Research on Publishing CIM Model Change Events through OPC UA

Zewu Peng[1], Churan Deng[1,*], Bojian Wen[1], and Qingping Xu[2]

[1]Information Centre of Guangdong Power Grid Corporation, 510600 Guangzhou, China.
[2]Weihai CIMSTech Co., Ltd, 264200 Weihai, China

Abstract. This paper studies how to publish the change events of CIM model using OPC UA, and proposes a method to publish CIM model change events using OPC UA based on mapping CIM model to OPC UA address space. By comparing with the CIS GDA model change notification service function in the legacy CIS interface, it is confirmed that the OPC UA service completely covers the original CIS GDA model change notification service function. Base-Model-Change-Event-Type and the more detailed General-Model-Change-Event-Type, Semantic-Change-Event-Type are used to provide appropriate model change information for clients through OPC UA Event service. CIM model change events in data platform of distribution grid and utilization system are published using OPC UA, the research results have been verified.

1 Introduction

The IEC61970, IEC61968 and IEC62541 series standards developed by the International Electrotechnical Commission (IEC) have been widely used in the power industry. The domestic and foreign power companies are now using these standards in a large number of practical application system of power dispatching, electric power production and marketing. The data platform of distribution grid and utilization system of Guangdong Power Grid (referred to as DUSYS) is built entirely on international standards, mainly including two aspects: 1) information model, the Unified Information Model of Distribution and Utilization system (DUUM), is based on IEC TC57 Common Information Model (CIM); 2) data is provide through OPC Unified Architecture (OPC UA), the new version of Component Interface Specification [1]. The data platform models and collects the primary and secondary data of distribution grid and utilization system together to achieve full integration of distribution network production and operation data.

Through continuously aggregating data from multiple application systems in the field of distribution grid and distribution system, the DUSYS becomes a centralized data access point of grid model. To ensure that platform-based applications can quickly track model changes as platform-managed grid models change, the platform needs to issue model change notifications whenever grid model data changes. So that each application can start the corresponding model data synchronization process, tracking grid model changes in a timely manner, and ensure that the grid model data used is complete and correct.

In this paper, we research how to use OPC UA to publish the CIM model change data. Firstly, the CIM and OPC UA specifications are analysed to determine the management mode of CIM model in the OPC UA address space. Then a standardized publishing method of CIM model change events is determined, and a software implementation scheme is given.

2 OPC UA address space management for CIM model

The Common Information Model (CIM) is an abstract model that describes all the major objects of an electric utility, especially those related to electric operations [2]. The Common Information Model of Power System - TC57 CIM, provided by the International Electrotechnical Commission's 57th Technical Committee (TC57) has been most widely used. IEC TC57 CIM integrates the logical composition of the power grid, the physical entities of the equipment with the locations, document, records and measurement data required for power management. It uses information technology such as object-oriented modelling and XML to refine the information model, covers all dimensions of power information. This good information model has played an important role in the establishment of the standardized data centre for distribution and utilization power network management.

The data access services come with the Power CIM model were originally due to the EPRI's Energy Management System Application Programming Interface (EMS-API) project and were eventually defined as Component Interface Specification (CIS). The first edition of CIS service includes series of standards such as IEC 61970 402/403/404/405/407, which are based on

* Corresponding author: dengchuran@gdxx.csg.cn

the relevant standards of OPC and OMG and have been widely applied. With the introduction of the concept of smart grid, the demand for power system management and information exchange has been further improved. The CIM model interface system needs to be adapted to the needs of Internet access. The first version of the CIS service, which is mainly targeted at applications running on the LAN or intranet, has also been upgraded to use the OPC Unified Architecture(OPC UA) adopted directly by IEC 62541.

Compared with the first version of CIS service that data is difficult to be integrated by client applications due to model, real time, event and history data are separately described, the OPC UA specification of the new generation of CIS integrates the classic OPC data access specification into a single service-oriented architecture. OPC UA uses a consistent and complete address space, security model and service model to unify the metadata (information model), data, history and events into one address space for collaborative management, providing a series of secure and reliable services for data accessing, alarming, historical data accessing [3].

OPC UA uses reliable communication mechanisms, configurable timeouts, automated error checking and automatic recovery mechanisms to ensure efficient communications and data transfer across firewalls. And it also enhances the robustness of the architecture with technical advantages of unified model control, data security exchange, high reliability and platform independence, and can be adapted to many fields such as device data management and communication, enterprise modelling and information exchange.

The objects and related information provided by the OPC UA server to the client is referred to as the server's address space. In OPC UA, the rules and basic components of information model building are the OPC UA address space model. The UA Information Model uses notions to define its own, domain-specific types and constraints, as well as well-defined instances.

Although OPC UA has been adopted as a new generation of CIS service standards, it is not specifically formulated for power automation applications but rather as a common inter-industry standard for industrial control automation applications. Unlike the CIM model explicitly adopted as the power information model, the address space model defined by the OPC UA specification is an OPC UA proprietary information model description. Therefore, to apply OPC UA specification in power system, we must first solve the coordination and integration of power CIM model and OPC UA address space model, that is, managing CIM Model in OPC UA address space.

Table 1 shows the basic mapping schemes from CIM model to UA address space model [4].

Table 1. Rules of Mapping CIM Model to OPC UA Address Space Model.

CIM Model		UA address space model
package(Schema)		UA Object Node
data	Primitive	UA built-in basic types
type	Enumeration	UA Enum Node
	DataType	UA VariableType Node
	Compound	UA VariableType Node
object type		UA ObjectType Node
object Instance		UA Object Node
attribute of object		UA Variable Node (MeasurementValue.Value or StateVariable.Value)
		UA Property Node
type of association between objects		UA Reference Type
instance of association between objects		UA Reference

These mappings define how the CIM model is organized in the OPC UA address space and are also the basis for publishing CIM model change events via OPC UA.

3 Standardized publishing of CIM model change events

3.1 Event service of CIS GDA

In the first version of discarded CIS services, the GDA Event Service defined in IEC 61970 403 Generic Data Access (GDA) gives a standard way of publishing CIM model change events. GDA Event Services provide methods to inform clients that specific data changes have taken place and ensure consistent data access.

The GDA event module allows the client to receive a model data change event callback from the server. The structure of the model data change event is *ResourceChangeEvent*, in the structure, *version* is a unique integer that the GDA server assigns to each data change, the value of *version* in an event equals to the value returned from the GDA's *current_version*() method at the time of the event. *Affected* is a sequence of resource identifiers, *verbs* is a sequence of numbers to indicate how data changed. For each member of the affected sequence, there must be a member of the verbs sequence. The verb value defined by the GDA standard is a set of enumerations that represent the type of data change (Table 2).

Table 2. Enumeration of GDA data change types.

Verb	Meaning
VERB_CREATED	object created
VERB_DELETED	object deleted
VERB_CHANGED	attribute changed
VERB_CANCELED	operation cancelled
VERB_CLOSED	operation closed

Among them, VERB_CANCELED, VERB_CLOSED corresponding to the business process-related actions, such as closing the work order or cancelling the control request, are less practical used; VERB_CREATED, VERB_DELETED and VERB_CHANGED corresponding to the model creation, deletion, and attribute value changes, respectively, by setting the corresponding *affected* value to identifier of class or object, can describe model changes at schema or object level. If the *affected* sequence length is zero, the

client MUST assume that any or all of the data provided by the data provider may have changed.

Applications that concerns about changes in the CIM model typically use GDA Query service to get the required CIM model data during the initialization phase, then register GDA event listener to listen for changes. Upon receipt of the change notification, the GDA query service is used to obtain the up-to-date status of the change data so as to keep track of the CIM model data continuously to ensure the integrity and consistency of the CIM model data used.

3.2 Event service of OPC UA

3.2.1 Event model

The OPC UA event model defines a common event system that can be used in different areas to report the occurrence of an event through an event notification. The event is not directly accessible in the address space. Clients can subscribe to the corresponding events via objects and views. The node's *EventNotifier* attribute determines whether events can be subscribed to from the node.

OPC UA defines a hierarchy of extensible basic event types, with different fields for different event types. Clients can retrieve this information through the event type hierarchy that the server exposes in its address space. Using the event type hierarchy information, the client can create a filter based on the fields of interest and the type of event. OPC UA defines a series of standard event types that are represented in the address space as ObjectType node and directly or indirectly based on *BaseEventType*. *BaseEventType* uses UA properties to expose fields listed in Table 3.

Table 3. Property Set of BaseEventType.

UA Property	Value type	Meaning
EventId	ByteString	unique identifier generated by the server
EventType	NodeId	node ID of the event type
SourceNode	NodeId	node ID of event source
SourceName	String	description of event source
Time	UtcTime	event time
ReceiveTime	UtcTime	time when event receivedby UA Server
LocalTime	TimeZoneDataType	timezone
Message	LocalizedText	localized text description of the event
Severity	UInt16	indicating the severity of the event

Each concrete event type that is a descendant type of *BaseEventType* can extends specific fields according to its own characteristics.

In the OPC UA address space, the standard reference type *HasEventSource* associates event sources in a non-looping hierarchy whose child reference type, *HasNotifier*, is used to associate different node nodes of the *EventNotifier* to establish the hierarchy of event notification objects organization. When an event notifier references another event notifier, all events in the referenced event notifier are exposed by the reference.

The Server object, which is a standard object in the OPC UA Address Space, acts as the server's root notifier. It always provides server-wide events as if *HasNotifier* was declared for all event sources in the server. Clients can subscribe to the Server object to get all the events on the server.

3.2.2 Event service

OPC UA provides complete service support for subscribing, filtering and releasing events, including *View* services set, *Attribute* services set, *MonitoredItem* services set, and *Subscription* services set. The UA view is a publicly defined subset of address spaces created by the server. The view defaults to the full address space. The *view* service set allows the client to browse through the nodes in the view. *Attribute* service set are used for reading and writing node attribute values. The *MonitoredItem* services set defines services that allow clients to create, modify, and delete *MonitoredItems* used to monitor attributes for value changes and objects for events [5]. *MonitoredItem* monitors variable, attribute and event notifier. When it detectes an event that meets one of the criteria, a corresponding notification is generated. The *Subscription* service set is used by the client to create and maintain subscriptions. Subscriptions are entities that issue notification messages periodically for the monitoring items assigned to them. The notification message contains a common message header, followed by a series of notifications. The format of the notification depends on the type of item being monitored (i.e. variables, properties and event notifiers).

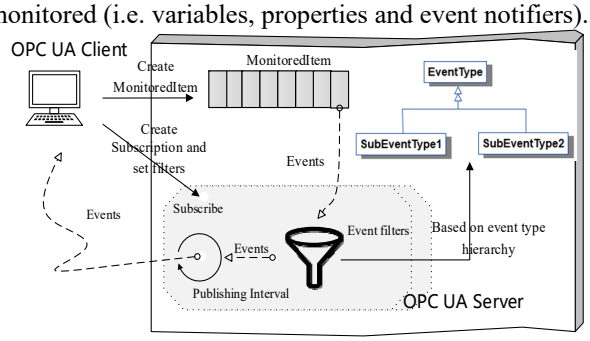

Fig. 1. Event interaction between OPC UA client and server.

Event interaction between OPC UA client and server starts with client's object nodes browsing in the in the address space through *View* service. By using the *Attribute* service to read the value of the *EventNotifier* property of a node, the client can determine which nodes generate events and create an event monitor entry for a specific node.

As shown in Figure 1, the client can create a filter based on the field of interest and the type of event by using the *View* service to retrieve the event type hierarchy information that the server exposed in the address space when creating the *MonitoredItem*. Finally, the client creates and maintains a subscription that contains a particular *MonitoredItem*, which can be notified by the server when a monitored item detects an event or alarm and sent to the client by subscription.

In the CIS GDA event service, the server sends events to the client through callbacks. Unlike CIS GDA, OPC UA uses polling for message exchange via one-way connections. This approach is firewall-friendly, and can reduce the development difficulty of the client. To ensure secure and reliable delivery of messages between clients and server, OPC UA has developed a fully-fledged subscription-publishing model that provides clear handling rules for subscription life-cycle maintenance, detection and retransmission of lost messages, message queue size management, etc. The relative services are *Publish* and *Republish*.

3.2.3 Model change event

In the OPC UA standard event system, an event of model change is used to indicate a change in the address space structure. Changes may be caused by adding, deleting nodes or references. Changes to the *DataType* attribute of a variable or variable type are also counted as model changes.

As shown in Figure 2, *BaseModelChangeEventType* is the basic type of OPC UA model change event. *BaseModelChangeEventType* does not contain detail information of the change, only tells a change have occurred, so the client needs to perform process as if all nodes have changed or each node maybe have changed. This is the same semantic representation of a GDA model change event when the *affected* member length is zero. *GeneralModelChangeEventType*, as a subclass of *BaseModelChangeEventType*, provides more detailed update details, contains information about the node being changed and the operation that generated the *ModelChangeEvent* (for example, adding a node, deleting a node, etc.).

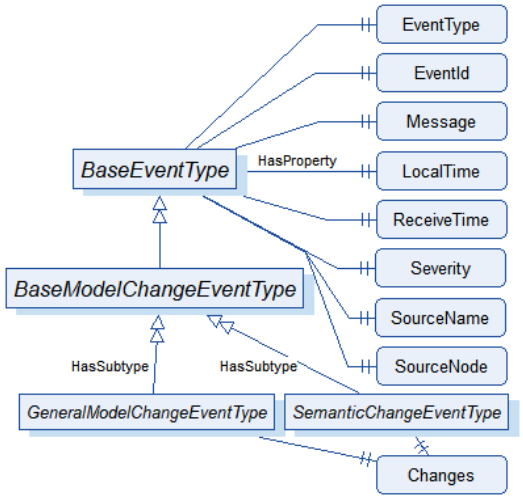

Fig. 2. Model change event types and properties.

An OPC UA server can publish model change notifications using types of model change events as appropriate, the specific granularity used is determined by the server based on performance requirements and update type. The server can also define subclasses to add additional information.

BaseModelChangeEventType does not extend any UA properties, *GeneralModelChangeEventType* adds a 'Change' property. The value type of 'Change' is *ModelChangeStructureDataType* . This datatype has such members: 1)*verb*, which describes the type of change that has taken place on the affected node; 2)*affected*, the node ID of the node that has changed. According to value of *verb* property, the client should assume that the affected node has been created or deleted, a reference has been added or deleted, or the data type has changed; 3)*affectedType*, if the affected node is an object or variable, *affectedType* contains the *NodeId* of the *TypeDefinitionNode* of the affected node, otherwise it is null.

The value of the *verb* member is a set of enumerations used to represent the type of data change, as shown in Table 4.

Table 4. Enumeration of OPC UA data change types.

Verb	Meaning
NodeAdded	"*affected*" node added
NodeDeleted	"*affected*" node deleted
ReferenceAdded	a reference to "*affected*" node added
ReferenceDeleted	a reference to "*affected*" node deleted
DataTypeChanged	When "*affected*" node is a *Variable* or *VariableType*, indicating the data type of Variable or *VariableType* has changed

As can be seen from the above table, comparing to data structure defined in CIS GDA Events(Table 2), the information provided by the OPC UA for model change event directly covers the change of the node reference (that is, the association between objects), which facilitates clients' processing.

3.2.4 Value change event

Different from CIS GDA event model, which publishes event with VERB_CHANGED when attribute changes, the change of value of variable in OPC UA address space does not cause model change event. In OPC UA, a change of value notification publication uses a different mechanism depending on the mode of value modelling.

The OPC UA Information Model defines two types of variables: data variable and property.

OPC UA data variable is used to represent the object's data, such as the temperature of the temperature sensor. According to the mapping rules in Table 1, the *MeasurementValue.Value* and *StateVariable.Value* attributes of the CIM model are mapped to data variables. OPC UA standard defines that changes in the value of data variables are published as data change notification which can be subscribed to by clients.

The OPC UA property is used to represent the characteristics of a node, such as temperature engineering units. According to the mapping rules in Table 1, attributes other than *MeasurementValue. Value* and *StateVariable.Value* are mapped as properties. OPC UA standard defines that property value change triggers semantic change event (*SemanticChangeEventType*, Figure 2).

3.3 Client processing

The client can listen directly to the root notifications of the OPC UA server to get the model change events in the entire address space. This can be achieved by creating an event monitor on the standard *Server* object node in the address space.

If only the field of the model change event itself is focused on, the filter may be set to limit the event type to the *GeneralModelChangeEventType* event type, and the field concerned is '*Changes'*.

4 Implementation of OPC UA publishing for CIM model change events

In the DUSYS, OPC UA Publishing for CIM model change events is implemented as a module of OPC UA server. The implementation scheme is shown in Figure 3. The server adopts the standard OPC UA hierarchical application architecture, realizes the mapping from CIM model data to OPC UA address space. Based on this mapping, CIM model changes are monitored, model change events are generated in time and posted to clients.

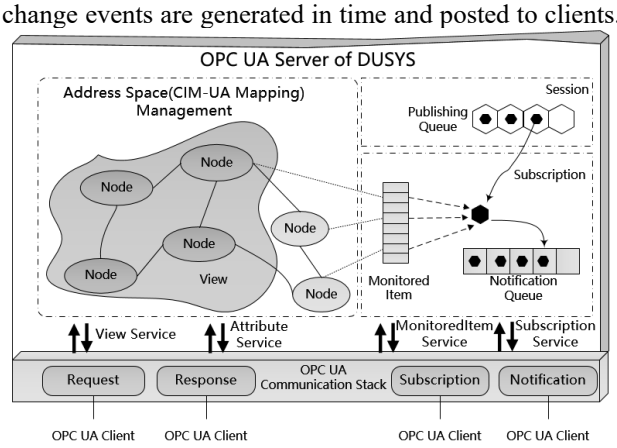

Fig. 3. Publishing OPC UA model change events.

The CIM model change events in the OPC UA server are generated according to the client using the node management service (*NodeManagement* service set) to add, modify, and delete nodes and references in the CIM model in the address space. The implementation of *NodeManagement* service set generates events when node management operation completed: *AddNodes* service creates events of *NodeAdded* type; *DeleteNodes* service create events of *NodeDeleted* type; *AddReferences* service creates events of *ReferenceAdded* type; *DeleteReferences* service creates events of *ReferenceDeleted* type.

When clients using the *Write* service of the *Attribute* services set to modify the "DataType" attribute of *Variable* or *VariableType* node, data change type of the event is *DataTypeChanged*.

When clients using the *Write* service of *Attribute* services set to modify properties of UA Variable node, event with *SemanticChangeEventType* type is generated if variable node not mapped from CIM attribute *MeasurementValue.Value* or *StateVariable.Value*.

To facilitate client-side processing, when the creation or deletion of property nodes mapped from CIM objects are caused by creation or deletion of CIM objects, the UA server does not generate CIM model change events for these property nodes, and only generated model change events for object nodes. Because the deletion of a CIM object node will result in relative reference to be deleted automatically, in that case, *ReferenceDeleted* events are not generated.

The OPC UA server of DUSYS enables full subscription feature of OPC UA subscriptions for all UA event types including model change events by creating message queues of *MonitoredItem*, data changes, and event notification for clients' subscription.

In publishing CIM model change events, notification message queues are configured as "queuing and buffering all notifications" or "only queuing and buffering recent notifications to be transmitted for the subscription", so the situation that events may occur at a faster rate than publishing rate (which is actually the frequency of client-side polls)can be handled. Each message in the notification message queue is assigned a sequence number that allows the client to detect the missing message, thus the standard mechanism of loss message detection and retransmission is supported.

5 Conclusion

Publishing CIM model change events through OPC UA is an important problem that must be solved when applying OPC UA to provide data accessing service for grid model data. Through analysis and comparison, it is confirmed that the OPC UA, as a new generation of CIS, has the ability to publish CIM model change events, and completely covers the original CIS GDA model change notification service function. The types and contents of events corresponding to different model changes are clearly given. The standardized publishing method and software implementation of CIM model change events are discussed in detail. Based on the mapping of the CIM model to the OPC UA address space, the OPC UA server generates corresponding events for the CIM model update operations and sends them to clients through the OPC UA services. Adopt event compression strategy is adopted to improve client processing efficiency.

The Scheme of publishing CIM model change events through OPC UA has been successfully applied to the data platform of distribution grid and utilization system of Guangdong Power Grid. It provides standardized CIM model change tracking method for platform-based power grid applications and satisfies data consistent requirements of distribution and utilization application, makes the management and exchange of power system data easier and more secure.

References

1. Y. Cao, J.G. Yao, S.Ch. Yang. Automation of Electric Power Systems. **35(17)**,1-4 (2011)
2. IEC. EMS-API part 301. (2009)
3. R. Hua. China Instrumentation. **2**,54-56 (2013)
4. S.Y. Xie, Q. Yang, Q.P. Xu. Automation of Electric Power Systems. **40(14)**, 115-121(2016)

5. IEC. OPC UA Part 4. (2011)