EXAMPLES OF INTER-APPLICATION COMMUNICATIONS IN DSO

Shengye Lu, Sami Repo
Department of Electrical Energy Engineering, Tampere University of Technology
firstname.lastname@tut.fi

Abstract

A DSO typically operates multitude software applications, and these applications need to communicate with each other. Traditional approach to implement inter-application communication is to establish dedicated links between each source and target applications, and data model from different data sources need mapping to each other. This approach does not scale well, as the number of links and mappings will grow exponentially with every new data source being added. To solve this problem, we use Service Oriented Architecture (SOA) technologies and Common Information Model (CIM) to establish the integration infrastructure that allows DSO applications communicating with each other in a loose-coupled way.

The paper will present the architecture, challenges and drivers of current approach of DSO software environment in general and how proposed SOA and CIM based approach could be used instead of it in case of example use cases. As this is an on-going research project, DSO applications are now being integrated gradually based on a series of use cases. This paper presents a few use cases currently under development. Use case 1: Data exchange between DMS and calculation engine – DMS exports network model to calculation engine; calculation engine generates load flow analysis result and sends back to DMS. Use case 2: Smart Meter alarm delivery – AMI generates alarms, which is collected by DMS; then DMS fetches related user data from CIS, and sends commands to Field Force Management system, which then sends result back to DMS after processing. Use case 3: Integration of Aggregator to DSO control center for voltage control use case.

Key words: CIM, SOA, electricity network management, aggregator

I. INTRODUCTION

A DSO typically operates multitude software applications. They are disparate applications, either legacy or new, from different vendors, and supported by dissimilar runtime environments. In order to implement utilities' business process, these disparate applications have to communicate with each other, exchanging or sharing data with each other.

In the past this kind of inter-application communication is implemented by establishing dedicated "point-to-point" links between each source and target applications using custom formats and protocols. Individual fields of data models from different data sources need map to each other. This approach does not scale well, as the number of links and mappings will grow exponentially with every new data source being added. Integrators have to take care of details of all relevant data sources.

In order to facilitate data exchange internally between different DSO applications and externally with other companies, two things are necessary:

1. A commonly agreed way of expressing the data content.
2. A loose-coupled way of integrating disparate applications.

For the first requirement, the Common Information Model (CIM) is selected. CIM provides the common format that can be used to express the data in electrical power system. Instead of mapping between every pair of source and target applications, each application only need to map data models between its internal interface and CIM interface, and exchanges data with each other using the same model, as shown in Figure 1.

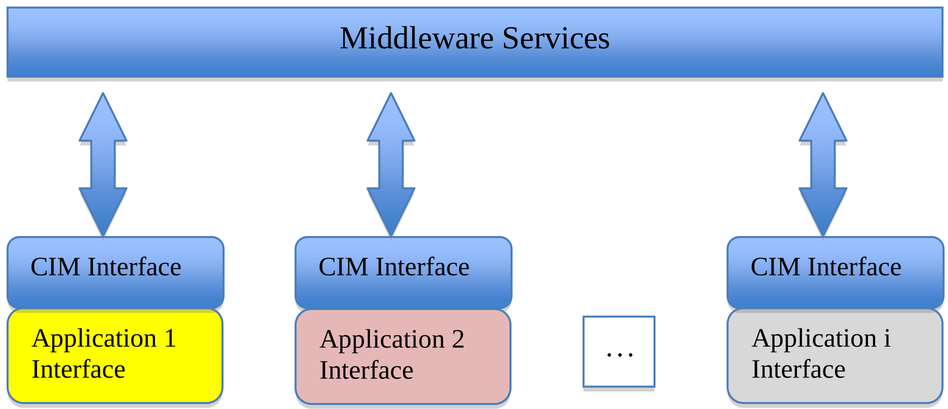

Figure 1: Each application maps data between internal interface and CIM interface for communication

For the second requirement, Service Oriented Architecture (SOA) is chosen. SOA is a computer system architectural style, which features loose coupling. Instead of establishing "point-to-point" links between each application, data are brokered among applications by one single middleware.

The paper will present how proposed SOA and CIM based approach could be used to facilitate enterprise integration in DSO software environment in case of example use cases. The paper is organized as follows: section II presents the IT architecture overview as well as the tools that our IT architecture is based on, i.e., CIM and SOA. Because migrating to CIM and SOA is our on-going project, DSO applications are being integrated gradually based on a series of use cases. Section III describes these use cases currently under development. Implementation is described in section IV. Section V describes future work, and finally section VI draws a brief conclusion.

II. System Design

Figure 2 depicts our proposed IT architecture. There are a lot of applications operated by a DSO (and its partners), such as Advanced Metering Infrastructure (AMI), SCADA, Distribution Management System (DMS), Customer Information System (CIS), Aggregator, Field Force management system, load flow calculation engine, etc. All of them are plugged into Enterprise Service Bus (ESB). The proposed IT architecture is built on top of two tools, SOA and CIM.

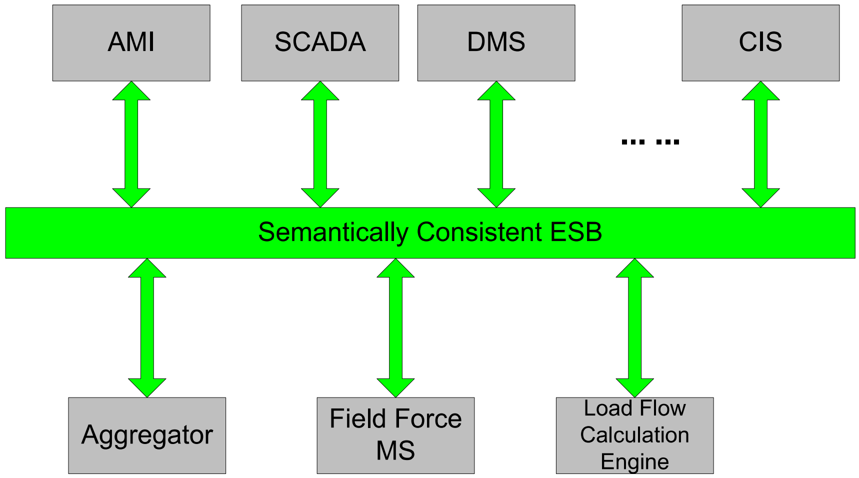

Figure 2: Our IT architecture overview

A. SOA and ESB

Service Oriented Architecture (SOA) is a software systems architectural style for creating and using software resources that are packaged as services. These services have well-defined interfaces, and they are available and discoverable on network, so that they can be combined and reused to create business applications. Independent services with defined interfaces can be called to perform their tasks in a standard way, and do not depend on the context or state of the other services. There are three flavours of implementation, including Generic-Typed Web Services, Strongly-Typed Web Services, and Java Message Service (JMS)[1].

Enterprise Service Bus (ESB) is an infrastructure that facilitates SOA through virtualization and management of service interactions between communication participants. It gives Application Programming Interface (API) that can be used to develop services and makes services interact with each other reliably. Technically ESB is a messaging backbone, which provides mediation, routing, message format transformation, logging/auditing, etc.

Using SOA architecture, applications like AMI, DMS, CIS, etc., are plugged into ESB with well-defined interfaces. Instead of exchanging data directly with each other, these applications publish/subscribe messages via ESB. ESB takes care of message routing and mapping. The sender does not have to know the exact location of the receiver or conform to the exact interface used by the receiver. Besides, data from the sender can be consumed by several receivers. These make message exchange more easily and efficiently.

B. CIM

While choosing SOA as the integration architecture, we use the IEC Common Information Model (CIM) to build message payload and interfaces.

The CIM is a set of standards that enable system integration and information exchange in electrical power system based on a common information model. The CIM model describes power system resources and services by using classes, attributes, and relationships between them, expressed in Unified Modelling Language (UML) notation. The model is not tied to a particular application's view of the world; instead it permits the same model to be used by all applications to facilitate information sharing between applications. The CIM consists of a series of standards. IEC 61970 standards, maintained by IEC TC57 WG13, deal with model exchange between applications in Energy Management System (EMS), or typically between TSOs. IEC 61968 standards, maintained by WG14, focus on supporting the inter-application integration of a utility enterprise. IEC 62325 standards, developed by WG 16, are related to energy market models & communications.

As an information model, the current CIM model contains 600+ classes with thousands of attributes and relationships [2]. An implementation does not need to include all of them. Therefore, profiles are defined to specify which elements must be included and which are optional in a particular use of the CIM.

In our project, to integrate DSO applications with CIM based messaging and SOA architecture, we need to do the following steps:

1. Define the contextual profile
2. Create message payloads XML Schema
3. Encapsulate the XML Schema (i.e., interfaces) in services WSDL
4. Plug the services into ESB

The step 1 is main issue. The typical workflow for step 1 is [2]: firstly, identify existing application interface attributes, and find correspondent classes and attributes in CIM model; then create a new profile by including all these related CIM classes, attributes and associations. In addition to creating new profiles, it is also possible to use CIM standard profiles. IEC 61968 standards define a lot of profiles that specify the messaging payloads for major business functions in utility enterprises. After the profiles are defined, by using tools like CIMTool [3] and ESB products, step 2 till step 4 can be finished straightforwardly.

C. Related Work

Adopting the CIM to facilitate utility enterprise integration has now been receiving more and more attention in industry. There are a lot of projects and initiatives going on inside and outside Europe. For example, ADDRESS [4] is the first European SmartGrid project to adopt the CIM. It focuses on the integration of active demand, promoting SOA architecture.

III. USE CASES

Because using CIM and SOA to integrate DSO applications is our on-going project, we are integrating applications gradually based on a series of use cases. This section describes these use cases currently under development.

A. Fault repairing

The first use case is “fault repairing”. This process involves message exchanges among several applications like AMI, DMS, CIS, and Field Force Management System:

1. Smart meter captures a problem from customer's residence. As a result, AMI - or more accurately, Automatic Meter Reading (AMR) gateway - raises an alarm, which is received by DMS.
2. DMS locates fault and priorities problem by consulting CIS for the customer data.
3. DMS issues a work order by sending commands to Field Force Management system, which then dispatches to correct work group.
4. Work group reports repairing work status. This is done by Field Force Management system sending report to DMS.

Using ESB and CIM to integrate these applications, the resulting message exchanges will be similar as in Figure 3. The messages will be constructed based on IEC 61968 profiles. As illustrated by Figure 3, all the messages are sent or received via ESB.

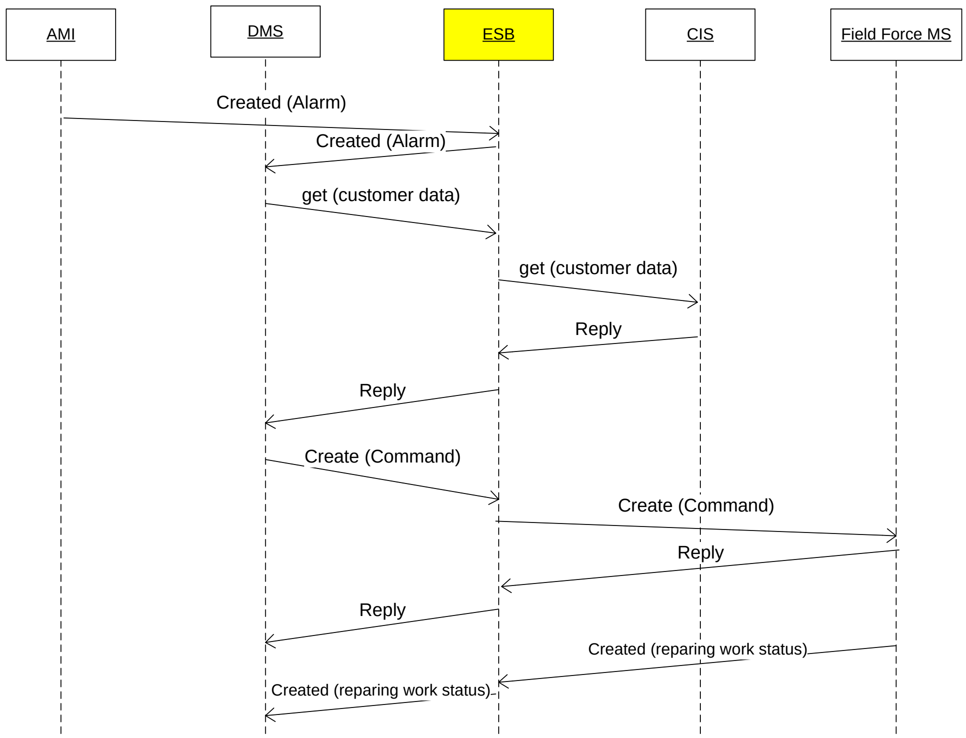

Figure 3: Data exchange in fault repairing use case

B. Model Exchange between DMS and load flow calculation engine

The second use case is model exchange between DMS and load flow calculation engine. In our project, we use ABB DMS 600 as the DMS, InterPSS OpenCIM [6] as the load flow calculation engine. Between them, there is Network Manager from Cybersoft Oy, functioning as a proxy. The data exchange is as follows:

1. DMS exports network model to Network Manager. The network model data contains ACLineSegment, Power Transformer, Switches, etc.
2. Network Manager exports the network model to OpenCIM.
3. OpenCIM carries out load flow analysis, and then exports the results to Network Manager. The result includes data about voltages and power.
4. Network Manager then forwards the load flow analysis result to DMS.

Unlike the other two use cases, the exchanged data in this use case will be mainly based on IEC 61970 profiles, i.e., IEC 61970-452 and 456.

C. Active voltage control in distribution networks

The third use case is about active voltage control in distribution networks [5]. Figure 4 shows all the relevant applications being integrated by ESB. The data flow goes as following steps:

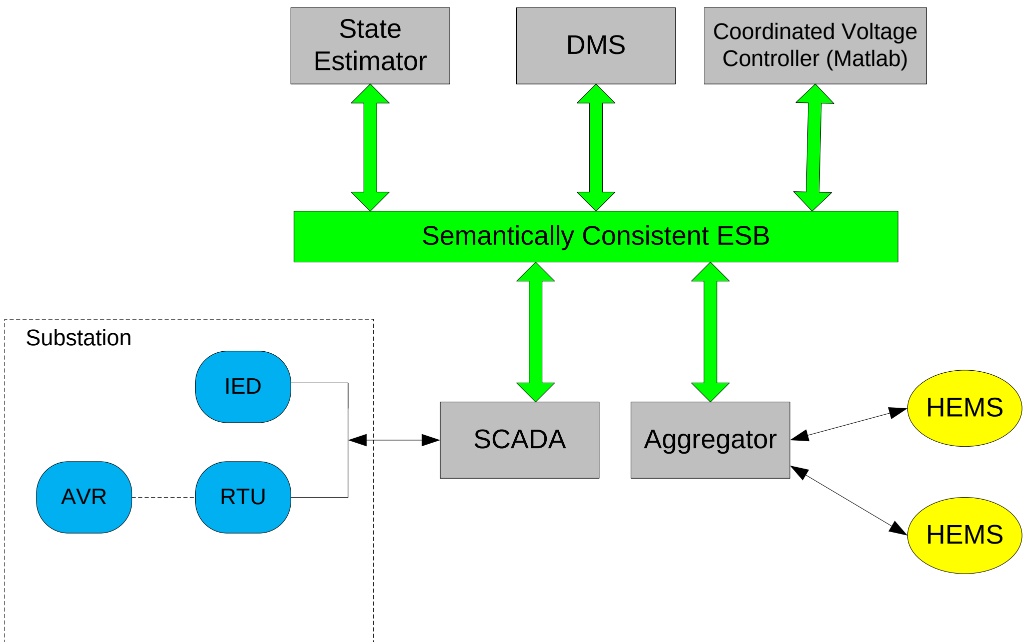

Figure 4: Voltage Control Use Case

1. Measurement data and status information from network and large generation units are collected by SCADA. SCADA publishes the data on ESB.
2. Home Energy Management System (HEMS) is measuring distributed energy resources (DERs) like small-scale generation and demand response at customer's home.
3. Aggregator collects all the measurement data from HEMSs, and then publishes available DERs for distribution network management on ESB.
4. DMS updates distribution network topology based on status information and the amount of available DERs based on HEMS information gathered from ESB.
5. State Estimator retrieves the measurement data from ESB. State Estimator executes state estimation. The state estimation results (e.g., maximum and minimum voltages in the network) are forwarded to Coordinated Voltage Controller via ESB.
6. Coordinated Voltage Controller determines whether control actions are needed. In the case of control action for DERs, DMS determines which customers are able to adjust their consumption or generation, and sends control command towards correspondent HEMSs via Aggregator. In the case of control action for network, Coordinated Voltage Controller calculates new set points for Automatic voltage regulator (AVR) of primary transformer tap changer and power factor controller of large generation unit, and sends the new set points to SCADA via ESB.

7. After getting new set points, SCADA sends the commands towards AVR and power factor controllers to change the set points. AVR and power factor controllers function as local voltage controllers, and bring network voltages back to acceptable levels.
8. After getting control command from DMS, HEMS directly controls customer's smart devices accordingly.

IV. IMPLEMENTATION

We will use a commercial ESB implementation, Microsoft Biztalk server, to integrate DSO applications based on aforementioned use cases. Besides use case B, all the other use cases can be integrated based on IEC 61968 standards. IEC 61968 parts 3 -9 defines CIM object model and normative XML payloads to exchange CIM derived data within a utility enterprise. Implementation profile IEC 61869-100 provides guidelines for using JMS and web services to integrate applications of the other parts of 61968 [1]. It defines normative message envelope and WSDL definitions for interfaces. However, 61968 standards only specify interface standards, do not care much about internal data structure. So we need to map the internal data of existing applications to and from CIM based interfaces.

For use case B, we have developed a network model simplification tool. This tool can parse network model in CIM XML file, and simplify the Node-breaker based network model to Bus-Branch model. The motivation is: The details of non-primary equipment like switches are unnecessary for load-flow calculation. Removing the details of these switches and extracting the correspondent bus-branch model can significantly reduce the volume of network model data.

V. FUTURE WORK

The objective of our research project is using CIM and SOA technologies to establish a new IT architecture for electricity network management. The new IT architecture can cope with increased system complexity, reduce cost, and facilitate interoperability with existing utility IT systems. Since a few interesting use cases have been identified in this paper, the next step is implementation. We will integrate more products and prototypes from our project partners, test against more use cases, to demonstrate the benefit of using this architecture.

For use case B, we are now trying to model the load flow analysis result using the CIM, so that the result can be easily used by other applications.

VI. CONCLUSION

This paper presents how we will use CIM and SOA based approach to facilitate enterprise integration in DSO software environment. The CIM provides a commonly agreed way of expressing the data content, while SOA provides a loose-coupled way of integrating disparate applications. Therefore, adopting CIM and SOA is a scalable and efficient way to enable inter-application communications in DSO. Since we are integrating DSO applications based on a series use cases, this paper also presents a few use cases currently under development.

VII. References

[1] Micheal Johnson, “IEC 61968-100 Implementation Profile Overview”, presentation in UCAIug Summit. 2011, November
[2] EPRI, “Common Information Model Primer”, First Edition. 2011, November.
[3] “CIMTool”, Available: http://wiki.cimtool.org/index.html
[4] “ADDRESS European Project Integration of Active Demand”, Cyril Effantin, presentation in CIM User Group meeting, 2012, May
[5] Kulmala, A., Repo, S., Järventausta, P. “Increasing penetration of distributed generation in existing distribution networks using coordinated voltage control”, International journal of distributed energy resources, 2009.
[6] “InterPSS OpenCIM”, Available: https://sites.google.com/a/interpss.com/opencim/