TITLE: Model Driven Transformation

PRESENTER: Alan McMorran

As utilities and vendors adopt open standards like CIM there is a growing need to either migrate existing legacy data and interfaces to use CIM or for vendors to support the import and export of their own, internal data structures into CIM. We will discuss how a Model Driven Architecture (MDA) can facilitate this process and can be combined with Model Driven Transformation (MDT) to provide a powerful architecture to support the transformation of data between multiple structures and formats including CIM, IEC61850, MultiSpeak and other power system data formats.

The MDA approach uses models to define the information within a system, file or database. It decouples this structure from its serialization format so that the data is described as abstract entities rather than being a particular line or column of text within a file; or as one or more columns and tables in a database. The same data can be serialized in multiple different formats without impacting its structure and definition. By having a well-defined data structure both users and developers find it easier to interpret, understand and use the data without having to understand the added complexity of a particular serialization format or technology. In most cases though an information model does not exist so we will discuss how information models can be derived from existing formats to support this architecture.

When an application or systems uses a Model Driven Architecture, data then becomes format-agnostic data objects and applications are written to process the data objects rather than requiring multiple interfaces to deal with the different possible sources. This approach also modularizes the software, separating the parsing of data from a file or reading from a database from the application logic itself, an approach already defined within the CIM which separates the serialisation as CIM RDF XML or CIM XSD messages from the information model in UML.

An MDT is written to map the structural elements within each model. We will show how these rich Modelto-Model languages such as Query View Transform (QVT) an open, standard language by the Object Management Group (OMG) can be used to define structural-level transformations between models such as CIM.