EPRI | ELECTRIC POWER RESEARCH INSTITUTE

Common Information Model (CIM)-MultiSpeak Harmonization 2nd Edition

2012 TECHNICAL REPORT

CIM-MultiSpeak Harmonization 2nd Edition

EPRI Project Manager
G. Gray

3420 Hillview Avenue
Palo Alto, CA 94304-1338
USA

PO Box 10412
Palo Alto, CA 94303-0813
USA

800.313.3774
650.855.2121

askepri@epri.com

www.epri.com

1026585
Final Report, December 2012

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

THE FOLLOWING ORGANIZATION PREPARED THIS REPORT:

Electric Power Research Institute (EPRI)

NOTE

For further information about EPRI, call the EPRI Customer Assistance Center at 800.313.3774 or e-mail askepri@epri.com.

Acknowledgments

The following organization prepared this report:

Electric Power Research Institute (EPRI)
3420 Hillview Avenue
Palo Alto, CA 94304

Principal Investigator
G. Gray

This report describes research sponsored by EPRI. EPRI would like to acknowledge the support of the following organizations:

- Bob Saint, NRECA
- Gary McNaughton, Cornice Engineering
- David Haynes, Aclara

This publication is a corporate document that should be cited in the literature in the following manner:

CIM-MultiSpeak® Harmonization 2nd Edition.
EPRI, Palo Alto, CA: 2012.
1026585.

Product Description

The CIM–MultiSpeak[1] harmonization effort sought to identify the degree of correlation that exists between the International Electrotechnical Commission (IEC) Common Information Model (CIM), which is based on IEC standards 61968, 61970, and 62325, and the MultiSpeak v4.1.5 standard for electric system interoperability. The initial effort focused on mapping attributes from the IEC 61968-9 profiles to equivalent profiles that were created for MultiSpeak v4.1.5. A proof-of-concept interoperability test example using the on-demand read test case was created that leveraged MultiSpeak web services to a CIM-based designed web.

Background

MultiSpeak and IEC CIM are standards that serve the utility application integration domain. However, the standards had different geneses and serve two slightly different constituencies. The CIM is being used by large investor-owned utilities, with an underlying assumption of larger information technology staffs capable of leveraging the information model and molding it to serve their purposes. MultiSpeak tends to serve smaller utilities such as municipals and cooperatives that have smaller information technology staffs. MultiSpeak started as an interface standard and was later reengineered into the unified modeling language. The challenge in the utility domain came when there was a need to integrate two different products or systems that were compliant with the differing standards. There is confusion in the marketplace about which standard to support, how to support both standards, and how to integrate systems based on the different standards, when the standards can take different approaches to solving a problem.

Objectives

The objective of this project was to determine the degree of harmonization that exists between IEC 61968-9 and MultiSpeak 4.1.5.

Approach

An initial attempt was made to analyze both the CIM and MultiSpeak models using an ontology tool such as Protégé. A

[1] MultiSpeak is a registered trademark of the National Rural Electric Cooperative Association.

method was developed using Enterprise Architect to export profiles as a Resource Description Framework Schema for import into an ontology tool and then run an analyzer on the classes to look for relationships.

When the import into the ontology tool met with limited success, a decision was made to map CIM profiles to MultiSpeak profiles. This required a MultiSpeak equivalent of the CIM profiles to be created.

After a MultiSpeak equivalent profile was created as an XSD, mapping could occur. Mapping looked for correlations between classes, attempted to make transformations if possible where correlation did not exist, and documented any gaps where a mapping could not be made. No assessment of mandatory or optional elements was made; all attributes were considered for mapping. Reviews were held as part of an IEC Working Group 14 for the IEC 61968-14 effort with CIM and MultiSpeak experts.

Results

A process was created whereby MultiSpeak profiles could be created. (MultiSpeak was one large model, and no CIM-like equivalent profiles existed.) A process was developed for mapping from CIM to MultiSpeak. UML for the MultiSpeak equivalents were developed such that XSDs could be directly generated from the model.

Correlations, transformations, and gaps between the profiles were documented, with a set of suggested changes for IEC Working Group 14 and the MultiSpeak Technical Committee to consider in future additions of their respective standards.

The mapping provides a guide and suggested solutions for systems integrators who are faced with integration applications or systems that use either standard. Documenting known gaps also allows systems integrators to focus on solutions rather than on finding the gaps in the first place.

The proof-of-concept integration of a well-known use case (on-demand meter reading) demonstrates the practical applicability of integration using an enterprise service bus with mapping not only to the respective meter read payloads but also to those message elements required for reliable systems integration.

Keywords

Common information model (CIM)
Harmonization
International Electrotechnical Commission (IEC)
MultiSpeak
Systems Integration

Abstract

This report reflects the effort to harmonize the International Electroctechnical Committee's (IEC) Common Information Model (CIM) for utilities, which incorporates IEC standards 61968, 61970, and 62325, 1st Edition, with MultiSpeak v4.1.5. Specifically, the harmonization is with the IEC 61968-9 Meter Reading & Control profiles. The objective of the harmonization effort was to determine the degree of correlation that already exists between these complementary models of utility application integration. MultiSpeak based profiles were created and mapped to existing 1st edition CIM profiles with the correlations, data transformations that may be needed, and any gaps in the models respective to the other. This harmonization may be utilized by utilities, systems integrators, and product vendors to better understand how to integrate a CIM-based product or system with a MultiSpeak-based product or system. The first edition of this report covered roughly half of the existing meter reading and control profiles. This edition completes that effort, which includes the remaining profiles and corrections to the initial effort.

Executive Summary

This report details the harmonization of the IEC CIM and MultiSpeak standards. The respective standards serve the same domain but slightly different constituencies. The needs of the differing constituents and the assumptions used to solve problems in the domain have led to differences between the standards. The differing standards lead to confusion in the marketplace and causes challenges for utilities and vendors seeking to integrate systems that may only support one standard or the other. This report includes a description of the initial attempt to port the respective models into an ontology tool for assessment, a proof of concept integration of the On Demand Meter Read use case utilizing both standards, and documentation of the harmonization of IEC 61968-9 profiles with MultiSpeak created equivalents.

This harmonization effort documented the correlations (where the standards align) transformations (where attributes may need to be modified) and gaps (where corresponding attributes, classes, or relationships do not exist) between the respective standards. This documentation can be used as a guide by product vendors and systems integrators to develop adapters that have a consistent interpretation, so any given integration effort should look much the same as the next.

The documentation also includes suggested mappings. While the mappings for any given implementation may vary somewhat based on the business requirements of a customer, e.g. some attributes may be optional and hence the customer may choose not to implement them, the transformations are suggested solutions to common systems integration problems, e.g. concatenating multiple attributes into a single attribute or performing logical functions on an attribute prior to mapping.

The harmonization effort demonstrated where the respective standards are closely aligned and where significant effort may be required in future additions to bring them to closer alignment.

Suggestions for changes based on the identified gaps have been contributed to the respective technical teams that support each standard. These suggestions will need to be vetted by the respective teams for consideration for updates in forthcoming editions of each standard.

As the standards become better aligned it means lowered cost for systems integrations that are based on these standards, eventually lowering costs for ratepayers.

In this update to the initial effort all of the 23 IEC 61968-9 1st Edition profiles were mapped to corresponding MultiSpeak v4.1.5 profiles.

Future work would include mapping the IEC 61968-9 Meter Reading & Control 2nd Edition profiles with the upcoming MultiSpeak (v5) that is currently under development.

Table of Contents

List of Figures

List of Tables

Section 1: Introduction

The Common Information Model (CIM) in the utility domain represents three standards related to utility application integration, IEC 61968, IEC 61970, and IEC 62325. These three standards and correlating documentation share and are derived from a common model. The CIM is an international standard and its users are primarily large utilities. While integration guidance is given in supplemental standards (IEC 61968-1 and IEC 61968-100 CDV), the standards implementation grew out of a presumption of a reference model, with utility staff capable of modifying and determining specific implementation customization. In contrast, MultiSpeak grew out of a need to serve a constituency that had much smaller IT staffs and hence grew out of a need to specify the interfaces between systems. CIM evolved as a reference model with interface specifications added later; MultiSpeak started as an interface specification that was reverse engineered into a reference model. While MultiSpeak is used internationally, it is not an international standard, and its constituent utilities tend to be smaller municipal and cooperative utilities.

With two different standards serving the same domain (utility application integration) while serving the unique needs of their respective constituents, causes problems for utilities, systems integrators, and product vendors as they try to determine how, and which, standards to support. This causes confusion in the marketplace and ultimately raises the costs of integration for all stakeholders.

Harmonizing the respective standards would identify where the standards align and where gaps exist. Harmonization does not seek to resolve the differences in the standard, only identify how they can be used together, usually through a mapping exercise. Once gaps have been identified and the business reasons for understanding the differences in the models, can suggestions then be made to the respective technical committees charged with maintaining the standards. Once the suggested changes have been vetted and included in successive editions of the standards, the costs for integration will gradually decrease.

The assessment will be covered in the following sections. The assessment went through three steps: 1) a determination and test to see if an ontology tool could be used to determine alignment between the models, 2) a mapping exercise to identify correlations, potential transformations, and gaps between the models at a smaller profile level, and 3) a proof of concept implementation where both MultiSpeak services and a service based on IEC 61968-100 CDV (Committee Draft for Vote – one of the steps before the document becomes a standard) were used to implement an On Demand Meter Read use case across an enterprise service bus. This proof of concept is the type of practical guidance that illustrates how a systems integrator would solve a problem of integrating systems based on the two standards.

Section 2: Ontology Assessment

Both the CIM and MultiSpeak are very robust representations of the utility application integration domain. On the surface they both represent entities such as customers, meters, and information related to distribution grid operation.

Ontology - An explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them.[1]

As such both CIM and MultiSpeak are *ontologies*.[2] Both standards are formally specified and have representations in UML. Additionally there are tools available that allows one to create ontologies and assess class relationships. A *reasoner* is used within the tool that examines the classes, attributes, and relationships of the ontology under assessment to determine if classes are equivalent or determine other inferred relationships. As both CIM and MultiSpeak represent ontologies the challenge becomes how to transform the models into something that can be imported into an ontology tool for an assessment.

This is accomplished by using a resource definition framework schema (RDFS). An RDFS is an XML file that uses a subject-predicate-object *triple* to not only define classes and their attributes, but also their relationship with other classes.[3] The important distinction between RDFS and other types of XML is that while XML can be used to create attributes that are self-defining and with meaning understandable by humans, RDFS XML allows a computer to determine meaning of the attributes in the class, attributes, relationship sense.

Both CIM and MultiSpeak have models that are maintained using SparxSystems Enterprise Architect and this product supports the capability to generate and export XSDs. Additionally there are also add-ins available to use either outside the Enterprise Architect environment or that integrate into the environment that allow the user to export snapshots of the model called a *profile*, in an RDFS format. Once the file has been exported into RDFS it can be imported into an ontology tool for assessment.

---

[1]Ontology.(n.d.).The Free On-line Dictionary of Computing. Retrieved October 07, 2011, from Dictionary.com website: http://dictionary.reference.com/browse/ontology

[2]*CIM Primer:First Edition*. EPRI, Palo Alto, CA: 2100. 1024449.

[3]http://www.w3.org/TR/rdf-schema/

In the ontology assessment portions of both the CIM and MultiSpeak models were exported from their respective UML models into an RDF file. The respective RDF files were then loaded into Protégé[4], a freely available Web Ontology Language (OWL) tool for assessment. Once the profiles were loaded into Protégé it became clear that there was an error because all of the classes that had been selected in the profile appeared to be equivalent to each other. There was no class hierarchy. For example, all classes in the CIM inherit from identifiedObject, yet this class was a peer to all of the other classes that had been imported as part of the profile. A similar challenge existed in the import from MultiSpeak. All classes in MultiSpeak inherit from mspObject, yet this class was a peer to all other imported classes.

To resolve this challenge the RDF export files had to be edited to resolve the discrepancies. It was soon revealed that the properties of the classes were also not being imported correctly. It soon became clear that the editing to get a correctly defined RDF file for import was going to be very time consuming. The alternative was to simply replicate the UML in Protégé; again a very time consuming prospect.

It was decided that in the interests of time a more pragmatic approach would be taken, that is, to create MultiSpeak equivalents of the existing CIM profiles and then map these using an XSD mapping tool.

[4] What is Protégé – OWL? http://protege.stanford.edu/overview/protege-owl.html

Section 3: Profile Mapping

CIM and MultiSpeak both use a class that other classes inherit from that also contains a universal identifier that is used throughout the respective models. Before beginning a profile mapping exercise it is important to understand these "high level" classes and their relationship to each other.

Starting at the Top

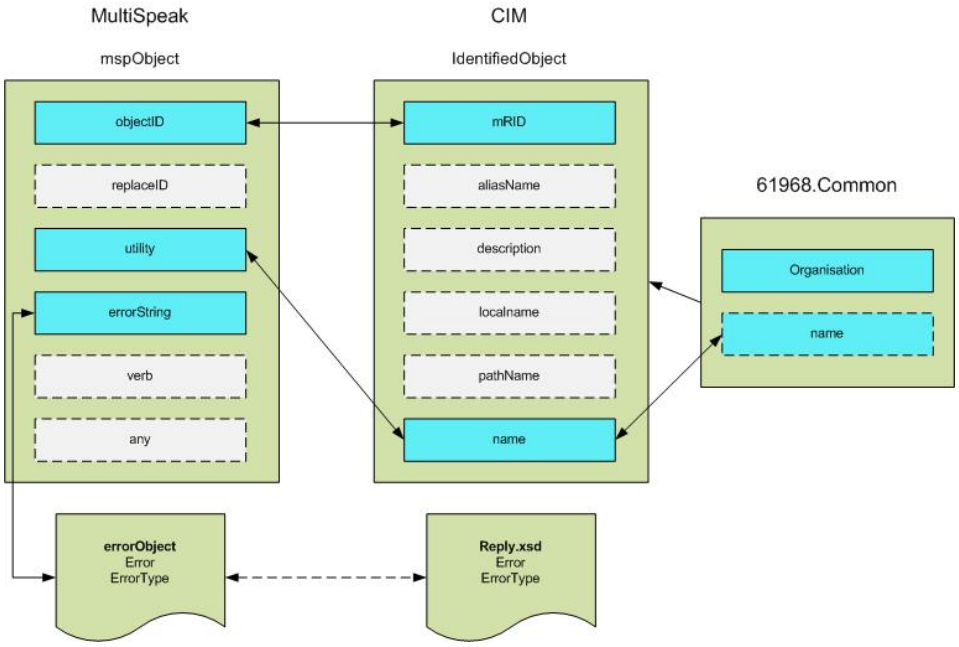

Figure 3-1
MultiSpeak mspObject to CIM IdentifiedObject Class Mapping. Dashed Lines Indicating Optional Attributes, Solid Lines, Required

When reviewing the respective models it is soon clear that they bear resemblances with regard to inheritance. Both CIM and MultiSpeak inherit from a high level class, identifiedObject, and mspObject respectively.

Both classes use a universal identifier throughout their respective models, CIM uses a Master Resource ID (mRID) while MultiSpeak uses objectID. These two attributes map to each other very nicely. AliasName, description, localname and pathname are all optional for the CIM as are replaceID, verb, and any for

MultiSpeak. (In v15 of the CIM AliasName, localname, and pathname have been deprecated). Obviously if these attributes are being used this would represent a greater challenge, but since they are optional they will not be considered for the harmonization effort.

In MultiSpeak the name of the utility is used to identify to which utility data belong. It is common for the constituents that MultiSpeak serves to share data so this identifier is important. This situation is less common for the utilities served by CIM, however this identifier can be accommodated by using the Organization class in CIM and using the inherited attribute "name".

The remaining challenge is errorString. errorString in the mspObject class is a holdover from previous editions of MultiSpeak. Nominally errors that are returned from a CIM errorType (returned as part of a web service call) could be mapped to the errorObject used by MultiSpeak web service calls.

With the relationship understood between the highest level classes in the respective models understood the profile mapping can begin.

Creating MultiSpeak Equivalent Profiles

The most significant challenge with mapping from CIM to MultiSpeak profiles is that MultiSpeak, as an XSD artifact, essentially contains the MultiSpeak model in its entirety (depending on MultiSpeak service). To do a profile-based evaluation then it follows that equivalent MultiSpeak profiles would need to be created.

The IEC 61968-9 profiles are available from the IEC. Each CIM profile can be copied from the pdf file available from the IEC web store,[5] into an XML editor such as Altova's XMLSpy for evaluation.

A CIM profile contains an element that reflects the name, then one of more structures within in the profile. For example, the profile MeterAssetConfig contains 0-to-many MeterAssets. Each MeterAsset element in the profile then contains the attributes for a specific MeterAsset.

Once the CIM profile has been reviewed a search for similar classes and attributes are conducted in the MultiSpeak model. Using the relevant classes, attributes, and associations in MultiSpeak an equivalent profile can be constructed in UML. In addition to classes and attributes, at times abstract classes, associations, or generalizations may need to be added to the MultiSpeak UML to facilitate crafting UML. The goal is that the XSDs are *model driven*. That is, the XSD can be generated from the UML with a minimum level of editing of the XSD once it has been generated.

[5] http://webstore.iec.ch/Webstore/webstore.nsf/Artnum_PK/43378

Using an open tool (CIMEA) which integrates into the Enterprise Architect environment can then be used to generate an XSD that can then in turn be used for mapping.

The one caveat to the model driven development is that the tool used for doing the XSD mapping, Altova MapForce, has a challenge when classes that are included in a profile have been set to abstract, so for the purpose of generating the XSD for mapping, abstract has been set to false.

MultiSpeak Equivalent of CIM Profiles

The following MultiSpeak equivalent profiles have been added and mapped since the release of the first edition of this report, CIM-MultiSpeak Harmonization, EPRI Report 1024443.

AuxiliaryAgreementConfig
CustomerConfig
CustomerMeterDataSet
EndDeviceAssets
EndDeviceEvents (Updated)
MeterAssetReading
MeterReadings
MeterServiceRequests
MeterSystemEvents
PricingStructureConfig
ReceiptRecord
SDPLocationConfig
ServiceCategoryConfig
ServiceDeliveryPointConfig
ServiceLocationConfig
SupplierConfig
TransactionRecord

UML Diagram Conventions

The UML diagrams that are provided are intended to show how the respective equivalent MultiSpeak profiles were constructed. The intent is to show the core classes and their relationships. Some classes may be derived types which are not shown, e.g. money. For these derived types it is suggested that the reader refer to the MultiSpeak data reference model. Also, in the figures, the UML that includes the simple types, e.g. double, are also not shown.

Further, whenever an object appears in red this represents a suggested addition to MultiSpeak that would be required to make a profile more consistent with the CIM. Most often this is the addition of an association, however, at times a there may be a suggested abstract class, typically the “parent” holder class in any given profile. For example, AuxiliaryAgreementConfig holds many AuxiliaryAgreements. Since there is no AuxiliaryAgreement in MultiSpeak, for the purposes of generating an equivalent profile to hold all of the subsequent

associated relationships, this abstract class has been added, merely as a means for suggesting how the equivalence might be created. This is for the MultiSpeak Technical Committee to then take under consideration for addition to future editions.

Table Conventions for Identifying Classes and Attributes

Within each table there is a column for CIM classes and attributes and a column for MultiSpeak class and attributes, and a column for any comments that may add clarifying information for the mapping. Additionally, within a given column the following convention apply to identify when a two attributes are being discussed, versus a class and any several attributes are being compared.

Attributes may or may not be indented. It depends in part on how well the names fit within the column.

Table 3-1
Example Convention to Identify Attributes of A Class

CIM	MultiSpeak	Comments
MeterAsset .mRID .category	Customer .objectID	

In the above table, mRID and category are attributes of the CIM class "MeterAsset" and objected is an attribute of the MultiSpeak class "Customer". The period is used to denote attributes of a class.

Auxiliary Agreement Config

This profile was created out of a need in a country where there was a retailer model, that is, some entities were authorized to resell electricity. This profile then tied those accounts to a parent account from the national electricity entity via this concept of an auxiliary agreement. For MultiSpeak, while this concept may become useful in the future to support retailer or third party models, there is no plan to develop this profile further. As it is this profile probably has this most significant gaps compared to other profiles that tend to have a much higher degree of correlation.

While the Auxiliary Agreement Config profile may eventually be useful in a third party or aggregator model, there are no plans to develop this profile further in MultiSpeak.

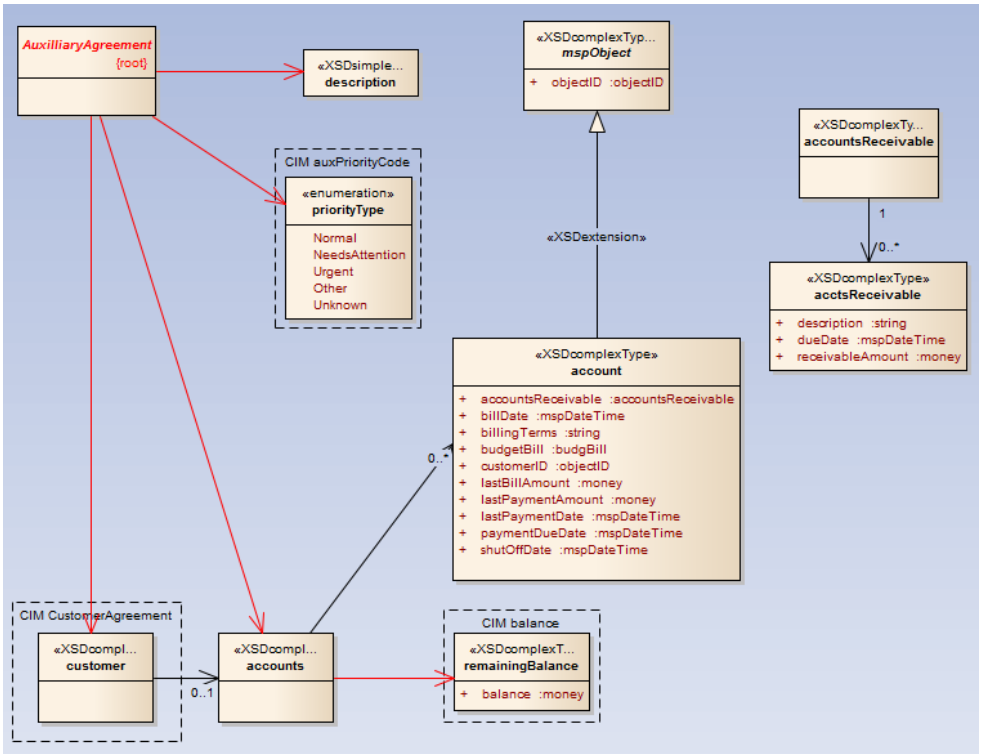

Figure 3-2
UML Representation of MultiSpeak Equivalent of AuxiliaryAgreementConfig

Table 3-2
AuxiliaryAgreementConfig Correlations

CIM	MultiSpeak	Comments
AuxilliaryAgreement .description	mspAuxilliaryAgreementConfig .description	

Table 3-3
AuxiliaryAgreementConfig Transformations

CIM	MultiSpeak	Comments
AuxiliaryAgreement	mspAuxiliaryAgreement	This is only noted because this MultiSpeak abstract class was added for illustrative purposes to build this profile.
CustomerAgreement	Customer	Nominally the objected could be mapped to mRID to identify a given customer, but a customer agreement does not exist in MultiSpeak.
AuxiliaryAccounts	Accounts.account	The unique identifier for each (mRID and objected) could be mapped to identify a given account.

Table 3-4
AuxilliaryAgreementConfig Gaps

CIM	MultiSpeak	Comments
AuxiliaryAgreement		Rather than detail each missing attribute, this profile the parent classes will be identified. The reader is directed to the CIM data reference model for more information
AuxiliaryAccounts		
Charges		
docStatus		
due		
lastCredit		
lastDebit		
Status		
CustomerAgreement		

Customer Account Config

Table 3-5
CustomerAccountConfig Correlations

CIM	MultiSpeak	Comments
mRID	customerID	customerID is type ObjectID
budgetBill	budgetBill	Type:String in both cases
billingCycle	billingCycle	Type:String in both cases
description	Description	By using the mspCPSM:IdentifiedObject class in MultiSpeak the high level attributes from CIM can be mapped
name	Name	

Table 3-6
CustomerAccountConfig Transformations

CIM	MultiSpeak	Comments
aliasName	objectName	
docStatus status	statusCode	statusCode is not normally associated with the account class. While there is no equivalent of docStatus in MultiSpeak, statusCode could have the CIM docStatus and Status classes concatenated and mapped to it.

Table 3-7
CustomerAccountConfig Gaps

CIM	MultiSpeak	Comments
category		Does not appears in the CIM project. It may be ServiceCategory which is the type of service the customer uses.(14v11-10v25 and 15v13-11 v-06)
lastModified .dateTime		From the Document class
title		From the Document class
	accountsReceivable	A/R information for this account
	billDate	Date of last bill
	calculatedCurrentBillAmount	Amount that the customer would owe if final billed today. This amount is based on the fixed charges, energyusage, fuel costs and taxes since the last bill.
	calculatedCurrentBillDateTime	Date and time that the calculated current bill amount was calculated.
	calculatedCurrentBillReadings	
	calculatedUsedYesterday	The value (expressed in currency) in energy, fuel costs, taxes, and prorated fixed charges the customer used yesterday. It may simply be the calculatedCurrentBillAmount from today minus the calculated current bill amount from yesterday.
	isPrePay	If the field is missing or false then the account is cycle or budget billed.
	lastBillAmount	The amount of the last bill to the customer.
	lastPaymentAmount	Amount of the last payment made.
	paymentDueDate	Date and time that the next payment is due.
	shutOffDate	Shut off date for non-payment.

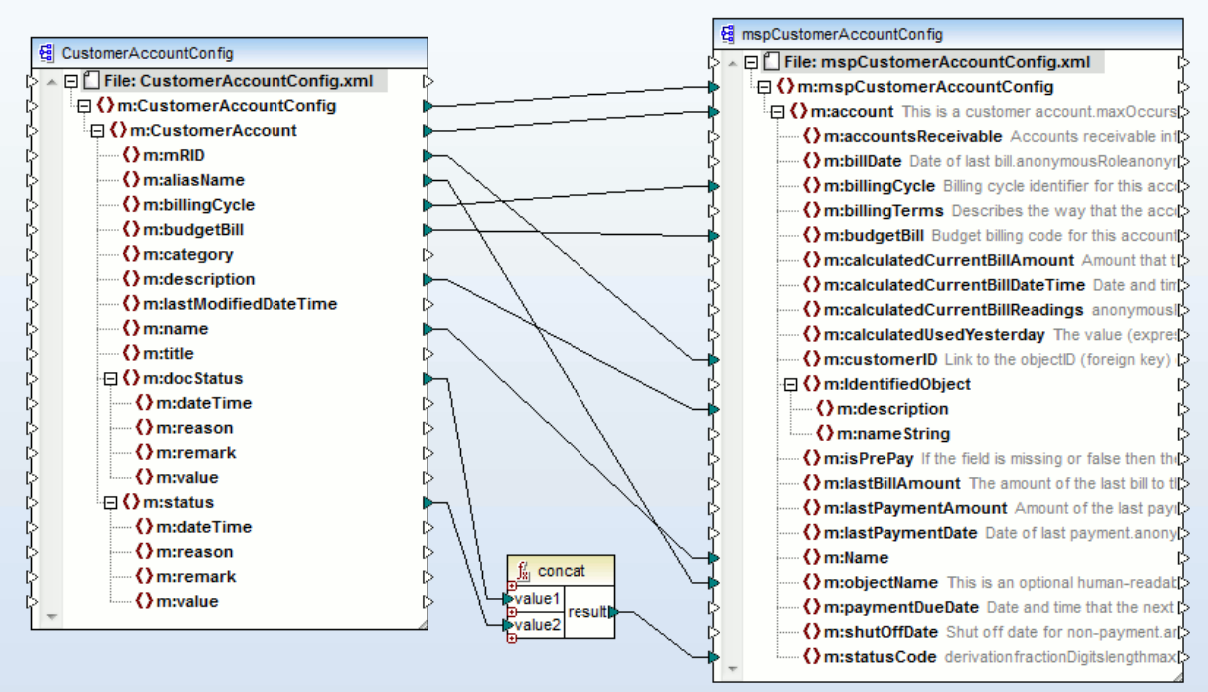

Figure 3-3
CustomerAccountConfig CIM to MultiSpeak Profile Mapping

Customer Agreement Config

There are significant gaps in this particular profile as MultiSpeak does not support the generic document or agreement structures. However, customer, account, and service supplier (see ServiceSupplierConfig) can be link.

Table 3-8
CustomerAgreementConfig Correlations

CIM	MultiSpeak	Comments
Customer mRID	Customer .objectID	
CustomerAccount .mRID	Account .objectID	

Table 3-9
CustomerAgreementConfig Transformations

CIM	MultiSpeak	Comments
Tarriffs .mRID	rateInfo .objectID	
TarriffProfiles .mRID	rates .objectID	
ServiceSupplier .mRID	resource .objectID	As noted in the ServiceSupplierConfig, this assumes an association is added from ContactInfo to resourceto create an equivalent ServiceSupplierConfig
validityInterval	dateTime	Not a good match as validitiyInterval is for the the length of the CustomerAgreement (start, end), while dateTime appliesto a specific rateInfo attribute

Table 3-10
CustomerAgreementCnfig Gaps

CIM	MultiSpeak	Comments
docStatus		This inherits from the generi statu class and applies specifically to documents
PricingStructure		
Status		
	commodityUnits currencyCode rateCode rateCodeDescription	Attributes of rates that provide additional information that is not included in CIM
	billingTerms	Provides additional information about how an account is billed

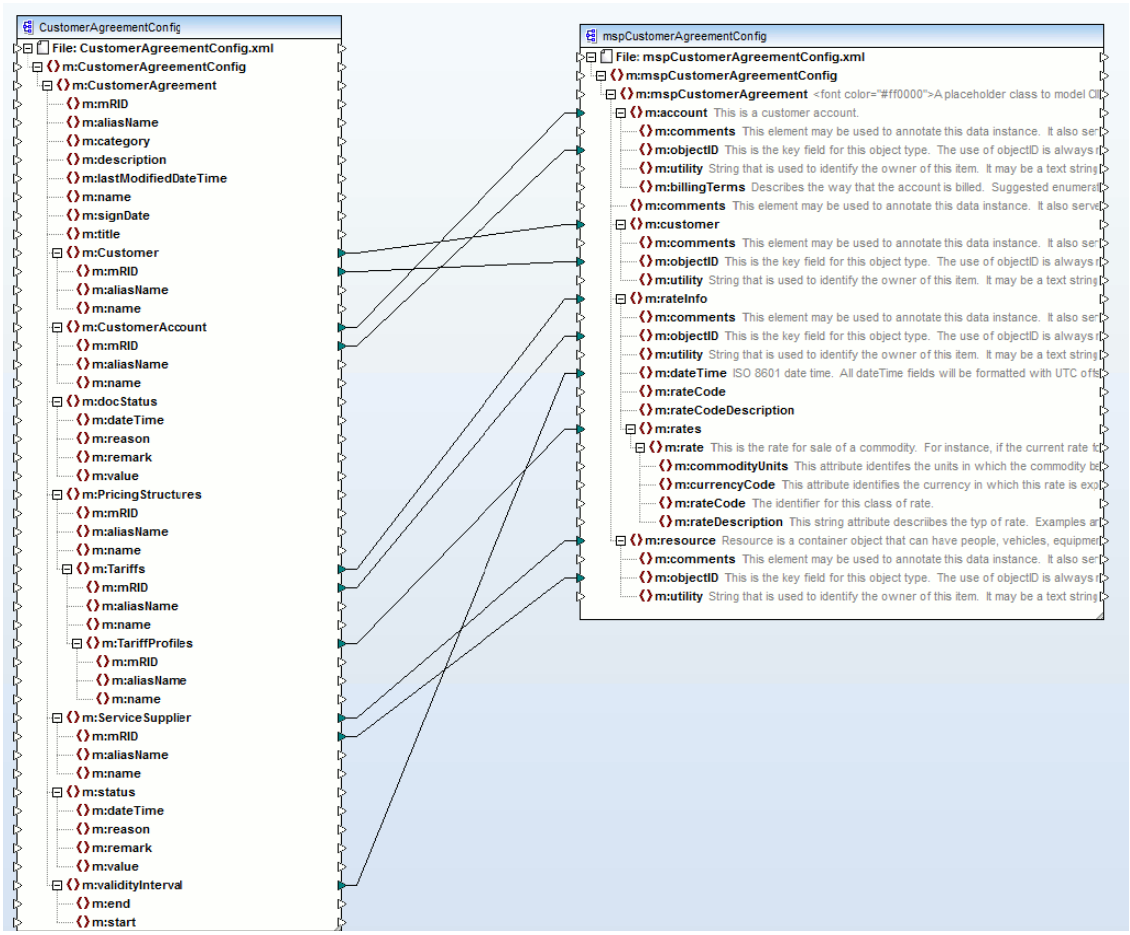

Figure 3-4
CustomerAgreementConfig CIM to MultiSpeak Mapping

Customer Config

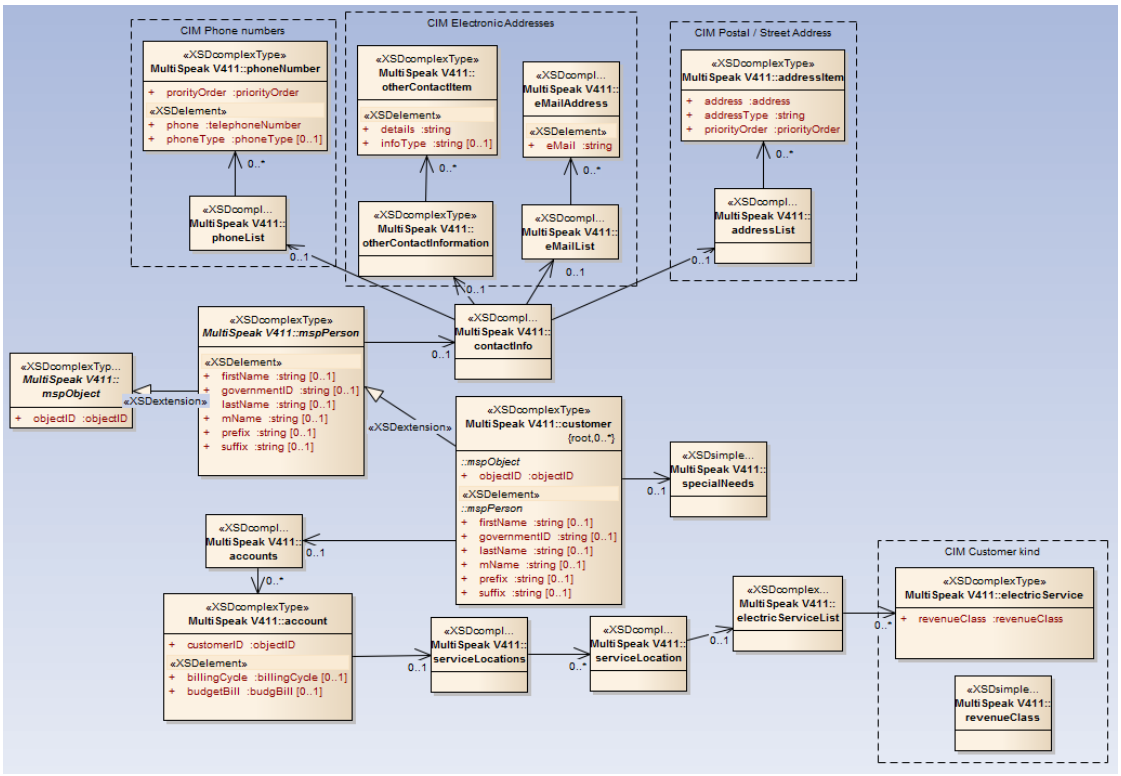

Figure 3-5
UML Representation of the MultiSpeak Equivalent of CIM CustomerConfig Profile

Table 3-11
CustomerConfig Correlations

CIM	MultiSpeak	Comments
Customer.mRID	Customer.objectID	
ElectronicAddresses email	eMailAddress.email	

For Address and Phone Number correlations, See Appendix A.

Table 3-12
CustomerConfig Transformations

CIM	MultiSpeak	Comments
Customer .specialNeed .vip	Customer .specialNeeds	specialNeed and vip would need to be concatenated and mapped to specialNeeds
Customer.kind	revenueClass	Issue with this CIM profile in that normally you would need to drill down into a specific account to determine the kind
ElectronicAdresses .lan .name .password .radio .userID .web	otherContactInformation otherContactItem .details .infoType	All of the CIM attributes are optional, and can be mapped to otherContactItem.details, or concatenated if multiple attributes are populated

Table 3-13
CustomerConfig Gaps

CIM	MultiSpeak	Comments
pucNumber		
buildingName		
code		
name		
withinTownLimits		
Status class		
streetAddress		Nominally not a gap in that MultiSpeak allows for multiple addresses to be stored per the addressList; it can support many addresses instead of only
	detailedAddressFields .region	
townDetail.name		
townDetail.section		

For phone number mappings see Appendix A, Table A-4 *Electronic Address related transformations.*

Customer Meter Data Set

The CustomerMeterDataSet profile is an aggregation of several other profiles as can be seen in the figure below.

Each of the high level structures shown in the schema view below are profiles themselves. Rather than replicate the mapping here the reader is referred to the respective paragraph for each profile.

Customer	– Customer Config
CustomerAccount	– Customer Account Config
CustomerAgreement	- Customer Agreement Config
EndDeviceAsset	- End device Assets
SDPLocation	- SDP Location Config
ServiceCategory	- Service Category Config
ServiceDeliveryPoint	- Service Delivery Point Config
ServiceLocation	- Service Location Config

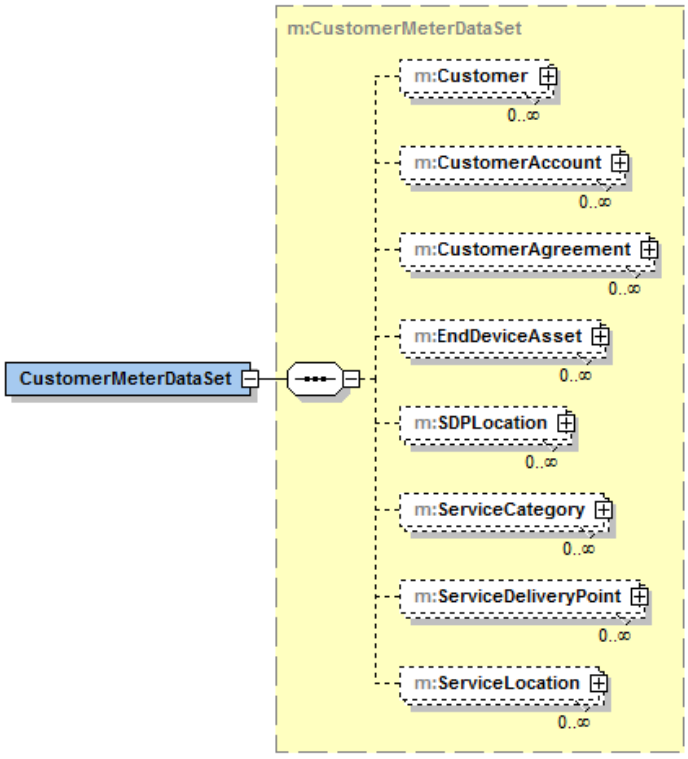

Figure 3-6
CIM Schema view of CustomerMeterDataSet

End Device Assets

There are six sections in the CIM EndDeviceAssets profile:

- ComFunction
- ConnectDisconnectFunction
- ElectricMeteringFunction
- EndDeviceAsset
- ReadingType
- Register -

Due to the size of this profile the component parts will be examined individually.

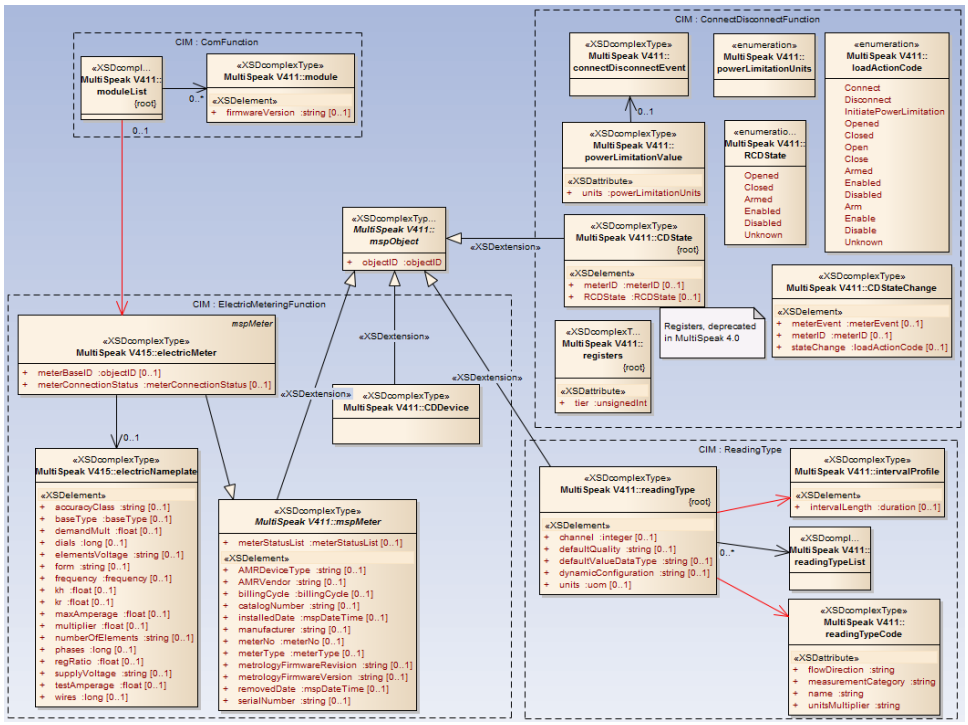

Figure 3-7
UML of MultiSpeak Equivalent of CIM EndDeviceAssets Profile

ComFunction

In the CIM End Devices are not only meters, but other types of devices in the distribution network that have communication capability. This section details that capability.

Table 3-14
EndDeviceAssets – ComFunction

CIM	MultiSpeak	Comments
ComFunction	Module	
.firmwareID	.firmwareVersion	

The strength of the ComFunction in CIM is that it has several attributes that more fully define a communications function where MultiSpeak only has firmwareID. However, MultiSpeak uses a moduleList relationship to module which allows MultiSpeak to associate many modules (communications functions) with a given asset where in CIM it is limited to a single communications function per asset.

Table 3-15
EndDeviceEvents – ComFunction Gaps

CIM	MultiSpeak	Comments
mRID		Unique identifier for the ComFunction (or MultiSpeak module)
amrRouter		
configID		
hardwareID		
password		
programID		
twoWay		
	moduleList	In the EndDeviceAssets profile CIM has o to many ComFunction where in MultiSpeak there are o to many moduleLists which in turn have o to many modules

ConnectDisconnectFunction

This is the functionality that allows the meter (or other End Device) to be remotely connected or disconnected, or in the case of meters, to enable load limiting model.

For the purposes of mapping the ConnectDisconnect portion of this CIM profile, an abstract class mspConnectDisconnect was added to the UML to hold the many connect/disconnect or power limiting related classes.

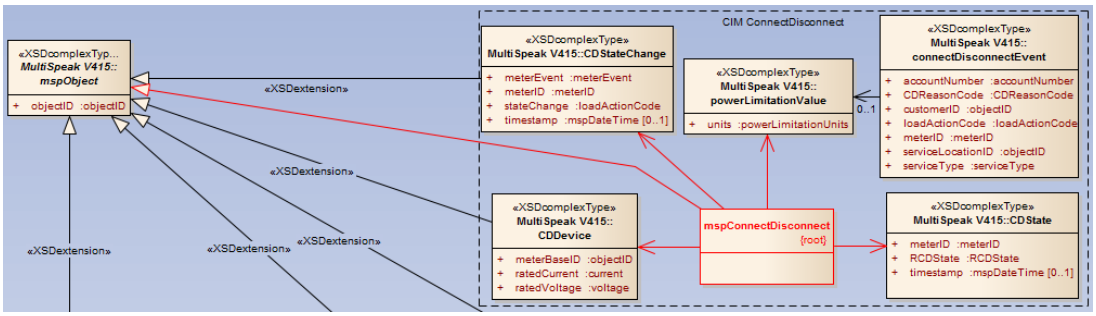

Figure 3-8
ConnectDisconnect Portion of the UML of the MultiSpeak Equivalent of the EndDeviceAssets Profile

Table 3-16
EndDeviceAssets – ConnectDisconnect Correlations

CIM	MultiSpeak	Comments
ConnectDisconnectFunction .disabled	CDState .RCDState	

Table 3-17
EndDeviceAssets – ConnectDisconnect Transformations

CIM	MultiSpeak	Comments
ConnectDisconnectFunction .mRID	mspConnectDisconnect .objectID	
rcdInfo .customerVoltageLimit .energyLimit .powerLimit	powerLimitationValue .units	Typically only one of these CIM attributes needs to be set at a time, therefore the inputs can use a logical OR to map to the MultiSpeak attribute

Table 3-18
EndDeviceAssets ConnectDisconnect Gaps

CIM	MultiSpeak	Comments
eventCount		
isConnected		
isdelayedDiscon		
isLocalAutoDisconOp		
isLocalAutoReconOp		
isRemoteAutoDisconOp		
isRemoteAutoReconOp		
rcdInfo .armedTimeOut .energyUsageStartDateTime .isArmConnect .isArmDisconnect .isEnergyLimiting .needs PowerLimitCheck .needsVoltageLimitCheck .usePushButton		

ElectricMeteringFunction

This portion of the profile defines the set of electric metering functions of the end device asset.

Table 3-19
EndDeviceAssets – ElectricMeteringFunction

CIM	MultiSpeak	Comments
ElectricMeteringFunction .mRID .firmwareID	electricMeter .MeterNo .meterologyFirmwareRevision	
.currentRating .demandMultiplier .voltageRating	electricNameplate .maxAmperage .demandMult .elementsVoltage	

Table 3-20
EndDeviceAssets – ElectricMeteringFunction – Transformations

CIM	MultiSpeak	Comments
ElectricMeteringFunction .disabled	electricMeter .meterConnectionStatus	
.kWhMultiplier .kWMultiplier	electricNameplate .multiplier	Logical OR the CIM attributes to the MultiSpeak multiplier attribute

Table 3-21
EndDeviceAssets – ElectricMeteringFunction Gaps

CIM	MultiSpeak	Comments
EletricMeteringFunction .billingMultiplier .billingMultiplierApplied .configID .currentRating .demandMultiplierApplied .hardwareID .password .programID .transformerCTRatio .transformerRatiosApplied .transformerVTRatio		
	eletrciMeter .meterType	

EndDeviceAsset

MultiSpeak's hierarchy of classes has a different focus, more meter centric where the CIM has generic classes. This is apparent in a CIM profile that uses ElectricMeterinFunction that is mapped to the MultiSpeak electricMeter. However, the generic EndDeviceAsset also maps to the MultiSpeak electricMeter. MultiSpeak does not have a generic "end device" class so an end device must be mapped to the closet match, electricMeter.

Table 3-22
EndDeviceAssets – EndDeviceAsset

CIM	MultiSpeak	Comments
EndDeviceAsset .mRID .amrSystem .category .description .formNumber .installationDate .serialNumber	electricMeter .meterNo .amrVendor .amrDeviceType .description .form .installedDate .serialNumber	meterNo is of type objected description class can be associated with the eletricMeter class
 .kH .kR	eletricNamePlate .kh .kr	

For the ActivityRecords mapping, see the paragraph on End Device Events.

Table 3-23
EndDeviceAssets – EndDeviceAsset Gaps

CIM	MultiSpeak	Comments
EndDeviceAsset .demandResponse .dstEnabled .loadControl .lotNumber .manufacturedDate .metrology .outageReport .purchasePrice .readRequest .relayCapable .reverseFlowHandling .timeZoneOffset		
	electricMeter .billingCycle .catalogNumber .manufacturer .meterType .meterStatusList .meterologyFirmware Revision .removedDate .meterBaseID	CIM has a generic status class, but this is one instance where it is not in a CIM profile
	electricNamePlate .accuracyClass .baseType .dials .frequency .numberOfElements .phases .regRatio .supplyVoltage .testAmperage .wires	

ReadingType

This paragraph will cover the reading types associated with the end device assets profile.

Table 3-24
EndDeviceAssets – ReadingType Correlations

CIM	MultiSpeak	Comments
ReadingType .mRID .defaultQuality .defaultValueDataType .dynamicConfiguration .unit .channelNumber .reverseChronology	readingType .objectID .defaultQuality .defaultValueDataType .dynamicConfiguration .units .channel .forewardChronology	Logical NOT to map these attributes
.intervalLength .name	intervalProfile .intervalLength .name	
.kind .multiplier	readingTypeCode .measurementCategory .unitsMultiplier	

Table 3-25
EndDeviceAssets – ReadingType Gaps

CIM	MultiSpeak	Comments
	readingType .measTypeCategory .measurementType	
	intervalProfile .profileID	
	readingTypeCode .flowDirection .name	

Register

Table 3-26
EndDeviceAssets – Register Gaps

CIM	MultiSpeak	Comments
Register .mRID .leftDigitCount .rightDigitCount		
	registers .tier	

End Device Controls

There is a high degree of correlation between the CIM EndDeviceControls and the equivalent classes and attributes in MultiSpeak.

Table 3-27
EndDeviceControls Correlations

CIM	MultiSpeak	Comments
mRID	objectID	customerID is type ObjectID
Description	description	Type:String in both cases
type	deviceType	Type:String in both cases
EndDeviceAsset .mRID	mspDevice .objectID	
name	Name	MultiSpeak uses a CIM equivalent identifiedObject class to pass attributes such as Name

Table 3-28
EndDeviceControls Transformations

CIM	MultiSpeak	Comments
demandResponseProgram	demandResponseEvent	The challenge is the intent of each of these parent classes. One is pre-event, one is post-event.
scheduledInterval.start schduledInterval.end	demandResponseEvent .duration	Nominally start/end can be used to calculate a duration but the question goes back to the intent of the information being captured
mRID	loadManagementID	loadManagementID is of type objectID
EndDeviceGroup	loadSection	The challenge here is a group of EndDevices versus what is an area of load within the distribution grid
EndDeviceGroup.mRID	loadGroupID	loadGroupID is of type objectID
groupAddress	loadGroup	CIM groupAddress is an integrer, MultiSpeak loadGroup is a string

Table 3-29
EndDeviceControls Gaps

CIM	MultiSpeak	Comments
DemandResposneProgram .type		MultiSpeak has classes for demandResponse events, not programs
drProgramLevel		Level of a demand response program request, where 0=emergency.
drProgramMandatory		Boolean indicating wether the request is mandatory
priceSignal	ZigbeePublishPrice	There is not a direct correlation between the CIM attribute and the MultiSpeak class although nominally priceSignal (floatQuantity) might be transformed to rateLabel (string)

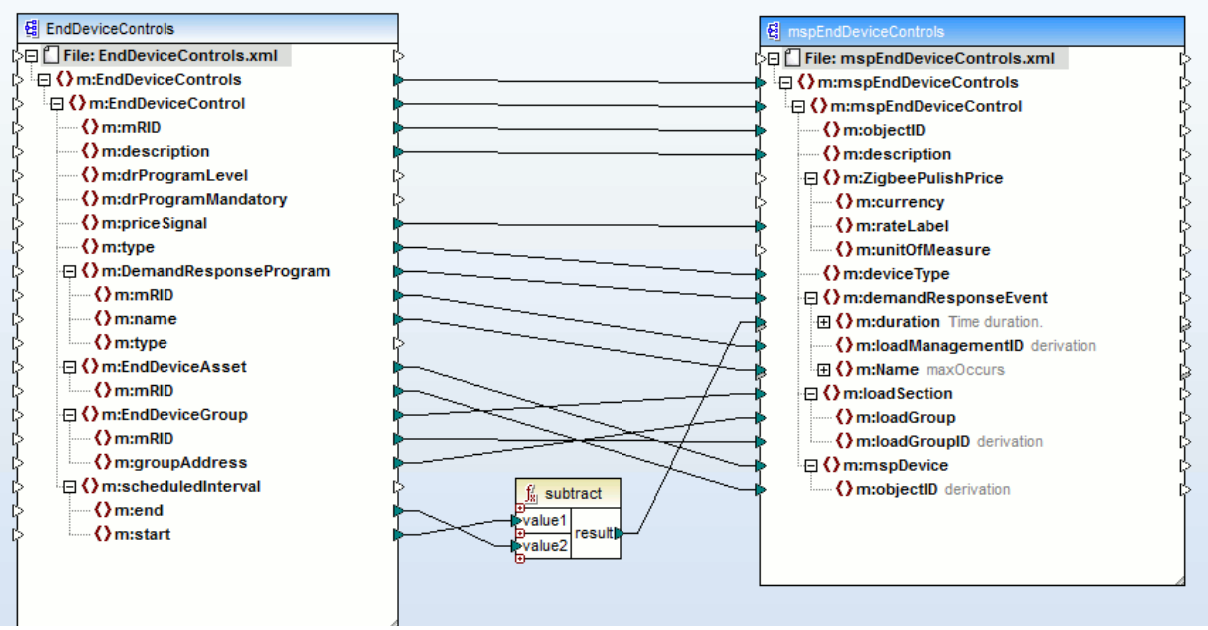

Figure 3-9
XSD Mapping of CIM EndDeviceControls to MultiSpeak Equivalent Profile

End Device Events

The biggest challenge with this profile is that CIM EndDeviceEvents is an abstraction meant to serve any "end device" whereas the event class in MultiSpeak is associated with metering. The other challenge is that MultiSpeak uses a wide variety of classes and enumerations to document different types of status depending on the context where CIM uses a generic, compound status class that can be in multiple contexts.

Another challenge is that there is not a good fit for CIM category attribute. MultiSpeak's outageEventStatus appears that it could be leveraged to serve this purpose, however there is normally no relation in MultiSpeak between an "event (for metering) and an outageEvent which is normally associated with outage management.

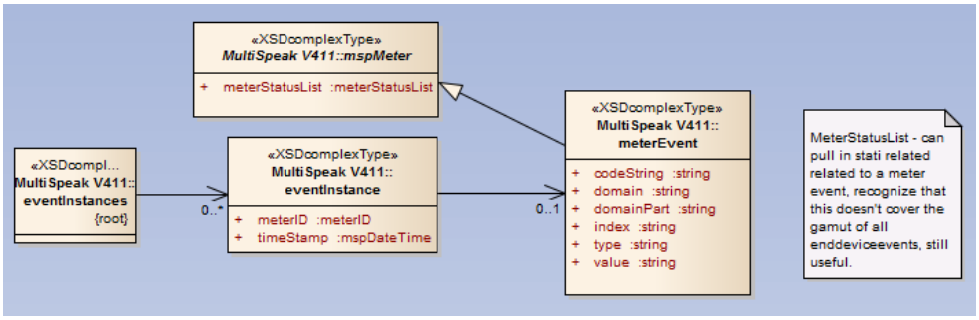

Figure 3-10
UML of MultiSpeak Equivalent of CIM EndDeviceEvent Profile

Table 3-30
EndDeviceEvents Correlations

CIM	MultiSpeak	Comments
mRID	objectID	
argument	argument	
userID	userID	
timeStamp	dateTime	
reason	comments	MultiSpeak comments normally does not have an association to event

Table 3-31
EndDeviceEvents Transformations

CIM	MultiSpeak	Comments
category	outageEventStatus	No direct correlation but some attributes in the outageEventStatus might be able to be mapped to category

Table 3-32
EndDeviceEvents Gaps

CIM	MultiSpeak	Comments
severity		No MultiSpeak equivalent

This mapping illustrates that CIM category can either be mapped directly to MultiSpeak codestring, or category can be decomposed into its subordinate parts

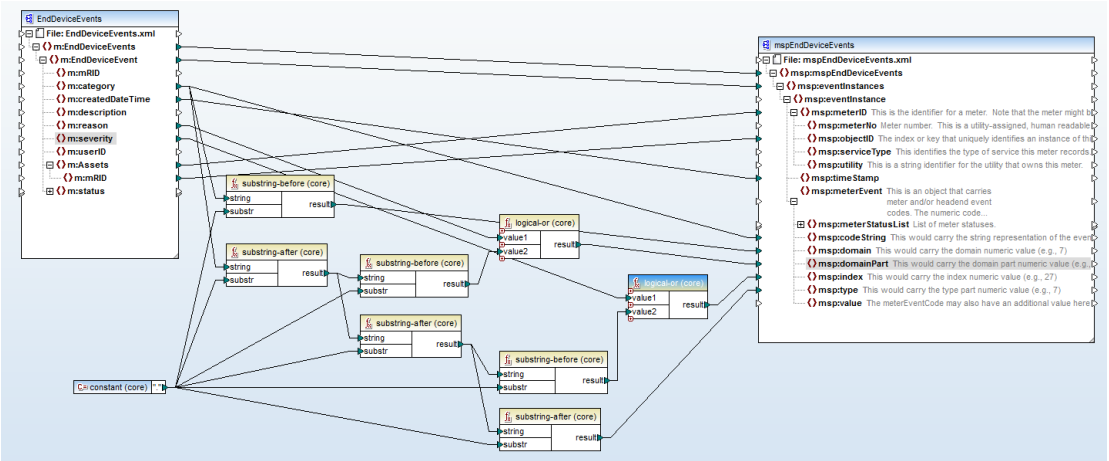

Figure 3-11
Mapping of the CIM EndDeviceEvents to MultiSpeak mspEndDeviceEvents Profile

End Device Firmware

The EndDeviceFirmware profile captures information about the particular functions that a device might have. This closely correlates to the MultiSpeak concept of Modules.

Table 3-33
EndDeviceFirmware Correlations

CIM	MultiSpeak	Comments
EndDeviceAsset	mspMeter	
mRID	objectID	
amrSystem	AMRVendor	
serialNumber	serialNumber	

Table 3-34
EndDeviceFirmware Transformations

CIM	MultiSpeak	Comments
DeviceFunctions	Module	MultiSpeak uses a parent class moduleList that allows for 1..* modules
mRID	objectID	
Category	moduleType	
ActivityRecords	mspAlarm	There is not a good match between these two classes although the intent appears to be similar
createdDateTime	eventTime	
reason severity category	eventCode	MultiSpeak has a specific code definition for events that are generated, however, it appears that the CIM attributes could be concatenated and mapped to this.
application configID disabled hardwareID password programID	description	While MultiSpeak allows for more than one instance of a module, the CIM DeviceFunction class has more granular information about a given DeviceFunction. However, these attributes can be concatenated and mapped to the MultiSpeak description attribute.
category	moduleType	Category of DeviceFunction is equivalent to moduleType of module

Table 3-35
EndDeviceFirmware Gaps

CIM	MultiSpeak	Comments
Status		CIM has a generic status class associated with the Activity records; no such status equivalent exists with mspAlarm
	sourceIdentifier	MultiSpeak has an additional identifier field for the device that has caused the alarm (CIM activity).

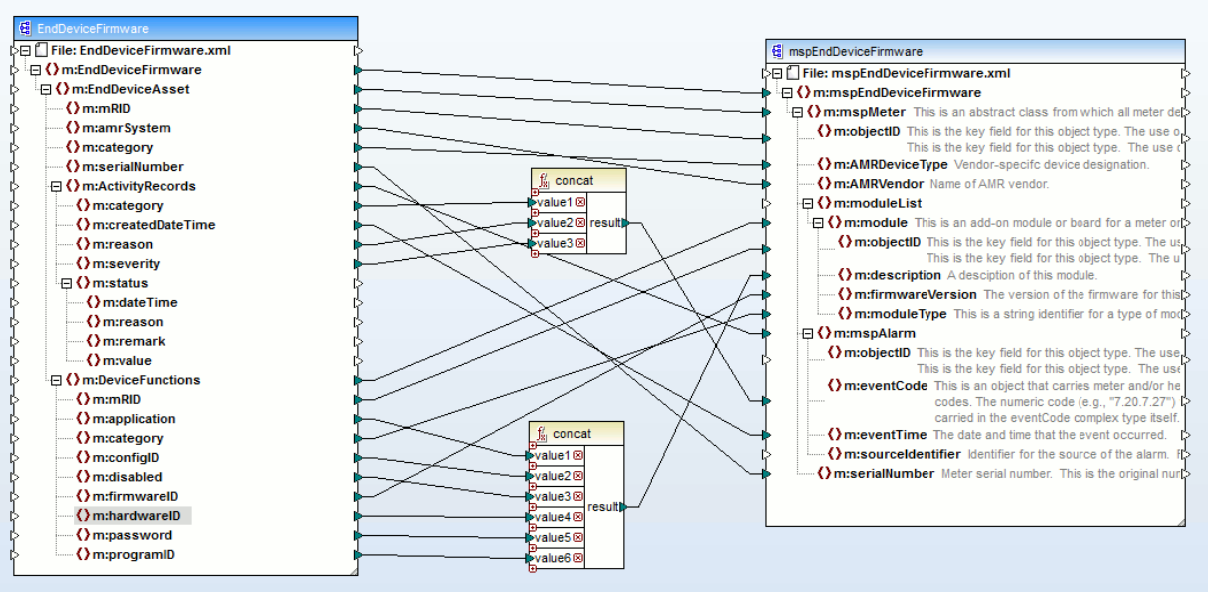

Figure 3-12
EndDeviceFirmware CIM to MultiSpeak Profile Mapping

Meter Asset Config

This profile passes the configuration information about a meter asset including its capabilities and in the CIM, the seals that are associated with the meter.

Table 3-36
MeterAssetConfig Correlations

CIM	MultiSpeak	Comments
MeterAsset .mRID .category installedDate .serialNumber .amrSystem	electricMeter .objectID .AMRDeviceType .installedDate .serialNumber .amrVendor	
kH kR	electricNameplate .kH .kR	
ServiceLocation .mRID	serviceLocation .customerID	
sealNumber	sealNumber	

Table 3-37
MeterAssetConfig Transformations

CIM	MultiSpeak	Comments
None	None	

Table 3-38
MeterAssetConfig Gaps

CIM	MultiSpeak	Comments
Seals .condition .kind .description .appliedDateTime		In addition to sealNumber (which correlates to MultiSpeak sealNumber), CIM provides addition attributes that indicate when the seal was applied and enumerations for the seal condition and the type (kind) of seal that was applied
MeterAsset .application .demandResponse .disconnect .dstEnabled .formNumber initialCondition initialLossOfLife .loadControl .lotNumber manufacturedDate .meterology .outageReport .readRequest .relayCapable reverseFlowHandling .timeZoneOffset .utcNumber	electricMeter .billingCycle .catalogNumber .manufacturer .meterNo .meterStatusList .meterType metrologyFirmwareRevision metrologyFirmwareVersion .removedDate	
Status		

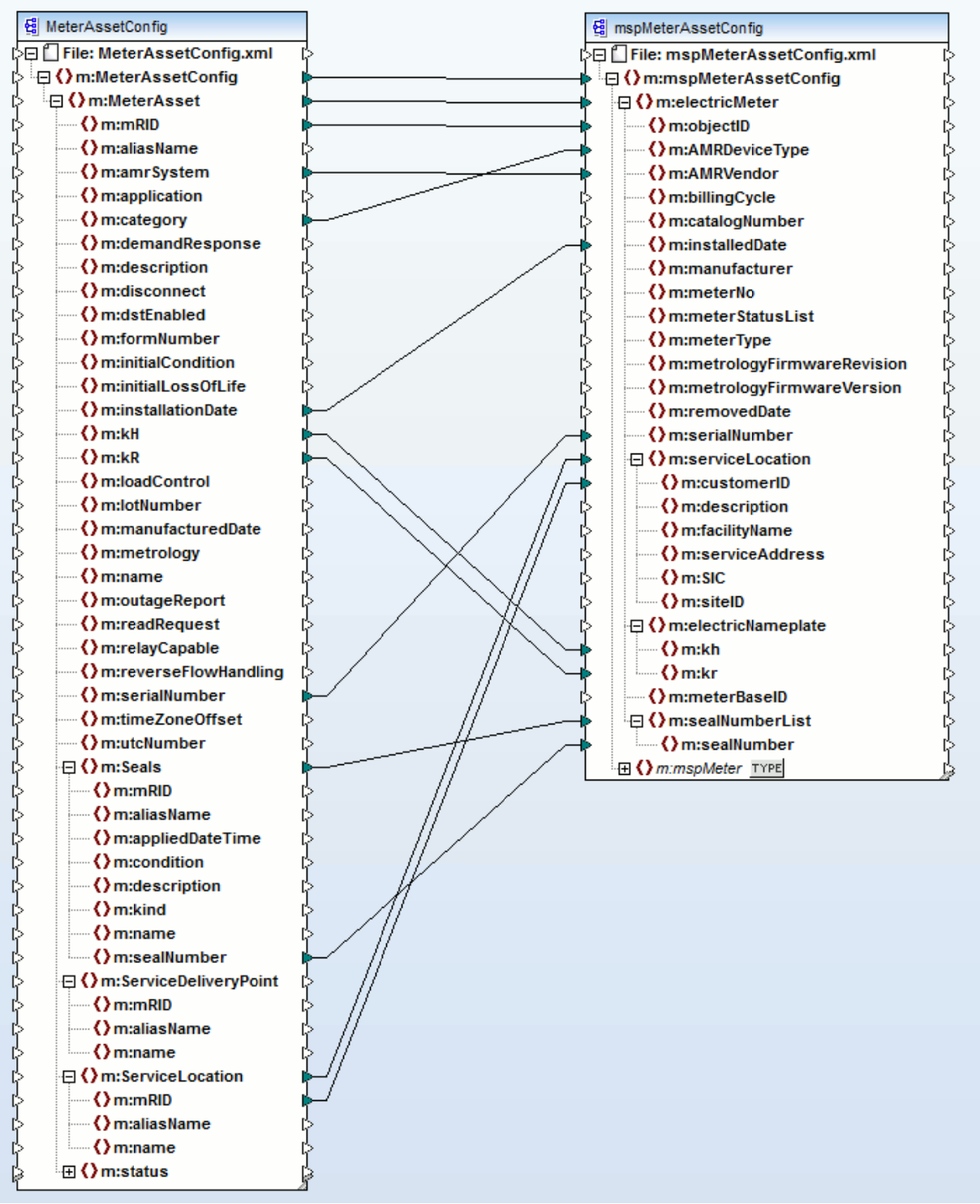

Figure 3-13
MeterAssetConfig CIM to MultiSpeak Profile Mapping

Meter Asset Reading

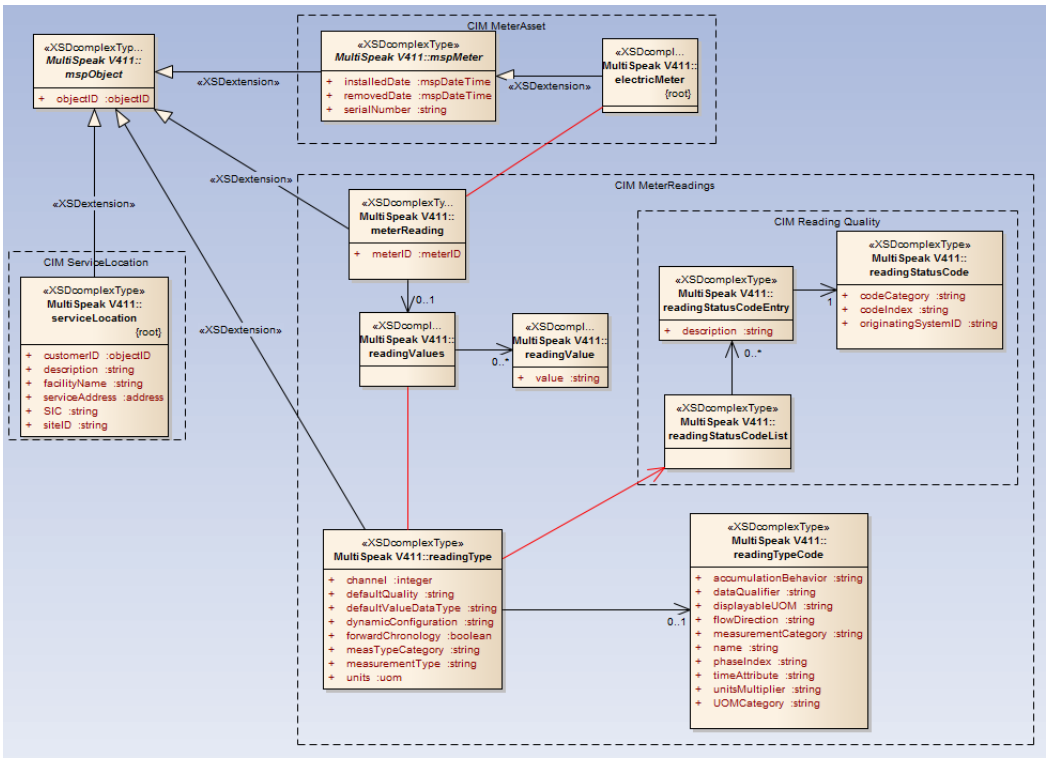

Figure 3-14
UML Mapping of MultiSpeak Equivalent to CIM MeterAssetReading Profile

Table 3-39
MeterAssetReading Correlations

CIM	MultiSpeak	Comments
MeterAsset.mRID .serialNumber	electricMeter.objectID .serialNumber	
MeterReadings	MeterReading	
ReadingType.mRID	readingType.objectID	

MeterAssetReading uses addresses attributes that can be found in Appendix A, Table A-1 Address Correlations.

Table 3-40
MeterAssetReading Transformations

CIM	MultiSpeak	Comments
Readings .timestamp .value	readingValues readingValue.value	
ReadingType	readingType	
ReadingQualities .quality	readingType .defaultQuality	

MeterAssetReading uses Service Location transformations which can be found in Appendix A, Table A-3 Service Location related transformations.

Table 3-41
MeterAssetReading Gaps

CIM	MultiSpeak	Comments
MeterServiceWork .kind .priority .requestDateTime		
Status		
	readingType .channel .defaultValueDataType .dynamicConfiguration .forwardChronology .measTypeCategory .measurementType .units	
	readingTypeCode .accumulationBehavior .dataQualifier .displayableUOM .flowDirection .measurementCategory .name .phaseIndex .timeAttribute .unitsMultiplier .UOMCategory	

Meter Readings

The CIM MeterReadings profile contains within in two main structures, MeterReading and ReadingType. Additionally, within the MeterReading there is also information about EndDeviceEvents, CustomerAgreement which identifies the customer, the MeterAsset which identifies the meter, IntervalBlocks, the Readings, and finally, the ServiceDeliveryPoint of the MeterReadings.

For the EndDeviceEvent mappings see the paragraph, End Device Events.

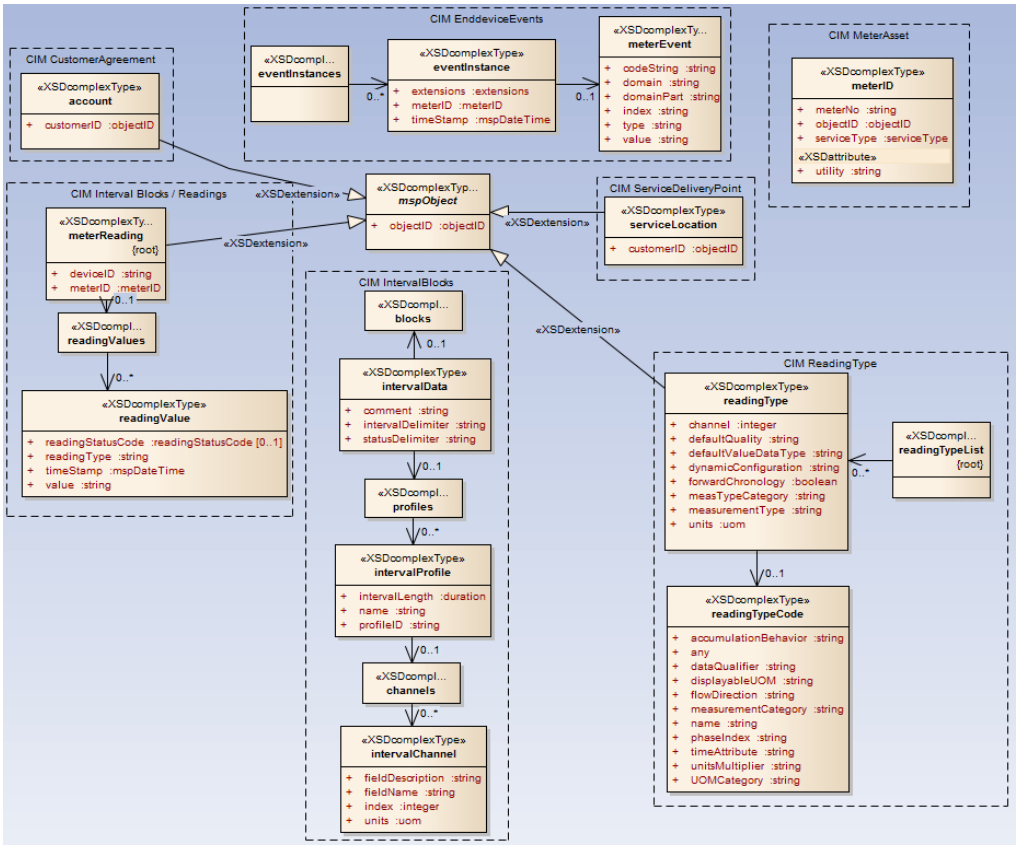

Figure 3-15
UML for MultiSpeak Equivalent of the CIM MeterReadings Profile

In MultiSpeak MeterNo is human readable meta data for the meter, the identifier is a meterID, a complex type that includes both an objectID and a serviceType

In CIM, MeterAsset has a status where in MultiSpeak the meter is identified first then the object caries the status.

Table 3-42
MeterReadings Correlations

CIM	MultiSpeak	Comments
ReadingType .mRID .channelNumber .defaultQuality .defaultValueDataType .dynamicConfiguration .multiplier .name .unit	readingType .objectID .channel .defaultQuality .defaultValueDataType .dynamicConfiguration .unitsMultiplier .name .units	
Readings .timestamp .value	readingValues .timestamp .value	

Table 3-43
MeterReadings Transformations

CIM	MultiSpeak	Comments
CustomerAgreement .mRID	customerID	A customerID in MultiSpeak is of type objectID, which can be used to identify a given customer
intervalLength	timeAttribute	
kind	measTypeCategory	
reverseChronology	forwardChronology	The intent is to indicate chronology, one is simply the inverse of the other
ReadingQualties .quality	readingStatusCode .codeCateogry .codeIndex .originatingSystemID	These correspond to IEC 61968-9 Annex C
IntervalBlocks	blocks	

Table 3-44
MeterReadings Gaps

CIM	MultiSpeak	Comments
	readingTypeList	MultiSpeak has the capability through the list class, to associate many readingTypes.
Pending .multiplyBeforeAdd .offset .scalarDenominator .scalarFloat .scalarNumerator		
	intervalData .comment .intervalDelimiter .statusDelimiter	
	profiles	
	intervalProfile .intervalLength .name .profileID	
	channels	
	intervalChannel .fieldDescription .fieldName .index .units	

While both CIM and MultiSpeak handle interval blocks, they appear to be handled differently. CIM has the Pending class and then intervalReadings with values, timestamps and associated ReadingQualities, where MultiSpeak has many more attributes for intervalData, profiles, and channels.

Meter Read Schedule

This profile creates a schedule and in the CIM, has a customer account, customer agreement, end device asset, or end device group associated with it.

Table 3-45
MeterReadSchedule Correlations

CIM	MultiSpeak	Comments
CustomerAccount .mRID	account .objectID	
EndDeviceAsset .mRID	mspMeter .objectID	
EndDeviceGroup .groupAddress	meterGroupList .meterGroupID	In MultiSpeak the meterList can also be used to define which meters belong to a group
ReadingType .mRID .name	readingType .readingTypeCode .name	
TimeSchedule .offset .disabled	readingSchedule .offset .isEnabled	

Table 3-46
MeterReadSchedule Transformations

CIM	MultiSpeak	Comments
TimeSchedule recurrancePattern .recurrancePeriod	readingSchedule periodicSchedule period .timeUnits .startTime	

Table 3-47
MeterReadSchedule Gaps

CIM	MultiSpeak	Comments
	serviceLocation	In MultiSpeak a serviceLocation can be associated with the readingSchedule which may provide important information about the area where the readings are occurring
window .end .start		The allowable window of time that the reading can occur
TimePoints .absoluteTime .relativeTimeInterval .sequenceNumber		
CustomerAgreement .mRID		
scheduleInterval .end .start		

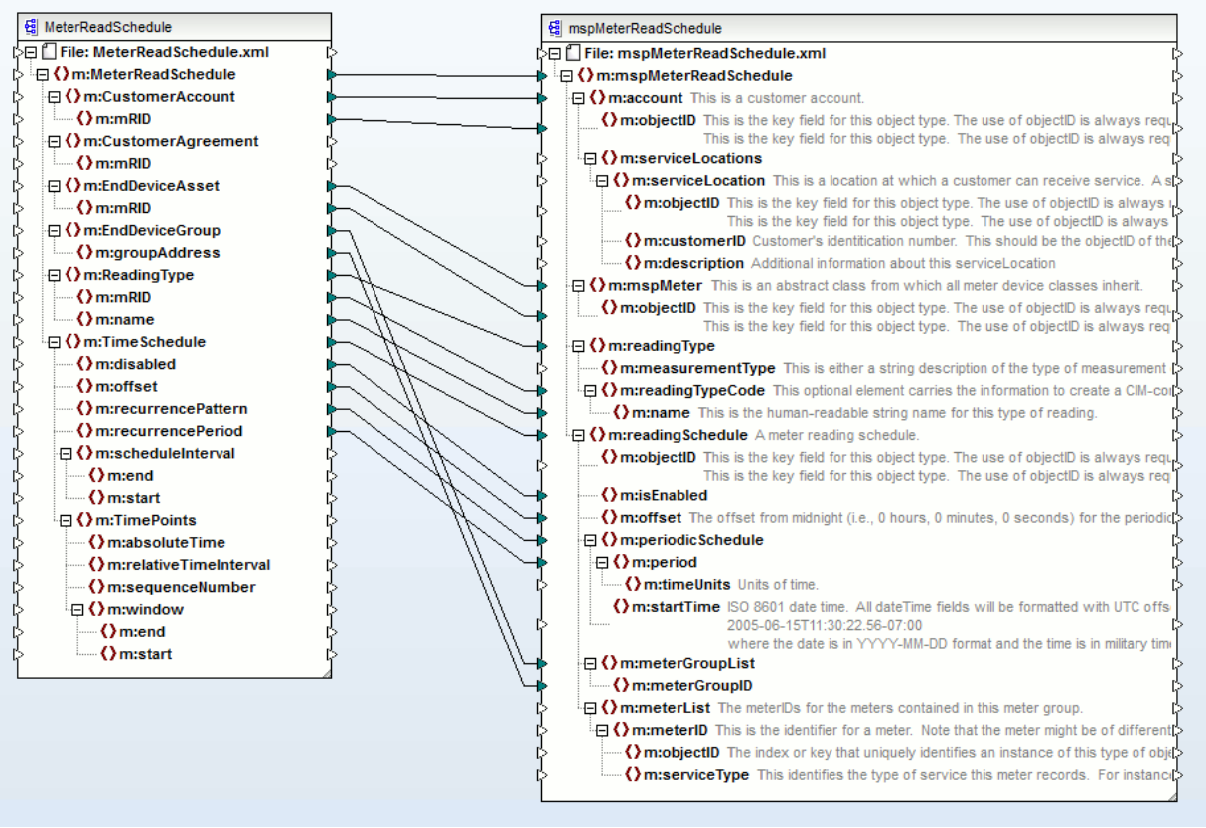

Figure 3-16
MeterReadSchedule CIM to MultiSpeak Profile Mapping

Meter Service Requests

Meter Service Requests profile handles service requests for meters which may also include a meter swap-out, and meter readings for both the old (final) and new (initial) meters, if the meter needs to be replaced. This is a fairly common scenario as even if a meter can be repaired it is usually brought back to the meter shop for the repair and a replacement meter installed in the field as a result of the service work.

For the CIM, ReadingType is handled a bit differently in this profile than in the MeterReadings profile. In this profile instead of having the full structure that is seen in the MeterReadings profile, there is only an mRID, name, and optionally, a unit. For the MeterAsset (representing the new meter) and OldMeterAsset (for the old meter) the ReadingType in the associated MeterReadings is a ref.

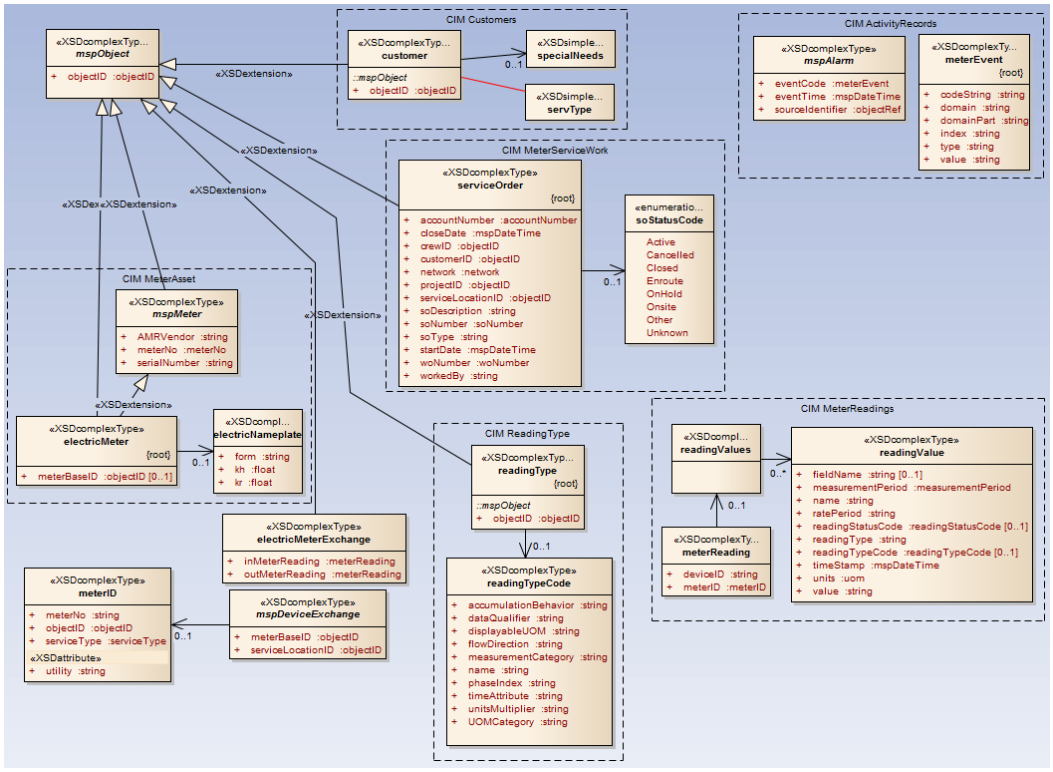

Figure 3-17
UML of MultiSpeak Equivalent of the CIM MeterServiceRequests Profile

Table 3-48
MeterServiceRequests Correlations

CIM	MultiSpeak	Comments
ReadingType .mRID	readingType.objectID	
Customers .kind	customer .servType	
MeterAsset .amrSystem .serialNumber .kh .kr	electricMeter .amrVendor .serialNumber electricNameplate .kh .kr	

Meter Readings

The CIM Meter Service Work profile also contains a set of meter reads for both the old and new meter (if appropriate for the profile). See the paragraph on MeterReadings for the associated mappings.

Table 3-49
MeterServiceRequests Transformations

CIM	MultiSpeak	Comments
amrSystem	amrVendor	
MeterServiceWork .kind	serviceOrder .soType	
Customers .specialNeed .vip	customer .specialNeeds	Concatenate CIM specialNeed and vip into MultiSpeak specialNeeds
CustomerAgreement .mRID	serviceOrder .accountNumner	MultiSpeak does not support CustomerAgreement but the mRID can be mapped to an accountNumber to identify the customer

Activity Records

The CIM Meter Service Work profile also contains a set of Activity records. See the paragraph on End Device Events to see how the category, severity, reason, could be mapped or substringed to the appropriate MultiSpeak classes and attributes.

ServiceDeliveryPoints

Addresses are included as part of the service delivery point information. For the address mappings see Appendix A, Table A-1 Address Correlations.

For more information see the paragraph on ServiceDeliveryPointConfig.

Table 3-50
MeterServiceRequests Gaps

CIM	MultiSpeak	Comments
Customer .pucNumber		
Status		
MeterServiceWork .priority .category .lastModifiedDateTime		
MeterAsset .formNumber		

Meter System Events

The intent of this profile is to capture events that may occur throughout the distribution network. There are three high level structures in this profile, ComMediaAsset (what the device uses to communicate), EndDeviceAsset, (the device itself), EndDeviceEvent (information relating to the event).

For EndDeviceEvent mappings see the paragraph End DeviceEvents.

The ComMediaAsset only contains an identifier for the communication device, and optionally, a serialnumber.

One challenge is that the “end device” in MultiSpeak is the meter, where in CIM it is meters or other devices that can communicate. In MultiSpeal mspMeter is an abstract class, which happens to contain everything needed to map to an EndDeviceAsset, but as an abstract class, would normally be inherited by electricMeter, which is not semantically the same.

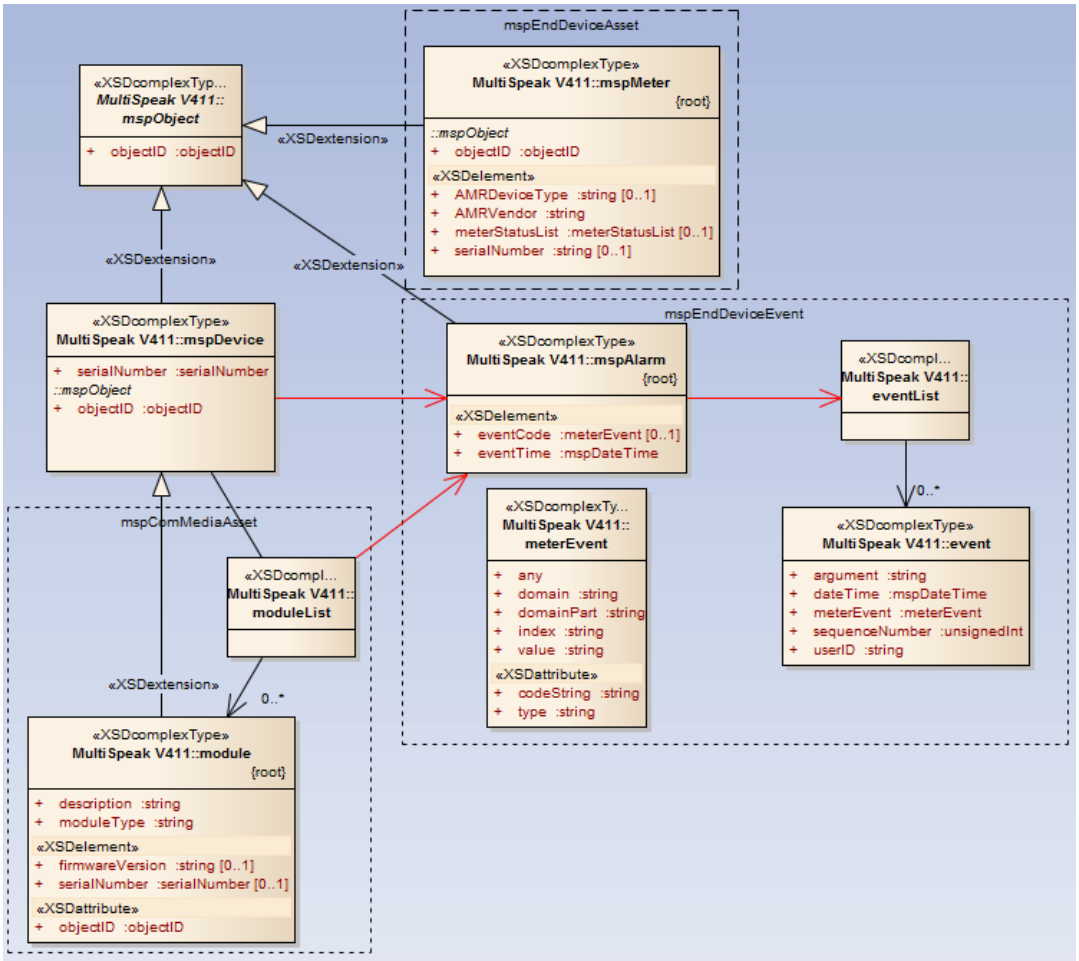

Figure 3-18
UML of the MultiSpeak Equivalent to the CIM MeterSystemEvents Profile

Table 3-51
MeterSystemEvents Correlations

CIM	MultiSpeak	Comments
ComMediaAsset .mRID .serialNumber	module .objectID .serialNumber	
EndDeviceAsset .mRID .serialNumber .amrSystem	mspMeter .objectID .serialNumber .amrVendor	

For information on the mapping of the EndDeviceEvents portion of this profile, see the paragraph, End Device Events.

Pricing Structure Config

The PricingStructureConfig profile is what is used to communicate tariff related information. In addition to core tariff, CIM also supports consumption and time-based tariff intervals.

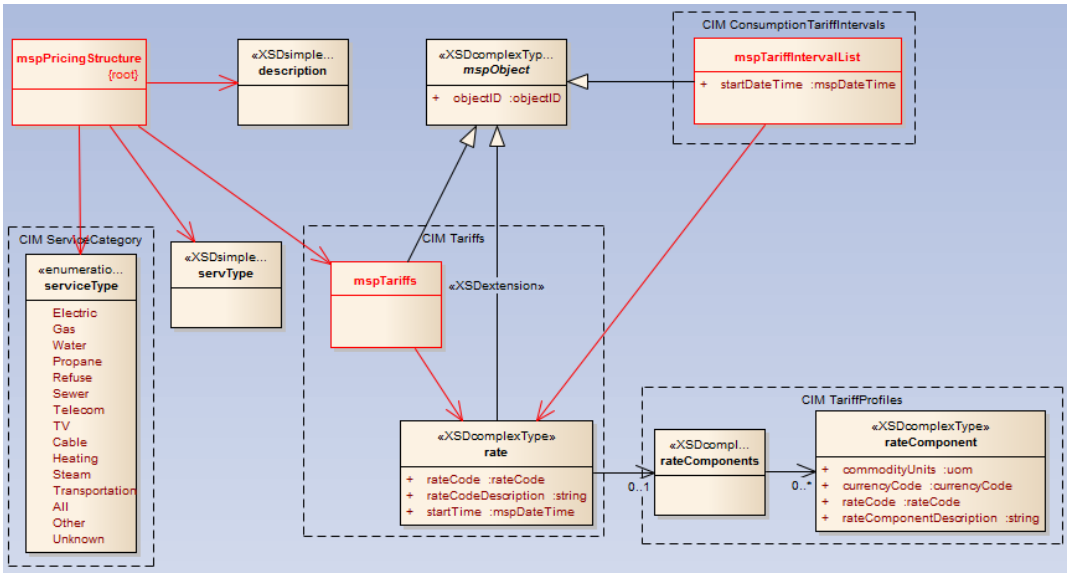

Figure 3-19
UML Representation of the MultiSpeak Equivalent of the CIM PricingStructureConfig

Table 3-52
PricingStructureConfig Correlations

CIM	MultiSpeak	Comments
PricingStructure .description .code	mspPricingStructure .description .rateCode	
Tariffs.description	mspTariffs .rateComponentDescription	

Table 3-53
PricingStructureConfig Transformations

CIM	MultiSpeak	Comments
PricingStructure	mspPricingStructure	Each of these "msp" classes were abstract classes added to be placeholders to facilitate mapping to the CIM profile
Tariffs .mRID .startDate	mspTariffs .objectID .startTime	
	mspTariffIntervalList	
PricingStructure . .revenueKind	mspPricingStructure .servType	Residential, Commercial, etc.
ServiceCategory .name	serviceType	serviceType in MultiSpeak is an enumeration of the many types of services that could be modeled
TariffProfiles	rateComponents	
fixedPortion .energyUnit .monetaryUnit	rateComponent .commodityUnits .currencyCode	
ConsumptionTarrifIntervals	mspTariffIntervalList	Using this abstract class as a placeholder the rate, rateComponents, and rateComponent classes could be used to model consumption or timeTariff intervals
TimeTariffIntervals	mspTariffIntervalList	
ChildCharges.name	rateCode	
Charges.mRID	mspTariffIntervalList .objectID	

Table 3-54
PricingStructureConfig Gaps

CIM	MultiSpeak	Comments
Status		
docStatus		Is of type CIM Status
variablePortion		CIM allows for the distinction of a fixed and variable portion of a tariff
fixedPortion		
code revisionNumber title		Each of these attributes are in the CIM profile, but are not defined in the CIM specification, nor in the CIM model.
Tariffs .endDate .lastModifiedDateTime		Nominally there should not be an empty date in a tariff so the startDate of the next tariff is the endDate of the previous tariff
ChildCharges .kind .variablePortion		
fixedPortion .energyUnit .monetaryUnit .multiplier .value		

The key gap in MultiSpeak in relation to the variable and fixed portion of the CIM profile is the lack of a corresponding value and multiplier that the energy or monetary units apply to. MultiSpeak has a currencyCOde and commodityUnits, but no attribute to say "how many".

Receipt Record

The ReceiptRecord profile is used to pass payment information from one system to another. The CIM profile defines both credit card and check transactions. MultiSpeak handles these two transaction types but also includes cash and ACH transactions. CIM also includes vendor or cashier information related to the transaction where MultiSpeak does not include this payment handling information.

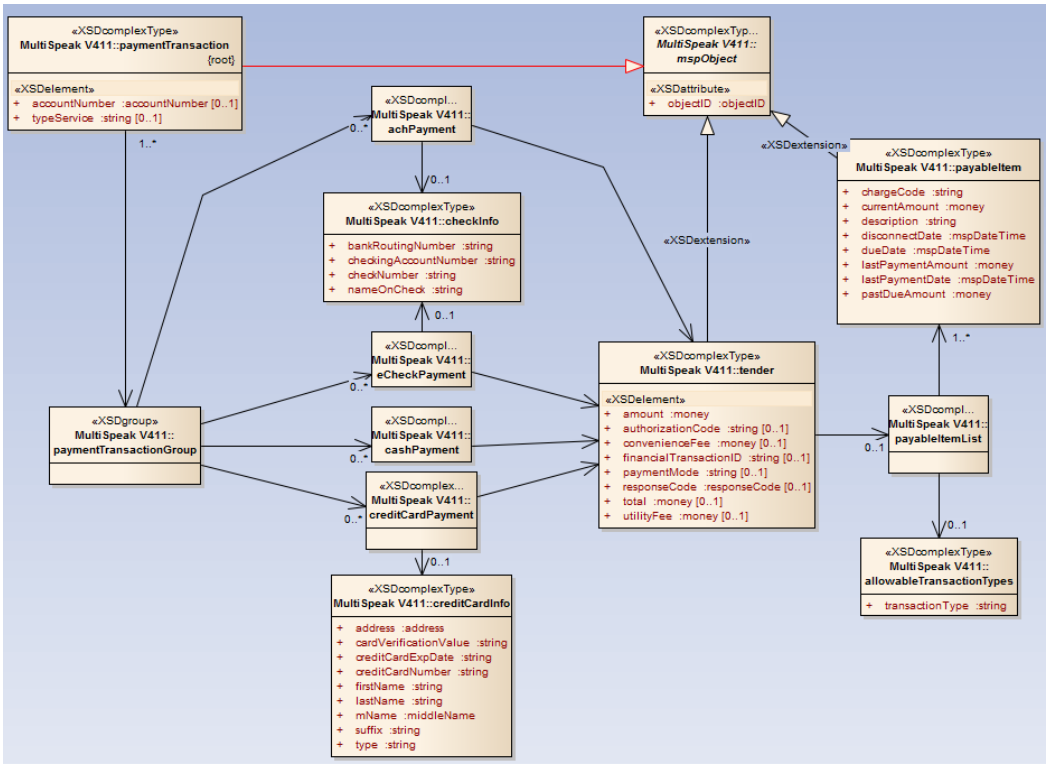

Figure3-20
UML of the MultiSpeak Equivalent of the CIM ReceiptRecord Profile

Table 3-55
ReceiptRecord Correlations

CIM	MultiSpeak	Comments
Receipt .mRID	paymentTransaction .objectID	paymentTransaction has an objected if it inherits from mspObject (see UML above)
Card .expiryDate .cvNumber .pan	creditCardInfo .creditCardExpDate .cardVerificationValue .creditCardNumber	
Cheque .chequeNumber .micrNumber	checkInfo .checkingAccountNumber .bankRoutingNumber	
bankAccountDetail .holderName .accountNumber	checkInfo .nameOnCheck .checkNumber	
line .amount .dateTime .note	payableItem .currentAmount .dueDate .description	

Table 3-56
ReceiptRecord Transformations

CIM	MultiSpeak	Comments
Receipt.description	paymentTransaction .typeService	
Card .accountHolderName	creditCardInfo .firstName .lastName .mName .suffix	The CIM profile has only a single name field, which would need to be substringed into the appropriate sub-fields.

Table 3-57
ReceiptRecord Gaps

CIM	MultiSpeak	Comments
CashierShift		
VendorShift		
	paymentTransaction .accountNumber	Odd that the account for which a payment might be made is not included in the CIM profile.
Receipt.isBankable		
	creditCardInfo.type	Visa, Mastercard, etc
	creditCardInfo.address	
Cheque .date .kind		Nominally the kind attribute could be used to denote normal check or ACH
bankAccountDetail .bankName .branchCode .holderID		
	cashPayment	MultiSpeak has significant improvements over CIM for this profile with support for both ACH and cash transactions
	achPayment	
line.rounding		

Table 3-57 (continued)
ReceiptRecord Gaps

CIM	MultiSpeak	Comments
	payableItem .chargeCode .disconnectDate .lastPaymentAmount .lastPaymentDate .pastDueAmount	
	payableItemList .total .utilityFee	
	tender .authorizationCode .financialTransactionID .paymentMode .responseCode .amount .convenienceFee	

This profile could be used to identify payers that may or may not be part of the utility for example, service stations or banks that take payments on behalf of the utility, or third party aggregators. The CashierShift or VendorShift classes are used to capture who is taking the payment.

SDP Location Config

The SDPLocation profile is somewhat different than Service Delivery Point configuration. SDPLocationConfig contains additional position point information in addition to mail address and contact information.

This profile uses an abstract class mspSDlLocation as a placeholder to tie the other MultiSpeak related classes used to build this profile.

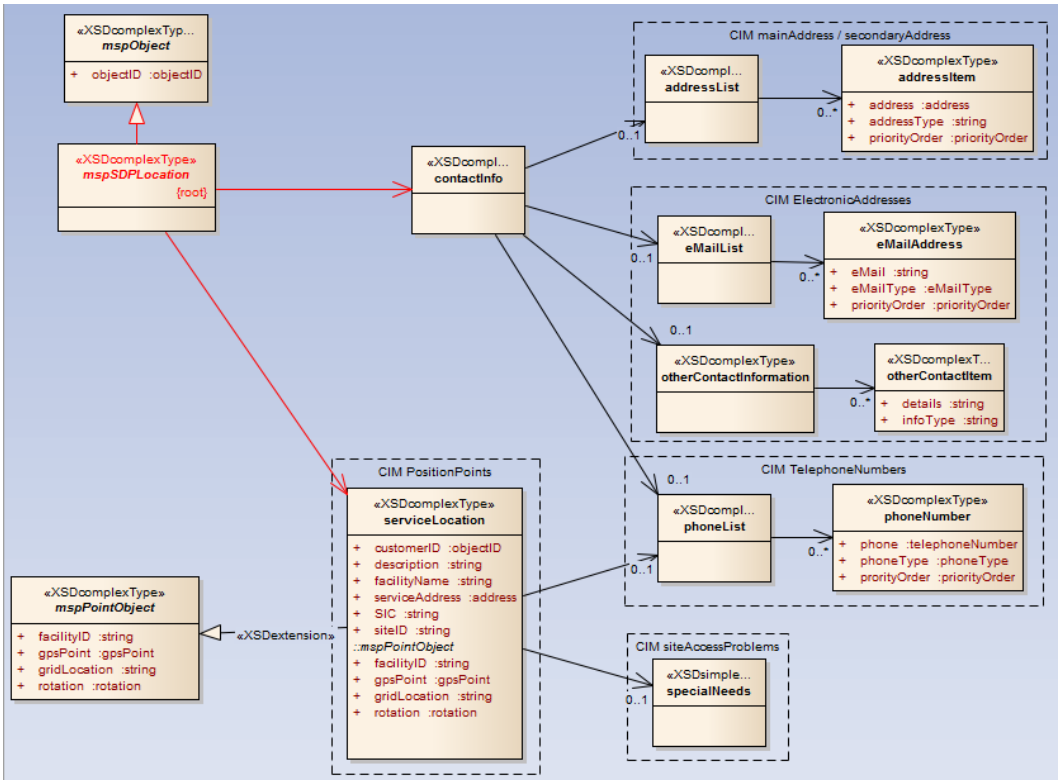

Figure 3-21
UML of the MultiSpeak Equivalent of the CIM SDPLocationConfig Profile

Table 3-58
SDPLocationConfig Correlations

CIM	MultiSpeak	Comments
PositionPoints .xposition .yPosition .zPosition	gpsPoint .longitude .latitude .altitude	
SDPLocation .corporateCode	serviceLocation .SIC	

For Address related mappings, see Appendix A, Table A-1.

For Phone related mappings, see Appendix A, Table A-2.

Table 3-59
SDPLocationConfig Transformations

CIM	MultiSpeak	Comments
SDPLocation .mRID	mspSDPLocation .objectID	A transformation as mspSDPLocation is an added abstract class used only for this profile
SDPLocation .accessMethod .remark .siteAccessProblem .description	serviceLocation .description	The three CIM attributes can be concatenated and mapped to the MultiSpeak description attribute

For electronic address related transformation mappings, see Appendix A, Table A-4.

Table 3-60
SDPLocationConfig Gaps

CIM	MultiSpeak	Comments
PositionPoints .sequenceNumber .isPolygon		The sequence number can be used to indicate a series of numbers and isPolygon, a Boolean, to indicate if the resulting set of coordinates closes.
SDPLocation .occupancyDate .category .direction .geoInfoReference		
Status		

Service Category Config

The Service Category Cpmfog profile is used to pass information about the different types of services that may be used by an organization (e.g. electric, gas, or water). This is the smallest CIM profile and essentially reflects an enumeration of these different categories. The main different with MultiSpeak is that the equivalent category, serviceType, is associated with objectRef. Within MultiSpeak objectRef can be used to refer to an account, a service, or a serviceLocation, all of which may have a serviceType associated with it.

Table 3-61
ServiceCategoryConfig Correlations

CIM	MultiSpeak	Comments
mRID	meterNo.objectID	
description	description	

Table 3-62
ServiceCategoryConfig Transformations

CIM	MultiSpeak	Comments
Kind	serviceType	Both attributes are enumerations of the types of services that are supported

Table 3-63
ServiceCategoryConfig Gaps

CIM	MultiSpeak	Comments
name aliasName		name and aliasName are both optional in this profile and nominally the Name class and identifiedObject class in MultiSpeak could be used to map these two attributes
	meterNo utility	Since serviceType is always associated with a meter, meterNo and utility are important attributes, especially as MultiSpeak supports the concept of meters being read by a utility on the behalf of another utility

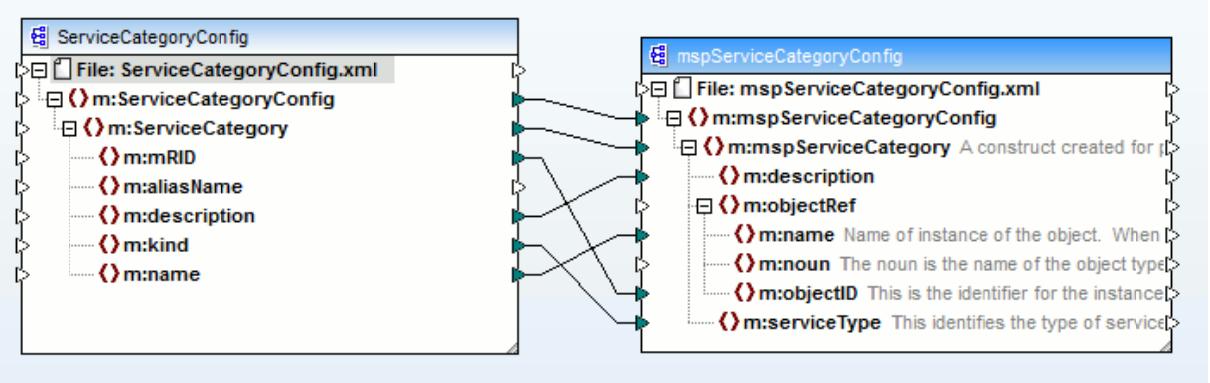

Figure 3-22
ServiceCategoryConfig Profile Mapping

Service Delivery Point Config

This profile contains information about a given service delivery point, e.g. phase, current, power, voltage, and related billing information. It also contains references to pricing structures (tariffs), SDPLocations, service category, service location, and service supplier.

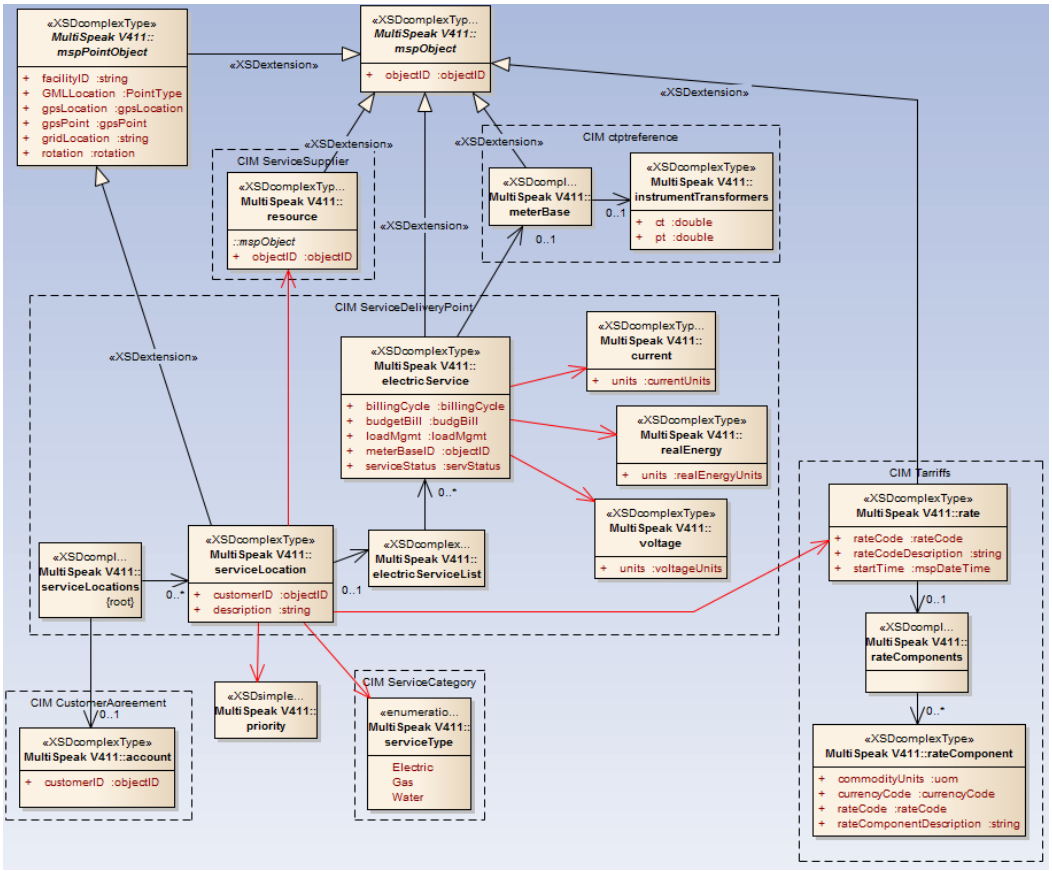

Figure 3-23
UML for MultiSpeak Equivalent of the CIM ServiceDeliveryPointConfig Profile

Table 3-64
ServiceDeliveryPointConfig Correlations

CIM	MultiSpeak	Comments
ServiceDeliveryPoint .mRID .billingCycle .budgetBill .description .loadMgmt .nominalServiceVoltage .ratedCurrent .ratedPower .servicePriority	serviceLocation .objectID electricService.billingCycle electricService.budgetBill .description electricService.loadMgmt voltage.units current.units realEnergy.units .priority	

Table 3-65
ServiceDeliveryPointConfig Transformations

CIM	MultiSpeak	Comments
CustomerAgreement .mRID	account .customerID	MultiSpeak 4.1.5 does not support customer agreements, but the mRID of an agreement can be mapped to a customer account
ServiceDeliveryPoint .ctptReference	instrumentTransformers .ct .pt	CIM uses a single attribute where MultiSpeak splits this between ct/pt.
.serviceDeliveryRemark	serviceLocation .description	serviceDeliveryRemakr could be concatenated with description and mapped to the MultiSpeak description field
PricingStructures .mRID .name	rate .objectID .rateCodeDescription	
Tarriffs	rateComponents	CIM tariffs has an identifier (from identifiedObject) and name, where in MultiSpeak 4.1.5 this is a holder for one or more rateComponent
TariffProfiles .name	rateComponent rateComponentDescription	

Table 3-65 (continued)
ServiceDeliveryPointConfig Transformations

CIM	MultiSpeak	Comments
ServiceSupplier.mRID	resource.objectID	ServiceSupplier in CIM identifies personnel (utility or contractor) that provides a service with associated contact information. Resource in MultiSpeak 4.1.5 can be used to identify similar contact information (by adding an association) but nominally this is for physical assets, not people.
ServiceCategory.name	serviceType	Normally in the CIM the mRID of a ServiceCategory would be used to uniquely identify the specific item.

See the paragraph for Supplier Config mapping to see how adding an association to MultiSpeak contact related classes very closely emulates the CIM ServiceSupplierConfig profile.

Table 3-66
ServiceDeliveryPointConfig Gaps

CIM	MultiSpeak	Comments
ServiceDeliveryPoint .checkBilling .consumptionRealEnergy .estimatedLoad .grounded .phaseConfig		checkBilling is a boolean that reflects that this is the result of of an inspection or that a previous billing was in error
Tariffs.mRID TariffProfiles.mRID		CIM uses mRID to identify tariffs and trariffsProfiles
	rate .rateCode .startTime rateComponent .commodityUnits .currencyCode .rateCode	MultiSpeak uses the attributes that contain the rate information rather than a identifier, e.g. objected like the CIM does

Table 3-66 (continued)
ServiceDeliveryPointConfig Gaps

CIM	MultiSpeak	Comments
	serviceLocation .facilityID .GMLLocation .gpsLocation .gpsPoint .gridLocation .rotation	MultiSpeak has a variety of ways to identify a location beyond the address that the CIM uses.
SDPLocations.mRID		

Service Location Config

There is a high degree of correlation between the CIM ServiceLocationConfig and the equivalent MultiSPeak profile, mostly due to the commonality of address related information between the two standards.

Table 3-67
ServiceLocationConfig Correlations

CIM	MultiSpeak	Comments
mRID	customertID	customerID is type objectID
City	city	

For address correlations see Appendix A, Table 3-69 Reused Address Correlations.

Table 3-68
ServiceLocationConfig Transformations

CIM	MultiSpeak	Comments
typeServiceLoc	SIC	typeServiceLoc is classification for the type of location according to the utility while SIC is a standard industry code. It may be more appropriate to use CIM Organisation.code to made to MultiSpeak SIC.

For ServiceLocation related transformations see Appendix A, Table 3-71 Reused Service Location related transformations.

Table 3-69
ServiceLocationConfig Gaps

CIM	MultiSpeak	Comments
inspectReq		There is an inspection class in MultiSpeak but this is more appropriate for capturing data related to an inspection, not flagging that one needs to occur.
	faciltyName	
	townCode	
	buildingNumber postOfficeBox region streetNumber streetPrefix streetSuffix streetType	The detailedAddressFields provide more address information and while used in other CIM profiles, e.g. ServiceSupplierConfig, this additional information is not used in this profile.

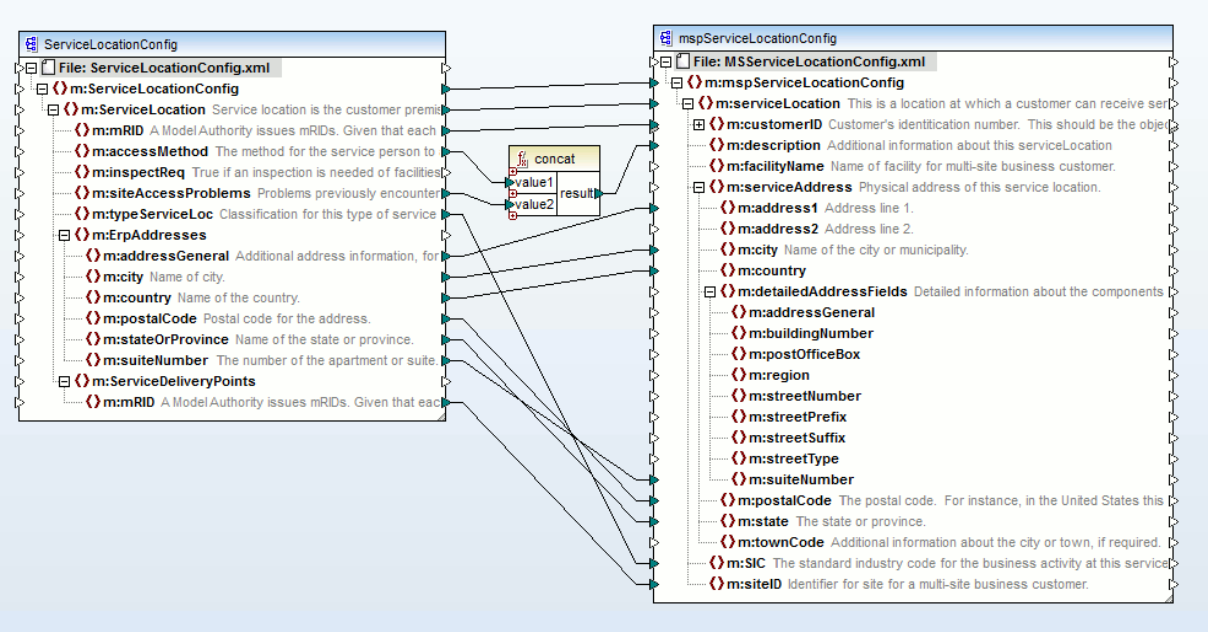

Figure 3-24
ServiceLocationConfig CIM to MultiSpeak Mapping

Of interest to the ServiceLocationConfig harmonization discussion is the Network class from MultiSpeak offers additional information that may be of use in reference to a service location. However, it should be noted.

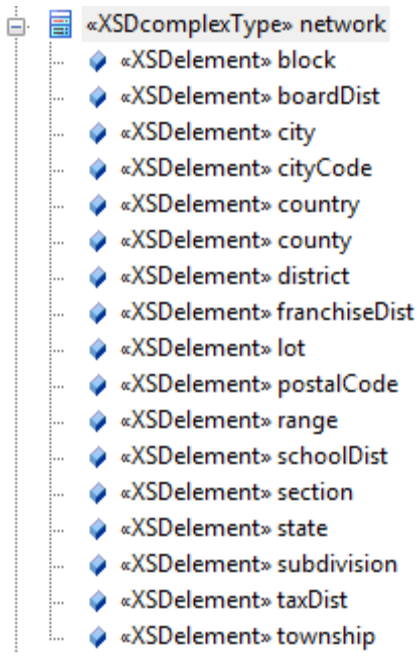

Figure 3-25
MultiSpeak Network Class

Supplier Config

The Supplier Config profile in CIM is used to transfer information about the service supplier, whether they work for the utility, a retailer, or other entity, and all of the contact information associated with the supplier. Since this information does not have an equivalent in MultiSpeak it would need to be concatenated and mapped to the utility field that resource inherits from mspObject.

MultiSpeak does not have this concept of a service supplier organization, but has numerous classes to capture both information related to a work item and classes to capture contact information. However, the contact information is associated with the high level mspPerson. Useful for mapping purposes is a container class, resource, available that is used for work related items that also inherits from mspObject. Creating a MultiSpeak equivalent of Supplier Config profile is simply a matter of adding an association from the resource class to the contactinfo class shown here. A notable gap, since MultiSpeak does not have this Supplier entity concept, is the issuerIdentificationNumber used by CIM to identify this organization (per ISO/IEC 7812-1 and ISO/IEC 7812-2) although nominally this could be mapped into the MultiSpeak objectID (concatenated with the mRID mapping from CIM).

In this profile both MultiSpeak and CIM use complete addressing information (where in the ServiceLocationConfig the CIM used limited address information), which maps very well. MultiSpeak using a construct where there can be many types of addresses which are tracked by an address type so it makes for a fairly clean hierarchy. CIM on the other hand calls out two types of addresses, street address and postal address in its profile. To map this second class MultiSpeak can simply instantiate a new instance of address and assign it a new type.

The other challenge with this profile is that the CIM captures other types of electronic address besides email addresses; LAN, radio, and web, which are not included in MultiSpeak. However, Multispeak has a generic otherContactInformation class that can be used to capture this information; again, it can be instantiated with a type to indicate the type of information being captured and the information, LAN address, web address, or radio address, mapped to the details attribute.

The final challenge is that the CIM profile uses the Status class construct and mRID, name, description fields (from identifiedObject) when it is not clear what value these classes and fields add to this profile. By tying the contact information to the high level ServiceSupplier this provides all the identification that the contact information should need to associate a given instance of contact information with its organization.

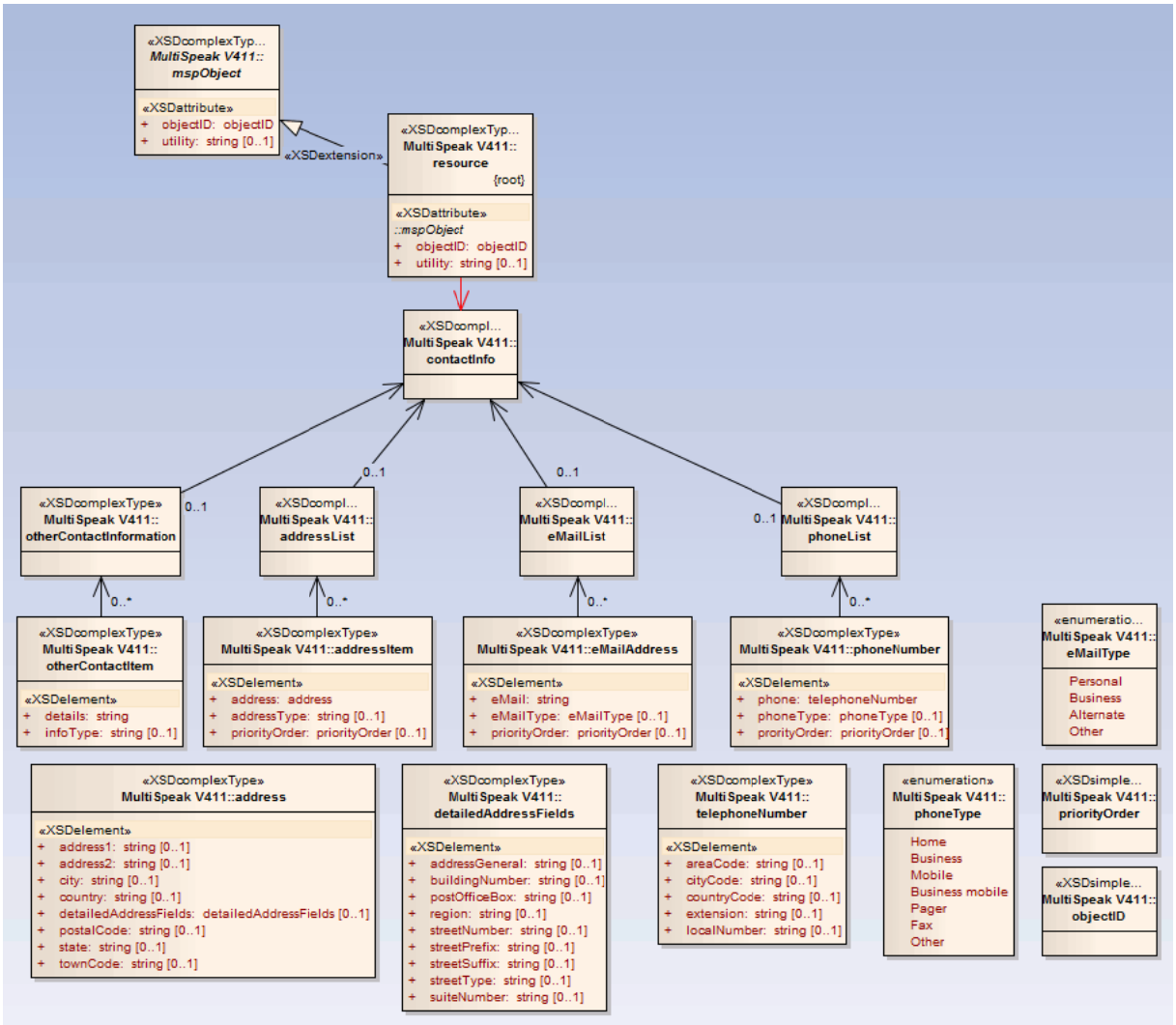

Figure 3-26
UML of MultiSpeak Equivalent Servicesupplierconfig Profile

Table 3-70
Service Supplier Correlations

CIM	MultiSpeak	Comments
mRID	resource.objectID	
email	eMail	

For the address related mappings for this profile see Appendix Table A-1 Address Correlations.

Table 3-71
ServiceSupplierConfig Transformations

CIM	MultiSpeak	Comments
kind name	resource.utility	Kind identifies whether the service supplier is a utility, retailer, or other. This information would need to be concatenated and placed in the utility attribute that resource inherits
postalAddress/ streetAddress	address	Not a transformation per se, but CIM calls out a specific class while MultiSpeak uses a concept where additional addresses can be added by instantiating a new class with the type being noted addressType
LAN web radio	otherContactItem	MultiSpeak does not have these equivalent fields but allows for additional contact information to be adding by instantiating new otherContactItems
Telephone. description	phoneType	Description is inherited in the CIM class, but the phoneType in MultiSpeak is not a direct correlation but is a contrstruct that allows MultiSPeak to add new phone records associated with an entity.

Table 3-72
ServiceSupplierConfig Gaps

CIM	MultiSpeak	Comments
Status		It is not clear what value having a Status associated with ServiceSupplier adds for this profile
issuerIdentifcation_ Number		This is used by CIM per ISO/IEC 7812-1 and ISO/IEC 7812-2, prefix number issued to an entity by the International Standards Organisation for the purpose of tagging onto electronic financial transactions
description, name, mRID, aliasName		In some cases this inheritance is passed down to subordinate classes as required where it appears this information could be optional.
	addressType phoneType infoType emailType	MultiSpeak uses these attributes to add additional instantiations of a given class of contact information
	priorityCode	MultiSpeak provides for the ability to have multiple types of contact information, but give information regarding the order in which they should be used

Transaction Record

The transaction record as a concept does not exist in MultiSpeak 4.1.5 so as can be expected that are some significant gaps in this profile. Additionally, as there is no equivalent root class for this profile in MultiSpeak, one was added (shown in red in the figure) to be the placeholder for the associations that MultiSpeak does support.

This profile could be used to identify payers that are not part of the utility for example, service stations or banks that take payments on behalf of the utility, or third party aggregators.

In CIM an account is associated with one rate tariff. In MultiSpeak an account can have multiple service locations that can have multiple services. Each service is associated with one rate tariff.

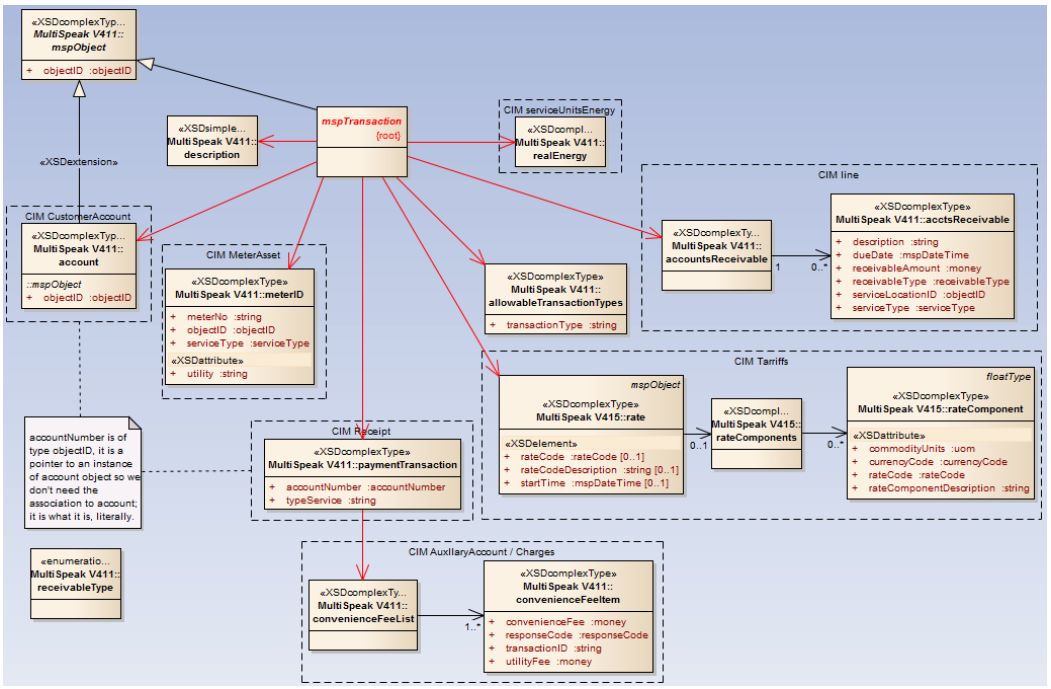

Figure 3-27
UML of MultiSpeak Equivalent of CIM TransactionRecord

Table 3-73
TransactionRecord Correlations

CIM	MultiSpeak	Comments
description	description	
CustomerAccount .mRID	account .accountNumber	
MeterAsset .mRID	meterID .meterNo	

Table 3-74
TransactionRecord Transformations

CIM	MultiSpeak	Comments
Transaction .mRID	mspTransaction .objectID	While this would normally be a correlation, mspTransaction is an addition to the model to facilitate the mapping of this profile
Transaction .kind	allowableTransactionTypes .transactionType	
Transaction .serviceUnitsEnergy .serviceUnitsError	mspTransaction .realEnergy	The transaction may be for energy, or a correction to the energy transaction so this is mapped using a logical OR
PricingStructures .mRID .name	rate .objectID .rateCodeDescription	
Tarriffs	rateComponents	CIM tariffs has an identifier (from identifiedObject) and name, where in MultiSpeak 4.1.5 this is a holder for one or more rateComponent
TariffProfiles .name	rateComponent .rateComponentDescription	
Charges	acctsReceivable	
.description	.description	
Receipt .mRID .description	paymentTransaction .accountNumber .typeService	
line.amount	acctsReceivable .receivableAmount	

Table 3-75
TransactionRecord Gaps

CIM	MultiSpeak	Comments
VendorShift		
CashierShift		
UserAttributes		
Transaction .diverseReference .donorReference .receiverReference .reversedId		
AuxillaryAccount		
	meterID .serviceType .utility	MultiSpeak provides more information to identify the meter beyond just the meterNo/objected
ChildCharges		
	acctsReceivable .receivableType .serviceLocationID .serviceType	Nominally CIM line class maps to MultiSpeak acctsReceivable, however MultiSpeak provides some additional attributes for receivables
line .note .rounding		

Section 4: On Demand Read Proof of Concept

On Demand Read is a well understood use case in the Advanced Metering Infrastructure (AMI) domain. There presumption is that a customer or a customer service representative on the behalf of a customer will initiate a meter reading in real time or semi real time. This use case demonstrates the data flow from the back-office (or service that may be provisioned for the customer via the back-office), through middle-ware, to the system that retrieves the reading from the specified meter. The high-level diagram for this concept is soon below.

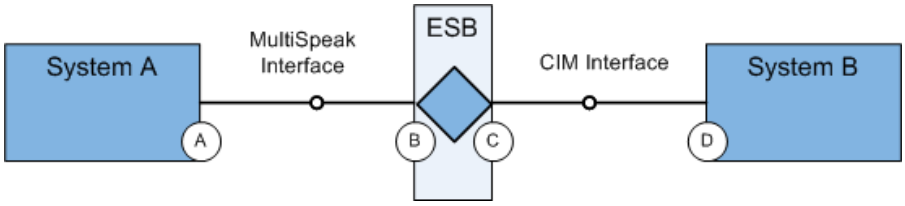

Figure 4-1
High Level Integration Concept

Normally in a MultiSpeak environment an InitiateMeterReadingsis used to retrieve meter readings from the Meter Reading (MR) server). Once the MR server has sent the request and the meter readings have been retrieved a MultiSpeak ReadingChangedNotification service is used to send the meter reading to the calling system. In this example then both System A and System B as shown above would be using MultiSpeak based services.

For this proof of concept however to retrieve the meter reading a CIM-based service would be used. At the time of this writing the CIM has a meterreading profile and of course the model in its entirety, but no standardized service call. However, a draft standard is available to guide this integration attempt, IEC 61968-100 (in committee draft stage at this point).

Using guidance from this document a GetMeterReadings service is crafted. Using the MultiSpeak and CIM-based WSDLs, the service are now ready to be "wired up" using OpenESB[6] as the proof of concept middleware. The wire diagram is shown below with InitiateMeterReadings show on the top left, GetMeterreadings on the right, and finally ReadingChangeNotification on the bottom left.

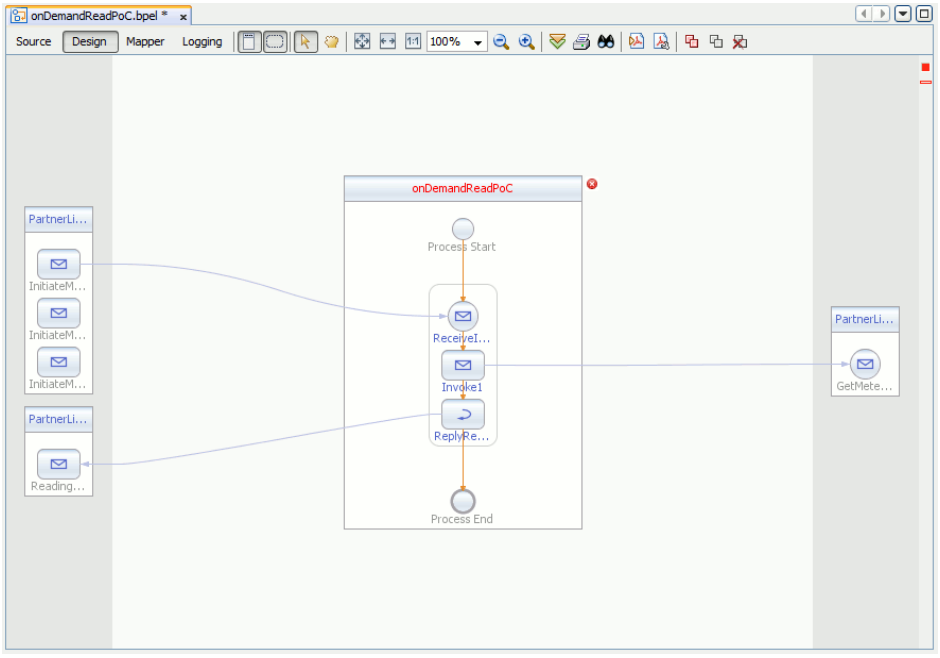

Figure 4-2
OpenESB Wire Diagram for MultiSpeak to CIM Proof of Concept

Once the services are "wired up" mapping of the payloads need to occur.

First of all both MultiSpeak and CIM use properties in their message header that are important for systems integration.

The following table notes the suggested mappings for the message headers.

[6] OpenESB – open source enterprise service bus http://wiki.open-esb.java.net/

Table 4-1
MultiSpeak – CIM Message Header Mapping

MultiSpeak attribute	CIM attribute	Comments
N/A - The verb is included as part of the operation naming.	Verb	Identifies the action to be taken
N/A - The noun is included as part of the operation naming.	Noun	Identifies the type of payload
MajorVersion MinorVersion Build Branch BuildString	Revision	For a mapping to occur the integer type version attributes need to be changed to string; the resulting attributes can then be concatenated and mapped to Revision
AppSource AppVersion Company	Source	identifying the source of the message, which should be the ID of the system or organization; for MultiSpeak the AppSource, AppVersion and Company may be concatenated and mapped to this field
MessageID	ReplayDetection (Nonce, Created)	A complexType made up of Nonce (a GUID or random number good for at least a day) and Created, a timestamp.
Timestamp	Timestamp	A timestamp generated by the source system to indicate when the message was created.
transactionID - The transactionID is not included in the message header, rather it is included in the operation definition for those operations for which it is required.	CorrelationID	Can be supplied on a request, so that the client can correlate a corresponding reply message. The server will using the incoming correlation ID on the outgoing reply.

Table 4-1 (continued)
MultiSpeak – CIM Message Header Mapping

MultiSpeak attribute	CIM attribute	Comments
AuditID N/A	User (UserID, Organization)	User is a complex structure that identifies the user and associated organization. Should be supplied as it may be required for some interfaces, depending upon underlying implementations. MultiSpeak AuditID is mapped to UserID
responseURL - the responseURL is not included in the message header, rather it is included in the definition of the operation for those operations for which it is appropriate.	ReplyAddress	responseURL is not part of the MultiSpeakMsgHeader, but part of the payload, but can be mapped to ReplyAddress
N/A	Comment	A field to add optional information about the service
Context	Context	Indicates the message pattern, e.g. Request, Response
N/A (The necessity of acknowledgement is inherent in the definition of the operations and is not included in the message header).	AckRequired	A flag indicating the an acknowledgement should be sent to the calling system
N/A	Property (Name, Value)	A way to optionally extend the header with other attributes that are not included in IEC 61968-100
UserID	N/A	This is an authentication parameter for the system that has sent the message.
Pwd	N/A	This is an authentication parameter for the system that has sent the message.
DefaultCurrencyCode	N/A	ISO 4217 code for the currency being used in the transaction

Table 4-1 (continued)
MultiSpeak – CIM Message Header Mapping

MultiSpeak attribute	CIM attribute	Comments
CSUnits	N/A	Any of the CS or coordinate system items are only appropriate if geographical data are being sent. In that case it is necessary to send the coordinate system name, CSUnits, and datum must be sent in order to be able to determine how to interpret the x,y,z coordinate data received.
CordinateSystemName	N/A	
CoordinateSystemAuthority	N/A	
CoordinateSystemAuthorityCode	N/A	
Datum	N/A	
SessionID	N/A	The SessionID and PreviousSessionID are for tightly coupled systems, e.g instance engineering analysis (DMS) and GIS. In this case, it is desirable to batch sets of edits into sessions which can be dealt with as transactions. The PerviousSessionID is then used to track the progress of a transaction and throwing exceptions if a particular transaction arrives out of sequence. This is necessary in installations where delivery is not guaranteed. MultiSpeak does not assume messaging middleware is in place to guarantee delivery.
PreviousSessionID	N/A	

Table 4-1 (continued)
MultiSpeak – CIM Message Header Mapping

MultiSpeak attribute	CIM attribute	Comments
ObjectsRemaining	N/A	The LastSent, ObjectsRemaining along with a calling parameter on some methods, lastReceived, enables two systems to coordinate chunking of data payloads. The sending system sends a chunk of data along with the LastSent header attribute and optionally an ObjectsRemaining. The receiver returns the value that it received in the LastSent in its subsequent call by putting that value in the lastReceived calling parameter. This goes back and forth until the sender returns a set of data with ObjectsRemaining set to 0.
LastSent	N/A	
RegistrationID	N/A	The identifier for an instance of a subscription for service. A potential subscriber applies to a server for a registrationID, which the subscriber uses to specify the data being requested. Subsequent data published pursuant to the subscription for service carries the registrationID so that the subscriber knows under which of potentially many subscriptions it is receiving this data.

Table 4-1 (continued)
MultiSpeak – CIM Message Header Mapping

MultiSpeak attribute	CIM attribute	Comments
AuditPassword	N/A	Used in conjunction with AuditID in MultiSpeak web services to authenticate a human user of an automation service and to provide audit logging capabilities for actions taken by a specific user.
Any Attribute	Any Element	

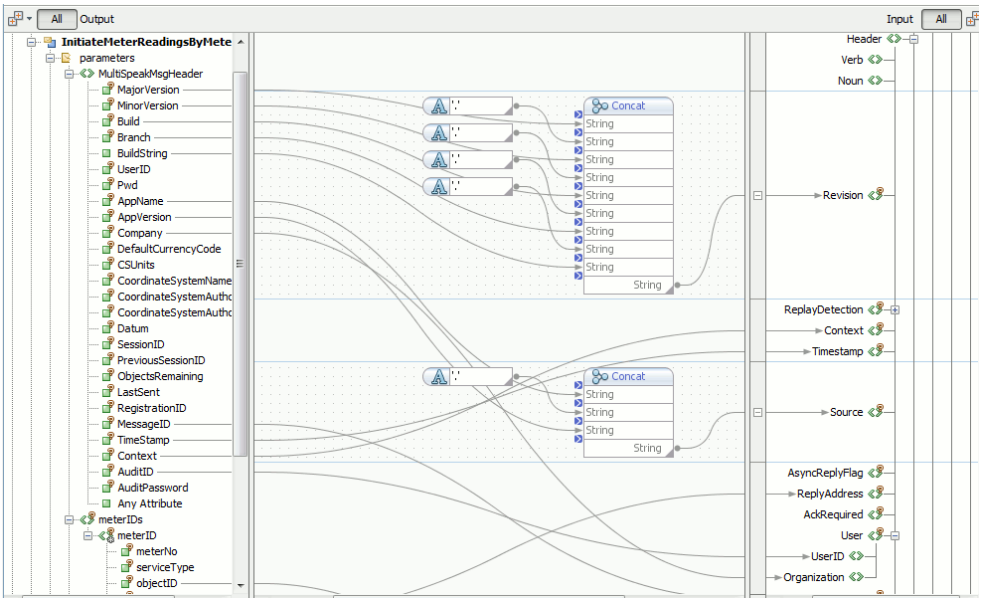

Figure 4-3
MultiSpeak – CIM Message Header Mapping

The meterID of the calling service is then mapped to mRID of EndDeviceAsset in CIM. Once the meter reading is returned it may be mapped via the forthcoming MeterReading profile harmonization.

The proof of concept demonstrates that it is possible to successfully integrate MultiSpeak and CIM-based applications in the same IT landscape. However, there are a couple of caveats. The main caveat being how closely the CIM and MultiSpeak profiles align. As has been seen profiles such as MeterSystemEvents and ServiceSupplierConfig have a high degree of correlation. However profiles that rely on the CIM document construct will show significant gaps as MultiSpeak does not currently support the generic document class construct. Finally any integration will fundamentally be based on the business requirements of the specific organization.

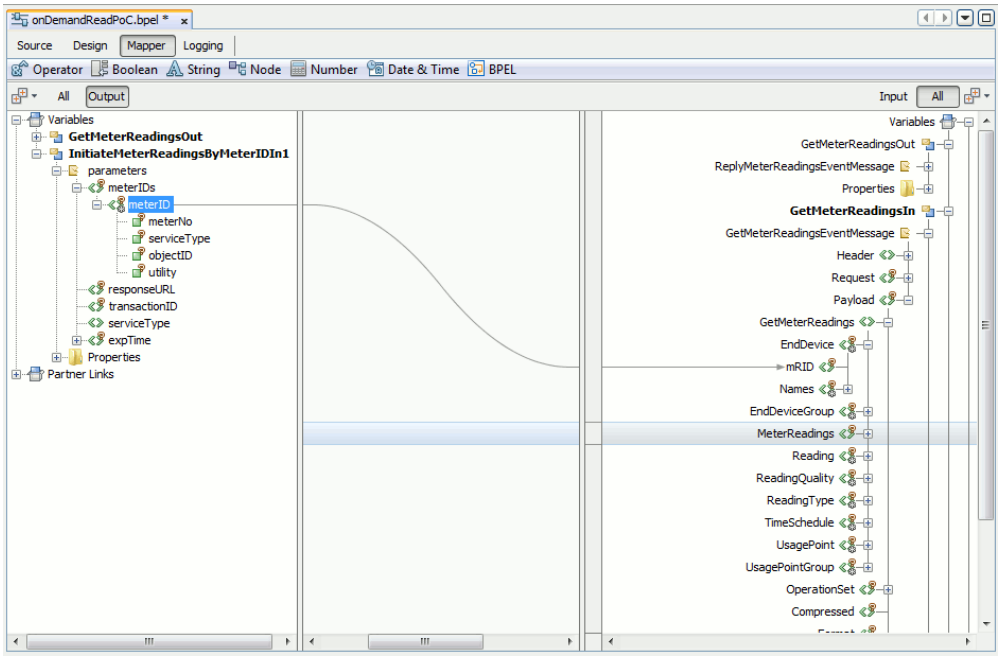

Figure 4-4
MultiSpeak meterID to CIM EndDeviceAsset.mRID Mapping

Section 5: Areas for Further Research

While this initial effort has developed a proof of concept for CIM to MultiSpeak integration and examples for creating MultiSpeak equivalents of CIM profiles, there are several areas that would benefit from more research. This effort included mappings of all 23 of the existing IEC 61968-9 1st edition profiles using MultiSpeak 4.1.5. The profiles also need to be mapped and validated against the 2nd edition of IEC 61968 profiles once they are published, which is expected in the first quarter of 2013.

This mapping was from CIM to MultiSpeak. For full integration there would also need to be a mapping from MultiSpeak to CIM. While for many attributes this would simply be a reverse of the CIM to MultiSpeak mapping, and gaps would still be gaps, this would validate those attributes where transformation occurred (e.g. string concatenation, logical functions). In practice the data that is passed may need to be validated, (looking for fixed length, special characters, keywords, etc) that determine how a string would be sub-stringed to map into the specific elements. This initial effort focused on IEC 61968-9 mapping and succeeding efforts may also want to examine IEC 61968-13 mapping.

While the mapping itself is of interest and provided specific guidance and useful findings that have been shared with both the CIM and MultiSpeak communities, the proof of the mapping would be for XSLTs to be created from the mappings and used in interoperation tests with the respective testing groups. For example, if a MultiSpeak service was transformed using an XSLT and passed the CIM interop test this would prove to the respective industries the value that the mapping provided and validate that CIM to MultiSpeak harmonization was viable. Participation in the interop may also reveal any errors that exist in the design of the mapping. Additionally, while the proof of concept investigated the use of a single use case, On Demand Meter Read, a more complete set of use cases could be developed showing the CIM to MultiSpeak and MultiSpeak to CIM through an ESB, providing a library of test integrations that the industry could leverage.

While the initial ontology investigation was put aside for pragmatic reasons, an ontological evaluation of the two standards could bear fruit. However this would take a significant effort, both because an RDF extract and import into an ontology would need to correctly record the classes, attributes, and class associations, but a reasonor would also need to be developed that could accurately assess the respective ontologies once they have been imported into an OWL tool. This would be a significant effort but one that may be used to further align the respective standards as they evolve.

Section 6: Summary

The initial research effort investigated the use of ontology tools for evaluating and comparing both CIM and MultiSpeak standards. After some difficulty in the creation and import of RDF files into the OWL tool, the effort was abandoned for a more pragmatic effort to map specific profiles of each standard.

This effort has resulted in mapping 22 of the 23 of the IEC 61968-9 1st edition profiles being mapped to MultiSpeak v4.1.5 equivalents. Since MultiSpeak equivalent profiles do not exist, a methodology for creating them was developed. The mapping effort noted correlations, gaps, and where transformation would need to occur to complete mapping from one standard to the other. The resulting gap analysis has been contributed to the respective standards bodies for evaluation and application to future editions. This should result in a gradual closing of the harmonization gap.

In addition, UML was developed that represents each MultiSpeak equivalent profile using the MultiSpeak standard as a data reference model[7]. CIMEA, a plug-in that works with Enterprise Architect was used to generate XSDs from the UML, following the principle of model driven development. The goal of model driven development being that an XSD that is generated from the model would need a minimum of editing before it could be used in a mapping application.

A proof of concept ESB integration following the On Demand Read use case demonstrated the practical application of using both MultiSpeak web services and CIM-based web services (following the guidance from IEC 61968-100) in an integration scenario that is commonly seen by system integrators. This example integration and the mappings created in the harmonization effort can provide useful guidance to systems integrators. By providing useful guidance, identifying the gaps and offering suggested transformation, this should contribute to lowering the cost of integration as the systems integrators do not have to do this investigation themselves.

[7] Both CIM and MultiSpeak are maintained in SparxSystems Enterprise Architect using UML.

Appendix A: Appendix

This appendix is to document structures that are used repeatedly, e.g. Address related classes. Often these classes and attributes are used repeatedly throughout many profiles. Rather than repeatedly document their use throughout the report, those structures will be documented once in this appendix and referred to here.

Table A-1
Address Correlations

CIM	MultiSpeak	Comments
poBox	postOfficeBox	
postalCode	postalCode	
addressGeneral	addressGeneral	
buildName	buildingNumber	Both fields are strings
number	streetNumber	
prefix	streetPrefix	
suffix	streetSuffix	
suiteNumber	suiteNumber	
type	streetType	
code	TownCode	
country	country	
name	city	CIM uses name although in the annotation it is clear that city name is the intent
stateOrProvince	state	State in MultiSpeak can be either a state or province.
areaCode	areaCode	
cityCode	cityCode	
countryCode	countryCode	
extension	extension	
localNumber	localNumber	

Table A-2
Phone Number Correlations

CIM	MultiSpeak	Comments
TelephoneNumbers.	phoneNumber.	
areaCode	areaCode	
countryCode	countryCode	
extension	extension	
localNumber	localNumber	

Table A-3
Service Location Related Transformations

CIM	MultiSpeak	Comments
ServiceLocation .accessMethod .needsInspection .siteAccessProblem	ServiceLocation .description	The three CIM attributes can be concatenated and mapped to the MultiSpeak description attribute.

Table A-4
Electronic Address Related Transformations

CIM	MultiSpeak	Comments
email	emailAddress .eMail .eMailType .priorityOrder	CIM only specifies a single email address. MultiSpeak uses an eMailList as a holder for 0 to many email addresses which can be ordered by type (personal, business, alternate, other) and priority (to specify the order in which they are used)
lan radio	otherContactItem .details .infoType	MultiSpeak does not support lan or radio directly. It uses the other ContactInformation class as a holder for many of types of contact information. In this case the infoType would be set to "lan", with the details the lan attribute from CIM
TelephoneNumbers .areaCode .cityCode .countryCode .extension .localNumber	phoneNumber .areaCode .cityCode .countryCode .extension .localNumber	CIM allow multiple phone numbers. MultiSpeak allows multiple phone numbers through the phoneList class with a 0 to many phoneNumbers. MultiSpeak also includes phoneType and priorityOrder so that if more than one phone number is used the priority in which they are to be used is captured.

Export Control Restrictions

Access to and use of EPRI Intellectual Property is granted with the specific understanding and requirement that responsibility for ensuring full compliance with all applicable U.S. and foreign export laws and regulations is being undertaken by you and your company. This includes an obligation to ensure that any individual receiving access hereunder who is not a U.S. citizen or permanent U.S. resident is permitted access under applicable U.S. and foreign export laws and regulations. In the event you are uncertain whether you or your company may lawfully obtain access to this EPRI Intellectual Property, you acknowledge that it is your obligation to consult with your company's legal counsel to determine whether this access is lawful. Although EPRI may make available on a case-by-case basis an informal assessment of the applicable U.S. export classification for specific EPRI Intellectual Property, you and your company acknowledge that this assessment is solely for informational purposes and not for reliance purposes. You and your company acknowledge that it is still the obligation of you and your company to make your own assessment of the applicable U.S. export classification and ensure compliance accordingly. You and your company understand and acknowledge your obligations to make a prompt report to EPRI and the appropriate authorities regarding any access to or use of EPRI Intellectual Property hereunder that may be in violation of applicable U.S. or foreign export laws or regulations.

The Electric Power Research Institute, Inc. (EPRI, www.epri.com) conducts research and development relating to the generation, delivery and use of electricity for the benefit of the public. An independent, nonprofit organization, EPRI brings together its scientists and engineers as well as experts from academia and industry to help address challenges in electricity, including reliability, efficiency, health, safety and the environment. EPRI also provides technology, policy and economic analyses to drive long-range research and development planning, and supports research in emerging technologies. EPRI's members represent approximately 90 percent of the electricity generated and delivered in the United States, and international participation extends to more than 30 countries. EPRI's principal offices and laboratories are located in Palo Alto, Calif.; Charlotte, N.C.; Knoxville, Tenn.; and Lenox, Mass.

Together...Shaping the Future of Electricity

Program:

Power Delivery

1026585

Electric Power Research Institute
3420 Hillview Avenue, Palo Alto, California 94304-1338 • PO Box 10412, Palo Alto, California 94303-0813 USA
800.313.3774 • 650.855.2121 • askepri@epri.com • www.epri.com