OPC UA and CIM: Semantics for the Smart Grid

Sebastian Rohjans, Mathias Uslar, and H.-Juergen Appelrath

***Abstract*—In our approach, we model an ICT-architecture for the utility domain for both transmission and distribution grids according to the IEC standards 61968 and 61970. To achieve this, we use three basic components. The first component is the Common Information Model (CIM) which is standardized by the IEC 61970/61968 standards. The OPC UA (Unified Architecture) is the second basic component. The OPC UA is also part of an IEC standardization process, called IEC the 62541 family. The concept of semantic web services is the last component. Our proposed architecture combines those three components to build a standards-based smart grid architecture and provides an intelligent, service oriented system for the utility domain. In most smart grid use cases, users know what they are searching for (e.g. the services mentioned in the NIST Interoperability study) but they do not know the proper syntax or system providing services. Hence, the user or system should find the matching service using semantic functions for the search. Our concept uses next generation techniques to become a state-of-the-art interface concerning information exchange between Energy Management Systems (EMS), Distribution Management Systems (DMS) and Enterprise Resource Planning (ERP) systems.**

***Index Terms*—CIM, OPC UA, Semantic Web Service, Smart Grid, SOA, Services, DMS, EMS, Architecture**

I. Introduction and motivation

THE utility domain is facing new problems in both transmission and distribution systems. Better reliability and more efficiency of the overall systems is one of the main goals of what is known as the smart grid. Characteristics to achieve are to enable active participation in the system for the consumer, accomodate all generation and storage options, enable new services, products and markets and to provide a proper power quality for what is known as the digital economy [3]. This should be accomplished by optimizing the asset utilization and operate the power grid more efficiently. As natural disasters and terrorist attacks become more likely and security by obscurity is by far no option, the new grid should become both anticipating and responding to system disturbances and achieve self-healing capabilities. This should make it possible to operate resiliently against third party attacks and natural disasters. The U.S. NIST[1] (National Institute for Standards and Technology) has stated that this is a problem with many challenges. The smart grid incorporates a lot of stakeholders, it has to deal with a very high complexity of the system, it should facilitate a whole industry transition, it brings up new security and safety issues and it has to be an enabler for a consensus on standards as well as their development and support. Technically, there are many challenges in terms of smart equipment, communication systems, data management, cyber security, information and data privacy and new software applications. To start dealing with the new systems and the transition process, requirements must be mature and readily developed. Furthermore, well-developed standards are already in place and mature architectures guide the development of the new smart grid [6]. This contribtuion provides an example based on IEC standards of how to deal with the new IT-architectures for the smart grid for both transmission and distribution. It focusses on the standards from the IEC 62357 seamless integration architecture of the IEC TC 57 committee [25] and takes into account the Common Information Model CIM (IEC 61970 and 61968) [7] and combines these with the new IEC 62541 OPC Unified Architecture (UA) which is the predecessor of the OLE for Process Control; a mature standard in the automation domain. This alone provides a proper architecture for a new generation of grid automation. Building on top of it, we extend the web services part of the OPC UA with what is known as semantic web services. The overall idea is to foster a better service discovery in the large and heterogeneous smart grid IT-landscape where different parties operating different systems have to work together.

We introduce the basic components of our architecture in the following three chapters. In chapter two a short introduction of the CIM will be given. Chapter three includes information about the OPC UA and semantic web services are described in chapter four. Chapter five explains the re-combination of the basic components mentioned. The state-of-the-art as well as our novel development approach is shown in chapter six. In chapter seven, an overview of future work is given.

II. Common Information Model

The Common Information Model (CIM) is used within the electric utility domain for both distribution and transmission energy management systems. The electronic model is developed using the UML (Unified Modeling Language), the lingua franca for modelling in software engineering, and the CIM is published by the CIM Users Group[2] and the IEC. The core packages of the model are specified in the IEC 61970-301 standard. Some additional packages are specified in the IEC 61968-11 [13] standard. The data model includes several main packages with different functionalities. These packages include sub packages and classes with their attributes and associations. This set of abstract classes, attributes and associations represents physical objects like devices and abstract objects like market operations.

Because of platform independence it is possible to model different domain specific problems and cases. The principle

S. Rohjans is with OFFIS - Institute for Information System, Germany, Oldenburg, rohjans(at)offis.de

M. Uslar is with OFFIS - Institute for Information System, Germany, Oldenburg, uslar(at)offis.de

H.-Juergen Appelrath is with OFFIS - Institute for Information System, Germany, Oldenburg, appelrath(at)offis.de

[1] http://www.nist.gov/index.html

[2] http://cimug.ucaiug.org/default.aspx

purpose of the IEC 61970 standard series is to standardize a set of guidelines to support the integration of applications in the control center (CC) environment, independent of suppliers and to support the information exchange with systems external to the control center environment [22]. Similar objectives can be found in the EPRI (Electric Power Research Institute) Control Center API (CCAPI) project because the standard series is based on this research project which had the intention to develop an integration framework for already existing systems and applications which have to exchange data and information. Part of this framework should be a common architecture and an information model which are independent of the underlying system. They should fulfill three primary objectives:

- Reduce the cost and time needed to add new applications to an EMS or other system;
- Protect the investment in existing applications that are working effectively;
- Improve the capability to exchange information between disparate systems both within and external to the control center environment.

Three main use cases for the CIM exist [23]. Because the data model can be used as a large domain ontology, one can build custom messages to be exchanged between systems in the utility based on CIM semantics and syntax, e.g. curve schedules based on XML serializations. Furthermore, the exchange of instance data of the power grid - the topology itself - is standardized. Profiles describing which objects from the CIM should be used to model distribution grids or transmission grids and how to serialize them using the RDF[3] (Resource Description Framework) standard from the W3C exist. Those are the two predominant uses of the CIM. The third deals with standardized interfaces between systems and takes interface specifications into account.

The IEC 61970-402 [8] contains a concept called "IECTC 57Views". This concept adresses the problem of a lack of instance object metadata in common interface description languages like Microsoft's Common Object Model (COM) or OMG's Common Object Request Broker Architecture (CORBA). The announcement and discovery of interface metadata is seen as a key issue in the case of constructing an interoperable infrastructure. Hence, the "IECTC 57Views" provides a hierarchical model including agreements on which CIM classes appear and which CIM associations are traversed and in which direction. The model is divided into two top nodes, the "IECTC 57PhysicalView" and the "IECTC 57EventingView". The "IECTC 57PhysicalView" represents a tree view including one or more hierarchies of classes, class attributes and object instances defined in the CIM. The "IECTC 57EventingView" represents a tree view including one or more hierarchies of event classes that consist of arbitrary collections of attributes; typically from the CIM.

Besides the IEC 61970-301 part of the series that defines the mentioned information model, there are several other parts. A large part of the standard family defines the Component Interface Specification (CIS). In these parts, interfaces which could be implemented by components or applications to exchange data in a standard way are specified. Apart from the framework and the common services, the following data access methods are taken into account:

- Generic Data Access (GDA)
- High Speed Data Access (HSDA)
- Generic Eventing and Subscription (GES)
- Time Series Data Access (TSDA)

The specifications for these interfaces are based on several OPC standards (OPC DA, OPC AE and OPC HDA) which are part of the next chapter and predecessors of the new OPC UA.

III. OPC Unified Architecture

The OPC UA is a server-client-architecture which is developed by the OPC foundation[4]. It is an improved version of the well known OPC specifications. The OPC specifications consist of several standards for different purposes. They are used for the exchange of real-time plant data among control devices. These control devices can be manufactured by different vendors because OPC specifications are based on Microsoft's Distribution Component Object Model (DCOM), thus giving a kind of interoperability. The main problem that led to the development of the OPC UA is that the existing specification only ensures the interoperability within local networks. The OPC UA should be a platform for interoperability between the existing OPC specifications using web services. In the context of the OPC standards, the OPC UA is the top level standard used to provide platform independent and service-based communication. So far, the existing OPC specifications use a platform-dependent and component-based approach.

The OPC UA is standardized as the IEC 62541 [15] standard series. Some parts are still in the standardization cycle being draft versions, other parts are already published as international standards. The parts of the series are divided into two groups. The first seven parts are related to core specifications like the concept, security model, address space model, services, information model, service mappings and profiles.

The parts eight to eleven are related to access type specifications like data access, alarms and conditions, programs and historical access.

Concerning the CIM, the current OPC specifications are already in use. The following parts of the IEC 61970 series are based on those specifications:

- IEC 61970-404 High speed data access (HSDA): OPC DA (Data Access) defines an interface that can be used to read and write real-time data [9].
- IEC 61970-405 Generic eventing and subscription (GES): OPC AE (Alarm and Events) defines an interface that can be used to monitor events [10].
- IEC 61970-407 Time series data access (TSDA): OPC HDA (Historical Data Access) defines an interface that can be used to access historical data [11].

The combination of the CIM and the new OPC UA is still an open issue which we will deal with in the later parts of this contribution.

[3]http://www.w3.org/RDF/

[4]http://www.opcfoundation.org

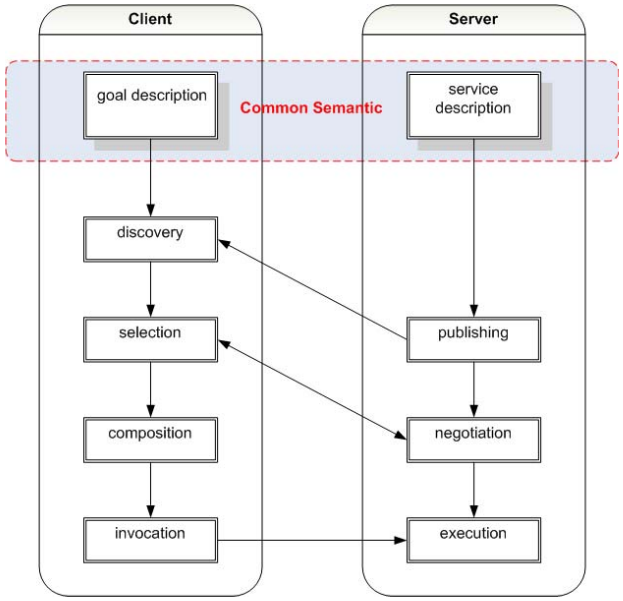


Fig. 1. Communication process for semantic web services [2].

IV. Semantic Web Services

Semantic web services are improvements of the known concept of web services. The improvement is to bring semantic content into the web services which only contain information about syntactic content so far. The semantic web services are amongst others specified by the W3C[5] and could be seen as a new approach in the course of the Semantic Web. Their intention is to exploit the full potential of the Semantic Web [5]. Web services are software applications which communicate with each other by using XML-interfaces to send messages via internet protocols. Each web service is identifiable by its Uniform Resource Identifier (URI). Three types of roles participate in a typical web service system [2]:

- Service Broker
- Service Provider
- Service Requester

The service provider uses the WSDL (Web Services Description Language) standard to provide its services to the service broker. In some cases, a small and local server is used, it uses the UDDI (Universal Description, Discovery and Integration) standard to register web services. The service request can also use WSDL to communicate with the service broker. It queries the broker's repository to find a QoS or requirements fitting service. In case of success, the service requester exchanges the data with the chosen service provider by using the SOAP standard for example.

In figure 1, the communication process between service provider (server) and service requester (client) is shown and extended to a semantic level. The individual steps will be explained in detail next.

The communication process starts with the publishing of the server. Therefore, it is necessary that the description of the offered web service is specific. Those descriptions should be written in formal machine-readable languages to support automation. Thus, it is also import for the client's first step within in the communication process to describe the desired service very specific, otherwise the discovery could fail even if there are fitting web services within the repository. During the next step, server and client communicate in a bidirectional manner. The discovery could result in a list of services which offer the desired functionalities, hence the client has to select one of the fitting services to negotiate a Service Level Agreement (SLA) with the server. To select a proper web service, the client could use predefined classes to achieve Quality of Service (QoS). There could be the need for the client to compose several web service because only a composition would lead to the desired functionality. During the composition process it is necessary to determine an order in which the composed web services must be invoked. The last step on the client's side is to invoke the appropriate web services and to communicate with the server to make it execute the web service. So far it is only possible to define the message format (e.g. SOAP) and the exchange protocol (e.g. HTTP using WSDL) but not to specify the message content. For this purpose the description level depicted in figure 1 could be used. The client could describe the desired functionalities in several ways and the server would also describe its web services much more specific in terms of semantics of the service provided. The descriptions of the services are functional and behavioral, concerning interfaces and non-functional requirements. Standards like WSML (Web Service Modeling Language) and WSMO (Web Service Modeling Ontology) can be used to solve this annotation and metadata problem.

In our case, these descriptions are used to facilitate the communication between a server deploying the CIM and different clients. The clients describe the desired functionality and the server can offer the semantically correct web service.

V. Combined Architecture: a new approach

In our approach we take the state-of-the-art concerning the GID (Generic Interface Definition) API of the CIM - as mentioned in chapter three - and extend it by using the OPC UA. In the next step we bring semantic to the OPC UA's way of communication. The resulting architecture is shown in figure 2. By combining the platform-independent architecture OPC UA and the platform-independent information (PIM) model CIM, we choose an appropriate way to face the upcoming requirements in the energy sector. As far as new services are needed, new processes come up combining old and new systems. Legal unbundling and distributed generation lead to new requirements to the IT landscape of a utility which can be mostly covered by Service-Oriented architectures using web services. This is our starting point to incorporate the CIM and the OPC UA into the utilities' infrastructure. This approach provides, an appropriate solution on a technical basis which is pretty much standardized. However, it still lacks some essential features as UA and CIM both cover syntax and semantics but the semantics are usually omitted when it comes to the point of finding and discovering appropriate services form different systems or parties. Therefore, we suggest combining the OPC

[5]http://www.w3.org/

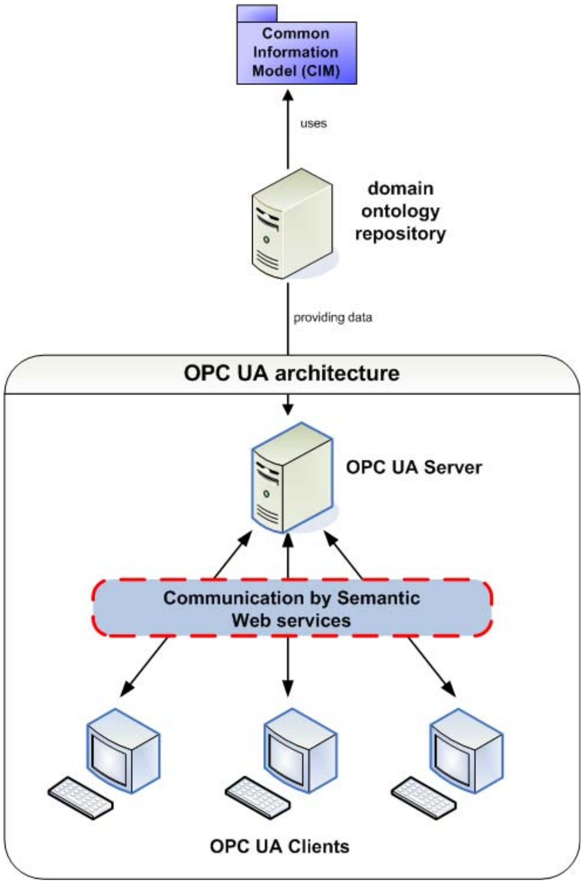


Fig. 2. Architecture of the approach.

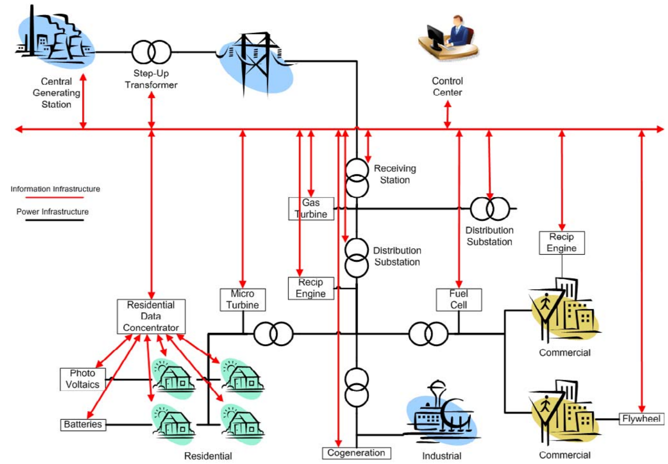


Fig. 3. Information and power infrastructure data and energy flow in the electric utility domain (adapted from [17]).

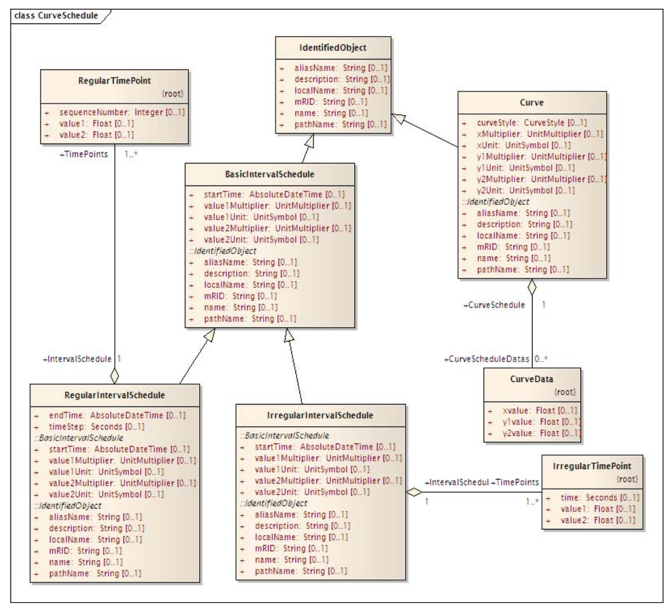


Fig. 4. Excerpt from the CIM UML data model in the Enterprise Architect.

UA and CIM with Semantic Web services. The OPC [18] server then provides services which are purely chosen based on their attributes like costs, QoS and availability. A typical service we intent to cover would be one to choose between different weather forecasts in order to do a better load forecast for distributed generation based on wind turbines. Different companies have different services which are better or worse at different times of the year, week or day. Based on the current situation, annotated services would provide the possibility to choose the best service based on the current context.

Our combined solution would cover both the important issues of being compliant to standards as well as finding the right services in a service-oriented architecture.

VI. Implementation

In terms of implementation, the architecture can be divided into two parts. On the one side the abstract OPC UA data model has to be used to implement a domain specific and CIM-based information model. On the other side, an ontology has to be designed to provide a basis for the semantic web services. This ontology is also related to the CIM semantics and the CIM domain ontology. Hence, we use the CIM as a provider of a common semantic for two different layers of communication within the electric grid to make it a smarter grid.

Figure 3 shows the two layers that are in the focus of our architecture. On the one hand, there is the power infrastructure and, on the other hand, the information infrastructure. In our approach, the communication for the power infrastructure is realized using the OPC UA and the communication for the information infrastructure is realized using the semantic web services. In this case, the CIM serves as basis to realize an interface for those two communication layer.

As mentioned before the CIM is maintained as a UML model. Figure 4[6] shows an example including some CIM classes, attributes and associations. In the following, this example will be used to show how it could become the basis for the OPC UA address space and the semantic web service ontology. The classes shown are used to model CIM compliant graphs which could be used to describe different forecasts.

[6]http://www.sparxsystems.com.au/products/ea/index.html

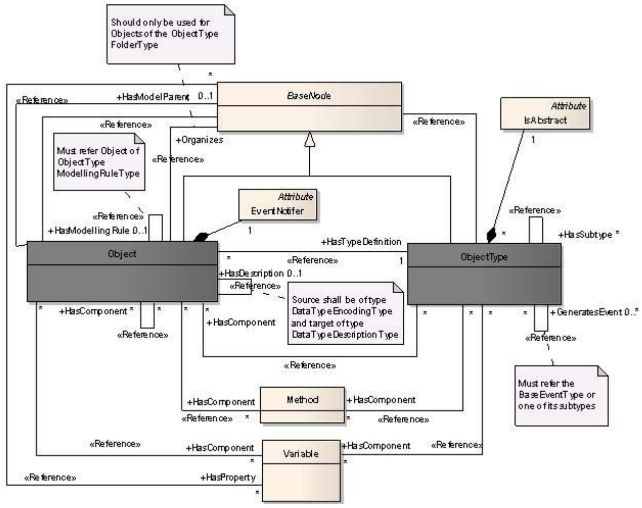


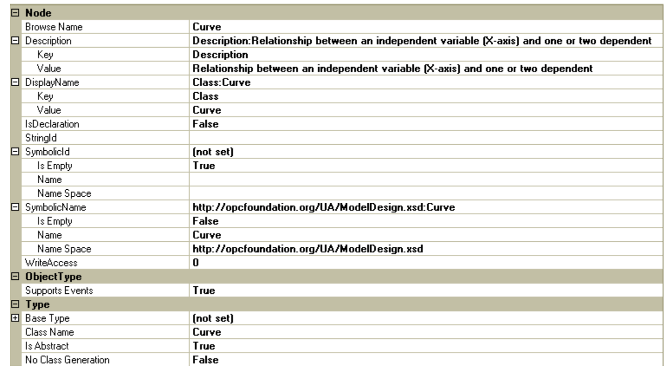
Fig. 5. UML meta model for *Object* and *ObjectType* [16].

A. OPC UA meets CIM

In the first step, we have to combine the CIM with the OPC UA. Therefore, it is necessary to model the CIM in an OPC UA server address space. The OPC UA standardizes the way how data can be exchanged, but not what kind of data is exchanged. Thus, we need an information model to be set on top of the abstract OPC UA model. In this approach the CIM (being a domain specific information model) is the first choice. We have to translate the CIM, which is maintained in UML into the OPC UA address space [19], [20]. A UML meta model for the UA data model can be found in the appendix of Part 3 of the OPC UA standard. The following OPC UA objects are taken into account for implementation and mapping:

- *BaseNode*
- *Reference and ReferenceType*
- *Predefined ReferenceTypes*
- *Attributes*
- *Object and ObjectType*
- *Variable and VariableType*
- *Method*
- *EventNotifier*
- *DataType*
- *View*

Figure 5 shows an example referring to the *Object* and *ObjectType* from the UA data model. For our implementation, we model the abstract CIM UML classes as abstract OPC *ObjectTypes*. The OPC UA *Objects* represent the concrete instances of the abstract CIM classes. Concerning other different modeling decisions, basic design decisions have to be made. For example, in specific cases it has to be decided to model CIM attribute either as *Properties* or *Data Variables*. Furthermore, the CIM associations have to be modeled as *References*, but because of the cardinalities, special OPC UA *ModellingRules* have to be created. Up to this point the modeling is server independent. The next modeling steps for the server's architecture are specific. Especially the design of the *Views* and the *Methods* is server specific and depends on the individual needs. A server can use the *Views* to give different



⊟ **Node**	
Browse Name	**Curve**
⊟ Description	**Description:Relationship between an independent variable (X-axis) and one or two dependent**
Key	**Description**
Value	**Relationship between an independent variable (X-axis) and one or two dependent**
⊟ DisplayName	**Class:Curve**
Key	**Class**
Value	**Curve**
IsDeclaration	**False**
StringId	
⊟ SymbolicId	**(not set)**
Is Empty	**True**
Name	
Name Space	
⊟ SymbolicName	**http://opcfoundation.org/UA/ModelDesign.xsd:Curve**
Is Empty	**False**
Name	**Curve**
Name Space	**http://opcfoundation.org/UA/ModelDesign.xsd**
WriteAccess	**0**
⊟ **ObjectType**	
Supports Events	**True**
⊟ **Type**	
⊞ Base Type	**(not set)**
Class Name	**Curve**
Is Abstract	**True**
No Class Generation	**False**

Fig. 6. The CIM class *Curve* modeled in the OPC UA address space.

clients or groups of clients access to parts of the model relevant to them. The *Methods* are also very important in this context, because a server can provide a lot of different functionalities implementing them to provide Smart Grid functions.

The CIM is a very large data model and it is difficult and often not necessary to use the whole model for all purposes. To make the use of the CIM more applicable, one commonly uses profiles which include only essential classes and associations of the CIM. On the one hand single companies use intra-corporate profiles and on the other hand large profiles exist which are partly standardized and widespread within the utility domain:

- **CPSM**: The Common Power System Model (CPSM) is used in the USA for the exchange of transmission system models [12].
- **CDPSM**: The Common Distribution Power System Model (CDPSM) is used in Europe for the exchange of distribution power system models [14].
- **UCTE**: The Union for the Co-ordination of Transmission of Electricity[7] (UCTE) profile is used in Europe for the exchange of transmission system models.
- **ERCOT**: The Electric Reliability Council of Texas[8] (ERCOT) profile is an intra-corporate data model.

The above mentioned concept of *Views* from the OPC UA adress space provides an opportunity to realize the profile concept of the CIM.

The CIM class *Curve*, which is shown in figure 4, is modeled in the OPC UA address space in figure 6[9] as an example. Therefore, the abstract mapping shown in figure 5 was the basis of the UML class, using the *CAS OPC UA Address Space Model Designer*. This tool is recommended by the OPC Foundation but is still in development. Some specific problems could not be solved, like needed interfaces, import and export functions. Thus, part of our future work is to develop a new tool solving our specific problems. We want to provide the resulting toolbox as open source programs.

B. Semantic Web Services meet CIM

The second step contains the evaluation of different semantic web service technologies as well as the selection of one to

[7] http://www.ucte.org/
[8] http://www.ercot.com/
[9] http://www.commsvr.com/UAModelDesigner/Index.aspx

```
instance Curve memberOf CIM_Class
     nonFunctionalProperties
          description hasValue "Relationship between an independent variable (X-axis) and one or two dependent
          variables (Y1-axis and Y2-axis). Curves can also serve as schedules."
     endNonFunctionalProperties
     hasParent hasValue IdentifiedObject
     hasAttribute hasValue {_"http://iec.ch/TC57/2005/CIM-schema-cim10#Curve.curveStyle",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#Curve.xMultiplier",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#Curve.xUnit",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#Curve.y1Multiplier",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#Curve.y1Unit",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#Curve.y2Multiplier",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#Curve.y2Unit",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.aliasName",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.description",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.localName",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.mRID",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.name",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.pathName" }

instance CurveData memberOf CIM_Class
     nonFunctionalProperties
          description hasValue "Data point values for defining a curve or schedule"
     endNonFunctionalProperties
     hasAttribute hasValue {_"http://iec.ch/TC57/2005/CIM-schema-cim10#CurveData.xvalue",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#CurveData.y1value",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#CurveData.y2value" }

instance BasicIntervalSchedule memberOf CIM_Class
     nonFunctionalProperties
          description hasValue "The schedule has TimePoints where the time between them is constant."
     endNonFunctionalProperties
     hasChild hasValue {RegularIntervalSchedule, IrregularIntervalSchedule }
     hasParent hasValue IdentifiedObject
     hasAttribute hasValue {_"http://iec.ch/TC57/2005/CIM-schema-cim10#BasicIntervalSchedule.startTime",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#BasicIntervalSchedule.value1Multiplier",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#BasicIntervalSchedule.value1Unit",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#BasicIntervalSchedule.value2Multiplier",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#BasicIntervalSchedule.value2Unit",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.aliasName",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.description",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.localName",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.mRID",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.name",
     _"http://iec.ch/TC57/2005/CIM-schema-cim10#IdentifiedObject.pathName" }
```

Fig. 7. A WSML mapping for CIM classes.

build a basis. In general, semantic web service technologies can be divided into two approaches [1]. The first one is a top-down approach. An ontology language is used to describe the desired services in a technology independent manner. Based on the results, a proper description language has to be chosen afterwards. Technologies like Web Service Modeling Ontology Language (WSMO) [4] and Web Ontology Language for Services (OWL-S) can be assigned to the top-down approach. The second approach uses a bottom-up modeling process. During that process an existing technology is chosen and extended by semantic information. Examples are Web Service Semantics (WSDL-S), Semantic Annotations for WSDL (SAWSDL) and the DARPA Agent Markup Language / Ontology Inference Layer (DAML+OIL). In our case, a top-down approach is the choice, because we aim for a very detailed service model. The COLIN (CIM Ontology aLigNment Methodology) methodology suggests the use of SAWSDL to annotate semantic metadata [24]. However, we try to improve that approach to have more detailed descriptions of the services by using a top-down approach instead of a bottom-up approach.

Figures 7 and 8[10] show the previous example modeled with the WSMT (Web Service Modeling Toolkit). In figure 7 a mapping between CIM information and the WSML (Web Service Modeling Language) concept is shown. We created a new WSML-Concept within the CIM namespace named *CIM_Class*. Another WSML-Concept, *CIM_Attribute*, was modeled to realize the CIM attributes. By creating the WSML-Attribute *hasAttribute* it is possible to allot instances of *CIM_Attribute* to instances of *CIM_Class*. It is possible to model inheritance by using the WSML-Attributes *hasParent* and *hasChild* as well. Furthermore, some non-functional properties are added already. In this case the descriptions of the CIM classes are added. They can be found in the UML model.

[10]http://sourceforge.net/projects/wsmt/

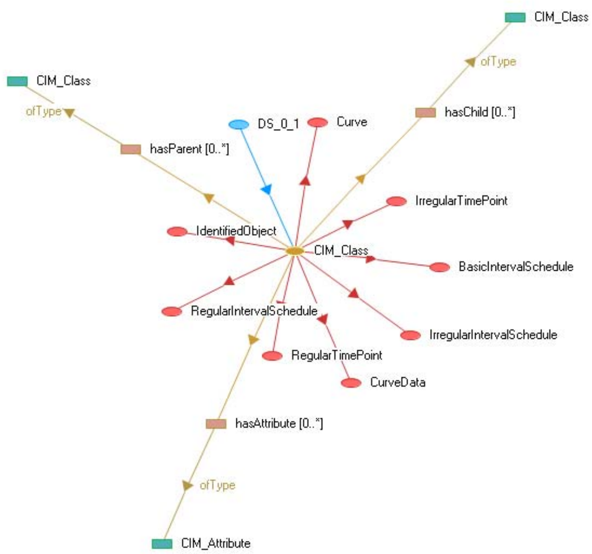


Fig. 8. Example for a WSMO realization.

After modeling part of the CIM information we extended our CIM Mapping Bench tool in order to generate WSML. Thus, we can parse the CIM XMI (XML Metadata Interchange) serialization, which is created by an export function of an UML tool, and automatically generate a WSML ontology.

During the next step, services have to be annotated with metadata. Therefore, we need several levels of machine-readable descriptions based on the developed ontology. For each service, we need a functional and a non-functional description as well as descriptions concerning the behavior and the interfaces of the service. Hence, the work focuses on a methodology which describes the process of creating the different descriptions by annotating the metadata. Especially in terms of choreography and orchestration, it is important to have very detailed and accurate descriptions with post- and pre-conditions.

In the last step a matching process has to be developed, which provides the automated discovery, combining and invocation of the services.

VII. Conclusion and Future Work

In this paper, we argued that the use of the large electric domain ontology and standard, the IEC 61970, or the Common Information model CIM with the new OPC UA (being a platform for implementation) would cover most of the important issues to Smart Grid achitectures like SOA (Service Oriented Architecture) at utilities when being enlarged with annotated semantic web services. We gave a short introduction to the CIM and its domain data model and ontology, covered the different existing OPC interfaces and gave an introduction to the next level of implementation of these services using the new OPC Unified Architecture. We argue that these new architectures for the electric utility domain may be appropriate for current automation needs while there are still new functions needed for the Smart Grid which cannot yet be met. Our feasible extension to meet these requirements to know which service actually provides which function under which conditions is the extension of the OPC UA with semantic web services. Our combined and standardized architecture will cover an implementation of an OPC server extending the standard web services implementation. This will lead to a better semantic discovery oriented system for electric utilities and Smart Grid functions and services from different parties.

The next step we are going to take is to combine the OPC servers with the whole CIM data model being the exchange data model for all Web service payloads. We then provide an extension using the WSMO ontology based on the function blocks of the IEC 61968 - IRM (Interface Reference Model for distribution) family which provides basic function blocks already identified for meter reading, maintenance, outage management, asset management, secondary IT and SCADA (Supervisory Control and Data Acquisition). Those function blocks will be the starting point for the final step of integrating more semantics into the services provided by the SOA [21] in an electric utility and thus building a Smart Grid.

References

[1] R. Ashri, G. Denker, D. Marvin, M. Surridge and T. Payne, *Semantic Web Service Interaction Protocols: An Ontological Approach*, in *The Semantic Web ISWC 2004*, Springer, Berlin, 2004.
[2] J. de Bruijn, D. Fensel, M. Kerrigan, U. Keller, H. Lausen and J. Scicluna, *Modeling Semantic Web Services*, Springer Verlag, Berlin, 2008.
[3] D. Von Dollen, *Report to NIST on the Smart Grid Interoperability Standards Roadmap*, Prepared by the Electric Power Research Institute (EPRI), 2009.
[4] D. Fensel, H. Lausen, A. Polleres, M. Stollberg, D. Roman, J. de Bruijn and J. Domingue, *Enabling Semantic Web Services: The Web Service Modeling Ontology*, Springer Verlag, Berlin, 2006.
[5] D. Fensel, M. Kerrigan and M. Zaremba, *Implementing Semantic Web Services: The SESA Framework*, Springer Verlag, Berlin, 2008.
[6] E. Guenther, *IEC Standardization "Smart Grid"*, Survey prepared for IEC SMB SG 3 "Smart Grid", 2009.
[7] IEC, *IEC 61970-301 Ed. 1: Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base*, 2007.
[8] IEC, *IEC 61970-402 Ed. 1: Energy management system application program interface (EMS-API) - Part 402: Common services*, 2008.
[9] IEC, *IEC 61970-404 Ed. 1: Energy management system application program interface (EMS-API) - Part 404: High Speed Data Access (HSDA)*, 2007.
[10] IEC, *IEC 61970-405 Ed. 1: Energy management system application program interface (EMS-API) - Part 405: Generic eventing and subscription (GES)*, 2007.
[11] IEC, *IEC 61970-407 Ed. 1: Energy management system application program interface (EMS-API) - Part 407: Time series data access (TSDA)*, 2007.
[12] IEC, *IEC 61970-452: Energy management system application program interface (EMS-API) - Part 452: CIM Network Applications Model Exchange Specification*, 2006.
[13] IEC, *IEC 61968-11 Ed. 1: System Interfaces for Distribution Management - Part 11: Distribution Information Exchange Model*, 2008.
[14] IEC, *IEC 61968-13 Ed. 1: System Interfaces for Distribution Management - Part 13: CIM RDF Model Exchange Format for Distribution*, 2008.
[15] IEC, *IEC 62541-1 Ed. 1.0: OPC Unified Architecture Specification - Part 1: Overview and Concepts*, 2008.
[16] IEC, *IEC 62541-3 Ed. 1.0: OPC Unified Architecture Specification - Part 3: Address Space Model*, 2007.
[17] IEC, *Draft IEC/TR 62357 second edition: TC 57 Architecture - Part 1: Reference Architecture for TC 57*, 2009.
[18] F. Iwanitz and J. Lange, *OPC - Grundlagen, Implementierung und Anwendung*, Huethig Verlag, Heidelberg, 2005.
[19] J. Lange, *Umstieg auf OPC Unified Architecture: Sanfte Migration oder harter Uebergang?*, in: *SPS Magazin*, 20, 2007.
[20] W. Mahnke, S.-H. Leitner and M. Damm, *OPC Unified Architecture*, Springer Verlag, Berlin, 2009.

[21] T. Schmedes, *Entwurf service-orientierter Architekturen fuer das dezentrale Energiemanagement*, in: D. Spath, A. Weisbecker, O. Hoe and J. Drawehn Hrsg. *Science Meets Business - Stuttgarter Softwaretechnik Forum 2007*, Fraunhofer IRB Verlag, 2007.

[22] M. Uslar, *Das CIM als Integrationsstandard fuer Versorger*, Proceedings of the 3rd GI-Workshop on Enterprise Application Integration.

[23] M. Uslar and F. Gruening, *Zur semantischen Interoperabilitaet in der Energiebranche: CIM IEC 61970*, Wirtschaftsinformatik, 2007, 49, S. 295-303.

[24] M. Uslar, S. Rohjans, S. Schulte and R. Steinmetz, *Building the Semantic Utility with Standards and Semantic Web Services*, in: *On the Move to Meaningful Internet Systems: OTM 2008 Workshops*, Springer, Berlin, 2008.

[25] M. Uslar, S. Rohjans, T.Schmedes, J. Gonzalez, P. Beenken, T. Weidelt, M. Specht, C. Mayer, A. Niesse, J. Kamenik, C. Busemann, K. Schwarz and F. Hein, *Untersuchung des Normungsumfeldes zum BMWi-Foerderschwerpunkt "E-Energy - IKT-basiertes Energiesystem der Zukunft"*, Survey prepared for the Bundesministerium fuer Wirtschaft und Technologie (BMWi), 2009.

Sebastian Rohjans has joined the OFFIS - Institute for Information Systems in late 2008. He holds a computer science degree from the University of Oldenburg with a minor in business and wrote his thesis about ontology-based integration. He is now working in industrial setting projects having the same scope. His research interests include semantic web services, the OPC unified architecture and ontology based mediation.

Mathias Uslar Since 2004, after having finished his studies of informatics with a minor in legal informatics and business, Mathias Uslar joined the OFFIS - Institute in Oldenburg Germany where he is working in the Utility Informatics branch. His main interests are standardization and utility EAI. He also brings in his expertise to both national and international standardization boards. He is also the director of the Center for IT Standards in the Energy Domain, abbreviated CISE (www.ccise.de).

H.-Juergen Appelrath is Professor of practical computer science at the University of Oldenburg. He studied and received his doctorate at the University of Dortmund and was assistant professor at ETH Zurich. Prof. Dr. Dr. h.c. Hans-Juergen Appelrath is on the board of OFFIS, on the advisory board of various research institutes and member of the Supervisory Board of several companies. In 2007 he was awarded the honorary doctorate from the TU Braunschweig.