

APPLICATIONS OF OPC AT BEPCII

Huijuan Xu, Jijiu Zhao, IHEP, Chinese Academy of Sciences,
Beijing, 100049, P.R.China

Abstract

The Run-time data and machine parameters of the BEPCII are distributed over different operating systems platform and stored with different application software. Some are stored in various SCADA logging files and some are stored in the EPICS archiver files. Now the EPICS data is stored in Oracle. No general method is provided to access these data. The OPC technology can solve this problem. We have developed EPICS/OPC Server and Oracle/OPC Server. With the help of these two servers and SCADA OPC Servers, it's easily to get the data mentioned above on Windows system. This paper describes the development of the two OPC Servers and OPC applications at BEPCII.

INTRODUCTION

The BEPCII project is for upgrading the Beijing Electron Positron Collider (BEPC) to obtain a higher luminosity. The BEPCII control system is developed with EPICS (Experimental Physics and Industrial Control System). There are two kinds of database in the BEPCII control system. One is the distributed real-time database running in EPICS [1] IOC to store raw data, the other is a relational database Oracle, which stores static and dynamic data including system configuration data, machine parameters beam parameters, historical and alarm data, etc. Though both of the two database systems provide user interface on the Web page and publish the running information, there was no common method to access them.

OPC OVERVIEW

OPC [2] stands for OLE [3] for process control. It is a series of standards specifications which originally based on Microsoft's OLE (Object linking and embedding) COM [4] (component object model) and DCOM [5](distributed component object model) technologies. These specifications define a standard set of objects, interfaces and methods for use in process control and manufacturing automation applications to facilitate interoperability.

Before the OPC technology is applied in the automation control field, the different applications required a unique driver for different device. Thus the programmers need to develop a device-specific chunk of code to talk to the device or network. Figure 1 shows the communication between applications and devices. It's a great waste of man power and money. The OPC technology solves this problem by creating a 'software bus' [6] as shown in figure 2. Applications only need to know how to get data from OPC data source (OPC Client) and devices only

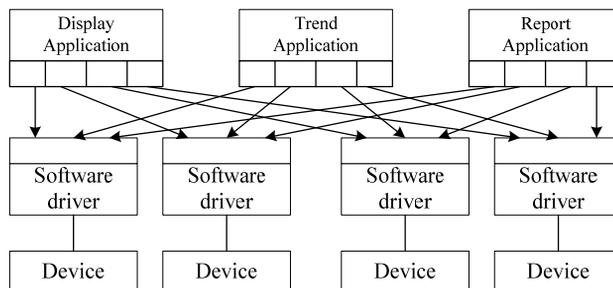


Figure 1: Traditional Data Access Method

need to know how to provide data in a single format (OPC Server). As a result, the applications are simpler, smaller and easier to use. Furthermore, OPC also supports

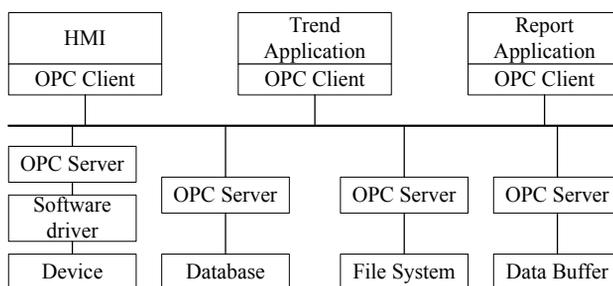


Figure 2: OPC Data Access Method

various data source such as database, file system, data buffer and is flexible enough to be used between these data source and high level applications as HMI, Trends, Reports, etc.

There are several OPC specifications released currently. Here we use OPC Data Access (DA) specification version 3.0. The OPC DA specification defines three objects: server, group and item. The server object provides management of itself and group objects, which may contain one or more group objects. The group object provides management of itself and item objects, which may contain one or more item objects. The item object contains several properties, such as value, quality and timestamp.

THE OPC SERVER DESIGN

The OPC Server exposes data from different data providers via the standard OPC interfaces. Thus data from different data sources such as EPICS, Oracle and so on can be integrated into the same development environment. Different servers can be developed to meet different requirements. The OPC Group provides a convenient way for an application to organize the data it needs. In addition, the OPC Group can have different update rates which are common for the contained items. Thereby, data which are used frequently can be added in the group with proper

xuhuij@mail.ihep.ac.cn

update rate. The OPC Item provides a connection point to a value in the data source and is used to create a connection between the server and the real data [6].

EPICS/OPC Server

The EPICS IOC database stores all data coming from the I/O drivers. Generally, one VME IOC keeps data of a single subsystem. Thus each subsystem can be developed as an OPC Server. One EPICS process variable can be mapped to an OPC item. As mentioned above, every item contains three properties, value, quality and timestamp. The value is in the form of a VARIANT, which can be mapped to the VAL field to represent different type of data. The quality is based on the HIHI and LOLO field of the corresponding IOC record, which also can be set according to the requirement of users. The timestamp is the current time on the user’s computer.

The Softing [7] OPC Server development toolkit (OPC Toolbox C++ V4.0x) makes our development easier. An OPC server can be quickly established using this toolbox. The most important task focuses on the development of the interface between OPC and EPICS. Figure 3 shows

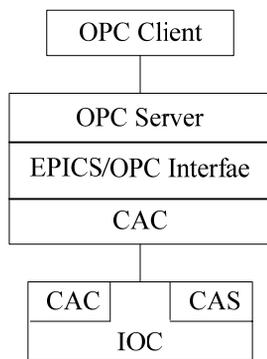


Figure 3: Structure of EPICS/OPC Server

the structure of EPICS/OPC Server. As we known, EPICS which use Channel Access to communicate over network has a lot of client functions. By calling these functions, the access to EPICS process variables is implemented. Considering the overhead of network communication, a channel access task is initialized when there is a data request and it is released when there is no request. During this period, the IOC database is accessed only when there is a data access request or there is a data update. Slowing down the group update rate can also ease the network load.

Another problem is that how to distinguish different data types. The EPICS record has various record types corresponding to different data types. By finding record type, e.g. ‘ai’, ‘ao’ in the record name, different data types can be identified.

Oracle/OPC Server

The historical data of the control system and various types of parameters of BEPCII are placed in the Oracle database. These Data are stored in table forms. The tables in the Oracle database of BEPCII provide the uniform

format. Each table stores the data information of an independent subsystem. Therefore, each subsystem can be developed as a single OPC Server. Each record in the table can be mapped to an OPC item. The PVNAME is the primary key of the table and can be set as item’s id. In the same way, LASTVALUE column corresponds to property value and LASTTIME column corresponds to property timestamp. The setting of quality is based on the actual need.

Network load is still a big problem. Here the client cursor is used to ease the load of Oracle server. Network delaying is another outstanding question. To solve the problem, the connection to database is created when an OPC Server program is run and disconnect when the program is closed. And only selecting operation is performed during this period. To reduce the selection time, the index and precise selecting sentences are used in the server program. Now it’s taken only seconds to get dozens data but still two minutes to get more than one hundred data. Solutions are still looked for to improve the response time.

DCOM CONFIGURATION

The two kinds of OPC Servers mentioned above are implemented as remote servers. Each operates as an autonomous program. The counterpart of the OPC Server is the OPC Client. When an OPC client programme is run, the relevant OPC Server is enabled. DCOM is used for data communication between OPC Servers and OPC Clients. It is a protocol that enables software component to communicate directly over a network in a reliable, secure and efficient manner.

DCOM handles low-level details of network protocols and all that we do is that how to configure it. The DCOMCNFG Program deals with it. On the server side, the security access and launch permissions must be changed to authorize access from connected clients. The ‘Distributed com on this computer’ option must be enabled and the OPCENUM properties must be set correctly. Note that on the identify page, ‘system account’ should be checked. It makes the configuration on client side much easier. Also settings of OPC Servers need to be changed. Table 1 and 2 give the details of DCOM configurations.

Table 1: Configurations of ‘My Computer Properties’

Parameter	Description
Default Properties Authentication Level	Connect
Default Properties Impersonation Level	Identify
Default Security Launch Permissions	Everyone, System, Administrator, Anonymous logon and Interactive User
Default Security Access Permissions	Everyone, System, Administrator, Anonymous logon and Interactive User

Table 2: Configurations of 'OPCENUM Properties'

Parameter	Description
Default Properties Authentication Level	None
Default Security Launch Permissions	Everyone, System, Administrator, Anonymous logon and Interactive User
Default Security Access Permissions	Everyone, System, Administrator, Anonymous logon and Interactive User

On the client side, only a few configurations should be done. Users only need to install the OPC Client program and use it directly.

SUMMARY

OPC is widely used in control system today. Some subsystems of BEPCII adopt SCADA software that also supports the OPC technology. With OPC Servers, a universal method is supplied to access data from different

data source. An OPC Client program is also developed to facilitate using. Simple graph function is provided in the OPC Client. Thus users could not only get the current value of data but also could get the dynamic curve.

REFERENCES

- [1] EPICS homepage: <http://www.aps.anl.gov/epics>
- [2] The OPC Data Access Custom Specification 3.0, <http://www.opcfoundation.org>
- [3] Kraig Brocksmidt, Inside OLE 2nd Edition, Microsoft Press, 1995
- [4] <http://www.microsoft.com/com/default.mspix>
- [5] <http://msdn.microsoft.com/library/en-us/dnanchor/html/dcom.asp>
- [6] Al Chisholm, A Technical Overview of the OPC Data Access Interfaces, ISA EXPO, 98
- [7] Softing Company: <http://www.softing.com>