Using CIM for Smart Grid ICT integration

Matthias Rohr, Andre Osterloh, Michael Gründler,
Till Luhmann, Michael Stadler, Nils Vogel
BTC Business Technology Consulting AG
Oldenburg, Germany
Email: Michael.Stadler (at) BTC-AG.com

Abstract: The eTelligence project explores and demonstrates various smart energy grid ideas by using modern ICT (information and communications technology). For this purpose, many new and heterogeneous types of smart grid systems have to be developed and integrated, such as a regional energy market, distributed energy management systems, and an advanced metering infrastructure. The future interaction scenarios of such new systems are still topics of research, which calls for an architecture easily supporting future changes. The integration capabilities of the eTelligence ICT architecture are based on standardized communication, especially using IEC 61970/61968 (Common Information Model, CIM) and an easily extensible market product description language also realized with CIM. Additionally, we present a process model for using CIM, and report our experiences from using CIM for integration.

Introduction

In an extensive field test the eTelligence project[4] [BBE09] explores and demonstrates various approaches of using modern ICT and advanced operation to improve the current energy supply system and to enable broad integration of renewable energy sources like wind, photovoltaic and biomass into the future power supply.

The IT infrastructure is responsible for collecting measurements from different sources (e.g., smart meter readings) from the field, storing these measurements, partly processing them in multiple systems and offering them to a number of other systems for further processing and for interaction/visualization purposes. It acts as a nervous system and has to be flexible towards changes. For example, the energy market place as a future utility application needs an extensible way of defining products to be traded and exchanging bids containing these products between the market participants and the market. Both purposes call for the use of a suitable standardized modeling and messaging solution. Within the project we chose the IEC CIM (Common Information Model) standard as a sophisticated approach.

[4] The project is co-funded by the German federal ministry of economics and technology (support code 01ME08010) and is part of the German E-Energy program.

This article gives information about the IT infrastructure and its design rationale, and the energy market place with a focus on the products traded. For both contexts we give a deep insight on CIM usage. We also describe both, our experience using CIM and a best practice process for supporting CIM usage. We start with a short introduction into CIM.

A short CIM Summary

In this section we briefly introduce the Common Information Model (CIM). For a deeper introduction into CIM we refer the reader to [Mc07][Us09].

In the field of standardization of system interfaces, data models for energy network management, and the integration of applications into the IT environment of an energy supply company, the IEC has adopted the Common Information Model from the Electric Power Research Institute (EPRI) as a base for the standards IEC 61968 [IEC07] and IEC 61970 [IEC09a]. This process started in 1996, and has since then been continued at international level. In the context of international standardization for electric utilities, CIM becomes increasingly relevant [EPR09, Us09]. This is, among others, reflected in the German E-Energy / Smart Grid standardization roadmap [DKE10].

The Common Information Model aims to reduce the time effort and expense associated with an integration of applications in an energy management system and to provide investment protection through the standardization in systems and the effective ensure operation of these systems [Us09].

CIM is an abstract UML model that defines a common vocabulary and a formal representation of knowledge as a set of concepts for the electric power industry. The abstract CIM model represents all the major objects of an electric utility enterprise typically needed to model the operational aspects. This model includes classes and attributes for these objects, as well as the relationships between them [IEC09a].

CIM can also be interpreted as an integration framework [Us09]. It helps to define the integration of a vertical value chain by using interfaces and data models for energy management systems. Three main use cases for CIM exist [Us09]:

- Exchange of network topologies
- Coupling of utility applications
- XML-based message exchange within a SOA (Service-oriented Architecture) using CIM-semantics

The interface reference model in IEC 61968-1 describes the use of a middleware and the information exchange model [IEC10]. By using the common semantics defined in CIM and by use of a suitable middleware, it is possible to reduce the costs for integrating power system operators’ software applications.

The eTelligence Approach of Integration

Our approach to integrating various software applications in eTelligence consists of a software architectural approach that has an emphasis on standardized communication and a corresponding structured process model for applying these standards.

Architectural Approach

The eTelligence architecture was designed in consideration of three requirements:

- **Interoperability:** As the future smart grid scenarios and systems to integrate are not clear yet, sufficient flexibility for adding new applications and interactions has to be provided.
- **Standardized communication:** Standards can potentially reduce integration costs and are therefore a core concern for smart grid infrastructures. In eTelligence it was a major goal to explore the broad application of IEC CIM and IEC 61850, which is used for field integration. Standardized communication also supports interoperability.
- **Low latency processing:** The value of information decreases with time. For instance, both the competition at energy markets and smart grid control mechanisms can benefit from low latency in end-to-end processing of sensor data information.

In the following, the major architectural means in eTelligence for satisfying these requirements are described.

Interoperability through two Integration Buses

For maximizing interoperability, the architectural landscape is structured into three layers interconnected by two ESB (Enterprise Service Bus) layers, as illustrated in Figure 8. The bus concept ties together event-driven services using a service-oriented architecture based on open standards and messaging [Ch04]. This allows for relatively independent development cycles of individual applications.

The field communication bus collects data from various sensors throughout the smart grid and disseminates control signals to those components. Therefore, it has to be reliable in basic messaging and sufficiently scalable to support communication to a large number of devices such as sensors. In detail, it has been implemented by using both, publish-subscribe messaging middleware and request-response field communication middleware. The publish-subscribe middleware is used for 1-N distribution of sensor data. For instance smart meter readings are published to multiple applications in parallel. Request-response middleware is used for more complex interaction, e.g., for steering and control of distributed energy resources. Already at the project's beginning, it was obvious, that the field communication bus was going to be subject to many changes, since many heterogeneous sensors and even more sensor data consumers were planned to be connected. We have chosen publish-subscribe as the dominant communication pattern to avoid bottlenecks for integration tasks at the field communication bus. This allowed developers of applications that need to consume the data to easily integrate their applications, even without downtime. The integration of these applications was even more simplified by implementing sensor data streams using standardized messages (e.g., CIM-based XML messages for smart meter readings).

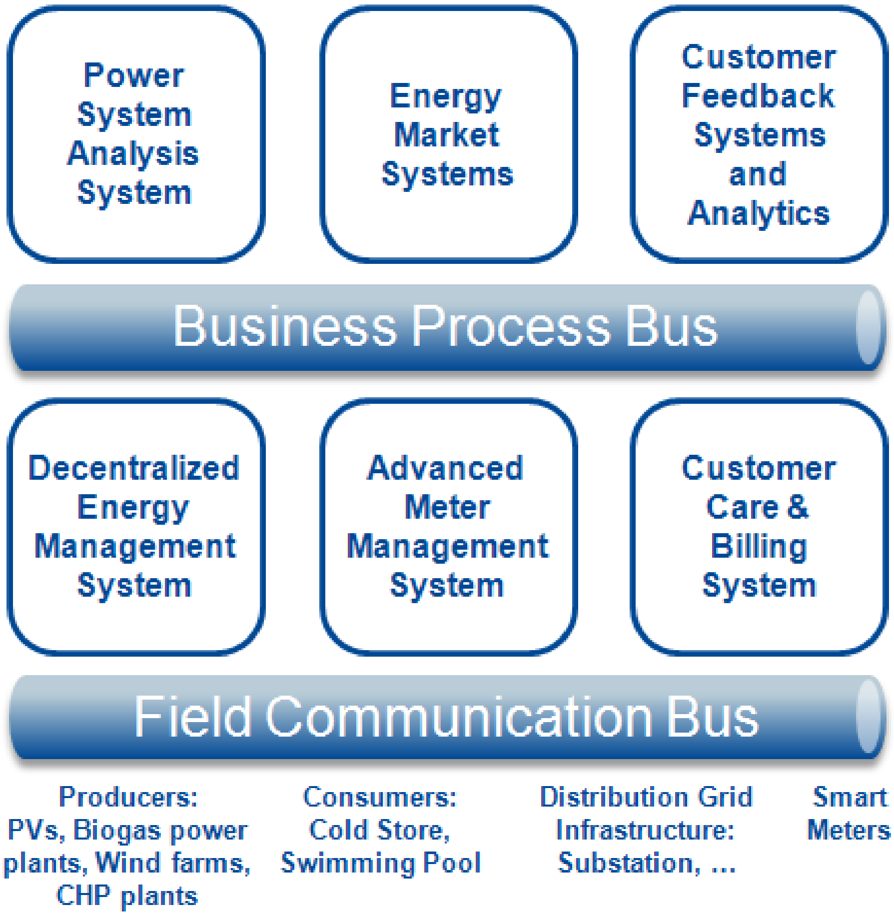

Figure 8: eTelligence architecture from a high-level conceptual viewpoint.

The second bus, denoted as business process bus implements business process execution besides plain request-response and publish-subscribe communication (e.g., user registration processes). Therefore, the bus was realized with two middleware products — one focused on plain high-performance publish-subscribe, the other focused on reliable process execution. Both request-response and publish-subscribe interaction between the business process bus and systems interacting with it are realized with IEC 61970/61968 (CIM).

Standardized Communication for Interoperability

Two standards have primarily been used in the eTelligence architecture: IEC 61970/61968 (CIM) and IEC 61850. CIM was mainly used on business level, but also in some cases for exchanging data with technical grid infrastructure (e.g., medium-sized power producers and consumers) over the field communication bus. IEC 61850 was used for communication on field infrastructure level, especially for communication with sensors in substations, medium-sized combined heat and power plants, photovoltaic plants and wind power plants.

Minimizing Latency

As mentioned above, latency requirements are an important aspect of energy management system landscapes. Especially connections between applications can have a significant impact on latencies.

In eTelligence, we selected three principles to minimize latencies: publish-subscribe communication, instant individual data item processing instead of batch processing, and parallel processing. Above, we described that we used publish-subscribe communication to simplify integration of new applications into the system. At many places in the architecture, publish-subscribe-based communication also contributes to low latency processing, compared to request-response communication. Request-response communication typically needs two messages (the request and the response), while an initialized publish-subscribe based communication requires only one message. In order to further reduce latency, data items (e.g., sensor data readings) are processed individually. In other words, no batch processing is used and each processing result is directly published to subsequent subscribers. Finally, data processing was organized such that a maximum amount of activities is executed in parallel. For instance, the field communication bus provides smart meter readings in parallel to different subscribing applications, such as data storage, billing, and infrastructure operations supervision.

Obviously, both parallel processing and instant individual processing have their disadvantages. Parallel processing can lead to synchronization issues, if the data is joined in subsequent steps. Instant individual processing (i.e., non-batch processing) is sometimes more complex (e.g., if data arrives out of order) and is often less efficient in terms of the ration between message payload size and header size.

The individual, instantaneous processing of sensor data and the broad usage of parallel processing via publish-subscribe is in contrast to a single central database paradigm. Modern SOA concepts (e.g., SAP's SOA platform called Enterprise SOA), address distributed databases, enabling that each application can be installed, configured, and used independently of other applications [He07]. At the same time, a company-wide integration strategy based on a central shared database can be problematic in terms of performance, reliability, and maintainability [SS03], especially for broad smart grid application landscapes. Therefore, our implementation explores both benefits and disadvantages of avoiding a centralized data repository in a smart grid scenario.

Methodology for efficient Integration using CIM

Since several years, we use a process model oriented methodology to realize IEC CIM-based communication interfaces together with IEC CIM-based message schemata and to coordinate the usage of different CIM schema versions within the enterprise.

Two roles are distinguished in the process model: first, a central CIM team, and second, software development projects that want to use the IEC CIM standard. The CIM team is a small company-wide expert group coordinating the evolution of shared CIM schema versions and supporting projects using CIM. The CIM team operates a central *CIM repository*, which is a file server providing several CIM model versions, CIM schema versions, extension models, CIM tools like the BTC CIMBench and used CIM message schemata in combination with versioning. Thereby, projects that share CIM models have a common place to refer to.

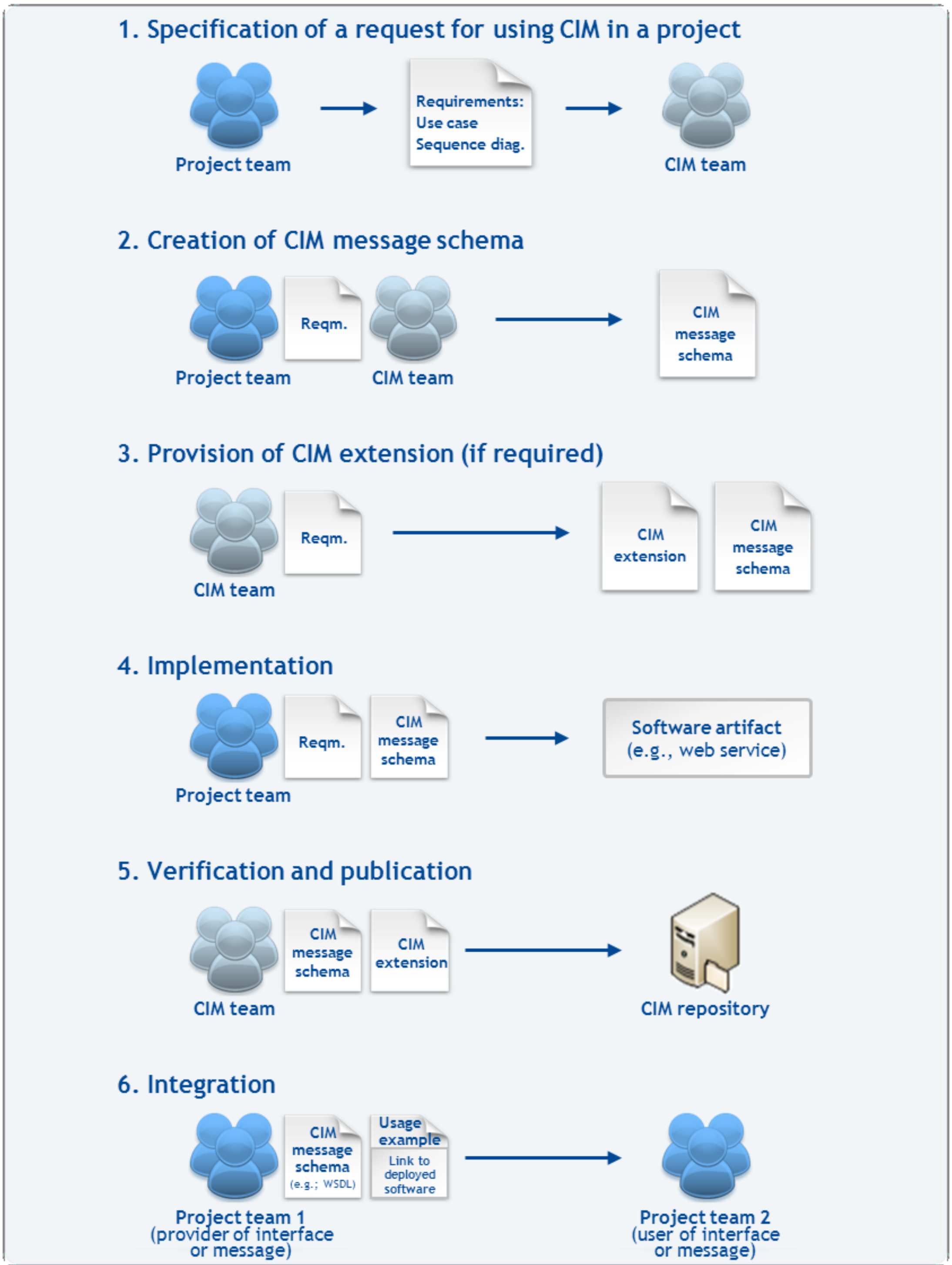

Figure 9: CIM schema process.

The process model, illustrated in Figure 9, requires a project to perform several steps in order to use CIM for messages and interfaces. In the following, the process is briefly sketched:

1. First, the project specifies the use case in which the CIM standard is to be used. This is done with UML Sequence Diagrams and a structured description that describes the use case.

2. The CIM team together with the project staff models CIM message schemata based on the project's use case specification. This provides several benefits for the project: The project members can benefit from CIM team's expert knowledge on the CIM standard, and the interfaces and message schemata will be similar to other CIM-based artifacts in other projects, since the same experts were involved. Additionally, this project step enforces an UML-based use case specification and a review (by the CIM team).
3. If needed, the CIM team provides a CIM model extension that satisfies the project requirements. This step is combined with the second step.
4. The project implements CIM standard usage and provides the resulting technical artifacts CIM message schemata and interface specification files (e.g., WSDL definitions) to the CIM team. Additionally, example messages for CIM message schemata have to be provided.
5. The CIM team checks the artifacts and deploys them to the CIM repository.
6. In case the development team provides an interface to other development teams, then both the CIM-based message schema for forward and response messages (in case of request-response) are provided together with example message instances that can directly be used at the deployed interfaces for testing.

The process for changing or extending existing artifacts and interfaces within the CIM repository is based on this process, but does not usually require to provide new use case specifications. Each change to the artifacts in the CIM repository is assigned a new schema version number and old schema versions are kept.

Furthermore, the CIM team is also a central coordinating instance for the adaptation to new CIM *model* versions. New *versions* of the CIM standard are provided by the IEC TC 57 working group about once a year. Frequent changes in the standard are a threat to interoperability and a central company-wide coordination allows reducing and controlling version conflicts. However, this does not mean that the CIM team is in charge to update the individual CIM message schemata whenever a new standard version is released.

In the eTelligence project and other projects, we experienced that the process model reduces integration costs through providing a simple step-by-step process and through specialization of the CIM expert group. We also observed that the process resulted in better interfaces and messages over the time. Obviously, the process only covers a smaller part of an integration project.

CIM in eTelligence

The previous section discussed the usage of standards and architectural principles in general in the eTelligence architecture. In the following, two more concrete examples for using CIM in the eTelligence system in the context of smart metering and market product description are presented.

Using CIM in Smart Metering

eTelligence implemented a complete electricity smart metering solution and explores new smart metering scenarios that are not yet in the market. Figure 10 sketches how smart meter data are distributed through different systems in eTelligence's ICT from a logical viewpoint. For simplicity, the two ESB (Enterprise Service Bus, e.g., [Ch04]) layers are not shown, but they organize a large part of the communication — with one major exception: interactive systems that communicate with data management systems via request-response have direct inter-system access to provide short response times. However, even in this cases, CIM-based messages are used.

Smart meters send pairs of timestamps and meter readings. In the processing chain, these messages are converted as soon as possible from a communication standard specific to power meters (SML standard) to messages conforming to the IEC CIM standard, more precisely to IEC 61868-9 meter reading and control [IEC09b]. In Figure 10 this step is performed in the field communication system.

After this, the messages are distributed in a publish-subscribe manner to four different target systems using the field communication bus. The target systems are the meter data management system (MDMS), the billing system, a live aggregation system that provides data to a public Web portal (average consumption of households), and finally to the smart meter live monitoring, performing data stream analysis in order to detect technical problems in the smart meter infrastructure in near real time. As mentioned in the previous section, data is provided in parallel to these four systems by the field communication bus enabling thereby parallel data processing. In one case, the parallel processing leads to a synchronization issue: the results from billing, i.e., determination of costs and prices for each consumption, have to be joined to data.

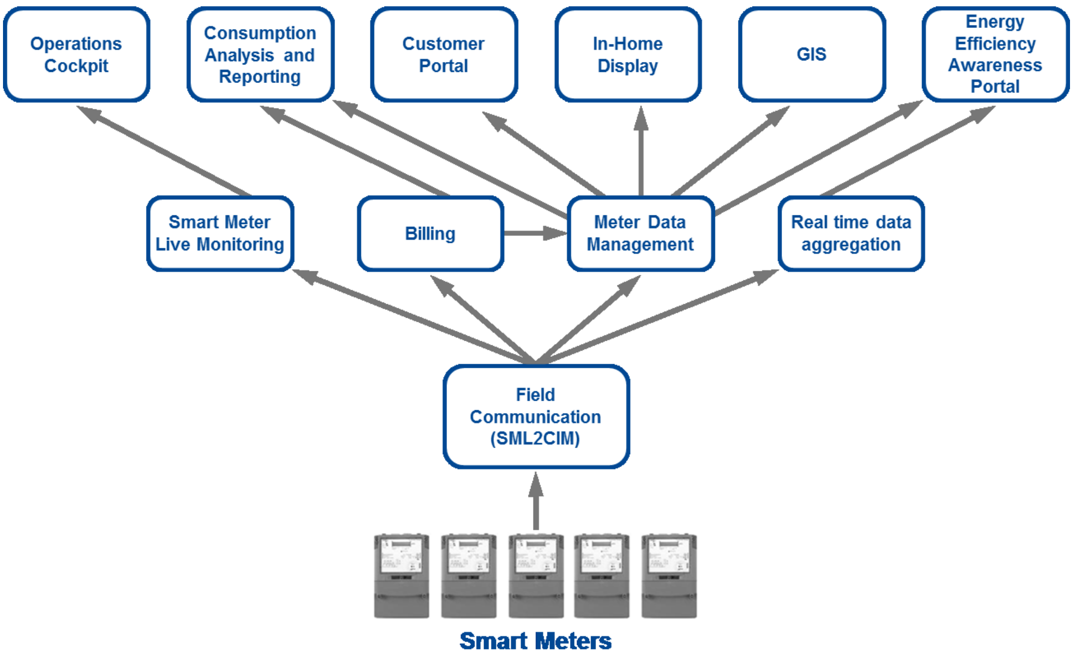

Figure 10: Logical data flow for smart metering.

In order to be able to receive data, subsystems interested in smart meter measurements need to conform to our CIM-based data model for smart meter readings and to register at subscriber at the field communication bus. This data model is stored in a company-wide repository for IEC CIM artifacts. This enabled us to develop all systems that use smart meter data independently.

Using CIM within the eTelligence Market Place

The previous section described the usage of CIM for standardizing messages of smart meter data. In the following, it is presented how CIM was used for a market product description language.

Within the eTelligence project, a regional energy market with EEX (European Energy Exchange) connection was developed, implemented and put into operation. All processes are handled automatically. Market participants interact with the market by means of software market agents who place their bids on the market place using a web service.

The following stakeholders' requirements towards products could be identified. Both, products based on active power and reactive power should be traded on the market place. Locality of generation and consumption had to be taken into account and also products should be differentiated by type of generation (e.g., from regenerative sources or from fossil fuel). Due to the project's experimental nature, it was important to keep flexibility with respect to product requirements arising later in the project.

Based on the requirements, we developed a product description language allowing definition of complex energy products for both *active power during a continuous period of time* and *reactive power during a continuous period of time*. This language uses the following additional parameters to characterize a basic product:

- is the product designated for sale by the owning party or should it be acquired by the owing party
- the minimum and the maximum amount of power traded
- the timespan of delivery
- the start and the end time of the trading period
- the product's price
- the location of delivery
- is the product an option or not

It is possible to add additional information to a product's description. The description of the basic active power product can contain information about the type of power generation, e.g., generation from renewables. Descriptions of basic reactive power products can be refined by information whether it is capacitive or inductive.

Complex products can be formed by linking basic products. They can be described as trees with basic products as leafs and link operators as inner nodes. Those operators describe how a product should be traded.

The following operators exist:

- all
- min(n), min_ordered(n)
- max(n)
- exactly(n)

For instance, min(n) applied on a set of m≥n products means that at least n of the m products given in the set have to be traded successfully on the market in order for the overall complex product to be traded. Otherwise no trade of this product or its parts shall be possible.

A detailed description of the operators and their semantics can be found in [SO10]. Using the product description language, it is possible to define products as schedules or load shifting potentials.

Example of a complex product

Figure 11 depicts a complex product as a tree. Every single node within the tree represents a product. While leaves represent basic products, inner nodes represent complex products. An inner node's label consists of a number within parentheses followed by a text and an optional additional number. The number placed within parentheses is the node's enumerative number, the root node being numbered as (1). The text, for complex products, is an operator name ("all", "min", "exactly", "max") followed by a number used as parameter to the named operator. The operator "all" does not need an argument.

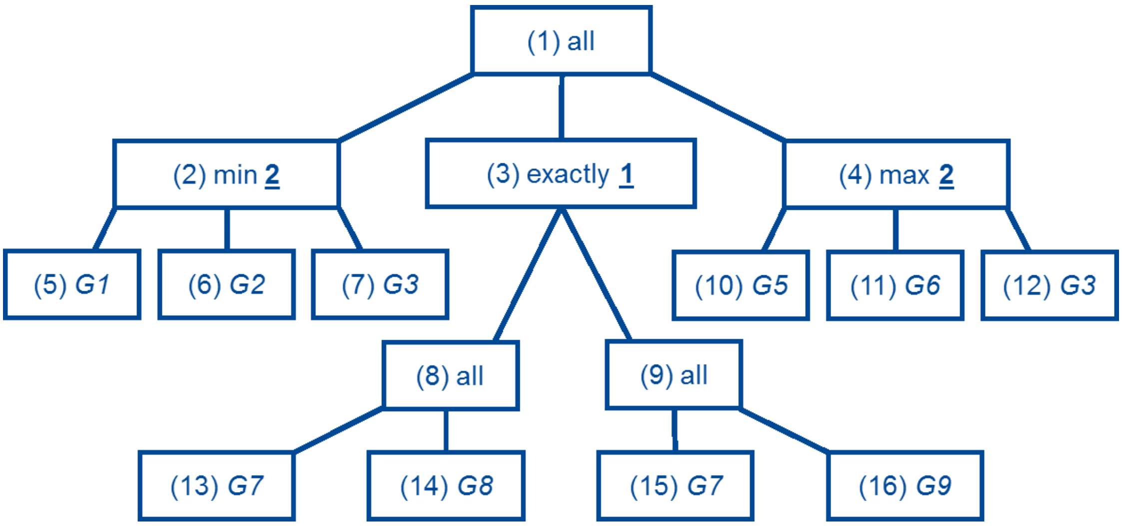

Figure 11: Complex product.

Basic products are labeled *G1* to *G9*. Same labels mean same power, e.g., for nodes (7) and (12). A basic product may be either one hour or 15 minutes of reactive power, respectively or either one hour or 15 minutes of active power, respectively. The example product defines that complex products represented by nodes (2), (3) and (4) all have to be traded successfully. The complex product represented by node (2) can be traded if a minimum of two out of the three basic products *G1*, *G2* and *G3* can be traded. The complex product represented by node (3) can be traded, if exactly one out of the two complex products from nodes (8) and (9) can be traded. Finally, the complex product represented by node (4) can only be traded if at most two out of the three basic products *G3*, *G5* and *G6* can be traded.

Embedding the eTelligence product description language in CIM

In order to embed the product description language in the IEC CIM, it was necessary to extend CIM. This was done in cooperation with the BTC CIM team and the eTelligence project partner OFFIS following the approach recommended by the CIM user group in [Co09]. Figure 12 shows the classes and enumerations used for the CIM extension. The classes which are part of the extension are labeled by <BTCExtension>.

The main task for extension was to individually decide whether an existing class could be extended for the purpose of our product description language or if a new class should be added. For example, the class GeneratingBid from the CIM Package Market Operations could be extended to the class ExtendedGeneratingBid. This class' extension, among others, contains the attributes capacitive, reactivePowerMax and reactivePowerMin, enabling a description of reactive power products. The class DeliveryTime was newly added. It serves the purpose of modeling delivery times used in a product. An in-depth description of the approach used during extension is contained in [SO10].

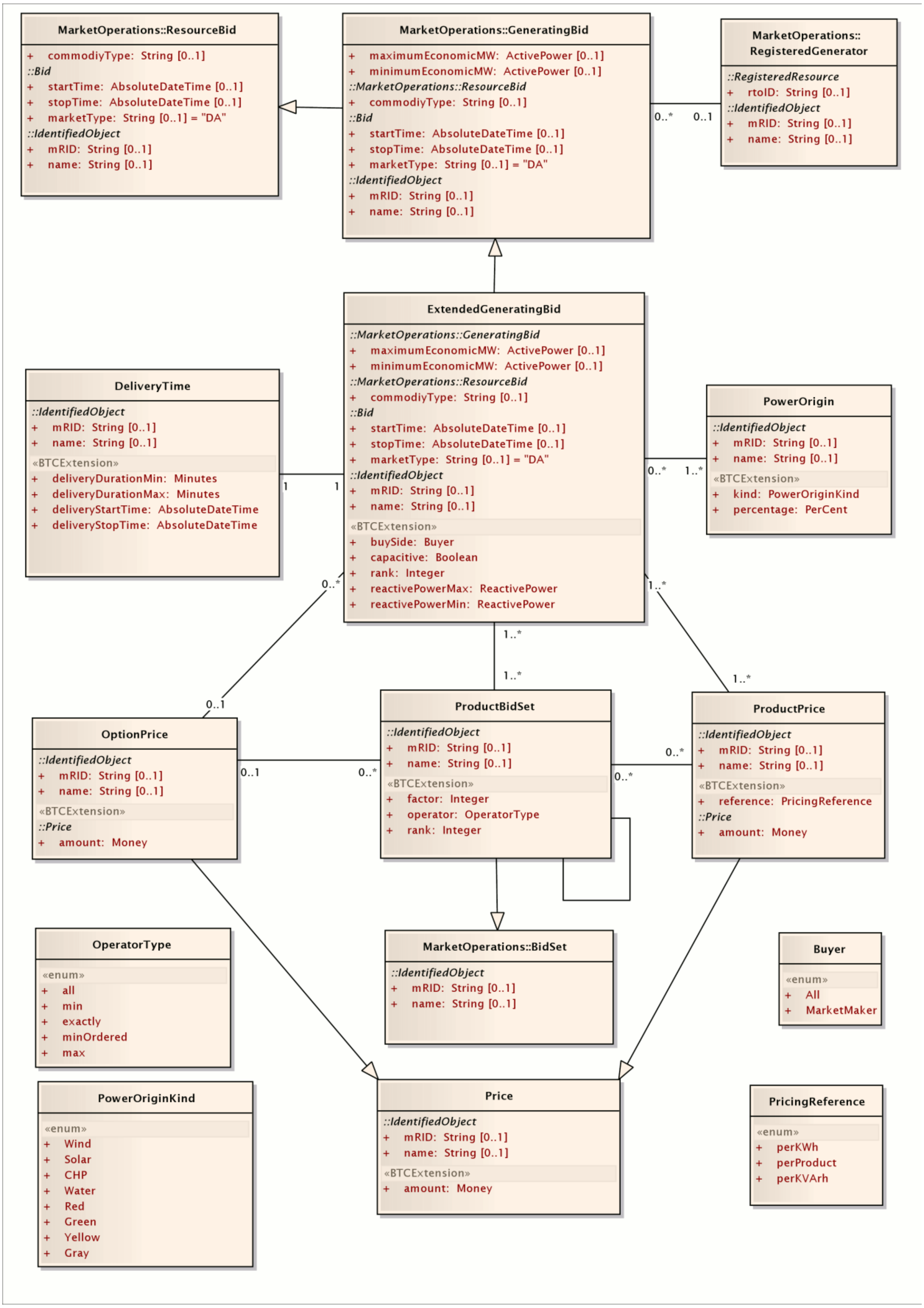

Figure 12: CIM – Extension [SO10].

The extended IEC CIM model was used to define the web service for accessing the market. This was done by generating XSD message schemata from the extended CIM using the CIM bench tool developed by BTC.

Describing bids in CIM

A product placed on the market place is called bid. Figure *13* shows a way of expressing a bid in CIM to buy reactive power using XML. Exactly one of two basic bids shall be sold. The XML message consists of a *ProductBidSet* containing two *ExtendedGeneratingBids* and an operator description. Figure 14 shows one of the two basic bids expressing that 42kVArh of capacitive reactive power can be delivered at the meter point designated Counter1 on 2009-11-17 from 03:00 to 03:15. The maximum price paid shall be 6.2 Euro cents per kVArh.

The bid is conveyed with the CIM message’s payload. For reason of clearness the message header and several attributes have been omitted from the presentation.

```
<ProductBidSet>
    <factor>1</factor><operator>exactly</operator>

    <!-- First bid -->
    <ExtendedGeneratingBid>
    </ExtendedGeneratingBid>

    <!-- Second bid -->
    <ExtendedGeneratingBid>
    </ExtendedGeneratingBid>

</ProductBidSet>
```

Figure 13: Skeleton of a complex bid expressed in CIM using XML.

```
<!-- First bid -->
<ExtendedGeneratingBid>
        <RegisteredGenerator><rtoID>Counter1</rtoID></RegisteredGenerator>
        <marketType>Demand</marketType>
         <capacitive>true</capacitive>
        <reactivePowerMin>
           <multiplier>k</multiplier><unit>VArh</unit><value>42</value>
        </reactivePowerMin>
        <reactivePowerMax>
           <multiplier>k</multiplier><unit>VArh</unit><value>42</value>
        </reactivePowerMax>
        <ProductPrice>
           <amount><unit>EUR</unit></value>0.062</value></amount>
           <reference>perKVArh</reference>
        </ProductPrice>
        <DeliveryTime>
           <deliveryStartTime>
              <value>2009-11-17T03:00:00</value>
           </deliveryStartTime>
           <deliveryStopTime>
              <value>2009-11-17T03:15:00</value>
           </deliveryStopTime>
        </DeliveryTime>
     </ExtendedGeneratingBid>
```

Figure 14: Basic bid concerning reactive power expressed in CIM using XML.

Conclusions from using CIM in eTelligence

During design and implementation of the eTelligence ICT infrastructure and the eTelligence market place, an important goal was to evaluate to what extent IEC CIM can be used for integration on different system layers of a smart grid ICT.

Misunderstandings of interfaces and message schemata are a typical source of faults in large software systems. We experienced that CIM-based messages and interfaces lead to relatively few misunderstandings between development teams that “meet” at an interface. This is because of two reasons: CIM-based schemata use common domain terminology and CIM enforces mandatory attributes. For instance, sensor readings have mandatory attributes both for the measurement unit (e.g., “Watt”) and for its multiplier (e.g., “Milli”). In some sense, domain knowledge from the CIM designers becomes available whenever CIM is used. Thereby, the standard and its documentation provided valuable guidance and structuring for designing messages and interfaces.

Some partners that were initially unable to use XML-based CIM message schemata and interfaces could be easily enabled by giving example messages to them. For this reason, we adapted our process to always supply example messages with a schema. Using the example messages, partners could directly verify whether the interfaces could be accessed. This helped to discover other potential faults, such as closed firewalls.

CIM-based messages are relatively verbose and large, even with respect to other types of XML messages. For instance, already for simple messages, “Smart Meter 15 reports a meter reading of 5 kWh”, many nested XML structures are needed. These messages require more bandwidth compared to size-optimized binary communication. However, today’s infrastructures provide sufficient bandwidth capacity and computational capacity.

The full potential of CIM becomes evident when parallel development using common interfaces is required. In eTelligence, CIM allowed us to independently develop more than 15 CIM-connected components after the interfaces had been specified. This was also supported by the architectural style using publish-subscribe and message-based communication. The process described in this article ensured that CIM-based artifacts were used, reused, and maintained in a coordinated and structured way.

Using CIM extensions, we were able to realize a rather complex, extensible language for defining the products traded on the eTelligence energy market. CIM was sufficiently flexible for such a language definition approach. The transfer of the language design into CIM was a straightforward process.

Based on our experiences, we recommend using CIM whenever additional subsystems or applications need to be integrated or flexibility towards changes in communicated information is required and for complex tasks such as defining languages such as our market product definition language. Surprisingly, CIM provides benefits w.r.t. maintainability even for transporting sensor data readings on lower system layers. However, for instance due to its connectionless communication, it appeared not well-suited for control purposes within the project. Other communication methods, such as defined in the IEC 61850 standards family are better suited for this task.

References

[BBE09] Bauknecht, Dierk; Bischofs, Ludger; Erge, Thomas; Klose, Thomas; Krause, Wolfram; Rüttinger, Hannes; Stadler, Michael: Das Leuchtturmprojekt eTelligence. *Energy 2.0*, publish-industry Verlag, 2009.

[Ch04] Chappell, Dave: Enterprise Service Bus. O'Reilly Media Inc., ISBN 0596006756, 2004

[Co09] Constable, B.: Extending the IEC Common Information Model (CIM) with Enterprise Architect, CIM User Group Meeting 2009, 2009.

[DKE10] DKE: Die deutsche Normungsroadmap E-Energy/Smart Grid. http://www.dke.de/de/std/KompetenzzentrumE-Energy/Seiten/Links.aspx, 2010.

[EPR08] EPRI - Electric Power Research Institute: The Common Information Model for Distribution: An Introduction to the CIM for Integrating Distribution Applications and Systems. *EPRI Technical Update Report 1016058*, *Electric Power Research Institute*, 2008.

[EPR09] Electric Power Research Institute: Report to NIST on the Smart Grid Interoperability Standards Roadmap. URL:http://www.nist.gov/smartgrid/, Electric Power Research Institute, 2009.

[FFG11] Frenzel, Martin; Friebe, Jörg; Giesecke, Simon; Luhmann, Till: Standardschnittstellen als nichtfunktionale Variationspunkte: Erfahrungen aus der EPM-Produktlinie. *Proceedings of Workshop on Produktlinien im Kontext: Technologie, Prozesse, Business und Organisation, Lecture Notes in Informatics (LNI)*, Gesellschaft für Informatik e.V. (GI), 2011.

[He07] Heidasch, Robert: Get ready for the next generation of SAP business application based on Enterprise Service Oriented Architecture (Enterprise SOA). *SAP Professional Journal*, 2007.

[IEC07] IEC - International Electrotechnical Commission: IEC 61968-11 and -12 System Interfaces for Distribution Management. International Electrotechnical Commission, 2007.

[IEC08] IEC - International Electrotechnical Commission: IEC 61968-14: System Interfaces for distribution management - XML Naming and Design Rule. International Electrotechnical Commission, 2008.

[IEC09a] IEC - International Electrotechnical Commission: IEC 61970-301 Energy management system application program interface (EMS-API) – Part 301: Common Information Model (CIM) Base. International Electrotechnical Commission, 2009.

[IEC09b] IEC - International Electrotechnical Commission: IEC 61968-9 Application integration at electric utilities – System interfaces for distribution management Part 9: Interfaces for meter reading and control. International Electrotechnical Commission, Edition 1, 2009.

[IEC10] IEC - International Electrotechnical Commission: IEC 61968-1 Application integration at electric utilities – System interfaces for distribution management. International Electrotechnical Commission, 2010.

[Mc07] Alan W. McMorran; An Introduction to IEC 61970-301 & 61968-11: The Common Information Model. Institute for Energy and Environment, Department of Electronic and Electrical Engineering University of Strathclyde, 2007.

[SS03] Schwimm, A.; Schelp, J.: Data Integration Patterns. Proceedings of the 6th International Conference on Business Information Systems, 2003

[SO10] Specht, Michael; Osterloh, Andre: Erweiterung des Common Information Model zur Modellierung von dezentralen Energieprodukten an einem regionalen Marktplatz. *GI Jahrestagung (1) 2010: 473-478*, 2010.

[Us09] Uslar, Mathias; Beenken, Petra; Busemann, Claas; Gonzalez, Jose; Kamenik, Jens; Mayer, Christoph; Nieße, Astrid; Rohjans, Sebastian; Schmedes, Tanja; Specht, Michael; Weidelt, Tobias; Schwarz, Karlheinz; Hein, Franz: Untersuchung des Normungsumfeldes zum BMWi-Förderschwerpunkt „e-Energy – IKT-basiertes Energiesystem der Zukunft“, 2009.