Integration Using the MultiSpeak® Specification

By: Gary A. McNaughton, Cornice Engineering, Inc. and
Robert Saint, National Rural Electric Cooperative Association

Introduction

Over the years many different approaches have been tried to ease the interoperation of applications. The first approach (See Figure 1) was data integration, ensuring that the data stored in different programs was consistent, but otherwise each program handled business processes using its own logic. Typically this occurred using off-line batch file transfers, or occasionally using a single, common database. File transfers in particular were usually not successful since updates were often inconsistently applied and databases eventually became unsynchronized.

The next wave of integration was application integration, where two programs exchanged data or services. In this approach, programs expose data to be shared or services that they will offer to another, usually via a point-to-point application programming interface (API). Both data and application integration can be thought of as examples of *tactical* approaches to solving IT problems. Both approaches still have value today and the development of extensible markup language (XML) and web services make such tactical integration even easier than was possible in the days of common, but often proprietary, APIs.

The most recent concept is for all the programs in an entire enterprise to share a common data model and, ideally, a common services bus. Enterprise integration goes far beyond the capabilities of point-to-point application interfaces, but requires a strategic approach to integration and can be difficult, expensive and time-consuming to achieve.

Figure 1. Spectrum of integration.

Service-Oriented Architecture – Enterprise Architecture Fulfilled

The concept of a service-oriented architecture (SOA) has been around for decades. The premise of an SOA is that programs provide services for the use of other programs in a structured and well documented manner. The intention is to (i) achieve interoperability and (ii) share common IT services such as security, reliable messaging and transaction management across the entire enterprise, but to do so in a way that abstracts the details of the implementation of each program. In the past, SOA was

often implemented using proprietary APIs, middleware platforms, or Common Object Request Broker Architecture (CORBA) brokers. None of these approaches became widespread because of the difficulty and cost of deploying such solutions.

SOA is only now becoming accessible with the wide availability of XML and web services programming tools. XML successfully abstracts the details of implementation of data representations, regardless of the application programming language, platform or database used. Web services has been widely used in a variety of IT environments to expose program functionality or data in the form of abstract, but well documented, services – exactly what is needed to implement the structure of an SOA. What is needed to complete the picture, in addition to these infrastructure standards, are (i) an enterprise-wide semantic agreement on the data objects to be exchanged (often called a *data model*), and (ii) a clear definition of the services to be made available on the bus, which can be used to implement utility business processes. These two remaining requirements are satisfied by the use of the MultiSpeak® specification.

MultiSpeak's Role in Integration

MultiSpeak is an industry-wide open standard for integration of software used in distribution utilities and all portions of vertically-integrated utilities except for generation, power marketing and transmission modeling. MultiSpeak includes a common data model that is documented in the form of an XML Schema and an extensive set of web services to support common utility business processes. The web services are documented in the form of web services description language (WSDL) computer-readable files to ease the development of compatible interfaces.

In addition to a comprehensive data model and set of service definitions, MultiSpeak offers an independent interoperability testing laboratory. Utilities and vendors can make use of this testing service to ensure that interfaces actually meet the requirements of the specification. As a result, pairs of vendor products that have been interoperability tested typically can be integrated at a utility without the need for further customization.

The MultiSpeak specification has been developed by, and is maintained by, the MultiSpeak Initiative, an open collaboration of the National Rural Electric Cooperative Association and over 40 leading software vendors serving the electric utility market. MultiSpeak interfaces are in production use at over 200 utilities to date, including electric cooperatives, municipals, and investor-owned utilities.

Some of the issues that should be addressed when considering MultiSpeak as a foundation for integration are:

- Is it scalable?
- Should the integration be point-to-point or SOA-based?
- Should MultiSpeak be used for a tactical or a strategic implementation?
- Is MultiSpeak extensible?

Scalability

Scalability is an issue of implementation, not a function of the underlying data model used. Two issues factor into the scalability of an implementation: protocol design and choice of messaging paradigm. First, careful thought has been given during the design of the MultiSpeak protocols to ensure that large volumes of data can be handled efficiently. Many utility implementations have proven that the resulting protocol addresses scalability concerns. In terms of messaging, MultiSpeak web services have the same capability to be scaled as any web services implementation. Should a utility decide that a web services application is not appropriate for their particular situation, MultiSpeak can also be applied using an optional messaging framework over any message-oriented middleware platform.

Point-to-Point Integration versus Service Bus Integration

MultiSpeak is ideally suited to supporting either a tactical approach or a strategic, SOA-based integration architecture. Figure 2 illustrates the coverage of the MultiSpeak specification when applied tactically. When used in this manner MultiSpeak can be visualized as a set of point-to-point interfaces. Each box in Figure 2 is an abstract software capability and each line is a supported interface between software functions. For instance, the capability of an advanced metering infrastructure (AMI) system to detect customer outages is supported in MultiSpeak using the outage detection (OD) function. The capability of an interactive voice response (IVR) system to exchange information about outages reported via customer phone calls also is supported by this same software function, since an outage management system (OMS) needs the same data to handle the resulting outage, regardless of the source of the detected outage.

However, once an application exposes a web service, that service is available to any other program on the utility network – thus it really creates a service bus. Figure 3 illustrates the resulting service bus architecture. When implemented in this manner, MultiSpeak can support a strategic SOA-based implementation.

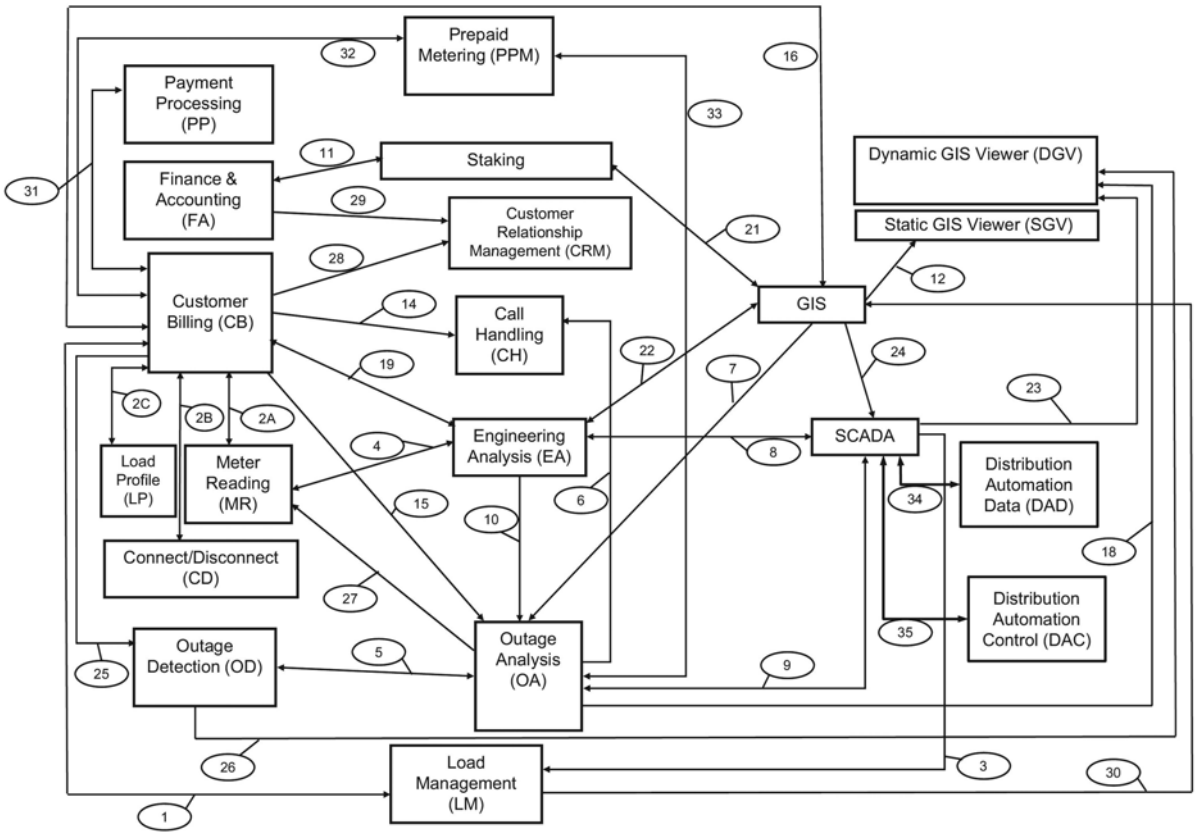

Figure 2. Point-to-point integration model showing how MultiSpeak can be used in a tactical implementation.

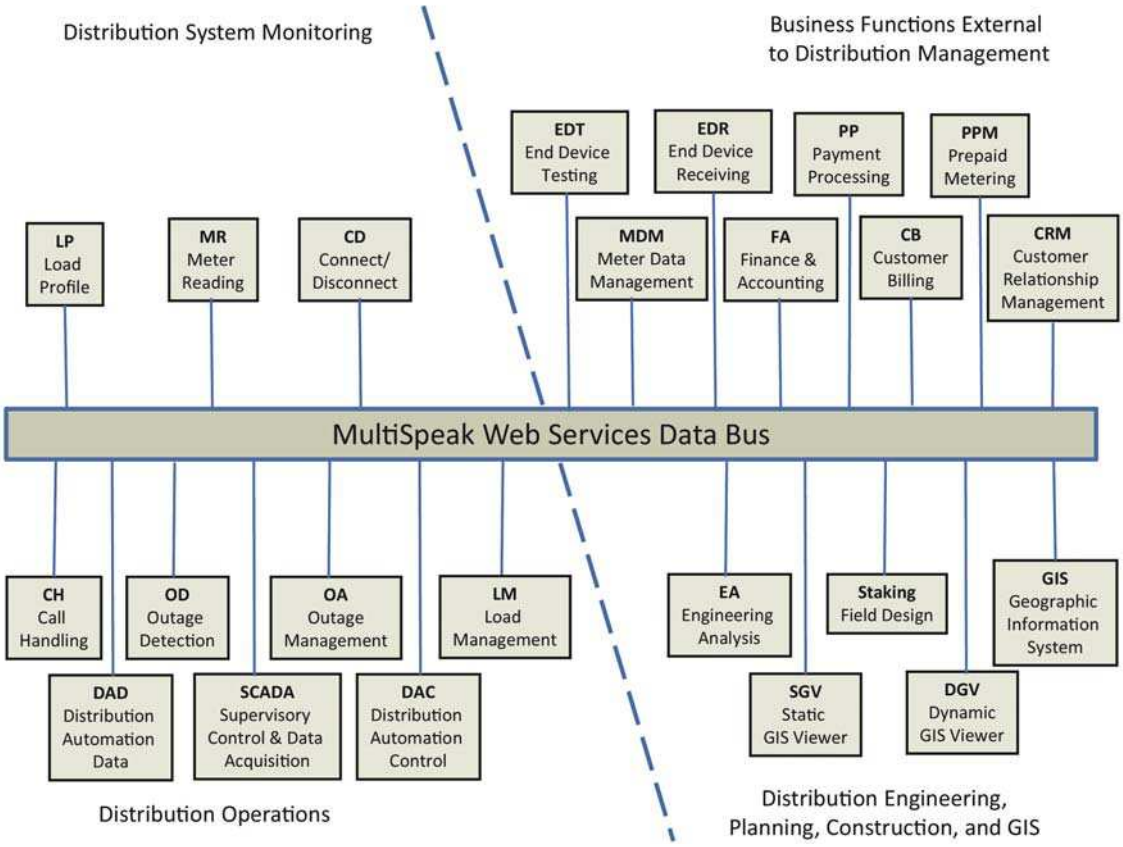

Figure 3. MultiSpeak shown as a service bus. Used in this manner, MultiSpeak can support SOA-based enterprise integration.

To clarify this relationship, an example is in order. Figure 4 shows the data flows that would occur among an OMS (the outage analysis or OA function), an AMI system set up to detect outages (the OD function), an IVR used to collect outage information from customer calls (a second instance of the OD function), an IVR acting to handle customer calls and confirm restoration (the call handling or CH function) and a supervisory control and data acquisition (SCADA) system (the SCADA function) that can send device status changes to the OMS in order to confirm feeder outages. The data flows for the same applications are summarized for the bus architecture in Figure 5.

This figure illustrates a key point about the bus character of the specification; once a consistent set of services is exposed by an application, reuse of services is possible. Another advantage of this structure is that the availability of granular services enables developers flexibly and quickly to string together services from multiple applications, even if they previously were not integrated, in order to support enhanced utility business processes.

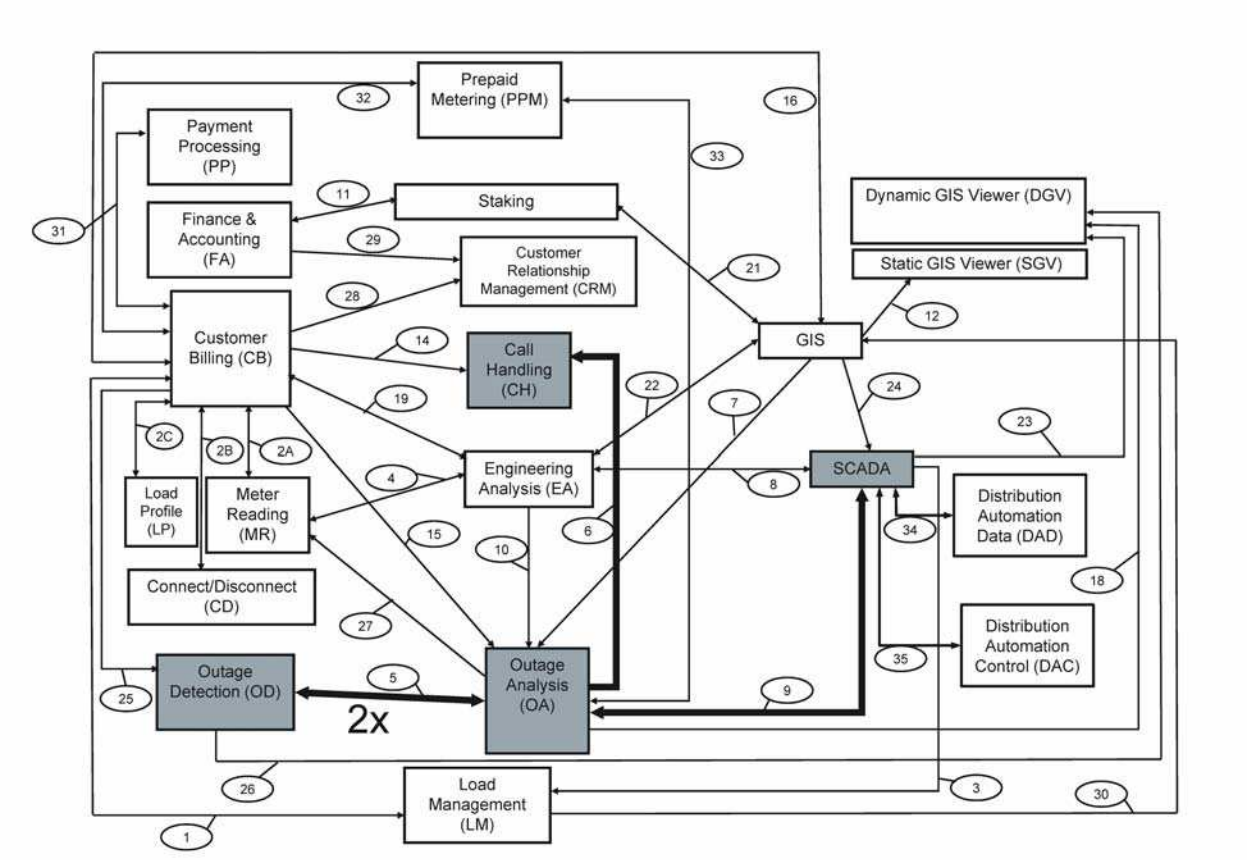

Figure 4. Outage Handling Integration Pattern

Another key point from Figure 5 is that during the development of the bus formulation of the web services, a comprehensive set of Meter Data Management web services was created using the previously defined services that touched any of the functions exhibited by AMI systems. The resulting MDM service interfaces are ideally suited to facilitating extensive AMI implementations.

Which Approach is Appropriate for My Utility - Tactical or Strategic?

Next one should consider the approach to implementing MultiSpeak-compatible application interfaces. Depending on the needs of the utility, MultiSpeak can be simply and quickly deployed for point-to-point application integration or can serve as the basis of an enterprise service bus.

Many small utilities, or those looking for a quick solution, can use vendor-provided web services and just turn on the point-to-point interfaces. This tactical approach results in fully functional integration between two applications often in only a few hours and at low cost. This is by far the most commonly used approach to date. An example of this implementation is Utility A that purchased MultiSpeak-enabled AMI and OMS applications. In this case, no development was necessary for complete integration.

The two applications merely needed to be set up to look for the other's web server; integration was complete in a matter of minutes.

In some cases, utilities have one MultiSpeak-enabled application but the others with which it should be integrated do not yet have MultiSpeak interfaces. In this case, the

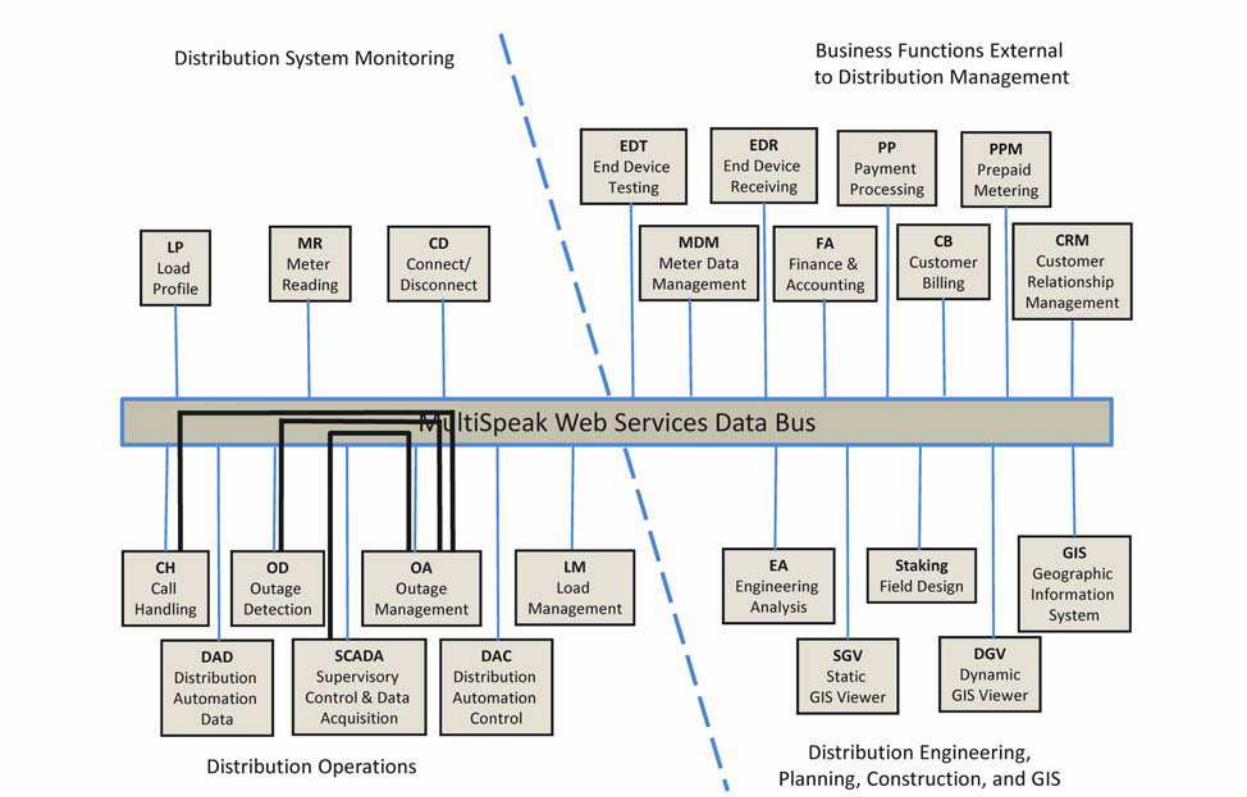

Figure 5. Outage Handling Integration Pattern in Bus Format

utility or a system integrator can develop an adapter for the non-enabled application. This situation will require some development time and care to ensure that the adapter appropriately implements the MultiSpeak data model, but still full integration can be obtained for little investment. An example of this case is Utility B that had a MultiSpeak-compatible IVR system, but their OMS did not support MultiSpeak. A system integrator was hired to write a MultiSpeak "wrapper" for the OMS. The development work was completed in a matter of weeks.

A similar situation exists where no applications have existing MultiSpeak web service interfaces. In this case both applications to be interfaced require adapter development. It should be noted however, that even in this more involved situation, the semantic foundation and service development work has been provided in MultiSpeak so the cost of interface development in such a case is a small fraction of what a traditional custom interface would cost for the same applications. Utility C was in this situation and

completed the MultiSpeak interface development for an AMI to CIS interface using in-house programming resources.

On the other hand, a number of utilities are beginning to look at the MultiSpeak data model and service definitions as the basis for a more strategic implementation. MultiSpeak is also well suited to serve as the basis for a service-oriented architecture. The SOA approach has several distinct advantages that may justify the additional development time and expense:

- **Uniform data model throughout the enterprise.** A common semantic understanding of the data (data model) eliminates repeated data translations and misunderstandings that can result when data must be interpreted by different departments or applications with different data structures.
- **A consistent set of services are available to all applications to use, even if no prior interface was defined between pairs of applications.** This fact streamlines support for new business processes and enhances agility of the organization.
- **The potential for reuse and flexible composition of granular web services.** Once a variety of web service methods are available on the service bus, business processes can be dynamically modified (composed) to fit specific needs.
- **Provisions for reliable messaging.** The tactical, point-to-point approach relies on the applications on each end of the interface to be constantly available. Although some messaging reliability features are built specifically into the MultiSpeak service protocols to minimize these problems, the addition of reliable messaging in the enterprise service bus can eliminate the potential for data loss.
- **Provisions for data security.** NERC CIP requirements often apply to the utilities that might apply MultiSpeak, but even if this is not the case, data security is becoming a necessary business practice in today's world. Once again, MultiSpeak makes some provisions for data security using secure sockets layer (SSL) encryption and authentication, but complex application interactions may require more than just SSL for complete security. An enterprise service bus can consistently provide additional security for all data communications, enterprise-wide.

Extensions to MultiSpeak

MultiSpeak may not currently support all of the applications that are needed by some utilities. For instance, since MultiSpeak was originally designed with the needs of distribution utilities in mind, power marketing and transmission network modeling are not currently supported. In addition, development is on-going for some utility applications, such as asset management and work management.

However, MultiSpeak was designed from the ground up to be extensible. It is possible to add an unlimited number of additional data objects to the data model and to extend any existing data object by the addition of an unlimited number of XML attributes and/or

XML elements, all without affecting interoperation with other applications that may be unfamiliar with the thus-defined extensions. Furthermore, it is possible to easily add additional web services (to support additional types of applications, such as work management) and to add new methods to existing web services. All of these extensibility mechanisms make it possible for any utility to build on the established, well-proven foundation of the MultiSpeak data model and service definitions to create those extensions necessary to meet their specific needs.

Harmonization with the IEC Common Information Model

There is an alternative data model for utility integration, called the Common Information Model (CIM), which is sponsored by the International Electrotechnical Commission Technical Committee 57. CIM is broader in scope than is MultiSpeak but is less fully developed. Many of the parts of the CIM standards have not been finalized and with a few minor exceptions, there are not well-defined profiles that would enable utilities to develop interoperable implementations. Furthermore, the emphasis in CIM development at this point is to finish the standardization of the data model and the use cases to support utility business processes. Because of this emphasis, efforts have not been focused on development of services to implement those use cases.

It is likely that both standards will co-exist since both bring value to the integration equation. Thus, there is value to working towards harmonization of the two standards. Both efforts have recognized this value and have begun a concerted, mutual effort to (i) bring the specifications together over time and (ii) develop bridging technologies to permit interoperation of applications that are compatible with the different standards.

Conclusions

MultiSpeak is a flexible, extensible specification which can be used as the basis for a wide variety of integration efforts from the simplest single point-to-point interface to a complete service-oriented architecture in the largest utility. Furthermore, for many utilities, the MultiSpeak specification is a complete solution to their integration needs that is available for implementation today. For others it is a solution that may need some customization, but it brings a robust foundation that supports the required extension with minimal development cost and risk. Even those utilities that eventually want to transition to a CIM-based solution in the future, should consider a near-term MultiSpeak solution with a bridge to a longer-term CIM solution as CIM gains maturity.

Any utility, regardless of size, that is entering into an integration initiative should consider adopting the MultiSpeak data model and service definitions as the foundation for their planned integration. More information about the MultiSpeak Initiative and specification can be found at www.MultiSpeak.org .