Research on Key Technology of the Address Space for OPC UA Server

Lu Huiming
Dept. Automation
North China Electric Power University, NCEPU
Beijing, China
luhuiming@ncepu.edu.cn

Yan Zhifeng
Dept. Automation
North China Electric Power University, NCEPU
Beijing, China
170769226@qq.com

***Abstract*—Due to a COM/DCOM component based technology; traditional OPC servers have technical defects on Web Data Exchange and Security Mechanisms, and highly rely on the Windows platform. This paper uses a new OPC Unified Architecture Specification to address the above issues, and integrates services of traditional OPC into one OPC server, which simplifies the function collaboration and enterprise deployment. By using code managed technology of Microsoft .Net and modular programming, the framework of OPC UA Server is constructed and the address space, security configuration and other basic modules are designed. Some classes such as the Node Manager, Data Manager, Browse Manager, Subscription Manager and View Manager of the address space are also developed for the server, with the Node and Reference technology of server's address space. An independent function module for OPC UA address space is also proposed in this paper.**

Keywords-OPC Sever; OPC Unified Architecture; OPC Address Space;

I. INTRODUCTION

At the mid-nineties of last century, OPC Foundation released its first Microsoft COM specification for Data Access (DA) that defined a series of access methods between the server and client, including read, write and subscription. The Foundation then developed some additional popular specifications for different types of data (Such as Alarm & Event and Historical Data Access), which soon became the dominated standard in industrial field. Due to a technological basis of Microsoft COM/DCOM, the existing OPC specifications have following insufficiencies:

1) Without a platform-independent nature, the traditional OPC can only be suitable for Microsoft's platforms;

2) The packets generated by DCOM are quite complex, and it is difficult to send packets on Internet because of the firewall [1];

3) It is hard to coordinate the independent servers of DA, A&E and HDA.

With the release of XML Web Service and the requirement of neutral communication, the OPC Foundation proposed XML-DA, a platform independent Web Service interface. Unfortunately, the size of XML message with the same information of DCOM is much larger than DCOM message, which results in a significant deterioration on the real-time performance of XML-DA compared to the COM version. In addition, XML-DA does not address the security issues when the data is transmitted over Internet. Therefore, XML-DA could not be regarded as an ideal replacement for the COM version.

OPC Unified Architecture, released by OPC Foundation recently, is a new direction of enterprise software architecture. Taking advantage of Web Service, XML, .NET technologies, OPC Foundation raised a definition of unified object and architecture for enterprise manufacturing model, which is a completely service-oriented architecture. Compared to the earlier OPC specifications, OPC UA raises a number of new features, such as built-in complex data, unified address space, cross-platform and abstract service functions [1], etc. The main advantages of OPC UA are:

1) Unified data access mode

Traditional OPC are independent for different applications. On the contrary, the OPC UA Server integrates data, alarm and event into a single address space. Client can get data, alarm and event applications only through one invoke [2].

2) Support complex data structures

The existing specifications only offer a simple hierarchical organization of item, whereas OPC UA offers Meta information models that can be easily extended. In OPC UA, you can add and delete the linkages between these data models. Therefore, client is unnecessary to understand the meaning of data with the detail description of the data model. This measure makes it easy to develop client software, and greatly improves the accuracy of the data significance.

3) Support multiple platform

Traditional OPC specifications are based on the Microsoft COM/DCOM technology. With the development of .NET and Web services, Microsoft does not focus on the COM technology any more. In addition, the vendors need a plat-independent specification to support OPC running on a non-Windows-based system. OPC Foundation has offered three different SDKs as C/C++、C# and Java. You can develop OPC UA on Windows, Linux and embedded devices.

4) Enhanced security mechanisms

Traditional OPC does not have its own security design, and totally depends on security of COM/DCOM. However, OPC UA defines a set of comprehensive security mechanism. Between the proprietary security channel of the client and server, it has two-way handshake to authenticate both

978-1-4244-5848-6/10/$26.00 ©2010 IEEE

certificates. It has primarily used the safety technology such as Public Key Infrastructure and X509v3 certificate[3], etc.

II. The Architecture of OPC UA Server

OPC UA provides a consistent and complete address space, service model and security model. It allows a single OPC UA server units the data, alarms and historical information into its address space, and provides interfaces for them by using a common set of services.

OPC UA server can be designed according to the framework shown in Figure 1, and it is mainly composed of seven parts shown in the Figure.

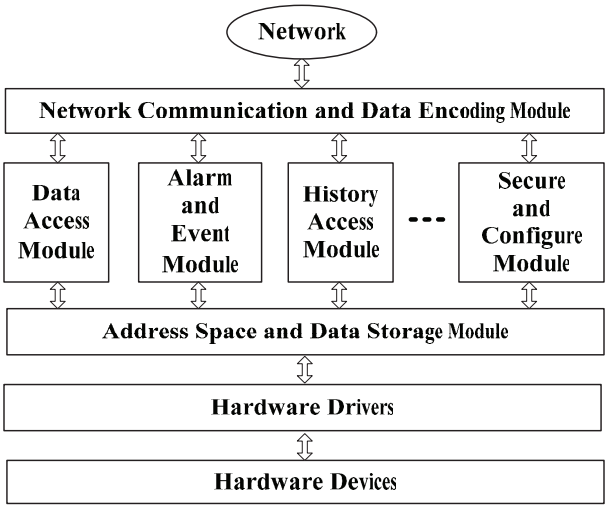

Figure 1. The Architecture of OPC UA Server

Network communications and data encoding module can also be called communication stack, which mainly completes three tasks: data encoding, data encryption and data transmission. We recommend that the developers use UA Stack which is provided by the OPC Foundation for this module. This not only allows the developers to focus on the functional design, but also ensure the interoperability of the program in different development environments.

Security and configuration module is responsible for configuring parameters when the sever starts (such as life cycle of the secure channel and Sampling interval, etc). It creates a special security communication channel between the server and the client, and authenticates the certificate of the client before exchanging data.

Data Access module provides services such as data reading, data writing and data subscribing [4].

Alarm & Event module browses the state of the equipments, and produces events and alarms [5].

History Access module provides functionality such as saving and browsing historical data and historical events [6].

Address space and data storage module forms the address space corresponding to the real objects. It creates the Views according to various tasks, also collects and stores the real-time data and historical data of the field device.

Hardware drivers are channels that OPC server connects with hardware. Different hardware need to develop different driver. It encapsulates communication details of under layer and provides an interface to server for invoking the functions.

III. Description and Realization of OPC UA Sever

Address space is the most important concepts in OPC UA, all the other functional blocks should be realized upon the address space. In traditional OPC specifications, each specification individually defines their own address space and services; however OPC UA integrates this model into a unified address space[7].

A. The composition of the address space

OPC UA server provides the client objects and related information related with the server's address space. The basic unit of address space is the Node.

1) Node and Reference

Nodes are mainly consisting of two parts, the Attributes and References. The Attribute is used to describe the node, and Reference defines the relationship with other nodes. The node's attributes are determined by the node type. However, there are some attributes are shared by all nodes. Node's public properties are shown in Table 1.

TABLE I. The Public Attributes of Node

Attribute	Description
NodeId	Unique identify and locate a node of the address space
NodeClass	Enumeration, defines the category of nodes
BrowseName	Identifies a Node when browsing the address space
DisplayName	Contains the name of the node that should be used to display in a user interface.
Description	A localized textual description of the node
WriteMask	Specifies which attributes are writable
UserWriteMask	Specifies which attributes can be modified by the user

Nodes only have public properties called the Base node which is the Meta data model of the address space. The base node is an abstract node type and can not be instantiated. All nodes of address space are derived from the base node, including the ReferenceType, Object, ObjectType, Variables, VariableTypes, DataTypes, Methods and View.

A Reference describes the relation between two Nodes. It should be noted that the reference is not derived from the base node, just a part of the node that does not contain any attributes. The semantics of Reference can only be expressed by reference types.

Reference cannot be accessed directly, only browsed through the source node to the target node indirectly. You can organize the address space of OPC UA to a mesh structure by references of nodes.

However, in order to improve the interoperability between server and client, it is generally recommended that the address space of server should be organized hierarchy, which is similar to the Windows folder structure.

2) ReferenceType

The semantic of References are represented by ReferenceType. There are two ReferenceTypes called hierarchical References and non-hierarchical References. Hierarchical References mainly used to organize address space, and reflect the field equipments in hierarchy architecture; non-hierarchical References are primarily used

to indicate the attributes of the equipments, such as the value of temperature sensor, etc.

The OPC UA specification defines a set of standard ReferenceTypes. They basically include all the common semantics of references. The following describes three most important ReferenceTypes: HasComponent, Organizes and HasSubtype.

HasComponent is used to define the components contained by Objects, and restrict these components to the object. That is, when a SourceNode is added to the Address Space, components of it will also be added to the Address Space; on the contrary, when deleting a SourceNode, the corresponding components will be subsequently removed.

Organizes is used to define the hierarchy of directory tree. It provides a convenient mechanism to organize and browse to the address space.

HasSubtype is used to define the type inheritance. The SourceNode of this Reference is the parent class, and the TargetNod is the subclass. It is used by the nodes and reference types to derive new types.

3) Object

The most important function unit in the Address Space is the Object. The Object nodes use HasComponent reference types, organize the Variable, Method together and produce Event.

Variable represents the object's data attributes. It has value, engineering unit, time stamp, quality stamp and other properties. Clients can read and write the value, also subscribe to changes of the value.

Method represents a method that is called by a client and returns a result that is implemented by the server.

Event represents the uncommon conditions occurred in the system such as alarm. Events used to represent the significant system changes, such as the user's actions, the setting value changes, etc.

4) View

In order to simplify the access of client to the address space, OPC UA defines the concept of View. The server can filter the corresponding reference types to consist a region as View by the filter's attributes of the View Node.

Each View defines a subset of the Nodes in the Address Space. Servers can organize their Address Space and provide views tailored to specific tasks or cases of use. The View Node acts as the root for all Nodes in the View, and it is used as entry point into the View. The default View is the entire Address Space.

5) Other Node Classes

ObjectType provide definitions for Objects. They can be simple or complex. Complex types expose the structure of the Object, whereas simple types define only the semantic for the Object. OPC Foundation defines a list of object information models that are used for different purposes.

VariableType defines the types for Variables. For example, the counter variable, historical data variables, etc.

DataType defines the types for Variables. It provides base types like Int32, Boolean, Double, and also specific types like enumeration data type and structured data type.

B. The Architecture and Implementation of the Address Space

The functions of Address space can be collaboratively implemented by Node Manager Class, Data Manager Class, Browse Manager Class, Subscription Manager Class and View Manager Class. These Classes handle the corresponding work and provide interfaces to outside.

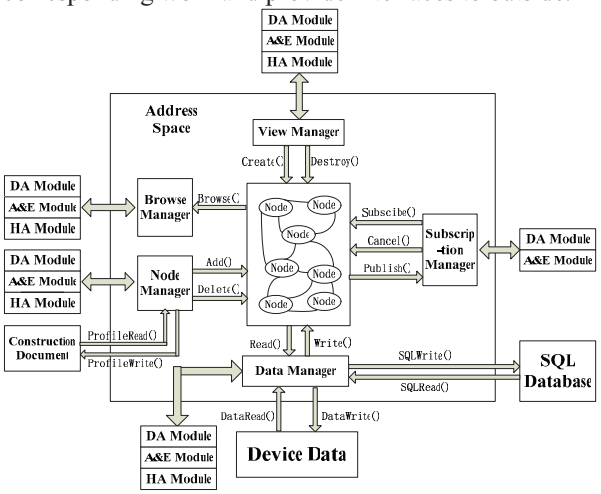

Figure 2. The Architecture of Address Space.

1) Node Manager Class

Node Manager Class applies memory space when server starts, and read the structure of the space saved in Construction document. When the server is running, it renews the organizational structure in real time, such as adding or deleting nodes. At last, it saved the changed organizational structure into the Construction document and free memory space when sever is close. In this paper, the organization of address space has used Multiple-tree structure. Multiple-tree hierarchical traversal algorithm may greatly speed up the access.

The key code of Node manager is as follows:

```
// apply memory space
nodesToRegister.create(count);
// Read construction document into memory
for ( i=0; i<count; i++ )
      g_NodeIds[i].copyTo(&nodesToRegister[i]);
// register the nodes
registerNodes(    nodesToRegister,
                  registeredNodes);
……
// add a reference and a new node to the source node
addNodeAndReference(  sourceNode,
                      pNewNode,
                      referenceTypeId);
```

2) Data Manager Class

Data Manager Class is responsible for read the data of device into address space, and provide operation interfaces for read and write variables and data of address space. For historical data and historical events, it is need to clean them from memory and store to SQL database [8].

The key code of Data manager is as follows:

```
// create an object to read the node and the value
UaReadValueIds nodesToRead;
UaDataValues          values;
// Get the number of nodes that want to read
count = g_VariableNodeIds.length();
nodesToRead.create(count);
// Read service
read( Max age,
      TimestampsToReturn_Both,
      nodesToRead,
      values,
      diagnosticInfos);          // Quality stamp
```

3) Browse Manager Class

Browse Manager Class provide operation for upper application module to browse address space.

The key code of Browse manager is as follows:

```
// Start browse from the root node
UaNodeId      startingNode(OpcUaId_RootFolder);
// Get the number of nodes
nodesToBrowse.create(count);
for ( i=0; i<count; i++ )
      exploreAddressSpace(startingNode, 1);
```

4) Subscription Manager Class

Subscription Manager Class provides operation for subscribe data and events, and provide an interface for external publishing.

The key code of Subscription manager is as follows:

```
// Set subscription publish interval
subscriptionSettings.publishingInterval = 500;
……
// Set filter parameters of Event
OpcUa_EventFilter* pEventFilter
          = (OpcUa_EventFilter*)OpcUa_Null;
……
// Create Monitored Item
createMonitoredItems(TimestampsToReturn_Both,
                     monitoredItemCreateRequests,
                     monitoredItemCreateResults);
```

5) View Manager Class

View Manager Class is responsible for creates View which upper application module requested, and filter the node of address space by filter parameter.

The key code of View manager is as follows:

```
// create the entrance node of a view
pView = UaViewSimple( name,          //View name
                      nodeId,        //Node ID
                      containsNoLoops,
                      eventNotifier);
// Set filter parameters of View
OpcUa_ViewFilter* pViewFilter
          = (OpcUa_ViewFilter*)OpcUa_Null;
```

C. Introduction to Development Tools

The development platform adopted by this paper is Visual Studio 2008 based on .net framework 3.5.

The full name of.NET is Microsoft.NET XML Web Services platform. XML Web Service allows application to communicate and share data via Internet without limiting the type of operating system, device or programming language. These provide foundation to implement platform-independent of OPC UA.

Managed code application can obtain common language runtime services. For example, automatic garbage collection, runtime type checking and security support, etc. These services help to provide platform-independent, language-independent and unified managed code applications. Use managed code and its compilation may avoid many typical programming errors which may lead to black holes of security and unstable programming errors. Programmers can spend more energy to concern the logical design of application and reduce the amount of code need to write, this means less development time and more robust program.

IV. CONCLUSION

OPC Unified architecture was a new standard for the automation of industrial data communication process. It is completely different from classic OPC standard which based on COM. OPC UA can support a small smart device, also run in a complex network system. It is expected that in the next few years, OPC UA will replace the OPC application such us DA、HDA and A&E servers based on Microsoft COM/DCOM. Due to the platform-independence and use of advanced web services technology, OPC UA can be deployed in field device, DCS, MES and ERP systems. The application of OPC UA technology will promote the data communication of industrial automation systems more standardizing and modularizing.

REFERENCES

[1] Wolfgang Mahnke , Stefan-Helmut Leitner , Matthias Damm; OPC Unified Architecture , ABB Germany , Springer Press; 2009, ISBN 978-3-540-68898-3

[2] Lee Neitzel ; OPC Unified Architecture Internals; The Instrumentation Systems and Automation Society ; 2004-10

[3] Annerose Braune, Stefan Hennig, Sebastian Hegler; Evaluation of OPC UA Secure Communication in Web Browser Applications; The IEEE International Conference on Industrial Informatics; Korea July 13-16, 2008

[4] OPC Foundation ; OPC Unified Architecture Specification Part 8: Data Access RC 1.31; 2006-07

[5] OPC Foundation ; OPC Unified Architecture Specification Part 9: Alarms and Conditions RC 1.00.18; 2009-06

[6] OPC Foundation ; OPC Unified Architecture Specification Part 11: Historical Access 1.00; 2007-01

[7] OPC Foundation ; OPC Unified Architecture Specification Part 3: Address Space Model RC 1.01.13; 2008-06.

[8] LU Huiming; ZHOU Zhao; LIAO Changbin; Historical data processing in real-time database system, Electric Power Automation Equipment, 2009-03