Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 170 (2020) 602–609

Procedia Computer Science

www.elsevier.com/locate/procedia

The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40),
April 6 - 9, 2020, Warsaw, Poland

An Investigation on Database Connections in OPC UA Applications

Selvine G. Mathias[a,*], Sebastian Schmied[a], Daniel Grossmann[b]

[a]*Zentrum für Angewandte Forschung, Technische Hochschule Ingolstadt, Esplanade 10, 85051 Ingolstadt, Germany*
[b]*Faculty of Engineering and Management, Technische Hochschule Ingolstadt, Esplanade 10, 85051 Ingolstadt, Germany*

Abstract

Open Platform Communications - Unified Architecture (OPC UA) as a communication protocol has been one of the front running IoT enablers in recent years. Many use cases in manufacturing domains demonstrate the vast inter-operability and cross-platform connectivity strengths of OPC UA. However, to consolidate its position in the IoT world, OPC UA needs to be able to bridge certain gaps, like interconnections across multiple database engines over different networks. This paper aims to investigate aggregation dynamics of OPC UA consisting of multiple database servers from a specific environment such as manufacturing. Prevailing databases over restricted networks such as production floors cannot be accessed by respective clients outside the system. In these cases, OPC UA services can provide an a selective but imperative view of data from these databases to the clients. The investigations in this paper are demonstrated with two different implementations of databases related to manufacturing with OPC UA. These applications show that using OPC UA as a means of connection and distribution of information from databases can be an effective solution to a common data view of all connections not necessarily of factory floor network.

Keywords: OPC UA; databases; implementations; SQL server; clients

1. Introduction

As the processing power and the connectivity of industrial embedded devices increase, a range of applications become feasible and available. Inter-operability, modularity and portability of systems and software is the trend of these IoT applications so that vastly different domains can be usefully interconnected. Digitization in almost all fields requires persisting technologies which can adapt to new connectivity approaches and transferability of techniques, like operating systems and programming language independence. New paradigms for information and communication technologies are emerging as well as evolving rapidly, namely, Industry 4.0, Cyber-Physical Systems (CPS), Big Data, etc. The network of physical processes and systems, interfacing and interacting software connected with hardware

* Corresponding Author. Tel.: +49-160-635-0951
E-mail address: SelvineGeorge.Mathias@thi.de

10.1016/j.procs.2020.03.132

components, devices and machines is expanding to involve new systems for better performance and delivery [4, 3, 2, 13].

As the systems evolve, so does the process of maintaining and sharing data across the modules of the system. This is a primary requirement for industries to stay competitive and sustainable. Modern industries have their own operative technologies for this evolution, like automation processes, management systems like Manufacturing Execution Systems (MES) and Enterprise-Resource Planning (ERP) with advanced data acquisition systems like SCADA. Over such a vast network of devices, applications and protocols, it is imperative that information transfer from one location to another is monitored, maintained and analysed. This exchange can happen in a multitude of ways depending on the kind of interconnections established using CPS and IoT [8].

This paper aims to use OPC UA as a simple information transfer tool between digital objects like databases and networks, rather than as a device communication protocol. This would further widen the scope of using simple systems for data exposure over networks rather developing standalone applications. The paper is divided as follows: Section 2 discusses the current state of technologies with regards to OPC UA followed by Section 3 which discusses the implementation procedure. Section 4 shows the example implementations while the last two sections provide inferences of the study and further works in these directions respectively.

2. Review of Technologies

2.1. OPC UA - An Overview

OPC UA is a machine-to-machine communication technology based on which multiple devices, ranging from sensors to programmable machines, can be linked over a single communication thread [9]. Supervisory systems such as SCADA, MES and ERP are also within the scope of connectivity of OPC UA, thereby completing the automation scenario through its simplified yet strong structure.

OPC UA primarily acts as a client-server communication model which preserves an object data model and the corresponding mechanisms for accessing those data using a Publish-Subscribe (Pub-Sub) model. The desired functionality provided by the services can be accessed by means of secure communication channels or through optionally encrypted Simple Object Access Protocols (SOAP) [7, 12]. Hence, data exchanges are independent from the underlying data source type, host operating system and network configuration. Clients and servers implement sets of services for handling the communication and exchanging the actual data. These services include *Discovery*, *Subscription*, *Historical Data Access*, *Methods* and *Events* [18] which is why OPC UA is a Service Oriented Architecture.

To elaborate on this structure, a communication stack is used on client- and server-side to decode and encode message requests and responses. Multiple communication stacks work together to provide seamless communication to different servers and clients.

The implementations of these services are based on Web-Services, namely the standard HTTP protocol. The issues of secure access and unwanted infiltration attacks over internet are addressed through *Authentication Certificates*, which are provided to only the relevant communication participants [1, 14].

Another important aspect of OPC UA services is to expose data in a structured form without complex description and types. This component of OPC UA manifests in the form of information models. An information model typically comprises of objects related to other objects, defined as a node structure with types called as *Object Meta-Model* and can be edited in formats such as XML. Thus, an object, PLC for instance, in OPC UA is an instance of a specific type defined as PLC-type under Object Types which is contained within the information model. The relations between objects provide structure and behaviour within the model space which in turn gives a virtual view of working within a communication field such as a manufacturing floor [11].

2.2. Aggregation in OPC UA

Another OPC UA specification that builds on the concept of providing only 'needed' data is an aggregating server. This service provides information from more than one servers, thereby providing one access endpoint to all those servers for a client [5, 17]. This concept is helpful in extracting only required parts of information from a whole set of servers, rather accessing each server and complicating retrieval. Such a flexible mechanism allows for multiple clients to have access to the same information while having different requirements of data extraction.

An aggregation server can also feed information into an MES, thereby limiting hard-coding of data exchange between machines on the field level and MES. In this case, a front-end MES acts as an aggregating server too, thereby exposing information from multiple servers in one place of access [6].

2.3. Databases with OPC UA

OPC UA provides access to past values of object instances in a separate specification called Historical Data Access [10]. Under this service, a user can query a view of the server within a specified time range. Thus, OPC UA provides a snapshot of its history if so desired. A typical example software implementing this specification is the Prosys OPC UA Historian [15]. Developed using Java programming and the Prosys OPC UA Java SDK, it is a data logging software where a user can configure source servers to which the Historian connects and the values that the Historian collects to a SQL database. This way, the user can access the data using again OPC UA HistoryRead Services incorporated in the Historian itself or directly from the database using SQL queries.

However in real scenarios, it is more convenient to store these values in a well known storage and access components such as relational databases. Through these, the tasks of accessing data, memory management and providing inferences is more convenient than accessing the direct communication work flow through OPC UA on the manufacturing floor. For example, an enterprise manager needing the information on plant floor to build yield reports, need not have plant floor communication access to exclusively retrieve data. As such, standard databases like ORACLE, PostgreSQL and Microsoft SQL servers provide an efficient way to maintain, access and divulge only the necessary information. Through MES where machine values and dynamic events are directly logged in, tables are created in databases which provide a tabular access. Another layer of data access in such a structure can be implemented to provide a superficial view of underlying data through OPC UA.

Though seemingly trivial given many commercial applications that offer exposure to databases using OPC UA, it is relevant to explore the connection meshes between databases and OPC UA to pursue the idea of relevant information extraction for a client. Hence, this topic has been the focal point of this study.

3. Process of Implementation

A flexible manufacturing system allows the system to react to changes, predicted or not. Part of the flexibility involves dynamic addition, removal or replacement of machines or products during production. These events are logged into specific documentations such as spreadsheets or databases [16]. Regardless of the kind of documentation, it is essential that archival access is maintained and accessible for further analysis. A direct access to such databases depends on the users or clients needing the access and not all clients can be given access to these systems. This section describes a data retrieval structure through OPC UA where databases or dumps are translated to objects and can be viewed through network access.

3.1. Access to a Database

The connectivity of OPC UA architecture across devices is dependent on the implementation of the SDK's along with required companion specifications. For the purpose of implementations in this paper, JDBC and ODBC drivers were used in Java along with Prosys OPC UA SDK for server development. The connectors derived from JDBC driver connect to Microsoft SQL servers while those from ODBC drivers connect to Oracle databases. The connectors include the authentication requirement from the server side. Only a user having an authenticated schema access to the database would be able to trigger the server for exposure to clients.

Once the connectivity is established, a resultset query reads all the tables from the database and table objects are created depending on the requirements. Not all tables need to be exposed at this step as objects which gives the server better constraints on data disclosure. This can be achieved by modifying the resultset query used to extract data. For this study, all the tables under a database were extracted.

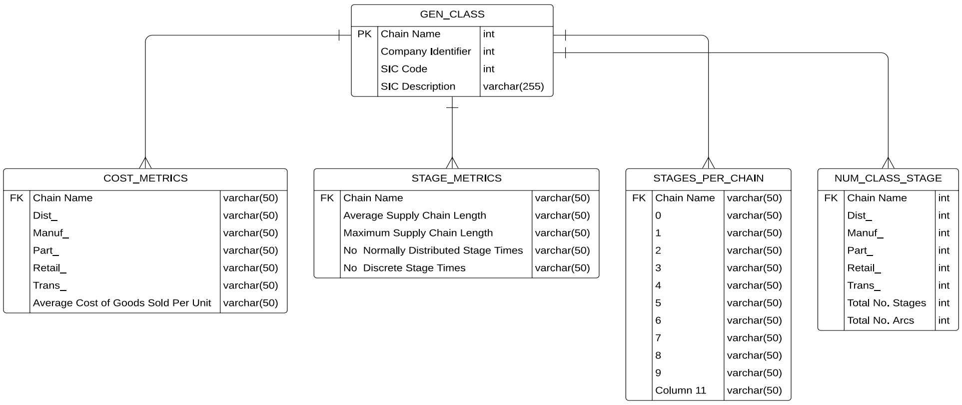

Fig. 1. The column-based ER diagram of a subset of tables from Supply Chain Database

3.2. Translation from ER to Information Models

As discussed in Section 2.1, information models form the crux of data transfer from servers to clients. A standard database has its own entity relationship (ER) model which connects one table to another, in other words, from one entity to another. This relationship enables users to extract required data using queries in SQL, but as an OPC UA object, each row of a table is exposed as an individual object for a clean view. The references are stored in the server for further aggregation in new servers, so the client needs to only access the aggregated servers for a restricted view of the data. This explicit model is created as an XML file with the help of standard UA Modeler provided by Unified Automation. An example of ER diagram giving column relationship is shown in Figure 1. for a subset of tables from database mentioned in Section 4.1.

Although not necessary in practice, this change of structure from ER to meta model ensures the ease of translating multiple tables having the same columns into multiple table objects on the server.

Once an information model is procured, either a standard UA server is implemented using this model or a more application specific approach can be used to begin transferring data to clients. Multiple servers can also be chained or triggered in parallel for a wider application decreasing the load on one server.

3.3. Aggregation of Databases

A simple aggregation server was developed for the purpose of combining the server results. At this stage, the complete address space of the target servers are read into the combined address space to be able to see all the data from the servers. Once aggregation is achieved, a customized UA Client can be developed or standard UA Client like UA Expert can be used to view the server data.

The aggregation is used in the context of pooling more security and address space layers. It is not an absolute requirement but assures that underlying servers are not overloaded with aggregation methods or queries. Additional layers of data transmission also ensures that batches of data passing through a server can be monitored if necessary before passing through to another server.

3.4. Data Flow Structure through OPC UA Layers

A conceptual view of information flow from databases to servers and then to clients is shown in Figure 2. Once the databases are connected, the severs are implemented as explained in the previous subsections with aggregated layers following if implemented. These stages are not completely necessary in an actual manufacturing environment, but serve as intermediates in sequentially providing data to clients while confining access to databases or data warehouses to limited and relevant clients only. The sequence of data flow is also not strict, as certain clients may have direct access to underlying servers, rather than only the aggregated servers.

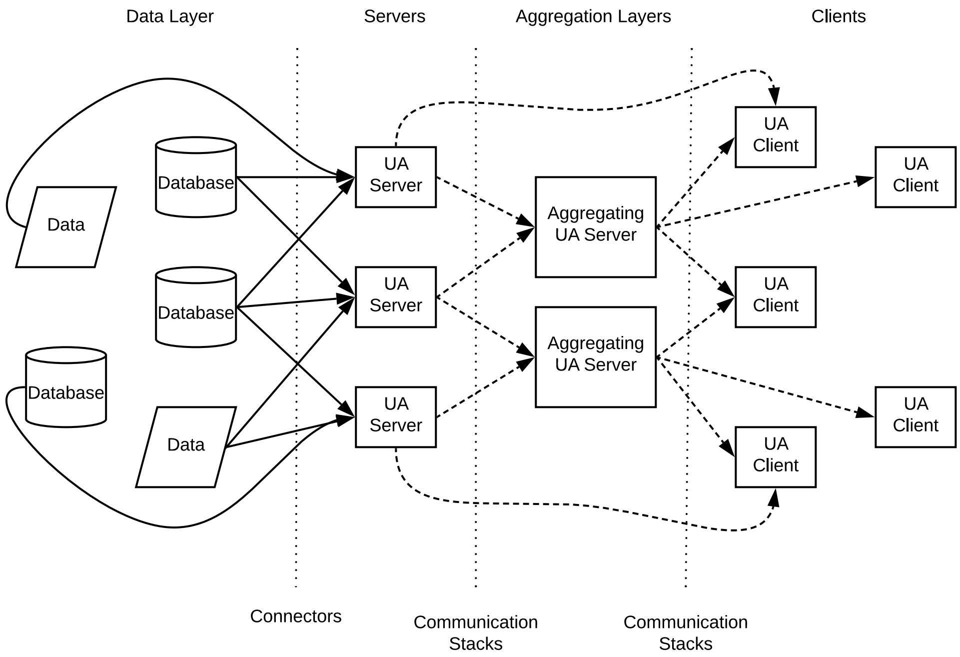

Fig. 2. Connection mesh from databases to clients

4. Example Implementations

This section discusses the implementations of the above mentioned process to expose data from databases to OPC UA objects.

4.1. Supply Chain Database Implementation

A supply chain is a network of retailers, distributors, transporters, storage facilities, and suppliers who take part in the production, delivery, and sale of a product that convert and move the goods from raw materials to end users. It describes the processes and organisations involved in converting and conveying the goods from manufactures to consumers.

This dataset [20] is an example for enterprise and management related data which can be exposed to clients outside the manufacturing or management network scope. It comprises of 38 multi-echelon supply chains across different companies which have implemented them in practice. Although this paper focuses on databases exposed from a single manufacturing company, this dataset can be considered synonymous with enterprise managers keeping survey of multiple supply chain models for the optimization of inventory and production. Such a dataset comprises of crucial information for potential suppliers, retailers and distributors looking forward to join the chain of production.

For the translation to OPC UA objects, the data was first fed to a Microsoft SQL server, since the available data was in the form of Excel sheets, followed by creation of an information model fitting the entity relationship data from the columns through UA Modeler. A part of the information model is shown in Figure 3.

The tables are referenced under the database object upon creation while each row of a table is referenced under the respective table object. Further ordering of tables is also possible by referencing tables under other tables or under other objects, but the retention of entity relationships is not under the scope of this paper.

4.2. Factory Floor Database Implementation

This archive contains 13910 measurements from 16 chemical sensors utilized in simulations for drift compensation in a discrimination task of 6 gases at various levels of concentrations [19]. As such, it can be considered synonymous

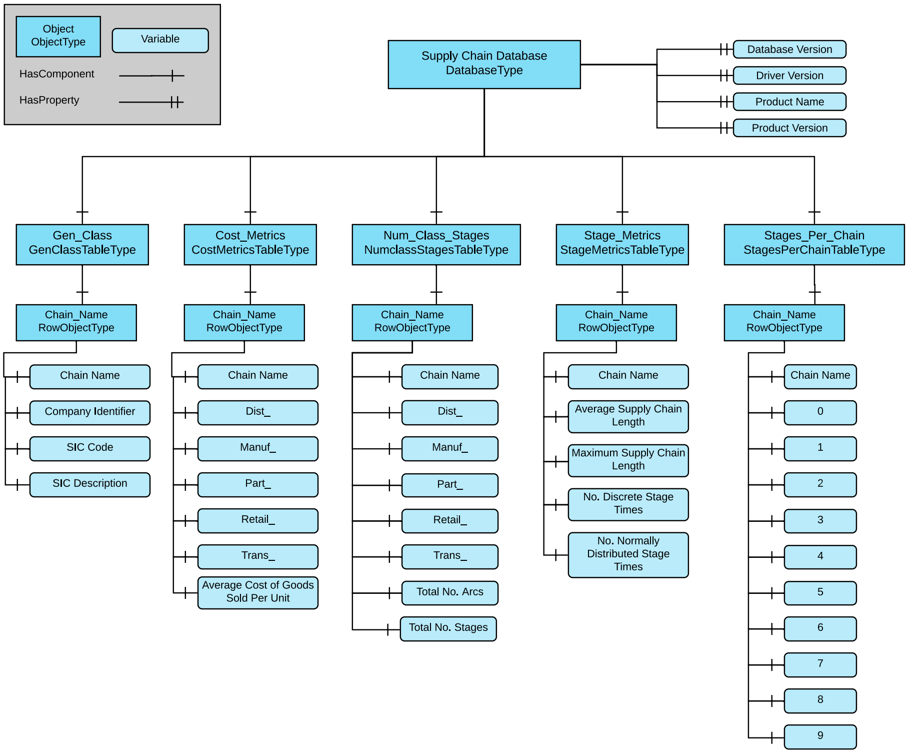

Fig. 3. Information Model of ER data from Figure 1

with in-house manufacturing tests during production where compressed gases are used. The data from this set was fed to an ORACLE server for implementation.

Another difference from the previous case is the absence of an explicitly created information model from UA Modeler for this database. Since the number of columns is very large, the manual task of creating an information model referencing each column and its subsequent row entities is a laborious. Hence, OPC UA objects such as database, tables and rows in this case, are created dynamically during server instantiation. This approach is useful when databases comprise of huge amount of archived and present data distributed in numerous tables and their subsequent columns. Hence, the creation of information models prior to server creation is ignored in this implementation for the purpose of avoiding complexity in server formation.

5. Observations

Figure 4. shows UA client connecting to two servers accessing databases and the aggregating server showing the consolidated address space for both databases. The address space displays properties of the respective database like driver and product versions with tables instantiated during server creation.

5.1. Approach Benefits

The approach of using OPC UA as an information retrieval tool rather than just a communication protocol highlights its vast applicability. Under the implementations discussed in Section 4, it is fairly straightforward to pass data from one network location to another just by means of sql connectors and servers. The following inferences can be drawn from the implementations:

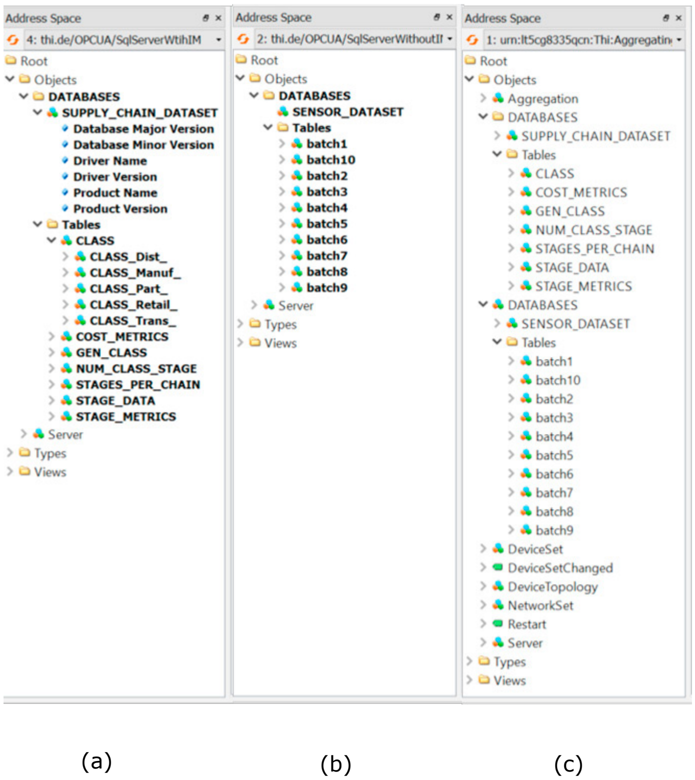

Fig. 4. Implementations (a) Database 1 (b) Database 2 (c) Aggregated View of Databases

- A complete database or a dump can be instantiated as an object in servers.
- An explicit information model in XML format as used in Section 4.1 is not absolutely essential for such an instantiation.
- Either all the tables or only certain tables can be exposed in a server. This enables more control on the amount and kind of data passed on to clients.
- By means of remote connections over HTTP Protocol of OPC UA, this data can also be viewed remotely.
- Any client using this server, even remotely, can be verified through authentication certificates from OPC UA. Thus, confidential database information remains within the boundaries of relevant clients without the need for a new schema access in the database.
- Aggregation servers are able to expose multiple databases from multiple servers.

5.2. Application Drawbacks

Some drawbacks have also been observed during implementations. These are primarily based on the structure and amount of data in the databases rather than the application or implementation themselves.

- The entity relationships which is evident in databases are not preserved in this approach. Since every database has its own ER structure, it is difficult to generalize this concept across servers.
- The aggregation of multiple servers has more latency in reading the complete address space if databases are huge. Thus, entire databases are aggregated at computational expense.
- The data extraction in this approach lies on the server side, rather than the client side. This can be remedied with query methods implemented in aggregation servers.

6. Conclusions and Future Works

A data transfer from databases to OPC UA servers was implemented in this paper. The use of OPC UA as more than just a bridge from devices to digital storage was shown by implementing its applicability the other way around, from digital storage to clients. Future applications could include the order mapping of data, that translates tabular data to more meaningful object-property form. This would also preserve the ER data from databases or at least provide a meaningful traceable order from highest element to the lowest element in the address space of the server. These concepts can be further applied to factory floor where devices and machines are ordered by a certain relationship, for example, sensors working under an industrial robot. Aggregations could also be replaced with other graph related databases which map data into dictionaries. With these techniques, computational complexity may decrease when exposing huge tabular data from databases.

These works are included in future directions for further development of OPC UA applications.

References

[1] Banerjee, S., Großmann, D., 2019. Opc ua and dynamic web services, in: Unknown (Ed.), ICCAE 2019, The Association for Computing Machinery, New York, New York. pp. 114–117. doi:10.1145/3313991.3313996.

[2] Dorofeev, K., Zoitl, A., 2018. Skill-based engineering approach using opc ua programs, in: 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), pp. 1098–1103. doi:10.1109/INDIN.2018.8471978.

[3] García, M.V., Irisarri, E., Pérez, F., Estévez, E., Marcos, M., 2017. An open cpps automation architecture based on iec-61499 over opc-ua for flexible manufacturing in oil&gas industry. IFAC-PapersOnLine 50, 1231–1238. doi:10.1016/j.ifacol.2017.08.347.

[4] González, I., Calderón, A.J., Figueiredo, J., Sousa, J.M.C., 2019. A literature survey on open platform communications (opc) applied to advanced industrial environments. Electronics 8, 510. doi:10.3390/electronics8050510.

[5] Grosmann, D., Bregulla, M., Banerjee, S., Schulz, D., Braun, R., 2014. Opc ua server aggregation — the foundation for an internet of portals, in: IEEE [International Conference on] Emerging Technologies and Factory Automation (ETFA), 2014, IEEE, Piscataway, NJ. pp. 1–6. doi:10.1109/ETFA.2014.7005354.

[6] Hoffmann, M., Büscher, C., Meisen, T., Jeschke, S., 2016. Continuous integration of field level production data into top-level information systems using the opc interface standard. Procedia CIRP 41, 496–501. doi:10.1016/j.procir.2015.12.059.

[7] Leitner, S.H., Mahnke, W., 2006. Opc ua - service-oriented architecture for industrial applications. Softwaretechnik-Trends 26.

[8] Lu, Y., 2017. Industry 4.0: A survey on technologies, applications and open research issues. Journal of Industrial Information Integration 6, 1–10. doi:10.1016/j.jii.2017.04.005.

[9] OPC Foundation, a. Opc unified architecture part 1: Overview and concepts: Release 1.04 specification.

[10] OPC Foundation, b. Opc unified architecture part 11: Historical access: Opc 10000-11 - ua specification part 11 - historical access 1.04.

[11] OPC Foundation, c. Opc unified architecture part 5: Information model: Opc ua part 5 - information model release 1.04 specification.

[12] Pfrommer, J., Ebner, A., Ravikumar, S., Karunakaran, B., 2018. Open source opc ua pubsub over tsn for realtime industrial communication, in: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1087–1090. doi:10.1109/ETFA.2018.8502479.

[13] Profanter, S., Dorofeev, K., Zoitl, A., Knoll, A., 2017. Opc ua for plug produce: Automatic device discovery using lds-me, in: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–8. doi:10.1109/ETFA.2017.8247569.

[14] Profanter, S., Tekat, A., Dorofeev, K., Rickert, M., Knoll, A., 2019. Opc ua versus ros, dds, and mqtt: Performance evaluation of industry 4.0 protocols, in: 2019 IEEE International Conference on Industrial Technology (ICIT), pp. 955–962. doi:10.1109/ICIT.2019.8755050.

[15] Prosys OPC Ltd, . Prosys opc ua historian. URL: https://www.prosysopc.com/products/opc-ua-historian/.

[16] Schmied, S., Grosmann, D., Denk, B., 2019. A systematic top-down information modelling approach for workshop-type manufacturing systems, in: Proceedings, 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Piscataway, NJ. pp. 1305–1308. doi:10.1109/ETFA.2019.8869377.

[17] Seilonen, I., Tuovinen, T., Elovaara, J., Tuomi, I., Oksanen, T., 2016. Aggregating opc ua servers for monitoring manufacturing systems and mobile work machines, in: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, [Piscataway, New Jersey]. pp. 1–4. doi:10.1109/ETFA.2016.7733739.

[18] Stopper, M., Katalinic, B., 2009. Service-oriented architecture design aspects of opc ua for industrial applications, Citeseer.

[19] Vergara, A., Vembu, S., Ayhan, T., Ryan, M.A., Homer, M.L., Huerta, R., 2012. Chemical gas sensor drift compensation using classifier ensembles. Sensors and Actuators B: Chemical 166-167, 320–329. doi:10.1016/j.snb.2012.01.074.

[20] Willems, S.P., 2008. Data set —real-world multiechelon supply chains used for inventory optimization. Manufacturing & Service Operations Management 10, 19–23. doi:10.1287/msom.1070.0176.