Distribution Grid Model Manager (GMM) Functional Requirements

3002025388

Distribution Grid Model Manager (GMM) Functional Requirements

3002025388

Technical Update, November 2022

EPRI Project Manager

P. Brown

EPRI
3420 Hillview Avenue, Palo Alto, California 94304-1338 ▪ USA
800.313.3774 ▪ 650.855.2121 ▪ askepri@epri.com ▪ www.epri.com

EPRI PREPARED THIS REPORT.

This is an EPRI Technical Update report. A Technical Update report is intended as an informal report of continuing research, a meeting, or a topical study. It is not a final EPRI technical report.

NOTE

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

Together…Shaping the Future of Energy®

ACKNOWLEDGMENTS

EPRI prepared this report.

Principal Investigator
P. Brown

This report describes research sponsored by EPRI.

The following organizations, under contract to EPRI, contributed to the preparation of this report:

Britton Consulting
603 Windmill Drive
Freeland, WA 98249

Principal Investigator
J. Britton

GridOptimize
11 Louis Allan Drive
Danbury, CT 06811

Principal Investigator
S. Coe

EPRI would like to acknowledge the support of the following organizations:

Ameren
Arizona Public Service
Consolidated Edison
Duke
ESB Networks
Exelon
First Energy
Great River Energy
Hydro-Québec
Pacific Gas & Electric
Salt River Project

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

Distribution Grid Model Manager (GMM) Functional Requirements. EPRI, Palo Alto, CA: 2022. 3002025388.

ABSTRACT

This report defines high-level requirements for a Grid Model Manager (GMM) tool, the main enabler of a Common Information Model (CIM)-based, consolidated approach to distribution grid model data management. It is intended to provide guidance to utilities in understanding and internally promoting standards-based, consolidated grid model management. It is also intended to help vendors gain an enterprise-wide view of required GMM functionality.

As distribution grid behavior becomes more complex, the number of applications across distribution operations and planning which rely on accurate electrical system models is growing. Distribution grid model data comes from a variety of sources, most of which serve engineering design, facilities records, or asset management purposes. This data needs to be transformed and aggregated into a consistent view of the distribution grid, from which the grid modeling needed by various grid simulation applications across the distribution organization can be created. This is the functionality a GMM tool provides.

The functional requirements for a GMM tool are driven by the role it is intended to play in an overall distribution grid model management picture. This role is described, at a business function level, in EPRI's GMDM information architecture. The functional requirements of a GMM are expressed in terms of software functions and their production or consumption of modules of CIM data. This report leverages both the GMDM information architecture and the CIM in its description of the essential functions a distribution GMM tool should implement.

Keywords
GMM
Grid model manager
GMDM
Grid model data management
CIM
Common Information Model

CONTENTS

ABSTRACT .... **v**

1 **BACKGROUND** **1-1**

2 **GMDM INFORMATION ARCHITECTURE: DRIVING THE GMM FUNCTIONAL REQUIREMENTS** **2-1**

3 **CIM 'META' CLASSES: THE GMM'S DATA BUILDING BLOCKS** **3-1**

- Proposed UML for CIM 'meta' Classes 3-1
  - Basic 'meta' classes 3-2
    - Model 3-2
    - ChangeModel 3-3
    - Assembly 3-3
    - ChangeSequence 3-3
  - Specification 'Meta' classes 3-3
  - Helper 'Meta' classes 3-4

4 **GMM FUNCTIONAL REQUIREMENTS** **4-1**

- Overview of Requirements 4-1
  - Requirement Notes 4-4
    - Manual data entry 4-4
    - Import of grid modeling from other sources 4-4
    - derived physical ChangeModels 4-5
    - visualization Models 4-6
    - situation data patterns 4-6
    - complex validations of Models 4-6
    - Export of grid modeling to consumers 4-6
    - Participating in data exchange environment 4-7
    - Operations, Procedures and AuditTrails 4-8
    - Repository 4-8
    - Object Registry 4-8
    - user workspaces 4-8
    - extensibility 4-9

5 **REFERENCES** **5-1**

A **THE GRID MODEL MANAGEMENT BUSINESS FUNCTION** **A-1**

- Descriptions of the Grid Model Management Business Function and Its Component Functions A-1

LIST OF FIGURES

Figure 2-1 The Grid Model Management business function of the GMDM information architecture 2-1
Figure 3-1 Conceptual model of the CIM's 'meta' classes........ 3-1

1
BACKGROUND

Grid simulations and analytics are playing an ever larger role in distribution operations and planning. Distribution operations is increasingly expected to maintain a real-time steady-state solution of the present state, study switching prior to initiation, and run analysis to determine optimum control settings. Distribution planning is expected to study the grid states and simulate the behaviors that occur in operations in addition to determining approaches for optimizing grid expansion and facilitating engagement of DER developers and end consumers.

Every one of the grid simulation and analysis tools which will enable distribution utilities to perform these activities will rely on accurate, complete, and up-to-date grid models. Distribution grid model data comes from a variety of sources, most of which serve engineering design, facilities records, or asset management purposes. This data needs to be transformed and aggregated into a consistent view of the distribution grid, from which the grid modeling needed by various grid simulation applications across the distribution organization can be created. This is the functionality a GMM tool provides.

The functional requirements for a distribution GMM tool are driven by the role it is intended to play in the overall distribution grid model management picture. They are expressed in terms of the application functions needed to allow a GMM to fulfil its role and the modules of data that those application functions produce or consume.

EPRI research over the last five years has explored and validated a consolidated, enterprise-wide approach to grid model data management for meeting distribution utility grid model data needs. As part of that work, an information architecture – the Grid Model Data Management information architecture - has been developed. Concurrently, efforts in the Common Information Model (CIM) working group have moved the CIM toward more comprehensively supporting the needs of enterprise-wide distribution grid model data management. So, initial versions of the two essential components - an information architecture outlining the functionality required of a GMM and the definitions of the data structures it uses in delivering the functionality - are in place such that a first version of GMM functional requirements can be described.

This report starts out by overviewing the GMDM information architecture's Grid Model Management business function which provides GMM business level functional requirements. It then describes the CIM 'meta' classes which define the data building blocks a GMM tool can leverage to meet the requirements. The report finishes by outlining, in summary form, the functional requirements a GMM should meet, using a liberal set of background notes to fill in requirement detail.

2

GMDM INFORMATION ARCHITECTURE: DRIVING THE GMM FUNCTIONAL REQUIREMENTS

The GMDM information architecture is a business function-based vision of how grid model data can be effectively managed across the distribution utility enterprise. The GMDM information architecture – and specifically the Grid Model Management business function it calls for – provide the context for the GMM requirements outlined in this report.

The Grid Model Management business function, shown in Figure 2-1, is composed of two major component functions which operate independently, but work together to support the management of grid data at the enterprise level. The Physical Model Management business function continually maintains up-to-date grid modeling which is then used by the Assembly Support business function to construct modelling to meet the needs of various network analysis activities.

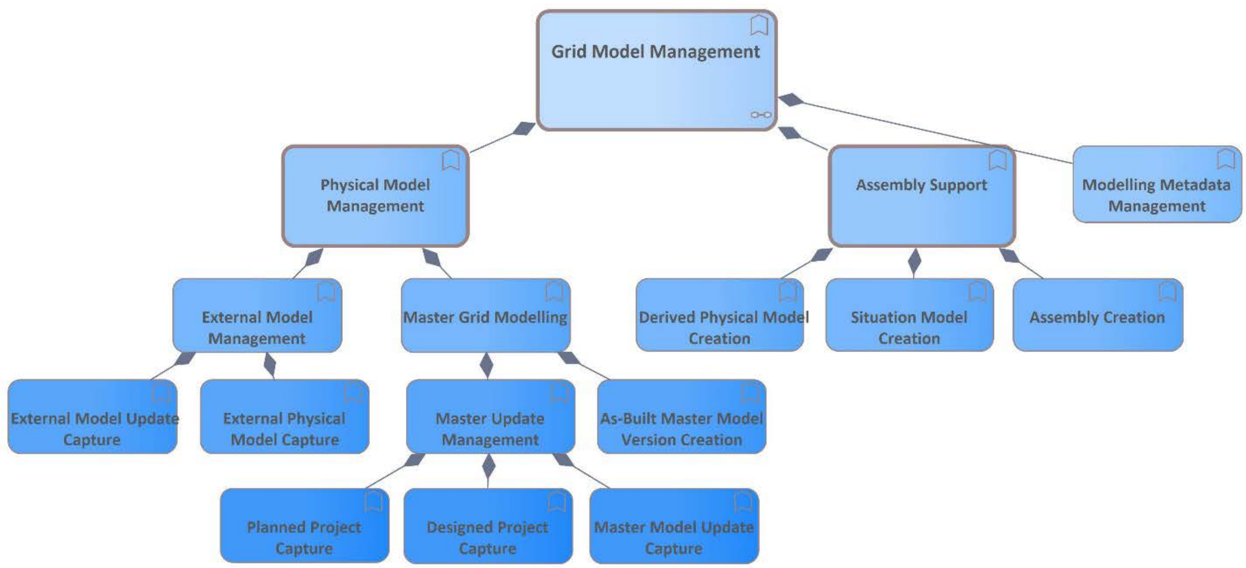

Figure 2-1
The Grid Model Management business function of the GMDM information architecture

The Physical Model Management function is split into two components, the major one of which is Master Grid Modeling whose job it is to maintain sufficiently detailed master reference data describing the grid for which the business organization has primary modeling responsibility. Master Grid Modeling, in turn, contains two component functions: Master Update Management, which maintains descriptions of prospective changes to the grid, and As-Built Master Model Version Creation, which creates new versions of the as-built by applying the prospective changes to previous as-built version as the changes become real. A second component function under Physical Model Management is External Model Management, which supports the import and utilization of external grid modeling from outside sources.

The Assembly Support function has three main component functions. The first is Derived Physical Model Creation which performs a variety of operations – like the application of changes, or the simplification or merging of modeling - to create new grid modeling. The second is Situation Model Creation, which creates operating state (load/generation, switch state, etc.) definitions based on input from a variety of sources. The third is Assembly Creation, which uses modeling created by the other functions to create the grid models and cases that consumer applications need.

The Grid Model Management business function outlines the activities a GMM tool must support to effectively enable utility-wide management of distribution grid model data.

A more detailed description of each of the functions shown in Figure 2-1 is provided in Appendix A.

3

CIM ‘META’ CLASSES: THE GMM’S DATA BUILDING BLOCKS

The CIM organizes grid model data and provides a basis on which a coordinated grid model data management strategy can be built. Two kinds of CIM constructs are essential to the activities of a GMM tool: constructs that support the representation of the grid and constructs that allow the management of modules of grid representation data.

The CIM is mature and field-tested in its modeling of balanced grid representations (and its support of unbalanced modeling is being refined in the latest CIM releases). Work on defining the classes that support the modularization of grid representation data has been underway for a number of years and is nearing completion. The concepts represented by those modularization classes – the CIM calls them ‘meta’ classes – are largely understood and agreed on in the CIM working group, though the actual Unified Modeling Language (UML) model has yet to be finalized.

Proposed UML for CIM ‘meta’ Classes

A high-level, proposed UML model of those ‘meta’ classes is provided in this report because they are essential to a discussion of GMM functional requirements. A diagram of the simplified UML model for ‘meta’ classes is shown in Figure 3-1.

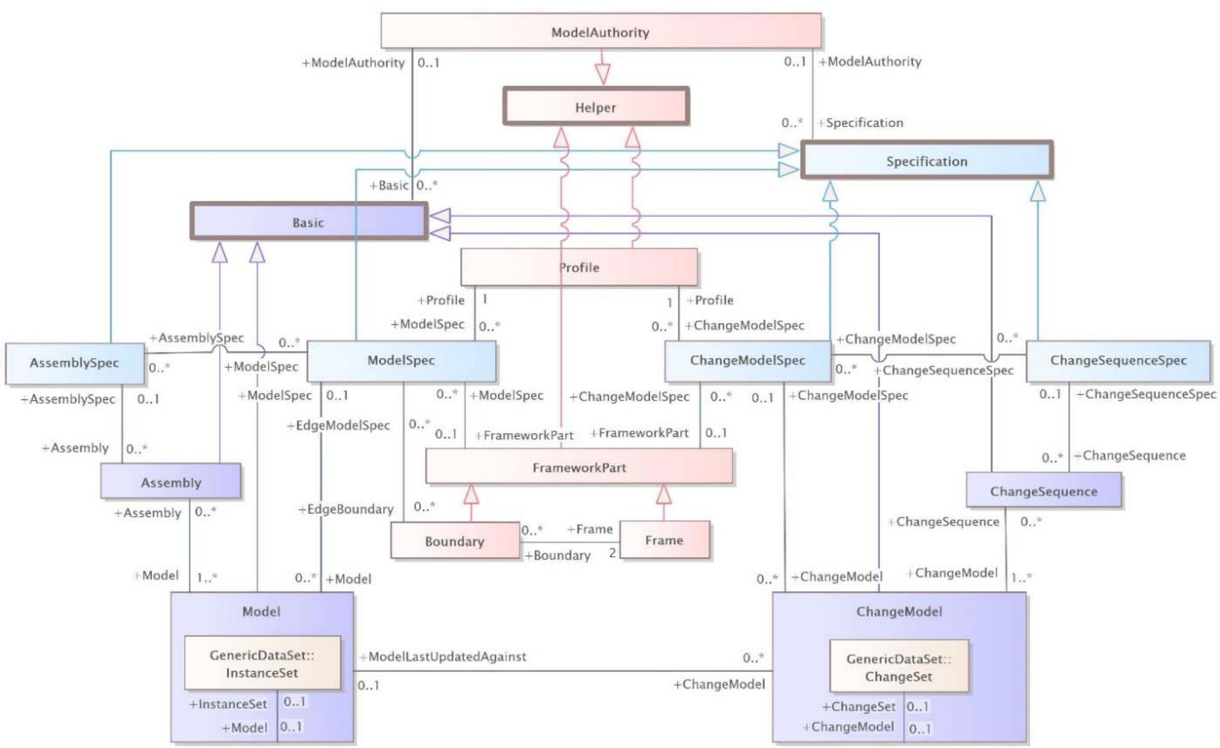

Figure 3-1
Conceptual model of the CIM’s ‘meta’ classes

The following sections describe features of the 'meta' classes relevant to the discussion of GMM requirements. For the purposes of this report, it is useful to divide 'meta' classes into three categories: Basic, Specification and Helper.

Basic 'meta' classes

The Basic 'meta' classes are the ones that are used directly in the exchange of grid representation data and there are four of them:

- **Model**
- **ChangeModel**
- **Assembly**
- **ChangeSequence**

Instances of all the Basic 'meta' classes share the characteristic of being immutable - once an instance of a Basic 'meta' class is created, it cannot be updated.

Model

A Model is a snapshot of the state of a profiled type of grid representation data. A Model typically is associate with a Model Authority which created it and typically 'fits' in a Frame or Boundary. Attributes of Model provide the 'context' data for its grid representation snapshot (which is contained in its InstanceSet). A Model contains a specific type (profile) of grid representation data. For grid model management discussions, it is useful to group the types of grid representation data as follows:

- Physical grid representation data describes the properties of the grid that are inherent its construction – in other words, the electrical properties of equipment, the way the equipment is connected, measured, and controlled, along with where it is geo-spatially located. The CIM 61970 EQ, SC, DY, GL, and OP profiles describe physical data.
- Situation grid representation data describes a grid operating state, including generation and load (energy in and out of the grid), control area net interchange, control settings, switch states, and operating limits. The CIM 61970 SSH profile describes situation data.
- Solution grid representation data describes a set of values of steady-state variables (principally voltages and flows) produced by a solution algorithm. The CIM 61970 TP and SV profiles describe solution data.
- Visualization data, which describes the layout of diagrams displaying any sort of grid representation data for human viewing. The CIM 61970 DL profile describes visualization data.

With the above types of grid representation data in mind, a number of types of Models useful in the discussion of GMM requirements can be defined:

- A **physical Model** contains physical data and, if it is associated with a Frame, contains object references which resolve to all objects in the Models of Boundaries which adjoin the Frame. Several categories of physical Models are useful to the GMM requirements discussion:
  - A **master physical Model** is the most detailed physical Model described for a Frame for which an organization has primary modeling responsibility. A master physical Model is

sufficiently detailed that the physical modeling of all consumer applications served by the GMM can be derived from it. There is only one current master physical Model for a Frame, and it is called the master as-built physical Model.
  - A **derived physical Model** is a physical Model derived as a result of the execution of one or more Operations.
  - An **edge physical Model** – a edge physical Model has references to objects in Models occupying only some of the Boundaries of the Frame in which the physical Model 'fits'.
- A **situation Model** contains situation data.
- A **solution Model** contains solution data.
- A **visualization Model** contains visualization data. A **physical visualization Model** is a special type of visualization Model which references objects in a physical Model.

ChangeModel

A ChangeModel is a snapshot of a collection of changes (all related to the same triggering event) which can be applied to a Model. Attributes of ChangeModel provide the 'context' data for the snapshot of grid representation changes contained in the ChangeModel's ChangeSet. Two types of ChangeModels are useful in the discussion of GMM requirements:

- A **master physical ChangeModel** – a ChangeModel whose prospective changes can be applied to a master physical Model. The changes described by a master physical ChangeModel can represent a variety of real-world situations, for example:
  - future construction projects (which can be expressed in general terms if they have resulted from a planning activity or in much greater detail if they describe a project after design engineering has completed).
  - field work that has resulted in changes that need to be made.
  - error corrections that need to be applied.

  Once a master physical ChangeModel is applied to the master as-built physical Model (which creates the next version of the master as-built physical Model), no further versions of a ChangeModel can be created.
- A **derived physical ChangeModel** – a ChangeModel whose prospective changes can be applied to a derived physical Model.

Assembly

An Assembly is a snapshot of a collection of Models.

ChangeSequence

A ChangeSequence is a snapshot of an ordered collection of ChangeModels. A ChangeSequence can be used, among other things, to define a construction project with stages – the construction project ChangeSequence would reference multiple ordered ChangeModels, each representing a stage.

Specification 'Meta' classes

There is a **Specification** class (named <Basic 'meta' class name>Spec) associated with each of the Basic 'meta' classes. The various Specification classes allow Basic class instances with the same purpose to be grouped together. Specification classes are the mechanism by which versioning of Basic class instances is done, although the notion of 'purpose' would certainly allow the Specification classes to be used to meet other requirements.

Because Specification 'meta' classes are used to define specific (business) purposes, the required attributes and associations of the Specification 'meta' classes are extremely likely to vary from one exchange environment to another. It is expected that the initial CIM 'meta' classes will include modeling that supports the addition of 'tags' to Specification 'meta' classes which allow implementations to describe both local attributes and local associations using standard CIM constructs. In contrast, the attributes of Helper 'meta' classes are likely not to vary nearly as much between implementations and are good candidates for direct modeling in the CIM UML.

There are likely a set of Specification 'meta' class attributes and associations whose use is so common that the initial CIM UML will represent them. Examples which are shown in Figure 3-1 are:

- Associations of ModelSpec and ChangeModelSpec to a specific Frame/Boundary or a specific Profile
- An association between an AssemblySpec and multiple ModelSpecs that could be used to describe a 'template' used to validate Assembly contents.

Helper 'Meta' classes

The **Helper** classes support ways of modularizing grid representation data. Helper classes are:

- **Profile** – a non-overlapping subset of CIM grid representation classes/properties/associations. A Profile describes a type of grid data at the level of a CIM profile standard. Currently, for exchange environments basing their exchanges on 61970 standards, a Profile object could be expected to exist for each of the following profiles: Equipment (EQ), Short Circuit (SC), Operation (OP), Dynamics (DY), Diagram Layout (DL), Steady State Hypothesis (SSH), Topology (TP), and State Variables (SV). For exchange environments basing their exchanges on 61968 standards, a Profile object would exist for each of the following profiles: Functional, Electrical Properties, Geographical, Assets, Asset Catalog, Customers, Steady State Hypothesis and Topology). (Note that this UML Profile class is not the same thing as the unique, machine-readable Resource Definition Framework Schema (RDFS) profile a given data exchange conforms to - that is described in the serialized InstanceSet or DataSet.)
- **Frame** and **Boundary** – children of the FrameworkPart class, Frame and Boundary, together are used to describe non-overlapping areas of the grid, allowing Models to be cleanly assembled.
- **ModelAuthority** – the entity which authored an instance of a Basic 'meta' class.

A detailed discussion of the data modularization the CIM's 'meta' classes support, along examples of their use, is provided in the EPRI report titled, "Common Information Model (CIM) Support for Distribution Grid Model Data Management"[1].

4

GMM FUNCTIONAL REQUIREMENTS

Overview of Requirements

The basic functional requirements for a distribution GMM tool are outlined below. Annotations providing additional detail for a number of the requirements are provided in the section following the requirements.

1. **Model Management Functional Requirements**

   1.1. **Maintain Master Physical Modeling** for the grid within the Frames/Boundaries for which an organization has primary modeling responsibility.

      1.1.1. **Maintain master physical ChangeModels**

         1.1.1.1. Support the creation and update[1] of master physical ChangeModels expressing prospective changes to the master modeling of the physical grid, using data supplied via:

         - Manual entry including the use of schematic displays.
         - Import of CIM fragments (incomplete modeling of a variety of types from individual sources).
         - Import of generalized modeling data from other sources (after validation and processing)

         1.1.1.2. Support the creation and update[1] of master ChangeSequences of master physical ChangeModels.

         1.1.1.3. Maintain history of versioned master physical ChangeModels and their master Change Sequences.

      1.1.1. **Maintain master as-built Models**

         1.1.1.1. Support the creation of new instances of versioned master as-built physical Models via the application of master physical ChangeModels to the previous master as-built physical Model.

         1.1.1.2. Maintain history of versioned master as-built physical Models.

   1.2. **Import External Modeling from Others** typically for the grid outside the Frames/Boundaries for which an organization has primary modeling responsibility.

      1.2.1. **Receive external pre-built modeling**

         1.2.1.1. Support the import of Basic 'meta' objects which adhere to the structure of the exchange environment.

[1] The use of the phrase 'support the creation and update' is used in several places in reference to Basic 'meta' objects. This is done to aid human understanding. Technically speaking, the 'update' is accomplished by creating a new version.

1.2.1.2. Support validation of received Basic 'meta' objects as to grid representation content and alignment with the exchange environment structure.

1.2.1.3. Support storage of these Basic 'meta' objects in the Repository.

1.2.2. Act as a component consumer

1.2.2.1. Support the import of Models and ChangeModels provided by other tools acting as GMMs under the component data exchange approach.

1.2.2.2. Support validation of these Models and ChangeModels as to grid representation content and alignment with the exchange environment structure.

1.2.2.3. Support storage of these Models and ChangeModels in the Repository.

1.2.3. Import generalized modeling

1.2.3.1. Support the import and validation of grid representations which are correctly serialized in the form of Basic 'meta' objects, but that do not align with the exchange environment structure.

1.2.3.2. Support validation of these Basic 'meta' objects as to grid representation content.

1.2.3.3. Support storage of these Models and ChangeModels in the Repository.

1.3. Build Consumer Modeling

1.3.1. Maintain derived physical Models

1.3.1.1. Support the creation of derived physical Models.

1.3.1.2. Maintain history of versioned derived physical Models as appropriate.

1.3.2. Maintain derived physical ChangeModels

1.3.2.1. Support the creation of derived physical ChangeModels, typically by differencing of physical Models.

1.3.2.2. Maintain history of versioned derived physical ChangeModels as appropriate.

1.3.3. Maintain visualization Models

1.3.3.1. Support the creation and update[1] of visualization Models and visualization ChangeModels.

1.3.3.2. Maintain history of versioned visualization Models and visualization ChangeModels as appropriate.

1.3.4. Maintain situation Models and situation ChangeModels

1.3.4.1. Support the import and storage of situation data patterns.

1.3.4.2. Support the creation of situation Models and situation ChangeModels from situation data patterns for specific physical Models.

1.3.5. Maintain Assemblies

1.3.5.1. Support the creation of Assemblies of Models.

1.3.5.2. Maintain history of versioned Assemblies as appropriate.

1.4. Perform Model Validation

1.4.1. Execute complex validations on Models

1.4.1.1. Support the routine or ad-hoc execution of complex validations on Models or combinations of Models.

1.5. Export Modeling to Consumers

1.5.1. Supply pre-built modeling to consumers

1.5.1.1. Support the provision of correctly serialized Models or Assemblies (or ChangeModels describing the difference between Model versions) to pre-built modeling consumers according to their requirements.

1.5.2. Supply components to consumers

1.5.2.1. Support the provision of correctly serialized Models and ChangeModels to component consumers according to their requirements.

1.6. Interact with Modeling Exchange Environment(s)

1.6.1. Act as supplier

1.6.1.1. Support the creation and update of the 'meta' objects appropriate to the coordinator or modeling authority role.

1.6.1.2. Share updates to these 'meta' objects with other applications participating in the modeling exchange environment as appropriate.

1.6.2. Act as participant

1.6.2.1. Support the import, storage of updates to Helper and Specification 'meta' objects.

1.6.3. Participate in multiple environments

1.6.3.1. Support participation in as many grid data exchange environments as necessary, in the appropriate roles.

1.6.4. Internally leverage Helper and Specification 'meta' objects

1.6.4.1. Support the use of the Helper and Specification 'meta' objects, as appropriate, to validate received modeling, to manage Basic 'meta' objects in the Repository, and to serialize exported modeling.

1.7. Implement Operations, Procedures and AuditTrails

1.7.1. Use an Operations-based approach

1.7.1.1. Support an Operation-based approach to model maintenance activities, defining the actions executed in the creation of all Basic 'meta' objects as Operations.

1.7.1.2. Support the definition of new types of Operations by users.

1.7.2. Support Procedures

1.7.2.1. Support the automated execution of Operations by means of Procedures.

1.7.3. Create AuditTrails

1.7.3.1. Track the history of Operations by which each Basic 'meta' object is created.

2. General Functional Requirements

2.1. Maintain a Repository structured to support GMM functional requirements

2.1.1. Support storage of 'meta' objects

2.1.1.1. Support the storage of all classes of 'meta' objects in a persistent Repository.

2.1.2. Support retrieval of 'meta' objects

2.1.2.1. Support the query and retrieval of any class of 'meta' object from a persistent Repository for use by various GMM activities.

2.2. Maintain an [Object Registry]

2.3. Support Individual [User Workspaces]

2.4. Provide [Extensibility]

2.4.1. Support Repository structure extensibility

2.4.1.1. Implement a model-driven Repository structure designed to enable extensions.

2.4.2. Commit to timely interface updates

2.4.2.1. Implement interface updates to support new versions of CIM standards in a timely manner.

2.5. Support for local requirements related to hosting and cyber security

Requirement Notes

Manual data entry

A GMM should support at least the following mechanisms by which grid data can be manually entered:

- Electrical schematics, including features to ease data entry like templates of symbols representing collections of CIM objects
- Object attribute dialogs
- Tables for bulk update

Initial data validation, including constraints on entered values and initial cross-attribute error checking, should done as data is entered. [Back to Overview]

Import of grid modeling from other sources

Aside from sources supplying fragment input for master model creation, interactions between suppliers of grid modeling and a GMM tool fall into three categories:

- Supply of pre-built modeling data – suppliers provide pre-built Models or Assemblies, or even ChangeModels expressing the difference between two versions of a Model (this is done as a means of reducing the amount of data required to be exchanged). An example of this could be a neighboring TSO routinely providing a GMM tool at a transmission utility with the model from its latest database build.
- Supply of component data - suppliers provide components from another application playing a GMM role. In this case, the provided data is most likely to include the current version of a physical Model (typically the master as-built physical Model) and current versions of a set of ChangeModels which can be applied to it.
- Supply of generalized model data – suppliers provide correctly serialized grid modeling - in the form of Basic 'meta' objects – that does not align with the exchange environment structure. [Back to Overview]

derived physical Models

Derived physical Models are the result of an endless number of physical Model derivations that a GMM tool might want to perform. All derivations should be implemented as Operations, some of which could be built-in product Operations, others of which would be user-defined. Examples of typical physical Model derivations include:

- Application of one or more physical ChangeModels to an existing physical Model to create new physical Model.
- Merging of multiple physical Models to create new physical Model.
- Creation of edge physical Models.
- Creation of a physical Model with computed limit sets.
- Creation of a simplified physical Model from a more detailed physical Model:
  - Synthesis of per segment impedances into line impedances
  - Creation of a bus/branch model from a node/breaker model
  - Removal of specific types of modeling detail. [Back to Overview]

derived physical ChangeModels

One attractive GMM feature, that seems potentially difficult to implement, is the support for versioned derived physical Models and ChangeModels which mirror the versioned master physical Models and ChangeModels. The benefit of this mirrored modeling is significant and is illustrated by the following example. Assume that there are the following sets of versioned Models:

- A set that conforms to a master physical ModelSpec.
- A set that conforms to an operations physical ModelSpec. Models in this set are derived from Models associated with the master physical ModelSpec, with the derivation removing connectors and CTs present in the master physical model that are not supported in the Operations tool.
- A set that conforms to a planning physical ModelSpec. Models in this set are derived from Models associated with the master physical ModelSpec, with the derivation replacing the multiple ACLineSegments modeled for each line in the master physical Model with a single ACLineSegment with equivalent impedance. This is done to prepare Models for use by a Planning tool.

It would be useful if the master physical ChangeModels used to describe future changes to the grid and to update the master as-built physical Model could also be ‘mirrored’ in operations physical ChangeModels and planning physical ChangeModels. The ability of a GMM to do this sort of mirroring enables the use of ChangeModels as the primary means of communicating about and sharing grid model data across a utility enterprise. This has tremendous potential benefit to enterprise-wide data management, allowing the tools used in various parts of the organization to build their own study models for their own purposes, using their own ChangeModel representation of the same real-world change (like a construction project). [Back to Overview]

visualization Models

It is expected that multiple mechanisms will be supported for the creation of visualization Models representing the contents of physical Models. These include:

- Manual creation by means of Graphical User Interface.
- Automatic generation of schematics potentially organized by CIM containment structures and with layouts based on either equipment voltage level or geo-location. [Back to Overview]

situation data patterns

The situation data actually provided in grid representation (p, q, switch states, control settings) is typically derived from information like:

- Normal breaker/control state definitions
- Outage schedules
- Energy (load and generation) schedules
- Energy forecast curves
- Contingency lists

The source data is assumed to be structured into machine-readable patterns (which hopefully will one day be expressed as standard CIM data structures), many of which describe data which varies over time. Situation data patterns are provided to a GMM tool by applications (like load forecasting, outage analysis or market systems) that support other business functions. [Back to Overview]

complex validations of Models

The execution of localized data validation is called for during manual entry and the execution of Operations. The ability to perform complex validations at a larger scale is also GMM requirement. A GMM should support comparison of a Model against rules that validate the reasonableness of grid modeling, as well as being capable of executing power flows against Models (or combinations of Models) containing sufficient amounts of physical and situation data. [Back to Overview]

Export of grid modeling to consumers

Interactions between a GMM and consumers of grid modeling fall into two categories:

- Pre-built modeling data provision – consumers expect the GMM to provide Models pre-built to their requirements. These can be in the form of Models or Assemblies, or even ChangeModels expressing the difference between two versions of a Model (this is done as a means of reducing the amount of data required to be exchanged).
- Component data provision – consumers want to construct their own Models from components supplied by the GMM. In this case, the exchanged data is most likely to include the current version of a physical Model (typically the master as-built physical Model) and current versions of a set of ChangeModels (in which the consumer has interest) which can be applied to it. [Back to Overview]

Participating in data exchange environment

A data exchange environment is one where a collection of applications is sharing data in compliance with a single (and commonly understood) semantic model. In most grid model data exchange environments, there is likely to be a structure of Helper and Specification 'meta' objects that define the organization of the shared data. Each participating application will likely have some awareness of the structure – at least those parts of it related to the grid modeling it is supplying or consuming. GMM tools must always be 'structure aware' and able to act as both supplier and consumer of appropriate parts of the structure.

The Helper and Specification 'meta' objects making up the structure need to be created and maintained. The collection of objects comprising the structure will likely vary significantly from one environment to the next and no firm picture can be drawn of what the typical structure will look like or how it should be managed, until the initial CIM 'meta' modeling has been finalized. Two roles of suppliers, however, seem likely - coordinator and modeling authority – which will share the responsibility for supplying Helper and Specification 'meta' objects. The following is presented to provide an idea of what the responsibilities associated with the management of an exchange environment structure might look like:

- Helper 'meta' objects
  - Management of Profile objects. Profile objects would typically be predefined by CIM profile standards, and their management would likely be done by a GMM in the coordinator role.
  - Management of Frame and Boundary objects. Since Frames and Boundaries need to be defined such that their extents completely cover – in a non-overlapping fashion - the grid which is the subject of the environment's data exchanges, they are likely to be defined by a GMM playing the coordinator role.
  - Management of Model Authority objects. Since Model Authorities are inherently related to Frames and Boundaries, they are likely to be defined by a GMM in the coordinator role.
- Specification 'meta' objects
  - These objects define groups of Basic 'meta' objects (Models, Assemblies, ChangeModels, ChangeSequences) and it seems likely that their definition would be done by a GMM either in the coordinator role or the modeling authority role – the latter being the case when a GMM is the Model Authority producing the Basic 'meta' objects which belong to the Specification.

Some GMM tools will participate in more than one data exchange environment. (Transmission GMMs will almost assuredly have to do so as they interact with interconnection-level applications using one data exchange environment and distribution-level applications in another. Distribution GMMs may find themselves in a similar situation as they interact with both transmission counterparts and micro-grid operators.) GMM tools should support the ability to participate in multiple data exchange environments and to map between their structures and if necessary. [Back to Overview]

Operations, Procedures and AuditTrails

GMM support for both scripting (in support of process automation) and audit trails is essential. The CIM concepts of Operations, Procedures, and AuditTrails support these needs:

- **Operation** - a request to execute a function that accepts as arguments one or more 'meta' objects (or a list of 'meta' objects of the same class) and produces a 'meta' object (or a list of 'meta' objects of the same class) as output. Common Operations that a GMM tool needs to implement include:
  - Create a Basic 'meta' object.
  - Difference any two Basic 'meta' objects of the same type.
  - Retrieve 'meta' object from Repository based on query arguments.
  - Store 'meta' object in Repository.
  - A whole set of Operations supporting derived Model creation and import and export of grid modeling.

  Appropriate error-checking and validation actions should be defined as part of each Operation.
- **Procedure** – an ordered set of Operations.
- **AuditTrail** - a history of the Operations executed in the creation of a Basic 'meta' object. An AuditTrail should include the Operation name and the identities of the input arguments.

The initial CIM 'meta' model will provide data structures for describing Operations, Procedures and AuditTrails (despite the fact that they are not shown in Figure 3-1). It is possible that a future version of the CIM will standardize some types of Operations (because many, like those cited above, are universal), but for now the definition and naming of Operations is left up to each GMM tool. [Back to Overview]

Repository

A persistent Repository of 'meta' objects is at the core of a GMM. While GMM tools will structure their internal data stores to support their specific features capabilities, a GMM must store data in such a way that it can be retrieved and supplied in accordance with the CIM-based structures of the exchange environments in which it participates. The Repository must be able to store both 'meta' objects aligned with exchange environment structures and those that are not. [Back to Overview]

Object Registry

Data exchange environments very frequently have the need to map identifiers across applications. An Object Registry is an application which supports CIM-based mapping of object IDs across applications participating in an exchange environment and implements automated or manual features to facilitate the definitions of mappings. [Back to Overview]

user workspaces

A user workspace behaves like a private modeling space for the user. A GMM tool supports multiple user workspaces in which activities are taking place concurrently. Activities in one user's workspace do not affect activities in another user's workspace. A workspace supports the execution of any defined Operation. The state of a workspace can be saved by the user and recalled on demand. The result of activities in a workspace are not saved until a user explicitly does so. [Back to Overview]

extensibility

GMMs manage data which is exchanged using CIM-based data interfaces and the CIM continues to grow to meet industry needs, so GMM extensibility is critical for both internal GMM data storage and supporting data exchanges.

A GMM needs a model-driven Repository structure designed to enable updates to reflect CIM UML changes. Its model-driven structure should also enable local extensions necessary to support utility enterprise semantic models. CIM UML updates are often accompanied by updates to IEC CIM profile standards and GMM tool vendors should be committed to implementing export and import interfaces supporting new versions of CIM profile standards in a timely manner (at least as soon as customers request them). GMM tools must support the import and export of grid modeling aligning with several versions of CIM profile standards as it is almost certain that the various suppliers/consumers with which a GMM shares data will upgrade on different schedules.

5
REFERENCES

1. *Common Information Model (CIM) Support for Grid Model Data Management*. EPRI, Palo Alto, CA: 2022. 3002025386.

A

THE GRID MODEL MANAGEMENT BUSINESS FUNCTION

Descriptions of the Grid Model Management Business Function and Its Component Functions

Grid Model Management Grid Model Management provides the single source of truth for grid modelling; from a wide variety of source data it creates and maintains Physical Models and Projects as well as Situation Models and, from those building blocks, creates grid modelling tailored to each network analysis consuming business function.

1. **Assembly Support** Assembly Support constructs modelling to meet the needs of various network analysis activities based on Assembly Requirements supplied by grid model consuming business functions.

    a. **Assembly Creation** Assembly Creation uses Assembly Requirements to construct Assemblies tailored to meet the needs of specific network analysis activities performed within a variety of network model consuming business functions.

    b. **Derived Physical Model Creation** Derived Physical Model Creation uses Physical Models and Physical Model Updates as input to procedures that create Derived Physical Models for a variety of business purposes. Model derivation can perform a variety of services, like selecting and combining Models and Projects, simplifying, reducing, calculating additional parameters, creating audit trails, etc.

    c. **Situation Model Creation** Situation Model Creation uses Future Energy Patterns, Equipment Outage Plans, and other inputs to construct Situation Models for a specific, cohesive set of Physical Models.

2. **Physical Model Management** Physical Model Management performs the activities (creating, updating, receiving, statusing, versioning, etc.) that maintain the Physical Models and Projects (both internal master and external) used by other GMM business subfunctions.

    a. **External Model Management** External Model Management receives physical grid model data from other modelling entities, performs any necessary processing and manages the resulting External Physical Models and External Model Updates.

        i. **External Model Update Capture** External Model Update Capture receives physical grid model changes from other modelling entities, performs any necessary processing and creates or updates External Model Updates.

        ii. **External Physical Model Capture** External Physical Model Capture receives physical grid model 'snapshot' data from other modelling entities, performs any necessary processing, and creates or updates External Physical Models.

b. **Master Grid Modelling** Master Grid Modelling encompasses all the activities that maintain master physical modelling, including updating the As-Built Master Model and managing Physical Model Updates and various types of Projects that track the evolution of the grid over time.

    i. **As-Built Master Model Version Creation** As-Built Master Model Version Creation applies Master Model Updates to the As-Built Master Model (creating a new version of the As-Built Master Model) and then archives the Master Model Updates.

    ii. **Master Update Management** Master Update Management maintains the full range of prospective updates (from years-out Planned Projects to error corrections in the form of Master Model Updates to be applied immediately) to the As-Built Master Model throughout their lifecycle.

        1. **Designed Project Capture** Designed Project Capture creates a Designed Project reflecting the impact of a completed Engineering Design Project. If a predecessor As-Planned Project exists, it is desirable that its power system objects are used in creating the Designed Project.

        2. **Master Model Update Capture** Master Model Update Capture brings in data about completed work or needed as-built corrections and creates Master Model Updates. If a Designed Project exists, it will typically be updated to create a Master Model Update. For changes related to completed work, Master Model Update Capture rationalizes information from As-Built Change with information from As-Built Facility Records.

        3. **Planned Project Capture** Planned Project Capture creates a Planned Project, available for use by various GMM business subfunction, from a Grid Planning Project produced by Grid Planning.

3. **Modelling Metadata Management** Modelling Metadata Management is responsible for performing actions related to Modelling Metadata which enable the Grid Model Management business function to participate in a model management environment.

About EPRI

Founded in 1972, EPRI is the world's preeminent independent, non-profit energy research and development organization, with offices around the world. EPRI's trusted experts collaborate with more than 450 companies in 45 countries, driving innovation to ensure the public has clean, safe, reliable, affordable, and equitable access to electricity across the globe. Together, we are shaping the future of energy.

3002025388

EPRI
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