A Proposed Service Mapping between the MultiSpeak® Specification and IEC 61968-9

Gary A. McNaughton, P.E.
MultiSpeak Project Technical Coordinator
Cornice Engineering, Inc.
PO Box 155
Grand Canyon, AZ 86023
gmcnaughton@corniceengineering.com

Keywords: Interoperability, MultiSpeak®, CIM, enterprise integration, semantic mapping

Abstract

There are two standards for enterprise integration in electric utilities: (i) the MultiSpeak® Specification, which is sponsored by the National Rural Electric Cooperative Association (NRECA) along with the MultiSpeak Initiative, a group of software vendors and utilities that support the goals of adopting and extending MultiSpeak, and (ii) the Common Information Model (CIM), which is an international standard maintained by Technical Committee 57 (TC57) of the International Electrotechnical Commission (IEC).

The MultiSpeak Specification is mature in its coverage of 25 software interface profiles including meter reading, connect/disconnect, meter data management, outage detection, prepayment metering, and demand response. MultiSpeak provides similar capabilities to those under development for the IEC 61968 distribution extensions to the Common Information Model (CIM).

IEC 61968-9, entitled "System Interfaces For Distribution Management – Part 9: Interface Standard for Meter Reading and Control", is the most recent release in the IEC 61968 family of standards. It gives guidance on the integration of meter reading, connect/disconnect, meter data management, outage detection, prepayment metering, and demand response systems with other enterprise systems. IEC 61968-9 has recently been issued as an international standard and currently is undergoing its first interoperability test.

A number of papers have presented general comparisons of the two standards, but until now it has not been possible to perform a detailed comparison of corresponding profiles of the two standards for integrating metering and control applications. This paper investigates the correspondence between the two standards in both data model and service definition. Furthermore, the paper illustrates portions of a mapping that can be used to make an electronic transformation between MultiSpeak Version 4.0.0, dated February 1, 2009, and the version of CIM that will be used for the 2009 CIM interoperability tests (61970cim14v04/61968cim10v21), dated March 1, 2009. The methodology used in this sample mapping can be extended to facilitate interoperability between MultiSpeak-enabled applications and those that support IEC 61968-9 with the goal of achieving an integrated system using applications that support the different standards.

1. BACKGROUND

Utilities seek to rapidly deploy smart grid technologies, but still wish to do so in a manner that minimizes customization and preserves future options. These goals require the use of a standards-based approach. Challenges exist, however, in that the appropriate standards are often incomplete or insufficiently comprehensive.

The National Institute of Standards and Technology (NIST) has issued its draft Framework and Roadmap for Smart Grid Interoperability Standards [1] that (i) gives guidance on the choice of standards for smart grid implementations, (ii) assesses the maturity of those standards and (iii) establishes a framework for moving the industry towards a unified, comprehensive set of interoperability standards. In the area of enterprise integration, NIST has chosen two standards for adoption and harmonization, the MultiSpeak Specification [2] and the Common Information Model (CIM), which has been developed and is maintained by Technical Committee 57 (TC57) of the International Electrotechnical Commission (IEC). As CIM is completed, it is documented in the family of IEC 61970/61968 standards [3] [4] [5]. Approaches to achieve interoperation of systems implementing the two standards were presented in [6] and [7].

Both MultiSpeak and IEC 61968 CIM focus on layers 4 (Semantic Understanding) and 5 (Business Context) of the GridWise Architecture Council's *GridWise® Interoperability Context-Setting Framework* [8]. Furthermore, there is considerable overlap in the areas of enterprise business function covered by the two standards. CIM has been targeted to larger investor owned utilities which typically have extensive information technology

staffs and have more complex IT environments, complete with legacy messaging middleware infrastructures. In contrast, MultiSpeak originally was developed for the electric cooperatives; smaller utilities that typically have limited IT staffs and rarely have messaging middleware of any kind. As a result, the CIM standards focus on message definition and content, leaving much of the transport and middleware as implementation issues. Such an approach would be inappropriate for smaller utilities and for the software vendors serving the cooperative market - many of which also have limited resources. As a result, MultiSpeak standardizes on web services as a means to transport data and does not assume the existence of a messaging middleware infrastructure. The fact that MultiSpeak has been designed to work even in the absence of the services offered by messaging middleware implies that the message exchange protocol must deal with issues that CIM interfaces do not need to address, such as message order and guaranteed delivery. Despite these additional features, there is nothing about MultiSpeak that restricts it to use in small utilities or IT environments that lack middleware.

The latest international standard in the CIM family is IEC 61968-9 (Part 9) [9], which deals with meter reading and control interfaces. There is a considerable overlap between the capabilities of Part 9 and MultiSpeak.

MultiSpeak has extensive capabilities for (i) meter reading (as part of its MR service interface), (ii) remote control of connect/disconnect devices (in its CD service interface), (iii) detection of customer outages (in its OD service interface), (iv) meter data management (in its MDM service interface), (v) end device inventory receiving and testing (in its EDTR service interface) and (vi) pre-paid metering (in its PPM service interface).

Part 9 supports (i) meter reading, (ii) end device events (such as outage detection or meter health notifications, (iii) synchronization between systems (such as meter asset management, inventory or testing), (iv) end device controls (such as control of connect/disconnect devices or meter demand reset), and (v) meter service requests (installing and removing meters).

The mapping will be discussed in four parts: (i) message patterns, (ii) message header information, (iii) an illustration of mapping between CIM messages and MultiSpeak methods and (iv) data payload.

2. MAPPING MESSAGE PATTERNS

The first step in developing a mapping between MultiSpeak and Part 9 is to consider how the two standards create messages. The CIM supports a verb/noun/payload message pattern. The verbs that may be used in CIM messages are shown in Table A.1 in Annex A of Part 9 [8]. The verb indicates the action that is expected to be taken on the data carried in the message payload. The noun describes the type of object or data that is to be included in the payload. The verb and noun are specified, along with message control information, in the message header. The payload is the actual instance data to be exchanged. The means of transferring these messages is outside the scope of Part 9. It is the intention of TC57 that there will be a series of message exchange profile documents in the 61968-1-x series to describe different ways to carry messages such as a message queue or over an enterprise service bus, perhaps using web services. This separation of message definition and transport is intended to permit utilities with disparate messaging infrastructures to all use a similar message definition.

MultiSpeak originally supported a similar approach in its Version 2.2, which was issued in 2003. A verb/noun/payload messaging framework was developed that closely parallels that used by CIM today; this approach to messaging is still available for use today with MultiSpeak Versions 3 and 4. However, since small electric utilities rarely have extensive messaging infrastructure, it was decided to move to a web services approach, in addition to the messaging framework, during the transition from Version 2 to Version 3. The use of web services has been found to facilitate ease of implementation and more dependable interoperation among systems provided by different vendors.

MultiSpeak web services use a standardized verb/noun/modifier naming convention that parallels that used in CIM. For instance, a CIM message to request customer data for a specific customer might be called GET(Customer) with a payload that contains a customer object and a qualifying identifier for the customer instance of interest. The equivalent web service method in MultiSpeak would be GetCustomerByCustID. In this method, “Get” is the verb and “Customer” is the noun type of interest. Furthermore, “ByCustID” is the modifier that indicates that a specific instance of customer data is being requested; which customer is of interest is specified by a customer identifier included in the request. The customer data instance is carried in the response payload. Although the verb and noun mapping between the two standards is not always this transparent, it is possible to map between CIM verbs and MultiSpeak verbs and between CIM nouns and MultiSpeak nouns. A complete mapping of CIM verbs with MultiSpeak verbs is outside the scope of this paper. Similarly, it is possible to map between the CIM class in a CIM message payload and the MultiSpeak object in a web service method call, although the mapping is sometimes complex.

3. MAPPING MESSAGE HEADERS

The next step in developing a mapping between CIM and MultiSpeak is to consider how the two message headers correspond. Figure 1 shows a mapping between the MultiSpeak message header (left) and the CIM message header that was suggested in a draft of IEC 61968-1 and which is used in Part 9 (right).

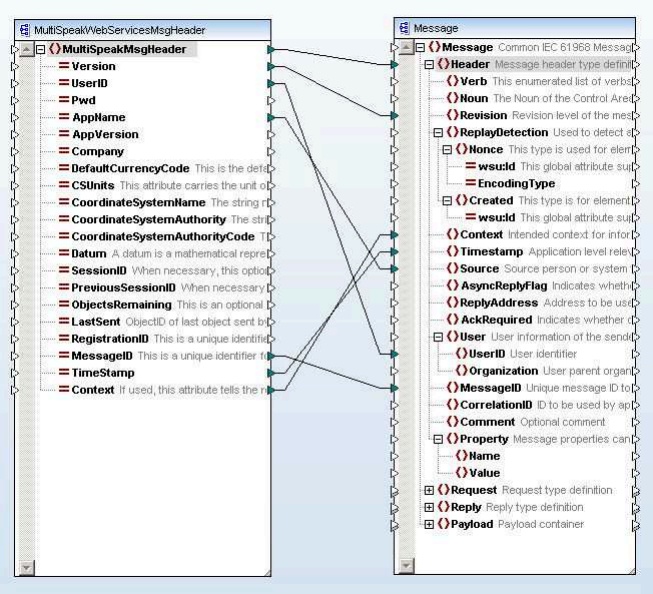

Figure 1. Message Header Mapping

It should be noted that, since CIM is transport-independent, and the information included here as message header fields might be specified elsewhere in the transport stack, the use of any of these message header fields is optional in CIM. At first glance, it does not appear that there is much common to the respective message headers, but a more detailed investigation shows that they are more alike than not. Table 1 shows the CIM header information for which there is no obvious corresponding item in the MultiSpeak header. Table 2 shows the MultiSpeak header information for which there is no obvious CIM equivalent.

Table 1
CIM Message Header Features Apparently Missing in MultiSpeak Message Header

CIM Message Header Feature	MultiSpeak Correspondence
Verb and **Noun**	The verb and noun are included in the web service method name.
ReplayDetection	CIM makes use of WS-Security replay detection to foil replay attacks. MultiSpeak does not explicitly use WS-Security, but the MessageID and TimeStamp can be used for this purpose.
AsyncReplyFlag	This specifies whether the reply should be asynchronous or not. This is not necessary in MultiSpeak; it is inherent in the service definitions.
ReplyAddress	The address for an asynchronous reply to be sent. This is included as the "returnURL" in MultiSpeak method definitions, where appropriate.
ACKRequired	This feature determines whether an acknowledgement is required. This is not necessary; it is inherent in the MultiSpeak service definition.
User.Organization	No equivalent in MultiSpeak messages.
CorrelationID	This is used to correlate a reply with a request. This is called "transactionID" in MultiSpeak and is included in the method definition.
Comment	No equivalent in MultiSpeak messages.
Property	Message properties to be used for extended routing. Not necessary in MultiSpeak.
Request/Reply	Message parameters used to define requests or replies. Inherent in MultiSpeak method definitions

Table 2
MultiSpeak Message Header Features Apparently Missing in CIM Message Header

MultiSpeak Message Header Feature	CIM Correspondence
Pwd	A password, used with UserID for user authentication. Outside the scope of Part 9.
AppName, AppVersion, Company	Identifiers of the software application that is sending the method call. No equivalent in CIM.
DefaultCurrencyCode	Used to establish a default currency code so that each currency instance does not need to be identified by type. Exceptions to the default are identified in the payload. No equivalent in CIM
CSUnits, CoordinateSystemName, CoordinateSystemAuthority, CoordinateSystemAuthorityCode, Datum	Used to pass information about a GIS coordinate system. Outside the scope of Part 9.
SessionID, PreviousSessionID	Used to tightly link systems by establishing data exchange sessions. No equivalent in CIM.
ObjectsRemaining, LastSent	Used with protocol features to send unmanageably large sets of data in smaller groups. No equivalent in CIM.
RegistrationID	Used with protocol features to exchange information about subscriptions. Outside the scope of Part 9.

An examination of Tables 1 and 2 indicates that the items that appear to be missing in one of the standards in Figure 1 are either optional or not required because of the differences in messaging technology chosen.

4. CIM MESSAGES AND MULTISPEAK METHODS

The first two steps in the mapping addressed message construction and message header information. It is time to turn to determining how specific Part 9 messages map to the corresponding MultiSpeak web service methods.

Table 3 shows the correspondence between selected message types defined in Part 9 and the equivalent services in MultiSpeak. The first column in Table 3 is the message type as defined in Part 9. The number in curly brackets ({XX}) refers to the number of the figure in Part 9 that contains the sequence diagram for the specific message definition. The second column lists the initiating message in the business process identified in column 1. Column 3 shows which component in the CIM reference model issued the message identified in column 2; acronyms are expanded in the table in the Notes section. The fourth column identifies the corresponding MultiSpeak web service method. The fifth column shows the MultiSpeak functions that publish the data included in the corresponding CIM message shown in column 2 and the function that subscribes to those data. Acronyms for the publishing and subscribing functions are expanded in the Notes section of Table 3.

Table 3
Correspondence between Selected IEC 61968-9 Message Types and MultiSpeak Service Interfaces

IEC 61968-9 Message Type (See Notes 1,2,3)	IEC61968-9 Sample Message	IEC61968-9 Reference Model Component Issuing Message	MultiSpeak Sample Web Service Method Call	MultiSpeak Service Interfaces Publishing/Subscribing
End device event				
• Outage detection (req/reply) {4, 5}	• GET (EndDeviceEvents) {4} • CREATE (OutageDetection) {5}	• OMS • OMS	• InitiateOutageDetectionEventRequest • InitiateODMonitoringRequestByObject	• OA/OD • OA/OD
• Outage detection (pub/sub) {6}	• CREATED (EndDeviceEvents)	• MS	• ODEventNotification	• OD/OA
• Meter health event (pub/sub) {9}	• CREATED (EndDeviceEvents)	• MS	• HistoryLogChangedNotification or MeterEventNotification	• MR/CB or MR/MDM
Synchronization between systems				
• Meter test [15}	• UPDATE(EndDeviceAssets)	• WMS	• MeterTestTransaction	• EDTR/CB
Meter reading				
• On-request meter read {25}	• CREATED(MeterReading)	• MS	• ReadingChangedNotification	• MR/CB
End device control				
• Remote connect/ disconnect {35}	• CREATE(EndDeviceControls)	• CIS	• InitiateConnectDisconnect	• CB/CD
Customer				
• Exchange customer account data {54}	• CREATE(CustomerAccountConfig)	• CIS	• AccountChangedNotification	• CB/MR
Meters				
• Exchange meter data {64}	• CREATE(MeterAssetConfig)	• CIS	• MeterChangedNotification	• CB/MR

Notes:
1) Req/reply – Request/reply message exchange pattern.
2) Pub/sub - Publish/subscribe message exchange pattern.
3) {XX} – Reference to the figure number in the Part 9 document.

CIM Reference Model Components
CIS – Customer Information System
MS – Metering System
OMS – Outage Management System
WMS – Work Management System

MultiSpeak Abstract Software Functions
CB – Customer Billing (CIS)
CD – Connect/Disconnect
EDTR – End Device Receiving and Testing
OA – Outage Analysis (OMS)
OD – Outage Detection
MDM – Meter Data Management
MR – Meter Reading

5. PAYLOAD MAPPING – METER READING

The last topic to discuss in determining the correspondence between Part 9 and MultiSpeak is the payload mapping. Space permits the discussion of only one payload type; the payload of most interest to developers is the meter reading. Meter readings may be of many kinds from simple residential monthly energy readings to complex multi-part interval readings. The CIM MeterReading class is intended to carry all kinds of meter reading data. In MultiSpeak the equivalent general-purpose meter reading object is the meterReading. In addition, MultiSpeak may also carry metered data in several types of specialized objects, including a formattedBlock (for carrying large blocks of XML-formatted data in a format similar to that in a comma-separated values file) and an intervalDataBlock, which is a specialization of the formattedBlock to more efficiently carry interval data.

For the sample payload mapping, a simple monthly energy meter reading will be considered. Hence the mapping will compare a MultiSpeak meterReading object and the payload from a CIM MeterReadings message, which carries a CIM MeterReadings class. Figure 2 shows an initial mapping between the two artifacts.

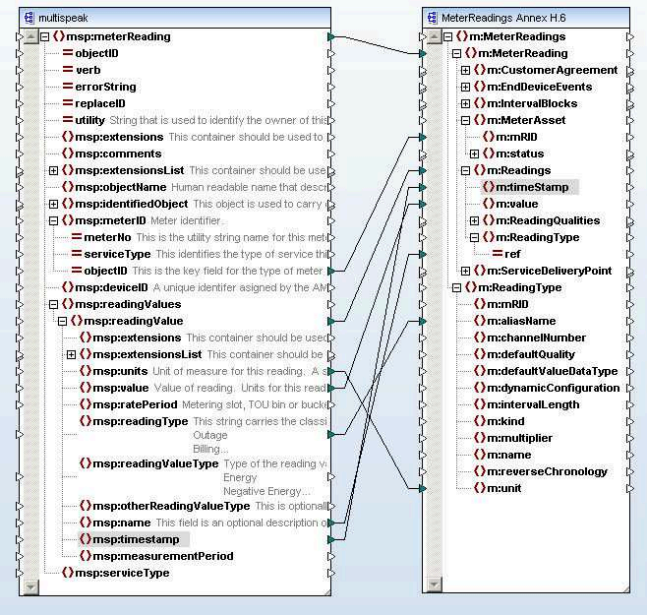

Figure 2. Mapping Between MultiSpeak meterReading and Part 9 MeterReading

Again, it appears on initial inspection that there are substantial portions of the two standard payloads that do not have corresponding components in the other standard. The MultiSpeak object carries information that is not included in a CIM Readings and the CIM MeterReadings has several additional classes that do not appear in MultiSpeak. However, a deeper analysis reveals that the two standards are closer than first believed.

The CIM message includes optional links to associated CIM CustomerAgreement, EndDeviceEvents and IntervalBlock classes, which have no direct equivalent in the MultiSpeak meterReading, however their functions are handled in other ways in MultiSpeak. For instance, there is no explicit concept of a customer agreement in MultiSpeak. Its closest match is the concept of rate class or rate code, which is associated in MultiSpeak with a service location, not a meter or meter reading. The CIM MeterReading can be used for a number of different purposes, some of which require an EndDeviceEvent or IntervalBlock. In the case where it is desired to send a CIM EndDeviceEvent, a MultiSpeak meterEvent would be used. Similarly, if interval data is to be sent, a MultiSpeak IntervalDataBlock or formattedBlock object would be used, not a meterReading.

Note, in Figure 2, that the CIM MeterReadings class contains an array of CIM MeterReading instances and an array of CIM ReadingType instances. The ReadingType class is used to pass information about a specific class of meter reading; each MeterReading then references the ReadingType class by its identifier (ReadingType.mrID). The composition of a ReadingType.mrID is described in detail in IEC 61968-9, Annex C.

Historically, MultiSpeak has sent sufficient information in the meter reading data so that a ReadingType description was not necessary. Thus, information like units of measure ("units") has been carried in the MultiSpeak meterReading, rather than being carried in a separate metadata object. For purposes of future harmonization, MultiSpeak has recently added a readingType object, identical in content to the CIM ReadingType class, and with a readingType.objectID that is matches the guidelines in 61968-9, Annex C for a ReadingType.mrID. Similarly, versions of MultiSpeak beyond V4.0.0 will add a ReadingTypeID element to the meterReading that can be used as a pointer to such readingType data in the same way as the CIM ReadingType.mrID carried in the MeterReading points to an instance of a CIM ReadingType.

Table 4 shows the detailed mapping of the base MultiSpeak meterReading using CIM classes. Table 5 details the MultiSpeak meterReading.meterID element, which identifies the meter used to collect the reading. Table 6 shows the details of the meterReading.readingValues element in MultiSpeak.

Table 4
Mapping of MultiSpeak meterReading Object into CIM Classes

MultiSpeak Parent Element	MultiSpeak Element/Attribute Name	Description	61968-9 Message Element	CIM Class.CIM Attribute
meterReading				
	objectID	Unique identifier for this instance of a meter reading. Inherited from mspObject base class	Same function is addressed in CIM IdentifiedObject.	MeterReading.mrID
	verb	This attribute gives the option to specify additional actions to be applied to this reading. Inherited from mspObject base class	Carried in verb portion of message header.	
	errorString	This attribute gives the option to return to the publisher error information about this reading. Inherited from mspObject base class	Outside the scope of Part 9.	
	replaceID	This attribute links an installation and removal transaction. Not applicable to a meter reading. Inherited from mspObject base class	No equivalent in CIM.	
	utility	Identifies the utility that generated this reading. Inherited from mspObject base class	Outside the scope of Part 9.	
	extensions	This element gives the capability to extend this object. Inherited from mspObject base class	Outside the scope of Part 9.	
	comment	Inherited from mspObject base class.	Not applicable.	
	extensionsList	This element gives the capability to extend this object. Inherited from mspObject base class	Outside the scope of Part 9.	
	objectName	A means to give a human-readable name to this object. Not applicable to a meter reading. Inherited from mspObject base class	Included in CIM IdentifiedObject.	MeterReading.name
	IdentifiedObject	This is a set of CIM naming fields that is carried in MultiSpeak for harmonization purposes. Inherited from mspObject base class	Naming fields inherited by MeterReading from IdentifiedObject.	
	meterID	Identifier for a meter. See Table 5 for detailed mapping.		
	deviceID	A unique identifier assigned by the AMI vendor. For instance, this might be a transponderID or endPointID.	Not in Part 9, but in CIM.	ComFunction.amrAddress
	readingValues	Carries the actual meter reading values. See the detailed mapping in Table 6.		
	serviceType	An enumeration of the type of utility service that this reading reflects. For instance, a meter reading for electric service would have serviceType = “Electric”.	Not in Part 9, but in CIM.	Service.kind

Table 5
Mapping of MultiSpeak meterReading.meterID Object into CIM Classes

MultiSpeak Parent Element	MultiSpeak Element/Attribute Name	Description	61968-9 Message Element	CIM Class.CIM Attribute
meterID				
	meterNo	This is the utility string name for this meter.	Not in Part 9, but in CIM.	MeterAsset.name
	serviceType	As described in Table 4, now applied to the meter.	Not in Part 9, but in CIM.	Service.kind
	objectID	Unique identifier for this instance of a meter. Note that if serviceType= ”Electric” and this objectID is”12345”, then this meterID refers to the instance of an electricMeter with objectID=”12345”.		MeterAsset.mrID

Table 6
Mapping of MultiSpeak meterReading.readingValues Object into CIM Classes

MultiSpeak Parent Element	MultiSpeak Element/Attribute Name	Description	61968-9 Message Element	CIM Class.CIM Attribute
readingValues				
	extensions	This element gives the capability to extend this object.	Outside the scope of Part 9.	
	extensionsList	This element gives the capability to extend this object.	Outside the scope of Part 9.	
	units	Unit of measure for the reading. A suggested enumeration is carried in the "uom" object, which comes from ANSI C12.19.	Included in ReadingType.mrID. 61968-9, Annex C suggests using field #11 to carry unit of measure.	
	value	Value of the reading. Units for this reading are contained in the "units" element.	This information is carried in MeterReading.Readings.Value.	
	ratePeriod	Metering slot, TOU bin or bucket.	Included in ReadingType.mrID. 61968-9, Annex C suggests using field #8 of the mrID to carry the measurement category.	
	readingType	An optional string field that carries the classification of the reading. Enumerations include: "Energy", Negative Energy", Current Demand", Max Demand", etc.	Included in ReadingType.mrID. 61968-9, Annex C suggests using the combination of fields #5 and 6 of the mrID to carry the UOMCategory.	Maps most closely to ReadingType.name.
	otherReadingType	This is optionally added as a means to explicitly incorporate readingValueType values not included above. One usage of this field would be to send values where readingValueType is set above to "Other".	Not directly supported.	
	name	This field is an optional description of the type of reading contained in this object.	Not applicable.	
	timestamp	The time the reading was taken.	Readings.TimeStamp.	
	measurementPeriod	The period over which a peak or an accumulated value was measured. The enumerations include "Current", "Previous", and "Previous Season".	Included in ReadingType.mrID. 61968-9, Annex C suggests using field #1 of the mrID to carry the time attribute .Could also be described by populating the Readings. Timestamp with ISO8601 time period elements.	

6. CONCLUSIONS

MultiSpeak and the CIM are both being used by utilities and their vendors for inter-application integration. Hundreds of utilities and their software vendors now use MultiSpeak to integrate automated metering infrastructure systems. With the issuance of IEC 61968-9 as an international standard, utilities and their vendors have been given guidance on how to use the CIM to perform similar integrations. In order to avoid redundancy and lack of integration between implementations of the two standards, it will be necessary for the industry to develop a means to translate between these two standard solutions in an automatic and comprehensive manner.

This paper outlines a process by which a mapping can occur and gives an example of such a mapping for one use case, monthly residential meter reading. Other mappings need to be completed before a comprehensive translation can be developed between IEC 61968-9 and the corresponding parts of MultiSpeak. Although the complete mapping will require substantial effort, it is achievable. Both the CIM and MultiSpeak communities will benefit from the completion of this work, and both should work together to accomplish the goal. One venue to complete and document the mapping work is the Part 14 team of TC57, Working Group 14. The author wants to take this opportunity to encourage feedback and participation in this process by any interested parties.

References

[1] National Institute of Standards and Technology, *NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0 (Draft)*, http://www.nist.gov/public_affairs/releases/smartgrid_interoperability.pdf , September 2009.

[2] MultiSpeak Initiative Participants, *MultiSpeak ® Version 4.0 Data Schema and Service Definitions*, Arlington, VA: National Rural Electric Cooperative Association, 2009.

[3] 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.

[4] Working Group 14 of Technical Committee 57, *System Interfaces for Distribution Management – Part 1: Interface Architecture and General Requirements,* International Standard IEC 61968-1, Geneva, Switzerland, International Electrotechnical Commission, 2002.

[5] 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.

[6] McNaughton, Gary A. and Robert Saint, "How NRECA's MultiSpeak® Specification Supports Interoperability of Diverse Grid Automation Systems", Proceedings of the 2007 Grid-Interop Forum, November 7-9, 2007, Albuquerque, NM, Gridwise Architecture Council, 2007.

[7] McNaughton, Gary A., Greg Robinson, Gerald R. Gray, "MultiSpeak® and IEC 61968 CIM: Moving Towards Interoperability", Proceedings of the 2008 Grid-Interop Forum, November 11-13, 2008, Atlanta, GA, Gridwise Architecture Council, 2008.

[8] GridWise Architecture Council, *GridWise® Interoperability Context-Setting Framework, Version 1.1,* dated March, 2008, www.gridwiseac.org .

[9] Working Group 14 of Technical Committee 57, *Application integration at electric utilities - System interfaces for distribution management – Part 9: Interface for meter reading and control,* International Standard IEC 61968-9, Ed. 1, Geneva, Switzerland, International Electrotechnical Commission, 2009.

Biography

Gary A. McNaughton is the Vice President and Principal Engineer for Cornice Engineering, Inc. He has a B.S.E.E. degree from Kansas State University and an M.S.E.E. degree from the University of Colorado. Prior to joining Cornice in 1995 he worked as a Plant Electrical Engineer for Union Carbide, at the Oak Ridge Gaseous Diffusion Plant, at Oak Ridge, TN, as a Transmission Planning and Protection Engineer for Colorado-Ute Electric Association, a generation and transmission cooperative, located in Montrose, CO, and as Staff Engineer, Manager of Engineering, and Assistant General Manager for Engineering and Operations for La Plata Electric Association, in Durango, CO. Mr. McNaughton currently serves as the Project Technical Coordinator for NRECA's MultiSpeak® Initiative. Mr. McNaughton also serves on Working Group 14 of IEC TC 57. He assisted in the development of IEC 61968-9 and serves as co-team leader for Part 14 of IEC TC57, WG14, which addresses harmonization of MultiSpeak and CIM. Mr. McNaughton is a registered professional engineer in the State of Colorado.

Acknowledgements.

The author gratefully acknowledges the contribution of Mr. David Haynes, of Aclara Power-Line Systems, who serves as co-team leader of the Part 14 MultiSpeak/CIM harmonization team and as a participant in the Part 9 team. Mr. Haynes also is an active participant in the MultiSpeak Initiative. His assistance has been instrumental in the development of the correlations that led to this paper.