OPC Unified Architecture: A Service-Oriented Architecture for Smart Grids

Sebastian Lehnhoff, Sebastian Rohjans, Mathias Uslar
R&D Department Energy
OFFIS - Institute for Information Technology
Oldenburg, Germany
{*lehnhoff, rohjans, uslar*}*@offis.de*

Wolfgang Mahnke
Industrial Software Systems
ABB Corporate Research Germany
Ladenburg, Germany
wolfgang.mahnke@de.abb.com

***Abstract*—In this paper, the OPC UA is introduced as a key technology for realizing a variety of Smart Grid use cases enabling relevant tasks of automation and control. OPC UA is the successor of the established Classic OPC specification and state of the art regarding information exchange in the industrial automation branch. One of its key improvements over the Classic OPC is that the area of application is no longer restricted to industrial automation but OPC UA can be applied almost in every domain facing challenges in automated control. This improvement stems from a more generic and object-oriented approach. For the adoption of OPC UA to Smart Grids, two of the most important data models – the Common Information Model (CIM) and the IEC 61850 – have been identified to be integrated into OPC UA communication. In this contribution, basic OPC UA features and functionalities (information modeling, communication services, and information security) are introduced and discussed in the context of Smart Grids.**

***Keywords*-Automation, Communication, Service-Oriented Architectures, Smart Grids, Standardization, OPC UA**

I. Introduction

The current transition process in the electric utilities domain and power systems in general aims at establishing the so called Smart Grid combining different views, definitions, and aspects of existing concepts in the electric energy domain. One aspect appears to be a commonality between all existing definitions: the increasing use of Information and Communication Technologies (ICT) in order to support automation and energy distribution functionalities for Energy Management Systems (EMS) and Distribution Management Systems (DMS). The result of an integration of those two technological scopes will be an increasingly complex and highly dynamic power system with a multi-dimensional infrastructure.

Looking at other domains, e.g. commerce, manufacturing or logistics, the introduction of ICT-based technologies and mechanisms lead to profound changes in the processes behind core services and businesses. When looking at industrial automation strong parallels to the utility domain become apparent. Both application domains were, respectively are characterized by monolithic software systems and their supported processes. In industrial automation, the introduction of re-usable software components replaced this approach.

However, aside from real and obvious advantages, this posed novel challenges, e.g. standardized interfaces became more and more important to avoid costly and labor-intensive integration work. Furthermore, the need for high performance Human Machine Interface (HMI) and Supervisory Control and Data Acquisition (SCADA) applications and their vendor-specific proprietary interfaces required the development of appropriate standards in terms of both syntax and semantics. For this purpose, joined vendor-driven initiatives like the OPC Foundation established basic building blocks.

Initially, the OPC task force aimed at creating a standard for accessing real-time data based on Microsoft's OLE/DCOM technology for Windows operating systems. This work resulted in the creation of the OPC Classic, which is the prevailing standard in industrial automation, today [1]. OPC Classic covers aspects like data reading, writing and supervision of procedures (OPC Data Access), sending alarms (OPC Alarms & Events) and accessing saved historical process data (OPC Historical Data Access). Furthermore, extensions to those basic building blocks exist (OPC Complex Data, OPC Batch, OPC Data exchange) and some first platform-independent specifications for web service-based interaction (OPC XML-DA).

In [2], the ten most important drivers to create the new OPC Unified Architecture (UA) are discussed, among which the end-of-life cycle of COM/DCOM, the rising need for Internet-based communication, platform-independence, and using a common information model are listed as the most prominent factors.

The following sections of this contribution provide an overview of OPC UA and focus on its application to Smart Grid communications for both automation and communication systems. Section II provides two relevant scenarios of automated control and monitoring in the context of Smart Grids. Section III discusses the OPC Unified Architecture. In Section IV, we introduce the mapping of common energy domain data models onto the OPC UA and finally in Section V, we argue for OPC UA adoptions to the Smart Grids scenarios. Section VI provides some concluding remarks and provides an outlook on our current and future work.

II. Smart Grid Use Cases

Future Smart Grids emphasize the (at least) partial coordination of a large number of stochastic appliances to balance consumption and generation of electrical energy. Due to the sheer number and complexity of the overall system automated means of measurement and (partially unsupervised) control are indispensable.

A vivid example of the need for automated monitoring and control is the supervision and operation of offshore wind farms using the example of the German "Alpha Ventus", which is the first German offshore installation that was constructed on the high seas. The pilot project is located some 45 kilometers from the coast of Borkum and provides fundamental experience not only in the construction of an offshore wind farm but also on the operation of such a system that may not be directly accessible due to varying weather conditions. Twelve 5-megawatt class wind power turbines are operating at the Alpha Ventus test field: six AREVA Wind M5000 turbines and six REpower 5M turbines, resting on two different foundations. Whereas the AREVA wind turbines stand on tripods, the REpower turbines are mounted on jacket foundations in a water depth of 30 meters. In order to provide the required wind yield – which is considerably higher compared to its onshore counterpart – to compensate for the significantly higher investment and operational costs individual turbines have to be monitored in real-time and appropriate control actions optimizing turbine configuration (angle, speed etc.) have to be calculated and taken immediately.

A second example that is becoming increasingly important is distribution grid automation. Future Smart Grids will consist of flexible appliances and controllable operating equipment actively contributing to power quality and guaranteeing high supply reliability for its customers. Main tasks will be operation and maintenance of distribution system assets in order to improve quality of service, reducing operating costs, and increasing operational efficiency. In liberalized markets, utilities are sometimes even required to report on reliability performance and issues of quality, or have to define explicit performance targets, which may be penalized in case of violation. Distribution grid automation may also support applications such as fault detection and location as well as providing safety-critical ancillary services in real-time. In order to meet customer expectations of supply reliability inexpensive technological concepts for system development and operation are necessary.

Both examples presented here illustrate and exemplify the need for a service oriented automation and control architecture within current, respectively future Smart Grid scenarios and will be picked-up in Section V.

III. OPC Unified Architecture

OPC UA defines a set of services in a Service-oriented Architecture (SOA)-based fashion for communicating data. In order to understand the services' functionality first the information modeling capabilities of OPC UA are described explaining the data the services have to deal with. Afterwards, the communication services are explained and information security is considered. Finally, the approaches on standardizing information models based on OPC UA are introduced.

A. Information Modeling with OPC UA

In addition to the transport of data, i.e. the communication, OPC UA also provides the possibility of information modeling. Information modeling allows enriching data with meta-data and thus exchanging information with known semantic rather than just exchanging pure data. Using the OPC UA mechanisms the information modeling can be done vendor-specific or using standardized information models. Especially when using a standardized information model, a new level of interoperability can be reached. Not only data is exchanged in an interoperable way, but information with a clearly defined semantic. To define information models OPC UA provides a meta-model, called address space model in the specification (Part 3 [3]). This model consists mainly of nodes and references between nodes. Depending on their meaning the nodes are separated into node classes. For each node class there is a set of attributes describing the node. Common attributes cover the name and an unique identifier (NodeId). Other attributes are only deployed on specific node classes.

- Objects serve structuring the address space.
- Object types define the semantic of objects, complex object types also the structure of connected nodes like variables and methods.
- Variables contain data, provided by the attribute called value. Another attribute defines the data type of the value, like String or Int16.
- Variable types define the semantic of variables and complex variable types in addition the structure of connected nodes, in case of variables in particular subvariables.
- Methods define the signature of methods that can be called via the OPC UA interface.
- Data types define user defined data types that can be used in variables. This contains the encoding of the data type, thus clients can ask during run time for the encoding and use the user-defined types.
- Reference types define the semantic of references between nodes. Each reference is assigned to a reference type. In addition to a set of predefined reference types like composition and inheritance, user-defined reference types can be defined with their own semantic.
- Views define an excerpt of the address space by connecting only nodes needed for a specific task.

Using complex object types complex structures can be defined in the address space, reused in each application of

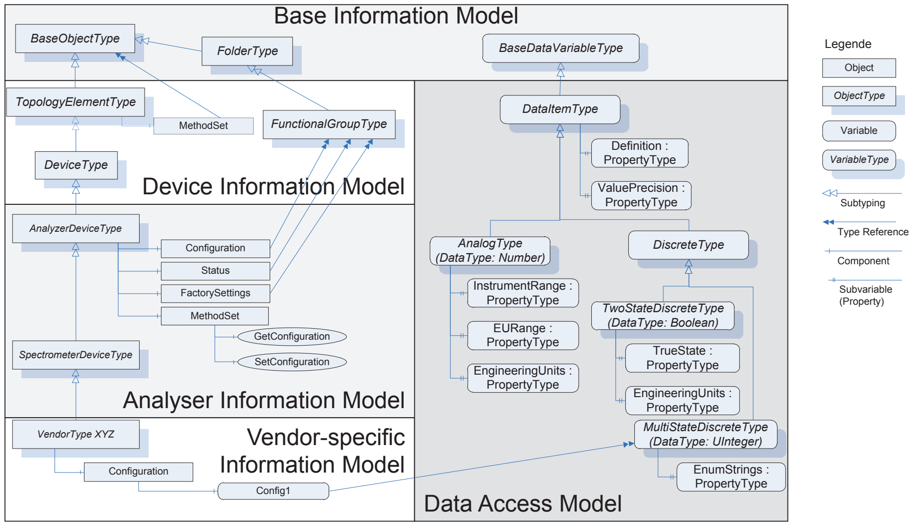

Figure 1. Example of OPC UA information modeling

the object type. This is comparable to a class in object-oriented programming languages. Accordingly there is support for inheritance of object types including the possibility to add characteristics to subtypes, like methods or variables. Based on the meta-model OPC UA already provides a base information model containing for example the base object type. The introduced concepts provide a variety of extension mechanisms. This starts at defining subtypes of the base object type or the base variable type, adding additional methods, variables or objects to object types or sub-variables to variable types, goes to the definition of user-defined data types and ends by defining user-defined reference types to bring additional semantic into the relations of nodes. Figure 1 contains an example applying information modeling using existing standardized information models based on OPC UA. The used notation is standardized by OPC UA and can be seen as the usage of UML (Unified Modeling Language) by applying UML stereotypes. The base information model defines the base types, where the device information model (also called DI - Device Integration) provides common types to describe devices like flow meter or temperature sensor, but also controller or IEDs (Intelligent Electronic Devices). Derived from that, the analyzer device model (also called ADI - Analyzer Device Integration) defines types of analyzer devices having concrete characteristics w.r.t. the configuration or supported methods (e.g. GetConfiguration in Figure 1). Each object of that type has the same structure and thus supports the same method. Finally, using subtyping vendor-specific extensions can be added, in Figure 1 indicated by VendorTypeXYZ. Variable types are already defined by the data access model which is part of the OPC UA specification. Those extended types offer the possibility to provide standardized information about the engineering unit ($^\circ$C, bar, etc.) or the precision.

B. OPC UA Communication Services

Communication in OPC UA is initially defined by abstract services (Part 4 of the specification [4]) that are mapped to different technologies (Part 6 [5]). This allows supporting new arising communication technologies in the future without changing the information modeling or the style of communication, just by defining another technology mapping.

1) Abstract Services: OPC UA provides a client-server based connection-oriented communication. To start a client one must establish a session. By doing this certificates and authentication information are exchanged between client and server. As soon as a session is established, the client can read and write data. This includes current data, i.e. attribute values of a node like the value of a variable but also other attributes to access the meta-data. The history of the data can also be accessed, for example the measured values of the last ten seconds or the last ten years. Using browsing or querying the structure of the address space can be accessed, including information about the nodes and references between them. In addition to those simple access methods OPC UA provides a subscription mechanism. Here, a value is not only read once but each change of the value is communicated to the client. In this exception-based communication approach only those data are submitted, that are needed. Using mechanisms like dead-band (small changes in the dead-band are not submitted, e.g. a temperature change from 22.0 $^\circ$C to 22.00001 $^\circ$C) and a sampling rate (defines the minimal time interval when a change should be propagated) the amount of submitted data is further reduced. Using this mechanism a client can for example easily show the current state of a wind power plant to the user. Mechanisms like heartbeat and acknowledgements guarantee a reliable communication, even if the underlying communication technology does not. In addition, OPC UA provides alarms and events. Events are transient and can be accessed via subscriptions. An event contains fields like Message, Severity and a unique identifier and can be extended by additional fields. Meta-data about the types of events are accessible in the server. Alarms contain states which can be read out of the server. They can be acknowledged and commented. An event is for example reaching a certain level of a boiler, whereas an alarm is reaching a critical level. Alarms and events are often shown to the user by specific alarm and event lists. The history of alarms and events is also accessible. The granularity of the services is service-oriented, i.e. a service call does not access a single value but a set of values at once. This reduces the number of service calls and the accompanying overhead.

2) Technology Mappings: To support different requirements, OPC UA offers different technology mappings of the abstract services. The mapping applies to the encoding of the data as well as the transport protocol. To allow a simple communication over the Internet to business applications like ERP (Enterprise-Resource Planning), a mapping to SOAP-based web services exist. The data can either be encoded in XML (Extensible Markup Language) or in an OPC UA specific binary format. The first mapping simplifies integration into the world of XML whereas the second option allows a fast and performing transfer of the data. To further increase the performance, a mapping to a self-defined TCP/IP based transport protocol is available. Using this variant the high requirements on communication in control systems can be fulfilled. In Figure 2 the different profiles and their technology mappings for transport and encoding are shown, including the used security mechanisms. The security mechanisms use the existing WS' specifications directly (in case of SOAP-based transport), or an adoption of them to the TCP/IP based transport protocol. Using an adequate architecture with a communication stack having interfaces based on the abstract services an application (client or server) can be built independent of the technology mapping. Additional technology mappings can be later added to the stack only. Such stacks are for example provided by the OPC Foundation.

C. Information Security

One particular aim during the development of the OPC UA was the aspect of information security. Unlike the predecessor Classic OPC and its standards, OPC UA cannot only be used in closed and isolated automation environments, which pose new requirements in terms of security. Automation systems can now be easily linked to company intranets or Internet-based systems which put them also in

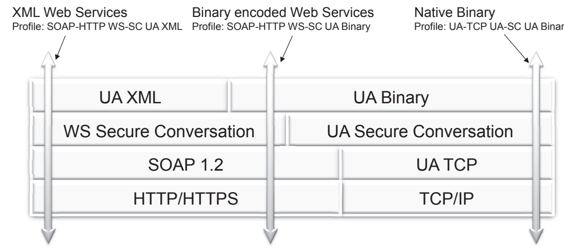

Figure 2. Technology mapping of OPC UA

the focus of malware and viruses. In order to cope with those problems, OPC UA provides a secure communication infrastructure. Within its scope, six protection goals are recognized: authenticity, authorization, confidentiality, completeness, availability, and traceability. Possible threats to OPC UA-based systems include different protection goals and communication layers and may be prone to message flooding, eavesdropping, and message spoofing. A generic approach for the security architecture was chosen at design time allowing implementing the needed functionalities within the security architecture. Based on the chosen technology mapping protection goals are addressed on different levels whereas the security architecture differentiates between transport, communication and application layer. Session services are used at application level in order to provide sessions based on secure channels from the communication level. Depending on the layer to be addressed, established non domain related security technologies are reused. SSL/TLS based specifications or WS Security, WS Secure Conversations, XML encryption, and XML Signature are used just to name a few.

D. Standardized Information Models

Standardized information models, i.e. specifications for a certain domain, have already been developed for different application areas. The OPC Foundation identified several potential cooperation partners (EDDL, FDI, ISA (S88 and S95), MIMOSA, PLCopen, and IEC TC 57 WG 13) in order to jointly develop Companion Specifications. Currently, (May 2012) three companion specification have been published but it can be assumed that further Companion Specifications will follow:

- **OPC UA for Devices:** Defined in a combined effort of the FDT-Group, Fieldbus Foundation, HART-Communication Foundation, OPC Foundation, and PROFIBUS-Nutzerorganisation (PNO), a generic model was developed representing devices. This model is the foundation for FDI (Field Device Integration), the current solution for field device integration, combining the advantages of FDT (Field Device Tool) and EDD (Electronic Device Description) [6].

- **OPC UA Information Model for IEC 61131-3:** In a joint effort of the OPC Foundation and PLCOpen and information model for the programming model of IEC 61131-3 is defined allowing a standardized mapping of function blocks, variables etc. defined in IEC 61131-3 to OPC UA [7].
- **OPC UA for Analyzer Devices:** Developed by a working group of the OPC Foundation the analyzer devices model defines a concrete model of several different types of analyzer devices like spectrometers or chromatographs [3].

Concluding, this introduction of OPC UA emphasizes that almost all requirements of a SOA are met. Beside the obvious service-orientation covering all required service characteristics, also the interoperability and loose coupling issues are dealt with.

IV. Smart Grid Data Model Mappings onto OPC UA

With its information modeling capabilities, OPC UA offers a high potential for becoming the standardized communication infrastructure for various information models from different domains. In the following, mappings of existing information models in the power domain to OPC UA are described: the Common Information Model (CIM; IEC 61970/ 61968) [8] and IEC 61850.

A. Common Information Model IEC 61970/61968

The mapping of CIM to OPC UA is already discussed within the IEC who are working on a draft version of the mapping (IEC 61970-502-8). In this context, CIMbaT has been developed at the OFFIS - Institute for Information Technology as a publicly available Enterprise Architect Add-In implemented with C# in Visual Studio supporting the generation of CIM-based address spaces [9].

In the design steps the user is offered to set OPC UA properties like *IsAbstract*, *SupportsEvents*, *Historizing*, *DataType* etc. for each CIM-class and their attributes and associations. That means, for example the engineer can override the default OPC data type integer or float. Furthermore, the decision whether a CIM-class attribute shall be mapped to an OPC *Property* or *DataVariable* is possible in the CIM-designer. The manipulable CIM-elements can be easily selected within a tree-structure. For each setting there is a specific stereotype within the CIM-UML-model which will be added to the updated CIM-element. These UA-specific stereotypes are unique and include a value like true or false. They will be recognized in the mapping. If the tool cannot find an UA-specific stereotype for a CIM-element, then the default values from the configuration file will be used. It is also possible to design UA *Views* and manage them. The *Views* are used for limiting the visible *Nodes* and *References*. The designed *Views* will also be saved as UA-specific stereotypes and mapped to an OPC *View-Node*. Because the changes can be saved as stereotypes in the CIM-model, the design choices are preserved without changing the CIM-model except regarding added stereotypes.

In coordination with the IEC and the decisions from the engineer the CIM-elements will be mapped as depicted in Figure 3. Thereby, the two branching arrows mean that the designer can choose between different options. It depends on his flavor of modeling and on the overall environment. Merging arrows only express that different CIM-elements may be mapped onto the same OPC UA structure.

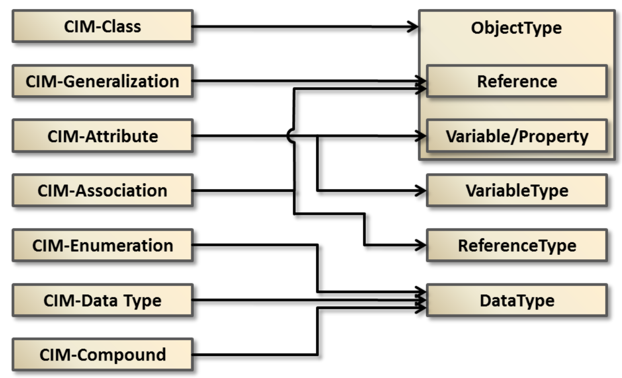

Figure 3. Mapping of the CIM data structure onto the OPC UA address space [9]

B. IEC 61850

Unlike the CIM the IEC 61850 does not only provide a simple data model but in addition mechanisms for the communication infrastructure like Functional Constraints (FC) for filtering the data, or timestamps and quality of the exchanged data. The IEC 61850 uses its own mechanisms to define its model and is not based on a pure object-oriented approach using UML (although the latest version of the IEC 61850 uses UML to document their approach [10]). Thus, the mapping cannot be performed in the same fashion as with the CIM. Different approaches may be chosen to map the IEC 61850 model to an OPC UA information model. For example, it has to be decided whether specific attributes of the IEC 61850 like quality and timestamp should be mapped the same way as all other attributes or handled specifically using the built-in OPC UA mechanisms having status codes and timestamps on each value. Furthermore, the FC defined for attributes in IEC 61850 could be made available in OPC UA using different modeling alternatives. In this section one possibility for the mapping is introduced. For the introduced mapping, the following decision were made and depicted in Figure 4:

- Logical Node (LN) Classes as defined in IEC 61850-7-x are generally mapped onto UA object types.
- LNodeTypes are generally mapped onto UA object types subtyping the LN Class.

- LN are generally mapped onto UA objects as instances of LNodeTypes.
- LN Data as the attributes of LN are mapped onto UA objects.
- CDC are also generally mapped onto UA object types.
- CDC DataAttributes as the attributes of CDC are mapped onto UA variables.
- CDC DataAttribute Types are the types of the CDC attributes and mainly mapped onto existing UA standard data types like Integer, Float and String.
- FC are mapped onto UA objects.

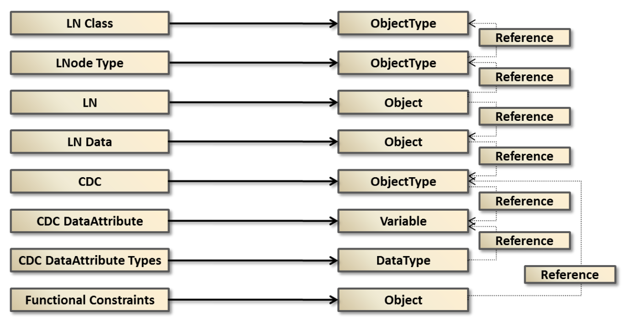

Figure 4. Mapping IEC 61850 data structure onto OPC UA address space [11]

In order to structure the objects three standard UA reference-types are used:

- *HasComponent* describes a part-of relationship between LN and its attributes as well as between CDC and its attributes. Furthermore, it is used for the grouping by FC.
- *Organizes* is used to group the CDC attributes by FC.
- *HasTypeDefinition* connects the LN attributes with the according CDC.

Mapping Example: The example includes the Logical Node Class (LN Class) MMXU and the Common Data Class (CDC) MV as well as their attributes. MMXU is a LN Class which shall be used for calculation of currents, voltages, powers and impedances in a three-phase system. It is mainly used for operative applications. The CDC MV represents measured values. We focus on only three attributes of the MMXU: TotVA (Total Apparent Power), TotVAr (Total Reactive Power) and TotW (Total Active Power). Also for the MV, we consider a limited number of attributes which can be divided by the FC. FC shall indicate the services that are allowed to be operated on a specific attribute. The attributes instMag (magnitude of a the instantaneous value of a measured value), mag (current value of instMag considering deadband), q (quality of the measured value), t (timestamp of the measured value) and range (range of the current value of instMag) belong to the FC MX (Measurands) and the attributes subEna (used to enable substitution), subMag (used to substitute the data attribute instMag) and subID (shows the address of the device that made the substitution) to the FC SV (Substitution). This is similar to modeling parameters for devices as defined in [6]. The mapping shows that it is possible to expose the IEC 61850 model in OPC UA. By providing the LN Class and the LNodeTypes in the UA address space, it is possible that pure OPC UA clients without any previous knowledge of the IEC 61850 can make use of the type model and design for example specific HMI elements for any MMXU.

V. Adopting OPC UA for Smart Grids

In the light of the general discussion on features of SOA and OPC UA in as well as the possibility to map certain data models relevant in the context of Smart Grids to the OPC UA, we come back to the scenarios introduced in Section II.

In order to fulfill the monitoring and control tasks, the offshore wind turbines of Alpha Ventus have been equipped with an OPC UA interface by the company Beckhoff Automation using an SDK by Unified Automation, a software distribution and development company specialized in OPC-based solutions and components for industrial automation[1]. Even though the wind turbines are stand-alone (that is self-controlled i.e. being able to run in basic feasible operation mode without supervision), onshore monitoring from the mainland is mandatory for safety reasons and necessary for optimal configuration and maximum yield. Hence, OPC UA was chosen due to its security model and mechanisms of authentication. With the integration of the OPC UA communication into Programmable Logic Controllers (PLC), the risk of failures has been minimized since no additional software had to be installed. In order to monitor, supervise and control these wind turbines the SCADA system had to be expanded by OPC UA client functionality fully supporting Alpha Ventus control.

An increasing amount of generation from distributed energy resources (DER) as well as from high-capacity consumption, e.g. from electric vehicles, poses novel challenges to distribution grid automation in terms of protection as well as automated control and monitoring. Substation automation systems in this area have to be modernized considerably. Major disadvantages to current systems are the high costs associated with proprietary and single-application-systems (it often happens that a system supports only a single function) as well as complex project set-up planning often causing high follow-up costs for updates and maintenance.

In order to allow for application scenarios in the domain of substation automation in future Smart Grids. a more flexible architecture is needed that supports functional detachment of software, hardware and communication. Thus, rendering efficient project planning possible. With the application of

[1] http://www.automationworld.com/opc-ua-connects-wind-turbines-offshore-wind-park

OPC UA capable systems, functions may be migrated on the fly among different devices supporting scenarios mentioned in Section II. The most prominent Use Cases in this scenario are:

- Reconfiguration of protection settings according to current DER set points taking into account the real-time requirements associated with the response characteristics of protection systems.
- Increasing availability through functional redundancy and dynamic device allocation supported by OPC UA-based separation of hardware and software (functions) in a standardized system.
- Adjustment to regulatory updates and directives through software patches easily deployed within an OPC UA client server system.
- Easy migration of functions on to new or updated hardware systems allowing for a more efficient project planning across hardware life cycles.

A consequent and consistent application of OPC UA in substation automation enables these use cases amongst others, due to various provided features. First, the concept of UA profiles enables both implementing embedded servers for small devices and complex servers for large SCADA systems. This also enables modeling simple as well as very complex information sets. Furthermore, UA fosters backend integration of different data models like CIM and IEC 61850 [12] which both play a vital role in the context of substation automation. Components, systems, and devices could be integrated seamlessly if the representing UA server provides information about the implemented data model. UA clients could use that information easily to establish a connection. Third, a comprehensive use of UA technologies in the context of substation automation offers a use cases dependent choice between XML-based data exchange for complex information, e.g. topologies being exchanged beyond firewalls and binary encoded data for time-critical communication like real-time control. This automation scenario is subject to ongoing R&D projects and we will present first exemplary results in further publications.

VI. Conclusions

In this paper we introduced OPC UA as a successor to the widely-used Classic OPC specification enabling existing, respectively future Smart Grid scenarios. It represents the state-of-the-art in communication in the domain of industrial automation. Due to the advances and upgrades of OPC UA over the Classic OPC its area of application is no longer limited to industrial automation. With its object-oriented generic approach it meets the requirements on data exchange in even the most challenging areas of application, e.g. future Smart Grids.

For the adoption of the OPC UA to the energy domain two dis-similar information models have been discussed, which rely on core data models for the representation of information in Smart Grids that can be integrated by OPC UA. Especially, the progress on the specification of a CIM-based information model has gone far and yields an advanced and sophisticated model. As a result, both the OPC Foundation as well as the IEC (IEC 61970-502-8) are working on mappings between the UA and CIM. In the same way, the OPC Foundation aims at implementing working groups that focus on companion specifications in the domain of Smart Grids, e.g. a mapping between the ZigBee Smart Energy Profile (SEP) 2.0.

In conclusion, the application of the OPC UA offers a high potential in the domain of Smart Grids and it is foreseeable that many companies will follow successful ”early adopters” like ABB, Beckhoff and Siemens in implementing the OPC UA.

References

[1] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC Unified Architecture*. Springer, 2009.

[2] J. Lange, F. Iwanitz, and T. J. Burke, *OPC: From Data Access to Unified Architecture*. Huethig, 2010.

[3] OPC Foundation, “OPC Unified Architecture Companion Specification for Analyser Devices - Release 1.00,” 2009.

[4] ——, “OPC Unified Architecture Specification Part 4: Services - Release 1.01,” 2009.

[5] ——, “OPC Unified Architecture Specification Part 6: Mappings - Release 1.00,” 2009.

[6] ——, “OPC Unified Architecture for Devices (DI) Companion Specification - Release 1.00,” 2009.

[7] OPC Foundation and PLCopen, “OPC UA Information Model for IEC 61131-3 - Release 1.00,” 2010.

[8] M. Uslar, M. Specht, S. Rohjans, J. Trefke, and J. M. González, *The Common Information Model CIM: IEC 61968/61970 and 62325 - A practical introduction to the CIM*. Springer, 2012.

[9] S. Rohjans, K. Piech, M. Uslar, and J.-F. Cabadi, “CIMbaT - Automated Generation of CIM-based OPC UA-Address Spaces,” in *IEEE SmartGridComm 2011*, Brussels, 2011.

[10] A. Apostolov, “IEC 61850 Substation Configuration Language and Its Impact on the Engineering of Distribution Substation Systems,” in *Congreso Internacional de Distribución Eléctrica (CIDEL2010)*, 2010, pp. 1–6.

[11] S. Lehnhoff, W. Mahnke, S. Rohjans, and M. Uslar, “IEC 61850 based OPC UA Communication - The Future of Smart Grid Automation,” in *17th Power Systems Computation Conference (PSCC’11)*, Stockholm, 2011.

[12] S. Rohjans, K. Piech, and W. Mahnke, “Standardized Smart Grid Semantics using OPC UA for Communication,” *IBIS - Interoperability in Business Information Systems*, vol. 6, no. 10, 2011.