# IEC CIM architecture for Smart Grid to achieve interoperability
# International CIM Interop in March 2011

**Nada Reinprecht - IT Architect**
**IBM Global Business Services**
**60 City Road, SouthGate, Melbourne, VIC**
**E-mail nadarein@au1.ibm.com**

**Javier Torres**
**IBM Software Group**
**11501 Burnet Road, Austin, Texas**
**E-mail jrtorres@us.ibm.com**

**Marilza Maia**
**IBM Software Group**
**415 Market St, San Francisco, CA**
**E-mail mmaia@us.ibm.com**

**Keywords:** Smart Grid, Common Information Model (CIM), IEC 61970/61968, Interoperability Test, Work Management, Asset Management, Construction, Maintenance

### Abstract

This paper describes IEC CIM architecture and demonstrates how this architecture fits into Smart Grid architectural principles: loose coupling, shallow integration and layered approach. The CIM architecture is traced back to UN/CEFACT modeling methodology, Zachman formal architectural framework and Open-EDI model to ensure reusability and repeatability of its design solutions. It maps the CIM architecture layers with the Interoperability Framework to illustrate the interoperability character of the architecture.

To demonstrate how the above architecture works in practice to achieve interoperability, the paper describes International IEC CIM Interoperability test in March this year especially concentrating on the testing of IEC 61968 interfaces. Multiple vendors: TIBCO, Oracle, Alstom, IBM, Telvent, GE, Siemens and others were interoperating by implementing CIM compliant SOA based integration.

## 1. INTRODUCTION

Smart grid is about an automated optimization of the electricity delivery system including all its components: central and distributed generation, high voltage network and distribution system, energy storage installations, end-use consumers and their appliances, electric vehicles and other household devices. It is characterized by bi-directional flow of energy and the information. Automated optimization of smart grid requires large amount of information to be collected, exchanged and processed. Successful implementation of smart grid involves the collaboration of multiple technologies such as electricity, communication, Internet and computing. One critical goal of the smart grid is to enable new technologies and support new business models as they evolve.

Such a goal requires a flexible integration architecture following principles of loose coupling, layered systems and shallow integration. As greater investment is made in the deployment of more smart grids and to addressing smart grid goals, standards will play a critical role both in protecting that investment and enabling interoperability between the millions of components that will have a role in future smart grids. The International Electrotechnical Commission [IEC] 61970 and 61968 series of standards, collectively known as the Common Information Model [CIM], have emerged as one such set of standards critical to smart grid.

Section 2 of this paper describes the main components of CIM architecture and illustrates how CIM architecture fits into smart grid architectural principles. Section 3 traces CIM back to other architectural frameworks such as UN/CEFACT electronic business modeling methodology, Zachman enterprise architecture framework and ISO Open-EDI reference model. This is to demonstrate how CIM architecture has re-used components of the above models to ensure robust and reliable framework which can satisfy smart grid requirements.

Implementation of the above architectural principles is demonstrated in CIM IOP in March 2011. Section 4 talks about this CIM interoperability test in general while Sections 5 and 6 offer the details of the CIM IOP reflecting especially to Maximo and webSphere implementation of tested CIM compliant integration.

## 2. IEC CIM ARCHITECTURE

IEC Technical Committee TC57 CIM architecture [Figure 1] is built into the core of smart grid architecture principles. It is a layered architecture which ensures implementation of standard methodology at each layer. Both information and context layers are semantic with the defined rules for translation to the implementable physical syntax. The architecture supports composition by allowing for bridging and extensions; therefore, it prevents a need for a universal database.

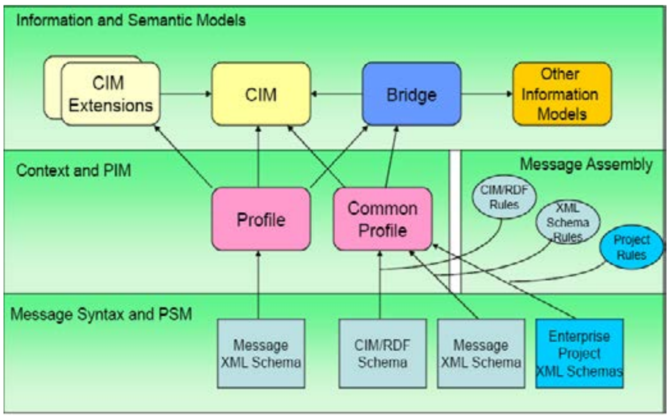


**Figure 1 CIM Layered Architecture**

Clear semantic definition of an interface which provides for conceptual understanding of exchanged information, allows for loose coupling as enables two or more integrating systems to interoperate without elaborate pre-arrangements. CIM compliant integration provides for minimum knowledge included in the message (shallow integration) thus enhancing value of composition. Common components of the integration for similar business processes are realized by common messages.

Service-Orientated Architecture (SOA) with Web Services providing concepts to define self-contained services in loosely-coupled system architecture was the chosen CIM integration architecture implemented in the IOP. The main characteristic of the CIM-compliant SOA implementation is a semantic built in Web Service Definition Language (WSDL) which enables an easy integration. Predefined WSDL provides product vendors with a contract to build against. This has enabled multiple vendors across countries and continents to interoperate by exchanging these messages, understanding their meaning and reacting to these messages appropriately.

IEC TC57 Working Group 14 is concerned with development of the IEC 61968 standard interfaces for distribution management. This includes network operations (Part 3), records and asset management (Part 4), work management (Part 6), customer support (Part 8) and meter read and control (Part 9). While the distribution management functionality is divided into the functional groups to facilitate standards development the standard interfaces span across the functional groups as well.

## 3. CIM AND OTHER STANDARDS

As a utility enterprise integration standard, CIM architecture draws from other electronic commerce and enterprise architecture methodologies.

Early standardized electronic data interchange models include ISO/IEC Open-EDI Reference Model illustrated in Figure 2.

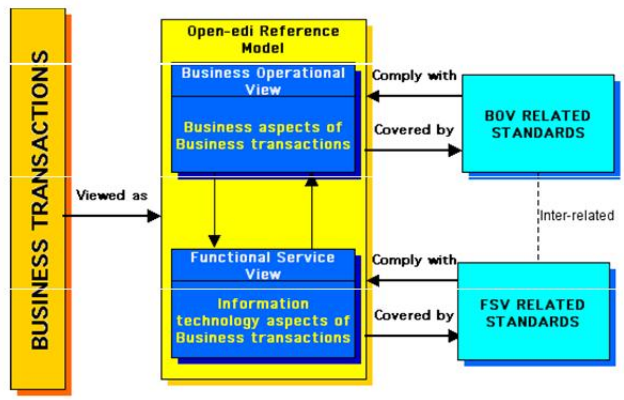


**Figure 2 Open-EDI Reference Model - ISO /IEC 14662**

Open-EDI Reference Model consists of two main views: Business Operational View (BOV) describing business aspects of business transactions and Functional Service View with information technology aspects of business transactions. BOV provided business context and information semantic of the message exchange. Functional service View has on the other side, provided technology implementation of the message exchange. There could be multiple technology implementations for the same business context.

UN/CEFACT modeling methodology (UMM) is the formal methodology for describing any Open-EDI scenario as defined in ISO/IEC 14662, Open-EDI Reference Model. The UMM provides a procedure for modelling collaborative business processes involving information exchange in a technology-neutral, implementation-independent manner. The UMM layered architecture consists of three layers: Business Domain View, Business Requirements View and

Business Transactions View. All three layers correspond to Business Operational View of Open-EDI model.

Drawing a parallel with CIM, the semantic information layer and contextual layer from CIM correspond to the BOV. Similar to Open-EDI and UMM, CIM has a clear separation between semantic and syntactical layer allowing for multiple technologies being implemented for a single logical design.

In Figure 3, it is illustrated how Zachman formal enterprise architecture framework relates to Open-EDI and UMM frameworks. From the diagram it can be seen that there is almost one to one relationship between Zachman architecture layers and UMM layers: context layer corresponds to Business Domain View, conceptual layer corresponds to Business Requirements View while logical layer corresponds to Business Transactional View. All three of them are part of Open-EDI Business Operational View. These layers deal with semantic description of the problem and there is no technology involved. Similarly, CIM layers semantic information and platform independent context, model the solution in technology neutral manner.

Functional Service View of Open-EDI corresponds to the CIM Syntactical layer. The clear rules exist for translation between semantic and syntactic layers. This ensures the repeatability and re-usability of all the above collaboration architectures.

CIM defines multiple different technologies for CIM compliant integration. Among them, at this stage the most used is SOA with CIM compliant web services to which vendors can build their interfaces.

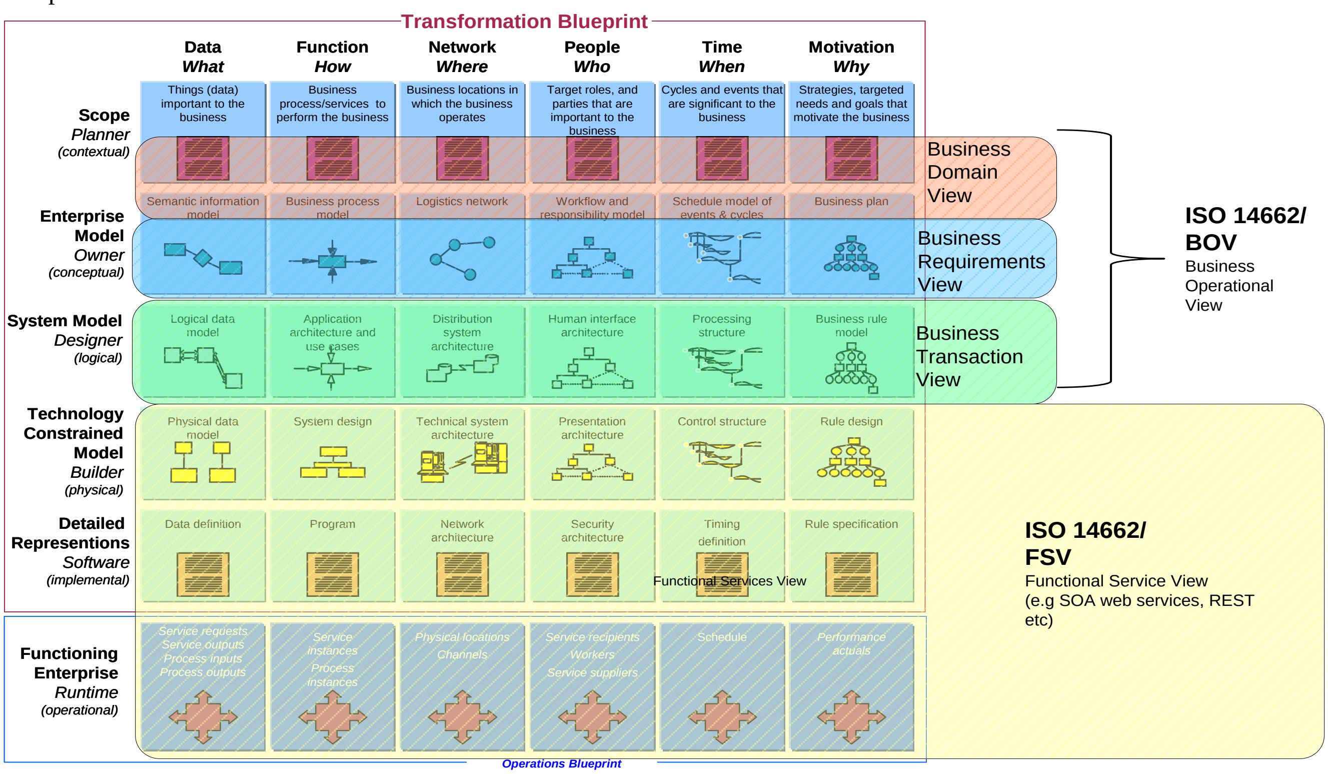


**Figure 3 Zachman architecture overlaid with UMM and Open-EDI**

## 4. SMART GRID STANDARDS IOP TESTING

As indicated in the above perspective, it is not just the standard that is important for interoperability, but also the testing process performed on the standards. This requires more than just testing for standards conformance of single products or applications. Looking at the Gridwise Interoperability context-setting framework in Figure 4, it is clear that interoperability has meaning at multiple levels and testing could be performed at many levels depending on the standard.

The CIM standards can be aligned to provide both syntactic and semantic interoperability. While, using the standards gives the systems/application both an understanding of data structures for the messages being exchanged, it also provides an understanding of the concepts contained in those messages. These are clear rules that govern the format or structure of information, and the definition or relationships of that information. CIM interoperability testing proves the ability of two or more systems/applications to successfully integrate with each other.

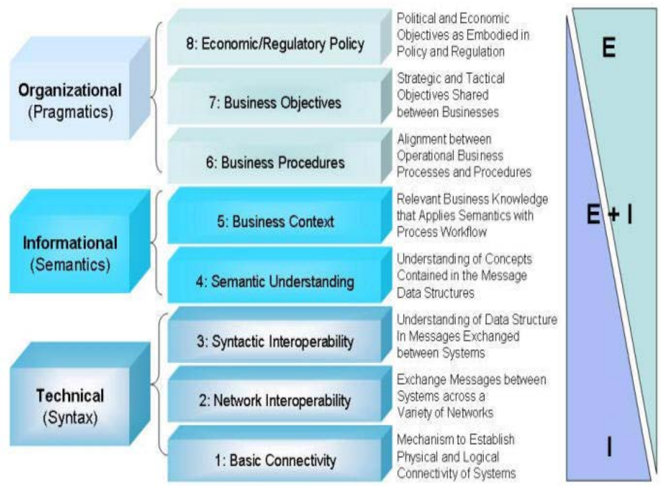


**Figure 4 GridWise Interoperability Framework**

As noted, interoperability testing for the CIM standards has already started on both the transmission and distribution side. The UCA IOP tests that took place in 2011 provide a good example for approaching such testing. The objectives of this interoperability event was to show that the participants could exchange messages or models using the IEC 61968 Part 3, 4, 6, and 13 standards through the integration patterns outlined in IEC 61968 Part 1. Service-Orientated Architecture (SOA) with Web Services providing concepts to define self-contained services in loosely-coupled system architecture was the chosen CIM integration architecture implemented in the IOP. The main characteristic of the CIM-compliant SOA implementation is a semantic built in Web Service Definition Language (WSDL) which enables an easy integration. Predefined WSDL provides product vendors with a contract to build against. This has enabled multiple vendors across countries and continents to interoperate by exchanging these messages, understanding their meaning and reacting to these messages appropriately. The remote infrastructure approach employed in this testing lead to reduced participation costs. This allows multiple participating systems to connect remotely and show they are generating or consuming a 61968 based message, thus validating interoperability and compliance to the specified IEC 61968 messages. Not only does this successful testing move the industry towards the notion that these 61968 standardized messages are complete, correct, and ready for real world use, but it also serves to bring multiple key parties together to define business processes, the appropriate test context, and potentially identify issues, updates, or corrections that can be fed back into the standard.

## 5. CIM IOP - MAXIMO

The IBM Maximo for Utilities application enables electric, gas and water utilities to efficiently and effectively manage their assets. Maximo contains modules for asset management, work management, crew management, and many other functional capabilities needed for comprehensive asset lifecycle and maintenance management. As with many enterprise systems, utilities will often have use cases that require the integration of Maximo to external systems. Whether to extract detailed asset information or, as was the case addressed in the CIM IOP tests, to initiate an action (i.e. maintenance work) on a specific asset.

To support standards based integration with external applications, a core and essential component used in the CIM IOP implementation is the Maximo Integration Framework. The framework architecture is made up of highly flexible components which provide the SOA technologies for data, transport, and communication necessary to expose Maximo content. Core to the integration framework, Maximo uses what are known as object structures to define the common data layer for all inbound and outbound data. The object structures are the basis to publish, query, add, update, and delete Maximo data via channels (for outbound transactions) and/or services (for inbound transactions). Out of the box, the framework includes predefined content for integration to some common business objects. As well as a tool kit to extend or define new integration content and integration points, along with support to transform integration content using tools like XSL and Java. To meet the varying integration requirements of external applications, the framework can be quickly configured and customized to support multiple integration approaches. As shown in figure 5 below, the integration framework facilitates scalable data exchange in real time or batch mode, synchronously or asynchronously over a variety of communication protocols, including: Web services, HTTP, JMS, and flat files. It is this robust framework, with its highly flexible and customizable components that provide support for industry standards like the CIM.

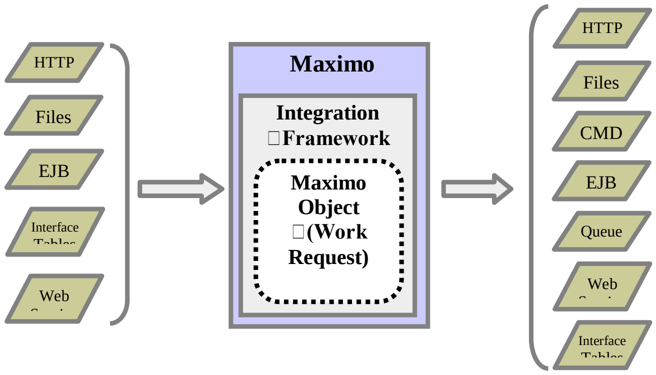


**Figure 5 Maximo Integration Framework**

The use cases chosen for CIM IOP testing centered around having an external system initiate a work request on a particular asset (for example, a DMS requests maintenance work on a transformer). This integration requires Maximo to accept and process inbound request data. The integration framework can facilitate inbound data exchange asynchronously or synchronously with three types of services: object structure, enterprise, or standard service. Of these three inbound services, Enterprise services differ in that they have support for additional content processing, business rules, and transformations. As such, they were the component used in the implementation for the CIM IOP test. Enterprise web services allow users to define a single service for each operation that is to be exposed by the framework. For the CIM IOP, a 'create' operation is defined to allow the external systems to request a new work order be defined in the system. The interested external system can call the enterprise service either asynchronously (XML over HTTP Post, web service over JMS, Direct JMS connection, etc) or synchronously (XML over HTTP Post, web service, etc). The framework uses the Apache Axis 2.0 engine to expose WS-I Basic Profile compliant web services.

However, since the inbound data is in the CIM XML format for a 'CreateWorkRequest' message and not in a native Maximo object format, the enterprise service needs to implement a data processing layer to transform and apply business processing rules before it reaches the system objects. Figure 6 depicts the synchronous enterprise service processing activities (though not every activity applies to every message transaction), which enable the framework to map the external CIM XML schema to the object structure XML for both the invocation and the response.

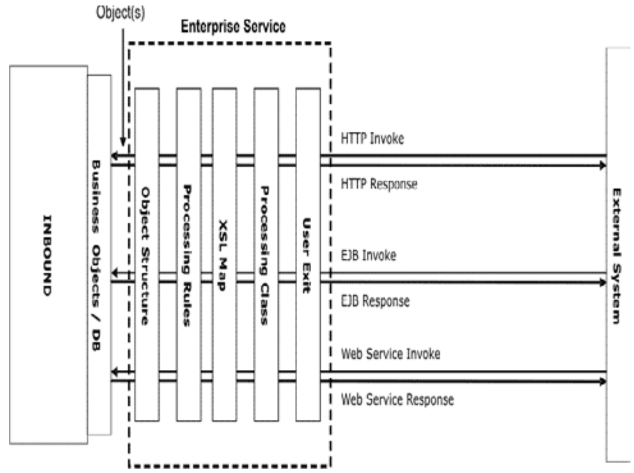


**Figure 6 Enterprise Service Processing Flow**

When the inbound request message enters the enterprise services processing flows, the framework can be configured to use various tools for customizing the message and framework behavior:

- The processing rules engine can filter and transform the XML message.
- The user exit Java classes can filter, transform data, and implement business logic.
- The data processing Java classes can filter, transform data, and implement business logic.
- The XSL Maps can transform data and perform mapping of the XML message to another format.

By extending predefined exit classes, users can modify/interact with the message at different points in the inbound/outbound processing. Looking specifically at the technique used in the CIM IOP implementation, the enterprise service has a place holder to implement an XSL file and to manipulate the data to be set to the object structure after the Java exit processing is completed. In this case, the framework needs to transform the Work Request format to the Maximo Work Order format. The XSL excerpts shown in figure 7 demonstrate how the message is translated within the process flow.

```
- <xsl:element name="WORKTYPE">
      <xsl:attribute name="changed">false</xsl:attribute>
      <xsl:value-of select="p:Work/p:kind"/>
  </xsl:element>
- <xsl:element name="WOPRIORITY">
      <xsl:attribute name="changed">true</xsl:attribute>
      <xsl:value-of select="p:Work/p:priority"/>
  </xsl:element>

- <xsl:choose>
  - <xsl:when test="p:Work/p:TimeSchedules/p:kind = 'estimate'">
    - <xsl:element name="SCHEDSTART">
          <xsl:attribute name="changed">true</xsl:attribute>
          <xsl:value-of select="p:Work/p:TimeSchedules/p:scheduleInterval/p:start"/>
      </xsl:element>
    - <xsl:element name="SCHEDFINISH">
          <xsl:attribute name="changed">true</xsl:attribute>
          <xsl:value-of select="p:Work/p:TimeSchedules/p:scheduleInterval/p:end"/>
      </xsl:element>
    </xsl:when>
  - <xsl:when test="p:Work/p:TimeSchedules/p:kind = 'request'">
    - <xsl:element name="TARGSTARTDATE">
          <xsl:attribute name="changed">true</xsl:attribute>
          <xsl:value-of select="p:Work/p:TimeSchedules/p:scheduleInterval/p:start"/>
      </xsl:element>
    - <xsl:element name="TARGCOMPDATE">
          <xsl:attribute name="changed">true</xsl:attribute>
          <xsl:value-of select="p:Work/p:TimeSchedules/p:scheduleInterval/p:end"/>
      </xsl:element>
    </xsl:when>
  </xsl:choose>
```

**Figure 7 Inbound XSLT Sample**

The XSL includes rules for basic mapping of the necessary fields from Work Request inbound message to a Work Order object representation in Maximo. For example, the work type for the new work order is picked up from the 'kind' field and the work order priority from the 'priority' of the Work object in the CIM WorkRequest message. There is also the ability to do more than simple mapping, the XSLT can do conditional mapping of certain fields in the Maximo Work Order business object based on values set in the incoming message. For CIM IOP, the XSLT inspects the 'kind' field from the TimeSchedules object of the incoming message. If that value is set to 'estimate', then the work order scheduled start and finish time fields are filled in. If the value is equal to 'request', then the target start and finish dates are filled instead. Aside from the XSL, Java processing classes are also implemented to provide extra business logic for the processing flow. Within these classes, we can check that the incoming message have either a valid asset identifier (MRID) or a valid location identifier (MRID). This is necessary since a work order can only be created on an asset or location that currently exists within the Maximo system. All of these processing rules can be customized to meet the integration requirements and agreement between the systems being integrated.

Although not shown directly in the CIM IOP, there are also use cases for enabling Maximo to send information to external systems. For example, after the work request has been created, enabling Maximo to notify the external system of status changes on that specific work request. The integration framework can facilitate this outbound data exchange asynchronously (through JMS queues) using publish channels or synchronously (through a direct connection) using invocation channels. Both of these outbound channels allow users to customize the integration to meet external system requirements for content format and communication protocol. For example, users can use the integration framework to notify the DMS system using web services when a work order it initiated is completed. Through event based integration, the framework can send updated work order notification in the format of a CIM CreatedWorkRequest message.

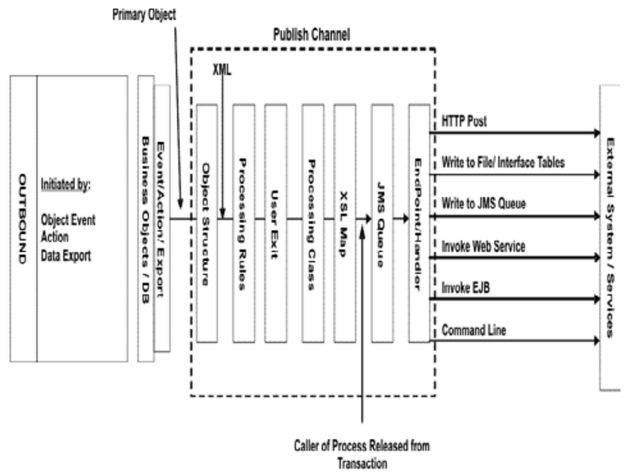


**Figure 8 Publish Channel Processing Flow**

Figure 8 above showcases the processing flow for asynchronous publish channels. Using a registered listener, an object event triggers the publish channel for a object structure (i.e. Maximo work order object). At which point, similar processing takes place as identified earlier for the enterprise service (processing rules, user exists, data processing classes, and XSL maps) to send the updated data to an external system. Once the processing flow completes, the framework places the message into the queue, where a system cron task picks up the message and sends it to an external system via a configured endpoint. The endpoint identifies the protocol to use (such as HTTP or Web services) and other property values (such as URL, user name, and password). As shown earlier, the integration framework provides multiple optional place holders within the outbound transaction flow to customize the publish channel. Processing rules can access, evaluate or change the values in XML and object fields, object sets, and integration and system controls. XSL files can be used to map publish channels to custom user-defined formats, manipulating the data that is sent to the external system after the Java exit processing completes. As with the enterprise service, the XSLT can be as simple or complex as needed to meet the integration requirements. Figure 9 depicts a portion of the XSLT used in the publish channel to convert Maximo Work Orders to CIM Work Request messages.

```
<?xml version="1.0" encoding="UTF-8"?>
- <xsl:stylesheet xmlns:m1="http://www.iec.ch/TC57/2010/schema/message" xmlns:m="http://www.ie
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:max="http://www.ibm.com/maximo" ver
    <xsl:output omit-xml-declaration="no" method="xml"/>
    <xsl:strip-space elements="*"/>
  - <xsl:template match="/">
    - <xsl:if test="max:PublishPLUSDWO">
      - <xsl:element name="m:ChangedWorkRequest">
        - <xsl:element name="m:Header">
            <xsl:element name="m1:Noun">WorkRequest</xsl:element>
            <xsl:element name="m1:Source">Maximo</xsl:element>
          - <xsl:element name="m1:MessageID">
              <xsl:value-of select="max:PublishPLUSDWO/@messageID"/>
            </xsl:element>
            ⋮
        </xsl:element>
      - <xsl:element name="m:Payload">
        - <xsl:if test="max:PublishPLUSDWO/max:PLUSDWOSet/max:WORKORDER">
          - <xsl:for-each select="max:PublishPLUSDWO/max:PLUSDWOSet/max:WORKORDER">
              <xsl:apply-templates select="."/>
            </xsl:for-each>
          </xsl:if>
        </xsl:element>
      </xsl:element>
    </xsl:if>
  </xsl:template>
- <xsl:template match="max:WORKORDER">
  - <xsl:element name="p:WorkRequest">
    - <xsl:element name="p:Organisation">
      - <xsl:element name="p:mRID">
          <xsl:value-of select="max:SITEID"/>
        </xsl:element>
      </xsl:element>
    - <xsl:element name="p:Work">
      - <xsl:element name="p:mRID">
          <xsl:value-of select="max:WONUM"/>
        </xsl:element>
      - <xsl:element name="p:kind">
          <xsl:value-of select="max:WORKTYPE"/>
        </xsl:element>
```

**Figure 9 Outbound XSLT Sample**

This is just a small view at the capabilities of the Maximo integration framework to integrate with external systems. The implementation route used for the CIM IOP may or may not suite all the requirements of a full scale implementation. However, combining the ease of configuration via the Maximo web interface along with the flexibility and capabilities of the integration framework will allow utilities to meet their integration scenarios and requirements. More detailed information can be obtained from the Maximo information center and referenced material. There are also sample Java classes or XSL files that can be downloaded from the IBM Service Management (ISM) community.

## 6. CIM IOP – WEBSPHERE

The IBM WebSphere BPM product suite provides a flexible and robust solution for integrating applications and human tasks. Included in the IBM WebSphere BPM product family is WebSphere ESB. As shown in Figure 5 below, the UCA IOP test topology included an Enterprise Service Bus (ESB) to provide a central connectivity solution for the IOP participating applications.

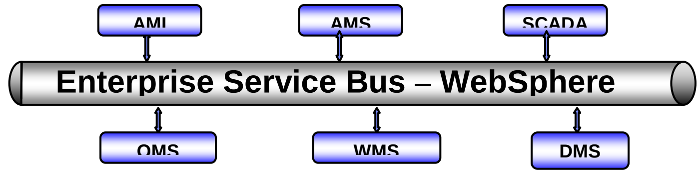


**Figure 5 - UCA IOP Test Topology**

During the IOP tests last March IBM's WebSphere ESB integrated Energy and Utility applications exchanging IEC 61968 Part 3 (Interfaces for Network Operations) and IEC 61968 Part6 (Interfaces for Maintenance and Construction) messages.

### Why Enterprise Service Bus

ESBs are the means by which SOA-based organizations connect service consumers with service providers, enabling them to route service calls, transform messages, mediate between protocols and distribute business events. In fact, ESBs are an excellent tool for helping organizations move closer to some of the broader goals of an SOA, including reuse, visibility, flexibility and business alignment. The ESB is unquestionably an essential component of the SOA. Without it enterprises are faced with issues, including:

***Duplicated connectivity efforts***—Lacking the ability to reuse services in an SOA, IT is perpetually relegated to rewriting code and duplicating transformation efforts.

***One-off and custom connections***—Instead of connecting service providers and consumers through an ESB, applications must be connected point-to-point—and hard-coded.

***Lack of control***—Service connections and transformations, rather than streamlined through an ESB, are haphazard due to the point-to-point connectivity.

### IBM WebSphere ESB

WebSphere ESB manages the flow of messages between service requesters and service providers. Mediation modules within the ESB handle mismatches between requesters and providers, including protocol or interaction-style, interface and quality of service mismatches. In an SCA-based solution mediation modules are a type of SCA module. The mediation modules perform a special role, and therefore have slightly different characteristics from other components that operate at the business level.

WebSphere ESB provides a flexible connectivity infra-

structure for integrating applications and services.
It supports advanced interactions between service endpoints on three levels: broad connectivity, a spectrum of interaction models and qualities of interaction, and mediation capabilities. The product supports connectivity between endpoints through a variety of protocols and application programming interfaces (APIs):

- Java Message Service (JMS) 1.1. Applications can exploit a variety of transports, including TCP/IP, SSL, HTTP, and HTTPS.
- WebSphere ESB supports Web services standards that enable applications to make use of Web service capabilities.
- SOAP/HTTP, SOAP/JMS, WSDL 1.1
- UDDI 3.0 Service Registry, through the WebSphere Integration Developer
- WS-* Standards including WS-Security, WS-Atomic Transactions

WebSphere ESB powers the SOA by reducing the complexity of integrating the applications and services. It integrates seamlessly with existing SOA and BPM platforms.

Along side WebSphere ESB is the WebSphere Registry and Repository promoting service reuse with control. And finally, WebSphere Integration Developer providing developer's with a visual and pattern-based mediation authoring tool.

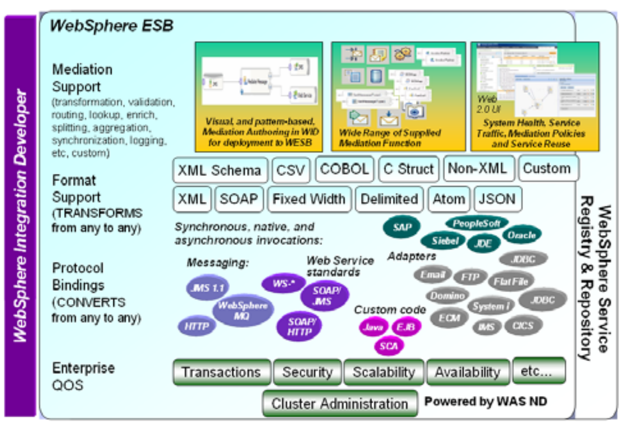


**Figure 6 - WebSphere Enterprise Service Bus**

**IEC CIM Industry Standards Data Models**

There has been an increasing need for Energy and Utility companies to exchange information on a regular basis between applications inside and outside the enterprise. The use of proprietary data formats creates the need to develop complex adapters and wrappers to transform the data exchanged between applications. For these reasons, utility companies and their software vendors are investigating the use of the International Electrotechnical Commission (IEC) Common Information Model (CIM) to manage the complexity of application interoperability.

The adoption of the CIM Data Models simplifies the interoperability between diverse applications and benefits future use of business rules, security, and overall service governance. Utility companies are defining a gradual and controlled CIM adoption roadmap and using an ESB as a single connectivity point.

To ensure compliance with the CIM standard, utility software vendors and utility companies have to be certified by the IEC CIM working group by participating in a semi-annual interoperability test.

In March 2011, one of the interoperability scenarios included the integration of multiple DMS systems with IBM Tivoli Maximo. As an example of the UCA IOP test case scenarios, a DMS system sent a request to IBM Tivoli Maximo to create a new work order based on a transformer asset alarm. The work request, as it traveled through the WebSphere ESB, was shown to the UCA IOP test witnesses using the WebSphere ESB Common Event Infrastructure capabilities. The response from IBM Tivoli Maximo was sent to the DMS system via WebSphere ESB and shown to the test witnesses.

The UCA IOP test witnesses were able to review IEC 61968-6 Work request and response messages in WebSphere ESB to ensure both DMS and WMS were CIM compliant.

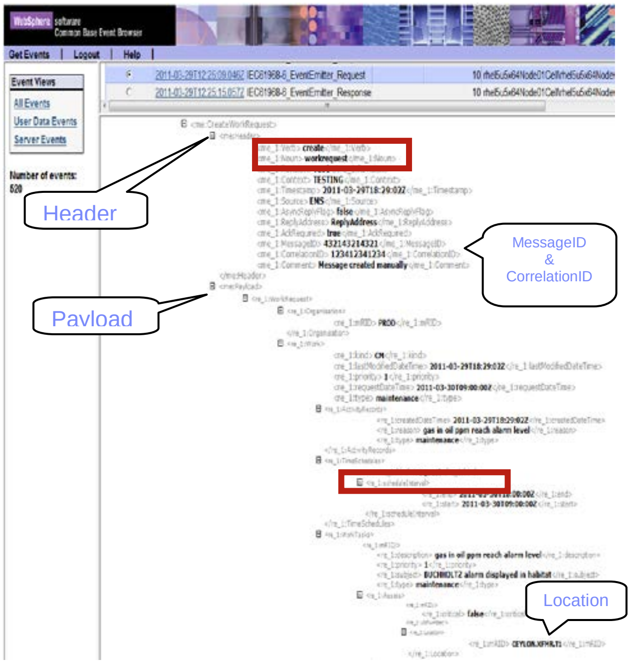


**Figure 7 - IEC 61968 - 6 - "WorkRequest" Message**

In summary, energy and utility companies are designing their integration topology to provide flexibility and an easy interoperability between systems based on an ESB connectivity pattern. Energy and Utilities companies can achieve efficiencies in skills, cost, and time-to-value across middleware products by adding WebSphere ESB to their standards-based IT environment.

## 7. CONCLUSION

An interesting two way relationship has developed between standards and smart grid: *standards are enabling smart grids* and *smart grids are the catalysts for standards* (and their acceleration to maturity). Smart grid requires repeatable and re-usable architecture to achieve interoperability of the large number of its components. CIM architecture draws from other architectural concepts including Open-EDI, UN/CEFACT and formal Zachman enterprise architecture framework and offers the integration solution to smart grid. In the CIM acceleration to maturity, interoperability tests are almost inescapable. Not only does this successful testing move the industry towards the notion that these 61968 standardized messages are complete, correct, and ready for real world use, but it also serves to bring multiple key parties together to define business processes, the appropriate test context, and potentially identify issues, updates, or corrections that can be fed back into the standard. This test verified that standard interpretation is shared making the implementation of each participating vendor work with the others in the network, showed the compliance of the implementation and therefore provided utilities with a selection guide to a number of suppliers.

## 8. FURTHER DEVELOPMENTS

Although CIM Interoperability test in March this year was a great success and demonstrated how multiple vendors can interoperate using CIM compliant integration, the test covered only limited functionality for which the standard messages were developed. The challenge in front of the industry is to develop more comprehensive CIM compliant messages that will cover majority of the utility requirements for integration. This and the follow up interoperability testing events to test and confirm developed messages are the major activities required in the wider implementation of CIM compliant integration for smart grid.


### Acknowledgements

March CIM IOP was run by CIMUg and hosted by EDF in Paris. Margaret Goodrich from SISCO has led the organization and implementation of this IOP.


### References


[1] Working Group 14 of Technical Committee 57, “61968-6_MaintenanceConstruction v14”, Draft Standard IEC 61968-6, July 2009

[2] Electric Power Research Institute (EPRI), ”Report to NIST on the Smart Grid Interoperability Standards Roadmap”, June 2009

[3] David Becker, Terrence L. Saxton, “The Missing Piece in Achieving Interoperability – a Common Information Model (CIM)-Based Semantic Model”, Grid Interop Forum 2007

[4] GridWise Architecture Council Interoperability Framework Team, “Interoperability Context-Setting Framework”, GridWise Architecture Council, July 2007 http://www.gridwiseac.org/pdfs/interopframework_v1.pdf .

[5] Working Group 13 of Technical Committee 57, Energy Management System Application Program Interface (EMS API) – Common Information Model (CIM), International Standard IEC 61970-301, Geneva, Switzerland, International Electrotechnical Commission, 2003.

[6] Working Group 14 of Technical Committee 57, *System*

- http://www-01.ibm.com/software/integration/wps/library/

- www.ibm.com/software/integration/wsesb - IBM's official WebSphere ESB product website
- http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.websphere.wesb.doc/doc/welcome_wps_ovw.html - The one stop guide for latest WebSphere ESB documentation.
- http://www.ibm.com/developerworks/websphere/zones/businessintegration/wesb.html - WebSphere ESB developer community portal
- http://www.ibm.com/developerworks/forums/forum.jspa?forumID=1672&start=0 – An interactive forum for the entire WebSphere ESB community

*Interfaces for Distribution Management – Part 1: Interface Architecture and General Requirements,* International Standard IEC 61968-1, Geneva, Switzerland, International Electrotechnical Commission, 2002.

[7] Working Group 14 of Technical Committee 57, *System Interfaces for Distribution Management – Part 11: Distribution Information Exchange Model,* International Standard IEC 61968-11, Geneva, Switzerland, International Electrotechnical Commission, 2002

[8] "*Maximo 7.1 Integration Guide".* 2007, 2008. IBM Corporation. 6 Nov 2011. http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.mam.doc_7.1/pdf/mam71_integration_guide.pdf

[9] Vander Weele, Barbara. *"IBM White Paper: IBM Maximo 7.1 Integration Framework Configuration Basics"*.Jan 2008. IBM Corporation. 6 Nov 2011. http://www.ibm.com/developerworks/wikis/download/attachments/105054383/Max_IF_ConfigWP.pdf?version=1

[10] Maximo Asset Management Information Center. Home page. IBM Corporation, 2008. Web. 6 Nov. 2011. http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.mam.doc_7.1/mam_welcome.htm

## Biographies

**Nada Reinprecht** is an IT Architect and Industry Consultant of IBM with an expertise in standards based utility integration. She leads development of 61968-6 Common Information Model (CIM) extensions for Work Management in IEC Technical Committee 57, WG14. Nada continues to be a member of United Nations Centre for Trade Facilitation and Electronic Business where she has been developing eCommerce standards. While working in the Australian Energy Market Operator, her main focus was development of aseXML (a Standard for Energy transactions in XML) for integration of Australian energy markets. Her expertise comprises of Smart Grid including Smart Metering, Electricity Markets, Network Operations and Enterprise Asset and Work Management. She has BSc in Electrical Engineering and Graduate diploma in Computer Science.

**Javier Torres** is an advisory software engineer on the IBM software group standards team. He serves as the team focal point for energy & utilities industry standards, advocating for the use of standards across IBM software and solutions. He participates in various industry groups such as CIMug, OpenSG, NIST SGIP and others. He holds an undergraduate degree in Computer Engineering and a Graduate degree in Computer Science.

**Marilza Maia** is a Senior IT Specialist working in the IBM Software and Service for WebSphere organization focusing in Telco and Energy and Utilities industry solutions. She led IBM's participation in the UCA International (Utilities Communication Architecture International) interoperability tests in March and November 2011. She has a Graduate diploma in Statistics and Computer Science.