

MultiSpeak Integration Strategies

Tom Taber and Scott A Koehler

Abstract

MultiSpeak is a software interface specification that supports the sharing of specific information between software applications, such as GIS, that are common in rural electric authority (REA) and municipal utilities. This presentation will discuss the data modeling considerations, interface requirements, and implementation issues involved in integrating MultiSpeak compliant software. The talk will focus on the MultiSpeak interfaces between Geographic Information Systems (GIS), Engineering Analysis (EA) applications, and Customer Information Systems (CIS).

1. Initiative and Specification

The MultiSpeak initiative is guided by software vendors and consultants for the purpose of developing and continuing to evolve a software application integration specification now implemented by many electric utilities, most predominantly cooperatives and municipalities. Leveraging industry standard data exchange software languages and protocols, this open collaborative process is helping to make cost-effective interoperability possible.

Why do Utilities Care?

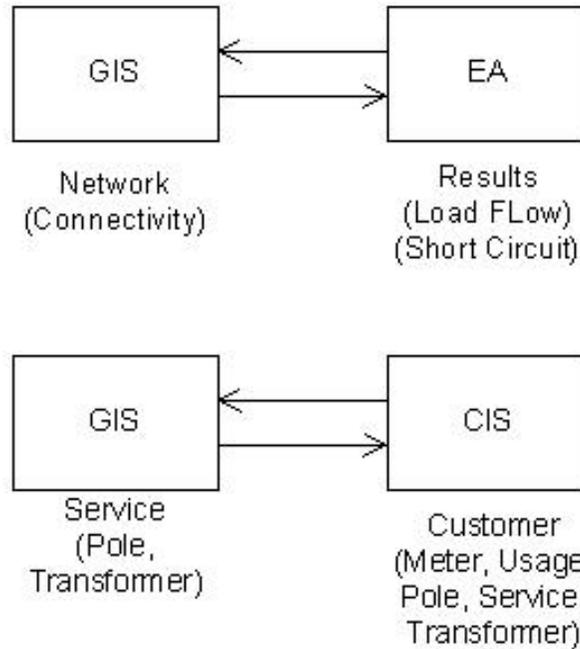
Utilization of MultiSpeak in the implementation of software integration can save time and money by simplifying maintenance, improving resource efficiency and reducing redundancy in data entry. Equally important is the fact that the specification is vendor independent. The biases that can emanate from one provider are lost when several vendors, representing each electric utility software function, are meeting to jointly help determine the future direction of the initiative.

Why do Vendors Care?

Standardizing the way in which vendors interface software allows them to concentrate on application development thus reducing the time to market for new products. This also allows for interface development that can be leveraged for multiple third party applications. Software vendors are not forced to adapt to proprietary languages and protocols. Instead they can utilize familiar standards such as XML, GML, and SOAP. Ultimately though, by lowering implementation costs, logic states vendors will be able to increase software sales.

GIS Specified Interfaces

Currently, there are six software function interfaces specified for GIS; Engineering Analysis (EA), Customer Information System (CIS), Outage Management System (OMS), Staking, Viewer, and SCADA. The two diagrams below illustrate two of these interfaces and the type of data that is shared. The arrows indicate that the interfaces are bidirectional.



MultiSpeak XML File – Example

Text View of Capacitor Information

```

<capacitorBank objectID="35.7" verb="New">
  <owner>Minerville REA</owner>
  <mapLocation>
    <coord>
      <X>2214862.97611417</X>
      <Y>399910.630805994</Y>
    </coord>
  </mapLocation>
  <sectionID>35.7</sectionID>
  <parentSectionID>21.844.1</parentSectionID>
  <phaseCode>ABC</phaseCode>
  <facilityID>CAP7</facilityID>
  <connectionCd>ShuntSameAsParent</connectionCd>
  <swType>Time</swType>
  <bankKvar>450</bankKvar>
  <volts>7.2</volts>
  <capacitorList>
    <Capacitor objectID="17.12" verb="New">
      <owner>Minerville REA</owner>
      <phase>A</phase>
      <kvar>150</kvar>
    </Capacitor>
  </capacitorList>
</capacitorBank>

```

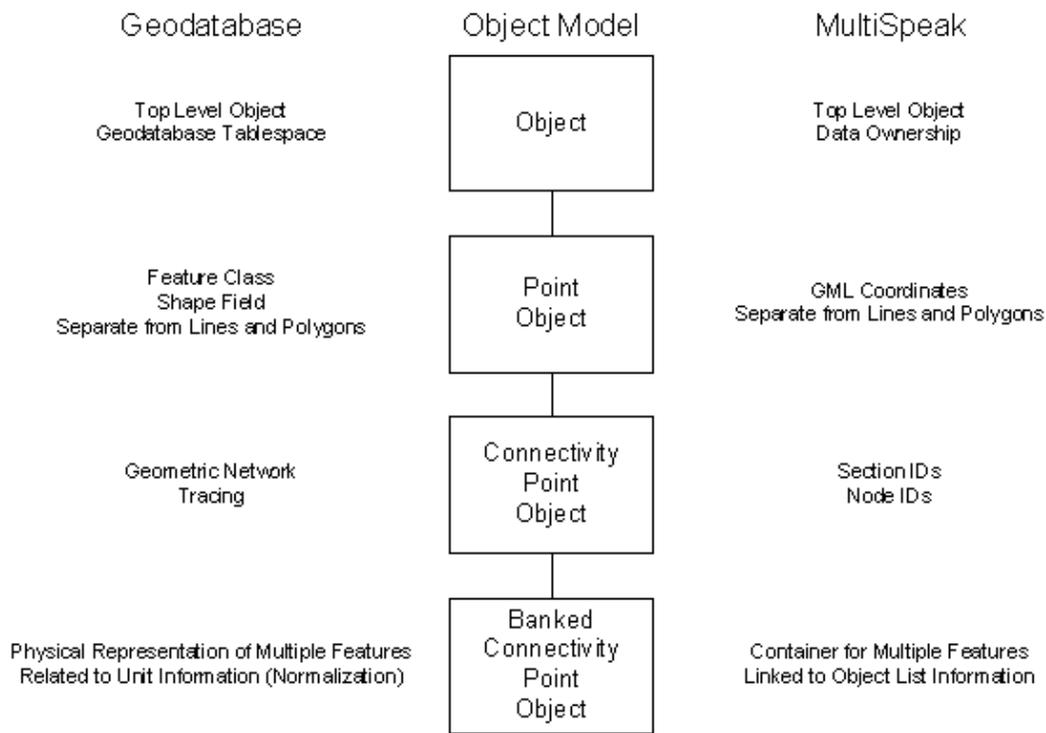
Grid View of Capacitor Information

XML							
▲ MultiSpeak							
=	documentType	incremental					
=	connectivityModel	sectional					
▼	substation	objectID=3467 verb=New					
▼	transformerBank	(17)					
▼	switchDeviceBank	objectID=43.153 verb=New					
▼	streetLight	(7)					
▼	serviceLocation	(67)					
▼	overcurrentDeviceBank	(8)					
▲	capacitorBank						
=	objectID	35.7					
=	verb	New					
(O)	owner	Minerville REA					
▲	mapLocation						
▲	coord						
(O)	X	2214662.97611417					
(O)	Y	399910.630805994					
(O)	sectionID	35.7					
(O)	parentSectionID	21.844.1					
(O)	phaseCode	ABC					
(O)	facilityID	CAP7					
(O)	connectionCd	ShuntSameAsParent					
(O)	swType	Time					
(O)	bankKvar	450					
(O)	volts	7.2					
▲	capacitorList						
▲	Capacitor	(3)					
	=	objectID	=	verb	(O) owner	(O) phase	(O) kvar
	1	17.12		New	Minerville REA	A	150
	2	17.11		New	Minerville REA	B	150
	3	17.5		New	Minerville REA	C	150
▼	ugSecondaryLine	(7)					
▼	ohSecondaryLine	(67)					
▼	ugPrimaryLine	(3)					
▼	ohPrimaryLine	(33)					

2. Data Modeling

Matching Structures

The current MultiSpeak specification, version 2.1a, is much improved over the previous release, version 1.1. One main improvement is the move to an object oriented schema format, a structure mimicking that of many GIS software packages. The concept of inheritance serves as the basic foundation of each structure.



Database Structure Independent

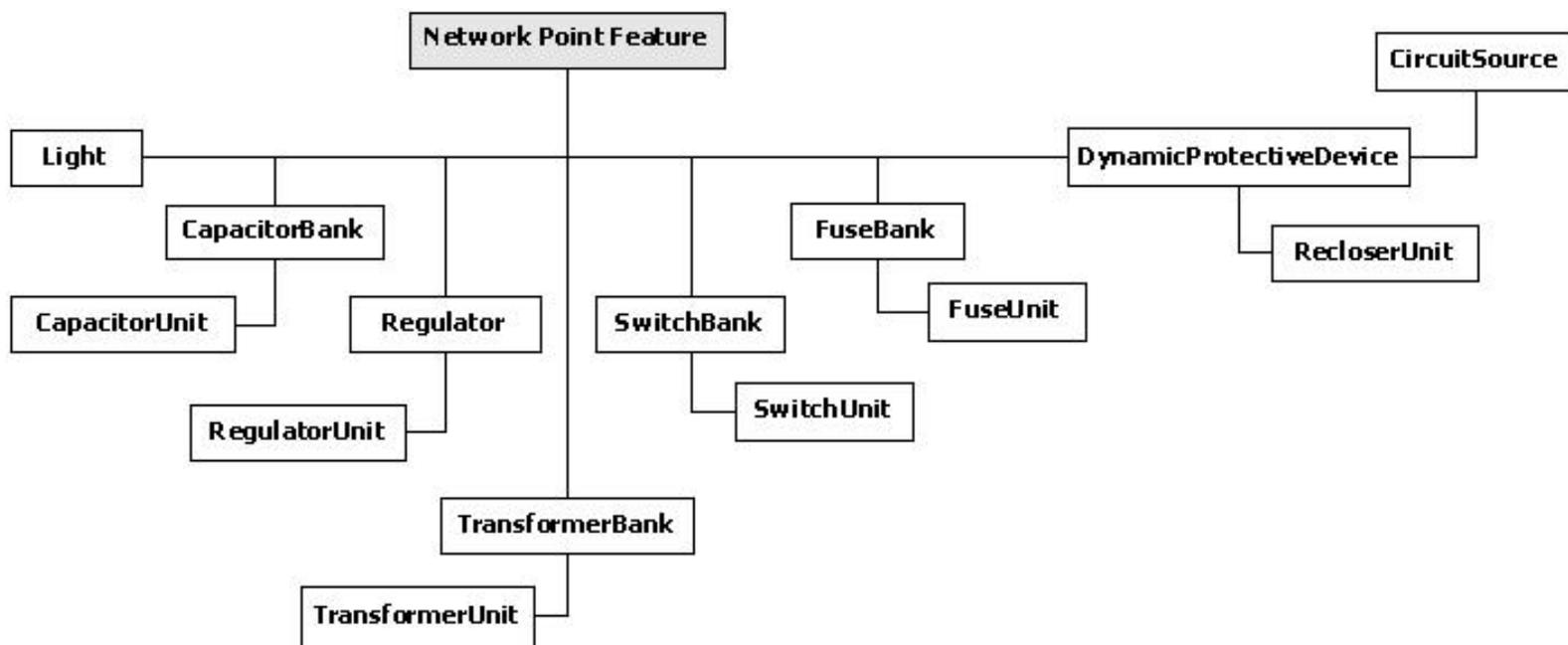
While the XML schema structure is similar to the structure of a Geodatabase, and while a Geodatabase can be modeled to copy the XML schema structure, there is no dependency between the two. When passing data between two independent applications via XML, translations of native database exports can and must occur.

Regardless of how vastly different an application's database structure might be from the XML schema, code can be written to translate the data into the correct format. While this is so, the more the application's database structure emulates the XML schema, the less translation that has to occur.

Flexible and Extensible

Software vendors and the MultiSpeak initiative will continue to grow with the electric utilities they serve. As advances are made in technology, software vendors will enhance their applications and the MultiSpeak initiative will improve the specification. As this occurs, it is important to have a Geodatabase database structure in place that can grow as needed and as desired.

Data Model – Example



Geodatabase Considerations

The Geodatabase provides a common data repository for editing, analyzing, viewing and maintaining a utility's GIS data. It provides a powerful architecture that supports versioning and geometric networks as well as being extensible through common programming languages.

Versioning

The Geodatabase allows multiple users to simultaneously access and edit data. This robust functionality is the basis for providing an enterprise wide GIS solution for a utility. This allows users to store both As-Built and Design versions of data in a single central repository. Utilities that use a MultiSpeak interface can export or import data that is in either a Design or As-Built state. This is particularly useful when integrating with an Engineering Analysis application.

Complex Electric Networks

The Geodatabase was designed to support and manage geometric networks. Electrical networks easily lend themselves to modeling through this technology. Networks support line and point connectivity as well as complex behavior and network related attributes.

The Geometric Network provides electric utilities a basis for managing an electrical feeder. A Feeder typically begins within a substation at a device, such as a circuit breaker or recloser, and then distributes electricity through a network of connected conductors and electrical devices. A GIS application that manages feeder features depends upon the geometric network architecture to manage feature connectivity, attributes, and behaviors.

For a MultiSpeak implementation a Feeder provides the most logical collection of features for export to any application that requires network data, such as EA, OMS, and Staking.

Extensible via ArcObjects

The ArcGIS framework and Geodatabase provide a library of powerful and extensible COM objects. These objects can be accessed through common programming languages such as C++, Visual Basic, and .NET. This

object-modeling framework provides a method for utility specific tools to be built onto existing ESRI technology. This allows vendors to leverage ESRI technology to build MultiSpeak compliant interfaces with third party software applications.

3. Interface Requirements

Specification (Not a Standard)

For better or for worse, the MultiSpeak initiative has developed a specification, not a standard. The term standard would imply that one very specific format has been developed. Any deviation from the standard would be a violation of regulations. The term specification implies that detailed guidelines have been developed to provide a recipe for interfacing software. While compliance testing exists to confirm whether or not a given application has followed the guidelines, there is no one set path to take.

Minimal Mandatory Data

When interfacing GIS with EA, there are very few mandatory fields that must be passed. Instead, a structure has been put in place that describes how data must be passed when a given utility desires to have it passed. In the GIS-EA case, this inherently prevents the specification from requiring a GIS system to export data that it does not maintain. Below is an example of the more common fields passed for a capacitor in a typical GIS-EA interface.

CapacitorBank

Shape (Coordinates) – A capacitor occupies a point in space and information about its location must be sent, but this information can be sent in different ways (i.e. 2 fields ‘x’ and ‘y’ or 1 field ‘x, y’.)

Network (Connectivity) – It is vital, thus mandatory, to send information regarding directional flow of a network, but this information can also be sent in different ways (i.e. section-based or nodal-based information.)

Phase (A, B, C, AB, BC, AC, ABC) – The phase of a capacitor bank is quite critical for an EA application, but it is not required that the GIS has this info. While a GIS will most likely track this data, if it does not, there are ways around this problem.

KVAR – The same rules that apply to Phase apply to KVAR and many others.

Default Values

For situations where data is not available in GIS, default values can be assigned. The assignment of defaults can be taken care of in the XML translation or directly within the EA application. Most EA applications, if not all, allow for the assignment of default values when pertinent information is not received. With some custom implementation effort, default values can be applied to an exported XML document. Fortunately, both options will not necessitate a modification to vendor specific interface applications.

Leverage XML Exports

XML files are a standard format for passing data on the Internet, but they also provide a powerful format for exchanging data between software systems. The MultiSpeak standard specifies using XML files with a predefined schema. This schema and file format can be used to leverage a MultiSpeak implementation to interact with software that is not MultiSpeak compliant.

Software Functions Not Specified

MultiSpeak is a specification for interfaces between applications, but does not place any limitations or requirements on the primary functionality of MultiSpeak compliant software. Therefore software vendors can employ a variety of strategies for entering, storing, accessing and manipulating data. The MultiSpeak standard only requires that a compliant application can package the required data in the required MultiSpeak format and be able to pass it (export) or receive it (import) from other MultiSpeak compliant applications.

MultiSpeak allows utilities shopping for MultiSpeak compliant software to determine the value of the functions provided by each vendor. This important distinction encourages greater support from the vendors who build software for the utility industry as well as providing a larger selection of software options for utility companies.

Non-Compliant Software Applications

Software vendors who develop MultiSpeak Interfaces are not limited to interfacing only with other MultiSpeak compliant applications. The same technology used to create MultiSpeak compliant XML exchange files can be leveraged for other applications that are not MultiSpeak compliant. The data formats for the exchange files will likely be different from the MultiSpeak specification, but the functionality for exporting and importing data will be very similar.

Internally, Custom-Developed Applications

Internally developed or legacy custom applications are two examples of applications that are not MultiSpeak compliant, but that may benefit from the technology used to support it. Many small utilities have internally developed or legacy CIS systems that they do not want to replace. A MultiSpeak compliant XML file from a GIS can be converted to an appropriate format accepted by the CIS by transforming the file using a stylesheet. Stylesheets can be created without the need to program and are a powerful way to leverage a MultiSpeak implementation for other uses.

4. Implementation Strategies

Components

Standardized Database Structure

Having a database structure that mimics the MultiSpeak XML schema format will simplify the interface translation process and thus improve performance and maintainability. Implementing one interface might be somewhat feasible in a custom interface environment, but implementing several interfaces is where a standardized database structure really pays off.

Standardized Import/Export Application

Having one application that standardizes the way information is extracted from GIS will also ease the pain of implementing external software applications. There is only one MultiSpeak schema that specifies the format for data between all interfaces. This being the case, an application that exports data from a GIS to EA will be able to satisfy an export to an Outage Management System (OMS). Purchasing or developing unique interface applications for each interface without investigating this option, could cost a utility a significant amount of time and money.

XSLT for XML Translation

MultiSpeak leverages XML as the format for passing files between systems. Therefore, leveraging a software application that exports data from a GIS into XML directly can be advantageous. If a utility has a data model structure unique to their business requirements, than a programming language like XSLT can be used to translate an exported XML into the XML schema format. XSLT is a much simpler language to learn and utilize than C++, Visual Basic, or other similar programming languages. In addition it does not need to be compiled or integrated into the GIS application.

Batch File Transfer

Passing data via a batch process is the simplest and fastest type of interface technology to deploy. Since the MultiSpeak specification version 2.1a details the options for passing data near real-time, there is room to grow from a batch process implementation. As with an implementation of GIS, it is typically more prudent to first get the application handling data properly before reaching for the next level of complexity. The approach is viable since the MultiSpeak schema format is the same whether data is being transferred in batch or real-time.

Implementation Realities

MultiSpeak was started as an initiative to remove as many barriers to software integration as possible, but it cannot account for all of the special circumstances that arise during an integration project. Integrating software applications from different vendors can be very difficult. Each utility has special business requirements and varying levels of facility data quality. MultiSpeak does provide a mechanism for quicker implementations, but it is not simply a plug and play implementation. Careful consideration should be given to both the business processes and data integrity when implementing any integrated software.

Finite Number of Elements Defined

The MultiSpeak exchange format specifies the most common data elements that will be passed between two systems. There may be a number of data elements not defined by the MultiSpeak specification that need to be passed for a specific implementation. These additional data elements will need to be addressed as additional work during the integration project.

Vendor Specific Requirements

Software vendor applications may have additional requirements for data formats. These requirements may be something as basic as the way that phasing information is stored in the database, such as phases being stored as phases present (A, B, C, AB, AC, BC or ABC) or as phase position (ABC, BAC, CAB...). There may also be requirements for storing data as character or numerical values.

Utility Specific Requirements

Utilities may also have requirements that need to be maintained by a new system. They may have complex company numbering systems that need to be supported. They may also need to continue to support integration to a legacy system, such as a CIS or OMS. They may have utility specific data for mapping applications. In addition to data requirements there may be unique business needs that affect the implementation and integration. All of these requirements will need to be addressed, but may not be covered by a generic MultiSpeak integration.

Implementation Goals

The MultiSpeak initiative was designed to create software interface standards that would be mutually beneficial to utilities and utility software companies. The initiative promotes cooperation between companies that, in some cases, are competitors. This delicate balance is maintained by having clear goals that ultimately benefit all parties involved. A few of these goals of the MultiSpeak initiative are to make interface implementations easier, cultivate vendor relations, and satisfy needs of electric utilities, which ultimately helps provide a foundation to minimize cost.

By implementing a MultiSpeak integration project, utilities should expect to minimize cost and risk while maintaining a high level of functionality for all of the applications that are implemented. Transferring data between applications reduces duplication of effort and minimizes the risk of entering errors into the data. Utilities can get more value out of the data that they maintain and make better business decisions based on it.

Conclusion

MultiSpeak provides a specification for interfacing utility specific software applications. The MultiSpeak initiative is designed to save utilities money by lowering implementation costs through standardized interfaces. The MultiSpeak initiative supports software vendors by creating mutually agreeable interfaces that can be used for multiple software applications. The interfaces are built using a schema defined by MultiSpeak in XML format. A database model is not dependent on the MultiSpeak specification, but a model that parallels the exchange format makes implementation simpler. The ESRI Geodatabase makes building a GIS interface flexible through versioning, geometric networks, and extensible COM objects. Ultimately, MultiSpeak can be beneficial to both utility companies and utility software vendors by reducing the total cost and risk of software integration implementations.

References

MultiSpeak, 2003. National Rural Electric Cooperative Association, Arlington, VA, online at <http://www.multispeak.org>.

Tom Taber
Project Manager
Miner & Miner Consulting Engineers
4701 Royal Vista Circle
Fort Collins, CO 80528
Phone: (970) 223-1888 Ext.135
Fax: (970) 223-5577
tom.taber@miner.com
www.miner.com

Scott A Koehler
Project Manager
Miner & Miner Consulting Engineers
4701 Royal Vista Circle
Fort Collins, CO 80528
Phone: (970) 223-1888 Ext.230
Fax: (970) 223-5577
tom.taber@miner.com
www.miner.com