PROPOSED MODEL FOR APPLICATION INTEGRATION IN ELECTRIC POWER UTILITY

Aldis CERNICKI MIJIC
Koncar - Croatia
aldis.cernicki-mijic@koncar-ket.hr

Ante MARTINIC
Koncar - Croatia
ante.martinic@koncar-ket.hr

Dorjan MOCINIC
HEP-DSO - Croatia
dorjan.mocinic@hep.hr

ABSTRACT

Traditionally, power system applications were designed as a discrete business function. The users of such application belonged to a single functional area in a utility organization. However, as business evolves, electric utility operations may involve multiple business processes across several functional areas. For example, to properly oversee safe and reliable operation of the distribution grid, distribution system operators (DSOs) must maintain operational power system models that span multiple service areas. Reliable operation and control of the distribution power systems, as well as the increasing open market activities requires that real-time operational data be readily exchanged among the entities to support the power markets.

The infrastructure needed to support the ever-increasing information demands for this industry must be both, flexible and expandable in order to bring about a seamless environment. In addition, many other entities (e.g. traders, marketers, independent power producers, suppliers, customers, etc.) must also be able to interact with the market entities to effectively support and further develop the evolving deregulated power markets. These changing business conditions have resulted in a requirement for integration of heterogeneous legacy power system applications as well as new applications inside and outside an electric utility organization.
The primary problem in power system data management is the wide variety of platforms, protocols, data management and exchange, and security technologies that need to be integrated. The goal is to define common modeling elements that can be mapped onto a variety of technologies as needed, using adapters around a core of integration infrastructure.

Currently Croatian utility runs a project in which different applications will be connected using off the shelf integration infrastructure. One of characteristics of the project is that SCADA system will also be integrated. There are many discussions about that this, in our opinion and from our experiences good integration architecture design, allows different applications from different entities to work together in modern and efficient way. This paper provides some details of the technology proposed for implementation of standards for Enterprise Application Integration within electric utilities. This article also deals with real project of application integration in Croatian utility, representing design details and problem solutions regarding required data throughput.

INTRODUCTION

The regular operation and control of distribution power systems has become more and more important for utilities under deregulated and competitive environment. For example, a utility needs a Customer Information System (CIS) to provide customer information, needs a Geographic Information System (GIS) to provide geographic information for the distribution network, and needs an Outage Management System (OMS) to identify faults and restore the system. Electric distribution is an extremely large enterprise and people are looking for more efficient integration solutions for exchanging and sharing information. In traditional architectures integration between systems is typically manual, labor intensive, and therefore expensive. In the past, each integration task has been treated differently from all other integration tasks. Over time, the lack of standards resulted in a software management nightmare. While previously there has been no standard way to handle these types of integration problems, utilities integrated applications anyway. Most frequently, utilities have chosen a mix of these short terms solutions, which resulted in many separate point-to-point links.
Successful integration of a utility's various systems requires a method that does not require existing applications to be disturbed. Typically, integration is performed by employing a run-time integration infrastructure and component adapters. The run-time integration infrastructure provides a common platform for component links.

ENTERPRISE APPLICATION INTEGRATION SOLUTIONS

Application integration models

The basic premise of application integration is linking two systems and passing of information between them. It is differentiated from data-level integration by the fact that interface is created with application rather than its underlying database. Much of the focus of this style of integration is on interfaces that are exposed by custom or packaged applications. Three principal models exist for achieving application integration: Point-to-Point, Hub-and-Spoke and Message Bus.

Point-to-Pont is complex, not practical and requires for business to be put on hold while the problem is addressed.

Even if it could be accomplished, the solution would only be temporary fix, as any subsequent change to the business would require re-working of many application interfaces.

Hub-and-Spoke is far easier to manage. Each application connects to a central hub or integration server, which in turn manages and coordinates the routing and processing of the messages between the other applications. The hub-and-spoke topology simplifies maintenance, as the hub becomes the central point of control or administration for the organization.

Bus topology is built on a common communication backbone, and the terminology comes from its networking origins. Applications are connected to the bus using adapters, and they put messages on the bus. The message bus itself handles transformations and routing of messages between applications. Message routing is primarily based on publish/subscribe paradigm. Bus topology is the most distributed integration architecture so data does not need to flow through the central point. This type of approach is ideal where large volumes of information need to be published to a range of receiving applications. Bus that connects applications over entire enterprise is called Enterprise service bus.

These three models of application integration are shown in the figure 1 and figure 2.

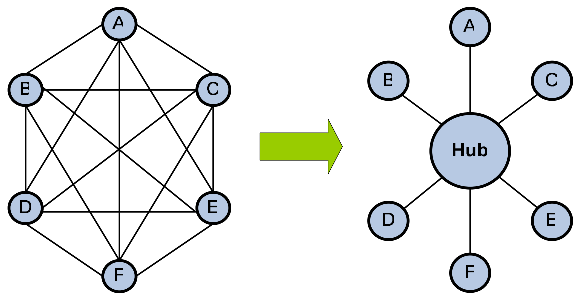

Figure 1. Point-to-point and Hub-and-Spoke architecture

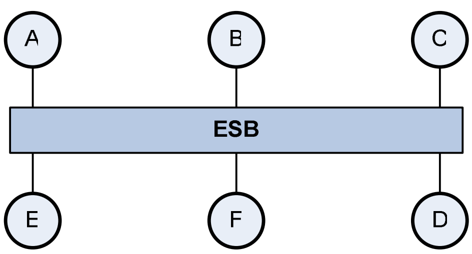

Figure 2. Enterprise service bus architecture

Service Oriented Architecture (SOA)

Enterprise service bus is most commonly implemented using SOA. SOA is computing paradigm that utilizes services as fundamental elements for developing applications or systems. Services are autonomous, self-describing, and open components, which support rapid, low-cost composition of distributed applications. Under SOA, any software component or a hardware device can be considered as a service. The service provider encapsulates its service implementations and has the ability to describe service capabilities, interfaces as well as qualities, and to publish these service descriptions to the outside world. Then these services can be discovered, selected, and bound by other distributed applications, which are called service consumers. Basic services, their descriptions, and basic operations, such as publication, discovery, selection, and binding, constitute the foundation of SOA. Since services may be offered and communicate over the local network, Intranet, or Internet, SOA provides a distributed computing infrastructure for both intra and cross enterprise application integration and collaboration.

In SOA services are usually exposed as Web Service. A Web Service is a specific kind of services that is identified by a URI, whose service description and transportation utilize open Internet standards. Laying on the groundwork of service-centric infrastructure and having all advantages of SOA, Web Services based framework surpasses other service-oriented computing paradigms by standardization and wide availability.

There are three basic components in a Web Services based architecture. They are service provider, service consumer, and service registry. The relationships among them are illustrated in figure 3.

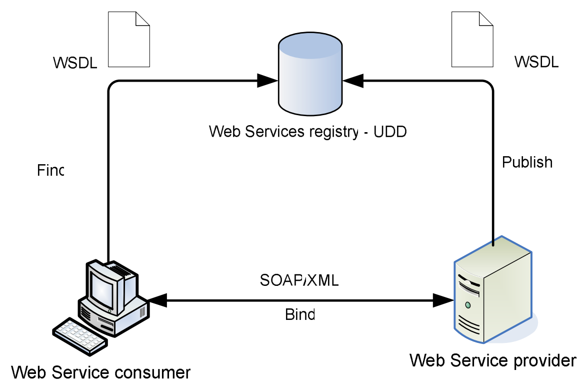

Figure 3. Technologies in Web Services architecture

Web Services technology is based on standardized Extensible Markup Language (XML). XML based Simple Object Access Protocol (SOAP) is used to exchange information between applications, and Web Service providers describe their service interfaces using XML based Web Services Definition Language (WSDL). Services are published to the service registry. Service consumers use the Universal Description, Discovery, and Integration (UDDI) APIs to locate services and get their interface descriptions. Owing to the open standardization of communication protocols, extensible information representation, pervasive Internet technology and advantages of SOA, Web Services provide an appropriate approach for electrical enterprises to build a loosely-coupled, language-neutral, and platform independent information integration infrastructure to link applications within utilities and across enterprises over the Internet.

WEB SERVICES FOR INTEGRATION OF POWER SYSTEM APPLICATIONS

In order to implement seamless application and information integration in a power system, electrical industry makes great efforts to standardize communication protocols and data models. Several standardization efforts of information models regarding power systems and equipments are published, for example CCAPI by EPRI, IEC61970 and IEC61850 by IEC, etc. These standards define the unified common information formats and services interfaces for the interoperation of power production, transmission, distribution, marking, and retailing functions. All these information standards accompanying with prevalent SOA and Web Services technology make it possible to build an open, flexible and scaleable infrastructure for application integration. A Web Services infrastructure for application integration in power system is illustrated in figure 4.

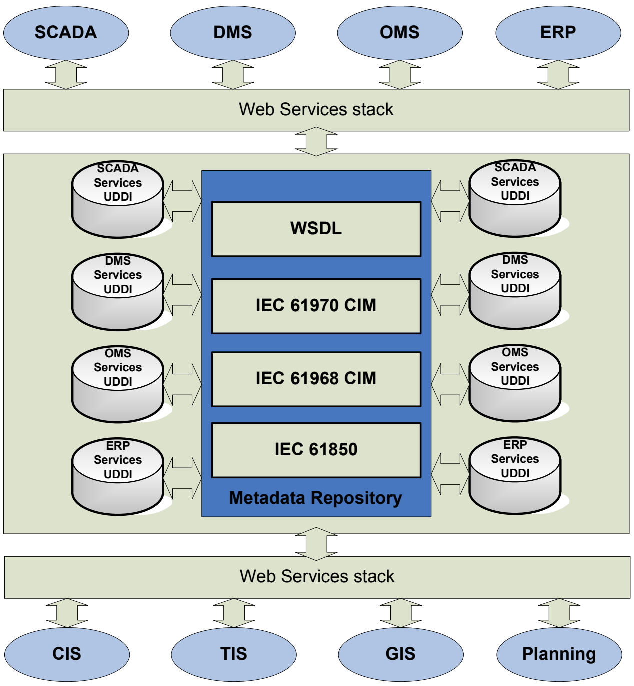

Figure 4. Web Services infrastructure for application integration

DISTRIBUTION MANAGEMENT SYSTEM INTEGRATION

When realizing a DMS by integrating many subsystems, it is helpful to use enterprise application integration techniques combined with service-oriented architectures. A service-oriented architecture could therefore be supported by using the CIM standard in a project employing an integration platform for connecting subsystems and it can be used to run processes via Web Services being orchestrated by business process modeling languages.

Common Information Model is an information model representing real-world objects found in the power distribution environments. The real-world objects are represented as classes, along with their attributes and relationships which identify their particular characteristics. The Unified Modeling Language (UML) is utilized to define the packages containing class diagrams and class specifications for the CIM.

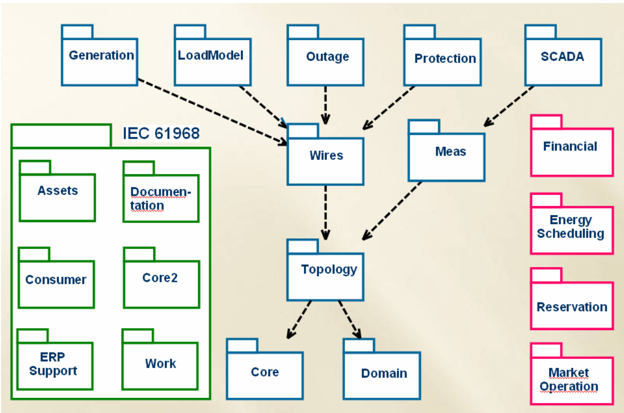

Figure 5. Package overview of the CIM

There are many uses for the CIM, an ontology that is being standardized through Technical Committee 57 of the International Electrotechnical Commission (IEC TC57). The most common uses have included application modelling, information exchanges, information management and system integration. Some of the common ways in which CIM is used are:

- To define database structures
- To define internal application data structures
- To define message structures (e.g. for communication within an ESB, often using XML)
- To define ESB and application interfaces (e.g. Web Services in a SOA using WSDL)
- For the encoding of models (e.g. using XML as defined through IEC 61970 and IEC 61968)
- For the encoding of updates to models
- For the encoding of data in request and response messages
- For the encoding of data in event messages.

REQUIREMENTS OF REAL-TIME DATA EXCHANGE

Application integration project in Croatian power utility is

focused on integration of SCADA and other decision support systems in order to enable real-time information access across multiple systems.
The main task is definition of suitable design of ESB architecture according the data flow in normal, high and peak activity state of power system. The normal activity state corresponds to typical condition of the power system. Data flow from the process to the SCADA system in this state can be taken as a constant An. According to recorded statistic of telemetry activity during normal activity state, these are typical values of telemetered data in this state (the values are expressed as percentage of database totals for particular items):

- 1 % of all signals are changed every hour (and every change is logged in an event and an alarm list),
- 1% of all measurands are change every second,
- All measurands are subjected to limit processing and each minute 0.01 % of these exceeds limits and are logged in alarm list,
- 1% of all calculated values, with an average of four operands each, are updated every second.

With this hypothesis, for our system (approximation for distribution area in Croatia) An = 155 items /second.
In the high and peak activity state, data flow (incoming items in SCADA system/second) jumps to its peak value at the beginning of the disturbance and fades away by e-function. To define data flow we applied Reference Data Avalanche (RDA) formula. RDA is based upon experience and includes the following characteristics:

- At normal activity state 1 to 10 percent of all measurands change per second,
- At beginning of peak activity the measurand flow is 5 times as high as in normal activity state,
- 30 percent of all status information change within peak activity state,
- Approximately 50 percent of all changes occur within the first 2 minutes of power disturbance.

The RDA is described by peak data flow at the beginning of the disturbance which fades away by a e-function within 10 minutes. With these assumptions the RDA formula for data flow is:

$$As = a * e^{-t/\tau},$$

where: a = (15% of all measurands plus 0,17% of all signals of the process database) / second, τ = 3 minute, $0 \le t \ge 10$.

Considering that peak data flow appears in $t = 0$ and total data flow is $A = As + An$, we can aspect 1670 items will be changed per second.
Whereas we conclude that in high/peak activity state data must be exchange in such frequency that web services architecture can not meet the needs and some other middleware technology must be used for communication to and within enterprise service bus. In this specific case application like SCADA should be integrated and connected to ESB in such a way that basic SOA concept is not undermined.

CONCLUSION

While each type of application and information integration solution requires same basic technologies, such as messaging, data translation, transformation, and routing, each business scenario also has unique requirements. Usage of Enterprise Service Bus based on paradigms and standards such as SOA, Web Services and CIM provides efficient and flexible integration infrastructure. When exchanging great amount of real-time data, asynchronous messaging using publish/subscribe paradigm will be far more efficient. To implement this kind of communication Web Services infrastructure has some drawbacks, mostly regarding performances. In this case, while data model stays the same, some other communication technologies and protocols must be used. The optimal solution will most often depend on implementing not one integration technology, but the right set of technologies, and getting them all to work together seamlessly. In order to minimize maintenance costs while maximizing reuse and agility, organizations will need to deploy integration technology strategically. Developing application integration architecture enables an organization to plan strategically and implement tactically.

REFERENCES

[1] Q. Chen, H. Ghenniwa, W. Shen, 2006, “Web-Services Infrastructure for information integration in Power System”, *Power Engineering Society General Meeting, IEEE*

[2] P.Shi, S.Gandhi, 2001, “Enterprise Application Integration”, *DiamondCluster International*

[3] I. Backer, 2001, "Common Information Model (CIM) CIM 10 Version", Final Report

[4] D. Sesnic, R. Jelusic, A. Cernicki Mijic, 1997, "Analysis of SCADA Systems in Transmission Area Zagreb Based Upon Proza Open Data", *Proceedings MIPRO ’97, 20th International Convention*, 205-209

[5] A. Martinic, 2005, “Middleware model for data acquisition in SCADA system”, *Master’s Thesis, Faculty of Electrotechnical Engineering and Computing, Zagreb, Croatia*