

There are several different aspects of the information integration problem including developing of adapters (how to convert incoming information to a demanded form) and a determination of formalism for expressing information models and their versatility, flexibility, and suitability for subsequent maintenance.

Prevalent part of various adapters is already implemented and also included in many solutions. Therefore, we will discuss various formalisms for information models specification in the following paragraphs.

3.1 Information Models for Integration

In this section, we will present several formalisms for designing information models which are essential for the information integration problem. The most common approaches together with their general description will be described. Key features of these approaches are flexibility and expressivity. Requirements on these features by an application often determine a target formalism.

Relational Database Schema. Relational database management systems (RDBMSs) are widely used for data storage for many years. RDBMS is a data storage with a collection of interrelated data files or structures. According to the name, relational databases were designed to store data. Tables themselves have no information about relations to other tables. Such information in a limited

form is stored in a schema. A schema provides information about the relationships between tables and field types. More complex relations among tables have to be stored in queries and many times also in the application implicitly. In other words, RDBMSs provide the solution of the information integration problem which consists of three parts—data storage, a schema, and corresponding application algorithm. Moreover, many of available RDBMS solutions provide implemented adapters to various data sources.

NoSQL Systems. NoSQL (Not only SQL) are systems typically suitable for massive amounts of unstructured data in situations, without the need to clearly define a schema, possibly also relaxing some other RDBMS requirements. Examples of architectures of this wide family of systems are a key-value model, column store, document database, and graph database.

In general, these systems provide greater flexibility. On the other hand, information integration is primarily handled by an application, i.e., also implicitly defined.

Ontologies. Ontologies are tightly coupled with the term conceptualization. A conceptualization is an intensional semantic structure that encodes implicit knowledge constraining the structure of a piece of a domain. An ontology is a (partial) specification of this structure. A conceptualization is a general model of a piece of a domain (which is language independent), but an ontology does not have to express all possible constraint because it depends on the requirements of an intended application.

The exploitation of ontologies for a representation of the information model is connected to Semantic Web Technologies today. Nowadays, a powerful set of formats is available with various expressivity (i.e., RDF, RDFS, OWL) together with complementary technologies - SPARQL, SWRL, reasoners, etc.

From a theoretical point of view, ontologies are the most suitable approach for a solution of the information integration problem as they provide explicit specification of a conceptualization with appropriate expressivity. However, we still see that it is difficult to properly deploy and use them in practical applications today.

4 Information Exchange and Integration: Semantic Big Data Historian

The solution demonstrating dealing with the aforementioned obstacles and challenges is presented in this section. The prototype illustrating the solution is a specific historian software architecture which addresses requirements for information exchange as well as integration. The prototype is named Semantic Big Data Historian (SBDH) and was proposed and developed within long-standing research at Czech Technical University Rockwell Automation Laboratory for Distributed Intelligent Control (RA-DIC).

The core functionality, as well as the main advantage of the historian, is the employment of Semantic Web technologies (more precisely an ontology in

OWL⁷) for explicit definition of knowledge. Thus, specific requirements for a historian architecture stem from the utilization of the ontology. Furthermore, the architecture is influenced by a historian target usage, i.e., gathering data and information from a shop floor and other involved systems as well as controlling a shop floor by appropriate feedback. Thus, the architecture has to be very flexible to process all required data and robust to provide a highly reliable solution. SBDH architecture consists of four layers, and a concept of the overall system is to provide a modular solution which may be adapted according to given needs and requirements for the software. The following listing provides a description of the four SBDH architecture layers:

- Data acquisition and control layer—this layer is responsible for the acquisition of data from relevant sources (e.g., sensors, users via a user interface, and any relevant software from higher levels such as MES/ERP) and providing a feedback to control a given process (e.g., controlling an actuator, calling a relevant services of the 3rd party system, etc.). The preferred way for communication is using previously mentioned OPC UA. Consuming of data streams is solved using Spark Streaming Custom Receivers⁸.
- Transformation layer—a transformation to a form of RDF triples according to Cyber-physical system Ontology for Component Integration (COCI) [8]. A very important responsibility of the transformation layer is to resolve semantic heterogeneity and to repair damaged data if possible.
- Data storage layer—transformed data in the form of RDF triples are stored in a triple-store in this layer. The storage respects nature of prevalent part of data, i.e., measurements from sensors. In general, two different file models are used in SBDH to provide more homogeneous data distribution across files in Cassandra—“vertical partitioning” model for data which are not time series (triples are partitioned according to a predicate of the triple) and “hybrid SBDH model” for storage of time series (triples are partitioned according to a triple predicate and a given sensor). More detailed description is available in [9]. It is obvious that this layer is not responsible only for simple data storage but is also responsible for conducting transformations of triples to a corresponding file model.
- Analytic layer—the last layer provides means to access data stored in the triple-store with the help of SPARQL⁹ and to implement analytic tasks. Apache Spark MLlib¹⁰ (library with implemented distributed machine learning algorithms) is used as a solution of analytic tasks.

The overall architecture is shown in the Fig. 1 and more details may be found in [10].

⁷ Web Ontology Language - <https://www.w3.org/OWL/>.

⁸ <https://spark.apache.org/docs/2.2.0/streaming-custom-receivers.html>.

⁹ <https://www.w3.org/TR/rdf-sparql-query/>.

¹⁰ <https://spark.apache.org/mllib/>.

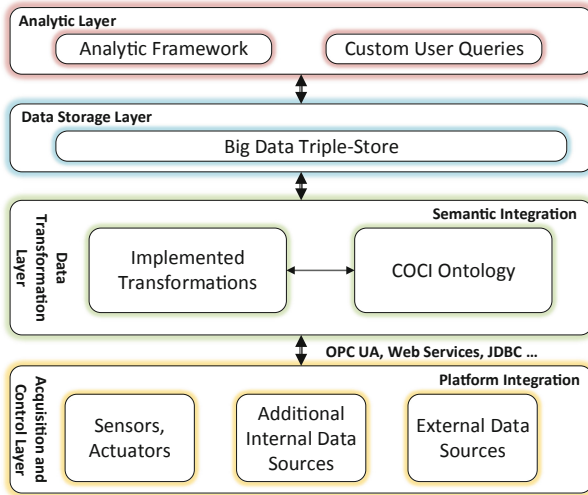


Fig. 1. Semantic Big Data Historian architecture

4.1 SBDH - Information Exchange

Based on our experiments as well as based on increasing adoption of Semantic Web technologies in industry, the exploitation of RDF format for not only data but also for information exchange seems to be a promising approach. On the other hand, comparing only the capabilities of various formats is not enough. Obviously, the adoption of these formats by manufacturers is one of the most important characteristics. Thus, OPC UA was chosen as the primary format for information exchange of SBDH.

The OPC UA information model can be used to express the information about the sensor, provided values, etc. Currently, the information expressed correspondingly to the OWL ontology (in RDF format) is used in OPC UA transferred values.

4.2 SBDH - Information Integration

For dealing with the information integration problem, SBDH exploits developed Cyber-Physical System Ontology for Components Integration (COCI). COCI is not a completely new ontology but is built on the top of Semantic Sensor Network (SSN¹¹) ontology, which has Dolce Ultra Light ontology as its cornerstone.

The most important concepts together with their relations are shown in Fig. 2. The concepts with the blue edge are from DOLCE Ultralight Ontology (DUL) and serve as general predecessors of all COCI concepts. There are also several SSN concepts (with the yellow edge) — general concepts from SSN ontology are reused such as SSN:Property, SSN:Process, and SSN:FeatureOfInterest

¹¹ <https://www.w3.org/TR/vocab-ssn/>.

instead of design similar concepts in COCI. Finally, there are shown the essential COCI concepts (with the green edge) representing entities related to an actuator.

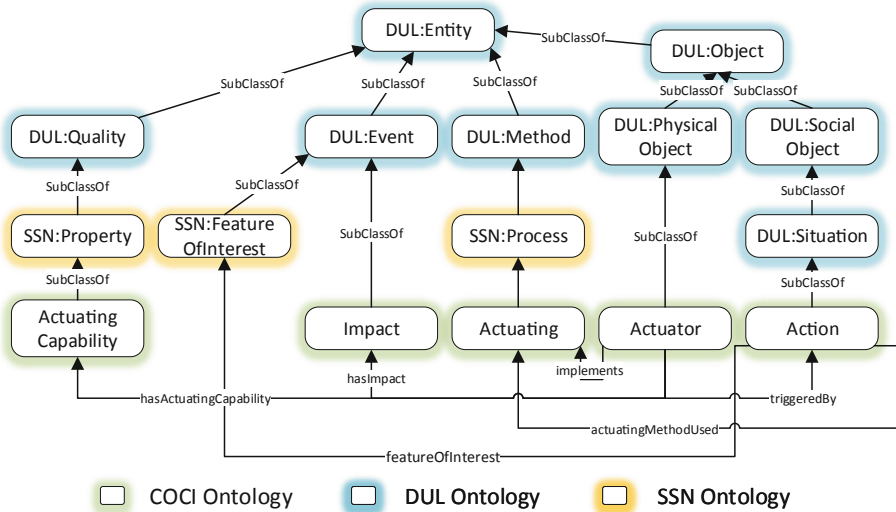


Fig. 2. Part of cyber-physical system ontology for components integration.

The cornerstone of SSN ontology (and of COCI respectively), DUL ontology, provides “a glue” for an integration of various data sources, i.e., the integration of various information. The example of the information integration using SBDH and COCI is presented in [10]. The example presents the integration of sensors from the hydroelectric power plant, information about the weather forecast, and information from the river catchment area.

5 Conclusions

In this paper, we describe the basic challenges and approaches to information exchange and integration problems. The information exchange solutions are primarily based on the exploitation of appropriate formats for communication with sufficient capabilities for given applications—appropriate formats were described. Next, the proper solution of the information integration problem resides in the utilization of suitable formalism for building information model and therefore appropriate formalisms were described with a short overview. Finally, a solution of the exchange and integration problems was demonstrated using Semantic Big Data Historian.

The information exchange problem is challenging primarily because of missing added meaning to exchanged information, i.e., many formats deal with a structure of given messages, not with their meaning. In other words, these

solutions are about the data exchange problem but not about the information exchange problem. Moreover, if the formats take a meaning of exchanging data into account, then there is still a big gap between exchanged information and their meaning in a given application.

Based on our experiments, a suitable solution for the information exchange problem is the utilization of Semantic Web technologies which enable to capture not only data or information but also to describe their meaning or relations among particular entities. Furthermore, everything is explicitly specified and thus easy to maintain and reuse. On the other hand, there is a significant impediment of exploitation of these technologies by manufacturing companies, and it is the relative complexity of Semantic Web technologies.

The information integration problem is complex primarily because of various expressivity of given information resources as well as a difficult understanding of the meaning of particular entities in given contexts. The suitable formalism for the solution of this problem seems to be the utilization of ontologies expressed in the Web Ontology Language. OWL is a flexible and versatile format for building information models and therefore is applicable to various problems. However, there is an identical obstacle to the information exchange solution. The OWL is a complex format and using it together with designing an ontology is a complex task. The good news is that there are reusable ontologies available for various domains, as demonstrated in our Semantic Big Data Historian.

Acknowledgment. This research has been supported by Rockwell Automation Laboratory for Distributed Intelligent Control (RA-DIC) and by institutional resources for research by the Czech Technical University in Prague, Czech Republic.

References

1. ANSI/ISA 95: Enterprise-Control System Integration [6 parts] (2000). <https://www.isa.org/isa95/>
2. IEC 62264: Enterprise-Control System Integration [5 parts] (2003)
3. Arnaud, R., Barnes, M.C.: COLLADA: Sailing the Gulf of 3D Digital Content Creation. CRC Press, Boca Raton (2006)
4. Brandl, D., BR&L Consulting Inc.: What is ISA-95. Industrial Best Practices of Manufacturing Information Technologies with ISA-95 Models. BR&L Consulting (2008)
5. International Electrotechnical Commission (IEC), et al.: IEC 62424. Representation of process control engineering-Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools (2008)
6. Drath, R., Luder, A., Peschke, J., Hundt, L.: AutomationML-the glue for seamless automation engineering. In: 2008 IEEE International Conference on Emerging Technologies and Factory Automation, pp. 616–623. IEEE (2008)
7. OPC Foundation: Unified architecture. <https://opcfoundation.org/about/opc-technologies/opc-ua/>
8. Jirkovský, V.: Semantic integration in the context of cyber-physical systems. Ph.D. thesis, Czech Technical University in Prague (2017)

9. Jirkovský, V., Obitko, M.: Enabling semantics within industry 4.0. In: Mařík, V., Wahlster, W., Strasser, T., Kadera, P. (eds.) HoloMAS 2017. LNCS (LNAI), vol. 10444, pp. 39–52. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64635-0_4
10. Jirkovský, V., Obitko, M., Mařík, V.: Understanding data heterogeneity in the context of cyber-physical systems integration. *IEEE Trans. Ind. Inform.* **13**(2), 660–667 (2016)
11. Mahnke, W., Leitner, S.H.: OPC Unified Architecture. Springer, Heidelberg (2009). <https://doi.org/10.1007/978-3-540-68899-0>
12. van der Wal, E.: PLCopen. *IEEE Ind. Electron. Mag.* **3**(4), 25 (2009)