INTEGRATING PLCS WITH AN OPC INTERFACE INTO AN EPICS BASED CONTROL SYSTEM

R. Fleischhauer, B. Kuner, G. Meyer, J. Rahn, C. Winkler, BESSY, 12489 Berlin, Germany

Abstract

At BESSY PLCs are used to control many beamline devices like valves, shutters or interlocks. A widely used way to communicate with PLCs is the Open Process Control (OPC) protocol. Typically an OPC server running on a PC based hardware provides the OPC interface to TCP/IP Ethernet, which OPC clients from other (PC based) hosts connect to. Since the BESSY control system is based on EPICS, it is desirable to use the various EPICS tools to access the PLCs. Starting from release 3.14 it is possible to run the EPICS database on PC hardware with Windows as operating system. A newly written device support links the records of the EPICS database to the OPC client library, enabling the PC-based IOC to act as an enhanced gateway between EPICS and the OPC world. This approach provides a seamless integration of PLCs in an EPICS based control system.

INTRODUCTION

In the experimental area of the of the third generation light source BESSY II programmable logic controllers (PLC) are used to operate the typical beamline hardware (i.e shutters, valves, interlocks, temperature sensors). In parallel the complex devices like monochromators and split mirror units are controlled by software based on the EPICS [1] software toolbox. Due to the fact that EPICS provides various tools to collect and display process data a reasonable mechanism to connect both worlds is desirable.

OPC OVERVIEW

OPC [2] (OLE [3] for Process Control) is a series of standard specifications that resulted from a collaboration of automation suppliers and Microsoft. OPC defines a standard set of objects, interfaces and methods for use in process control to provide interoperability. It is widely used in industrial environment and OPC servers are available for most PLCs on the market.

The OPC protocol is used to access process variables (PVs) of PLCs by application programs in a standardized way and saves the user of PLCs from writing special PLC device drivers. This work will typically be done by the manufacturer of the PLC who provides an OPC Server for the certain type of PLC, but there are also third party vendors.

The counterpart of the OPC Server is the OPC Client. It is typically an application program on a different computer communicating with the OPC Server to visualize, archive or manipulate data.

The communication between OPC Server and Client is based on DCOM (Distributed Component Object Model) [4].

DCOM is a set of Microsoft concepts and program interfaces in which client program objects can request services from server program objects on other computers in a network. DCOM is based on the Component Object Model (COM) [5], which provides a set of interfaces allowing clients and servers to communicate within the same PC.

DCOM is not only available for the Microsoft Windows operating system, but also for real time operating systems like VxWorks from Wind River Systems.

Part of the OPC specification is the “Data Access Specification” that defines a so-called Item for every PV that combines several properties, e.g. a value of a common data type, a quality and a timestamp. Each Item belongs to a Group. The Group defines parameters that are-common for the contained Items, e.g. the update rate.

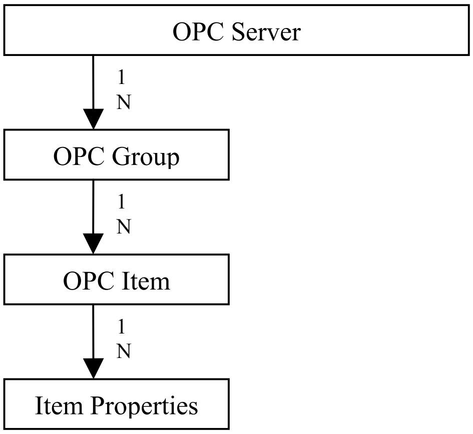

Figure 1: OPC Server Object Model

The OPC Items can be mapped to the EPICS process variables, i.e. the EPICS Records. A Record in the EPICS database is a process variable with a collection of predefined properties and a defined behavior, e.g. an ai-Record (analog in) is able to read a hardware channel and convert the value to engineering units. It has timestamp and quality information.

FROM OPC TO EPICS

The OPC Server

Figure 2 shows the setup of the BESSY installation. The OPC Server is located on PC hardware and communicates with the PLCs using Ethernet hardware and a proprietary communication protocol defined and provided by the manufacturer.

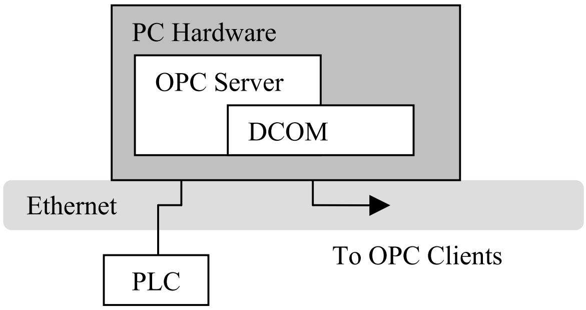

Figure 2: Access to a PLC with an OPC Server

Connecting the EPICS Database to OPC

Starting from EPICS Version 3.14 it is possible to run the EPICS database on PC hardware with Microsoft Windows as operating system.

The core EPICS application is called iocShell and provides mechanisms to load and run EPICS databases as well as some other command line functions. Within EPICS the standardized way to communicate with hardware is to implement a Device Support (comparable to a device driver). A special Device Support has been written to implement the communication path between the EPICS Records and commercially available OPC Client from the software vendor Softing (OPC Toolkit V3.01). Using this Device Support an EPICS system could communicate with an OPC Server as shown in Fig. 3. The OPC Server could be located on the same host computer as well as on a remote computer.

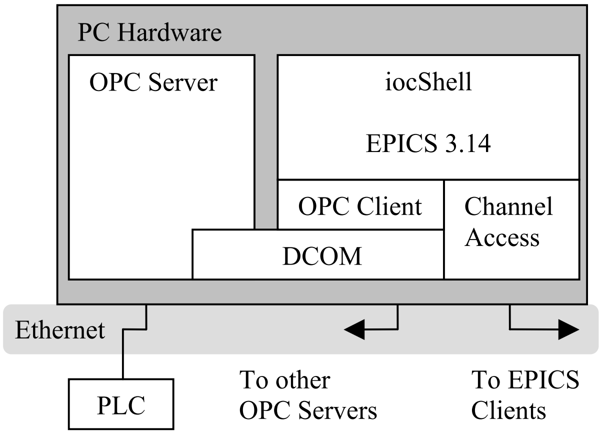

Figure 3: Data Flow from the PLC through the OPC Server to EPICS

IMPLEMENTED DEVICE SUPPORT

EPICS Device Support is a software layer which provides access from an EPICS Record to a certain hardware. In this context the OPC Device Support provides direct access for EPICS Records to items located on an OPC Server using the communication path shown in Figure 3.

Data Conversion

Data conversion between OPC and EPICS is implemented for all integer and float data types. Array support is not yet available.

Support for the following EPICS Record types has been provided:

- ai, ao
- bi, bo
- mbbi, mbbo
- mbbiDirect, mbboDirect
- longin, longout
- stringin, stringout
- (waveform Record will be supported soon)

EPICS "Out Records"

Reading EPICS Records are called "In Records" and writing records are "Out Records", bidirectional records are not supported due to the EPICS database design. On the other hand PLC I/O channels may be written both by the PLC itself and by a certain EPICS Record. Therefore the OPC Device Support for "Out Records" is designed to allow writing data to the PLC and updating the Record if the data has changed in the PLC.

Timestamps

Two choices as timestamp source of a PV are available. Either the timestamp generated by the OPC Server is used or the EPICS timestamp generated on the IOC when the Record is processed. The Record's TSE field determines the source used.

OPC Quality and EPICS Severities

OPC quality of an Item is mapped to the status and severity fields of the EPICS Record. The OPC Quality *Bad* will set the Record to *INVALID*.

Available Console Commands

For configuration purposes the following console commands within the iocShell are implemented:

- Set name of the OPC Server. Only one Server can be connected to a single EPICS computer.
- Set OPC Group name and the update rate for the Group. Only one Group is supported.
- Map OPC Item names from long names as used by the OPC Server to EPICS Record names limited to 29 characters.
- Browse the OPC Server: For debugging purposes it is possible to browse all OPC Items of the Server.
- Set level for debugging output.

Error Logging

For error logging the EPICS IOC log service is used. Error messages from all iocShells are sent via network to a log message server (iocLogServer) and additionally printed to the local console of the EPICS computer.

BESSY INSTALLATION

The following setup is the first installation using the OPC Device Support described above:

- 9 PLCs with approximately 6,000 PVs
- 2 OPC Servers
- Upgrade to 30 PLCs and 6 OPC Servers with 30,000 PVs on the way

FUTURE PLANS

Plans to improve the functionality of the Device Support are:

- Support of data value arrays (waveform Record)
- Support for other OPC Client libraries, e.g. to provide an implementation for the Linux OS
- Support for connections to multiple Servers
- Support for several Groups per Server

REFERENCES

[1] Experimental Physics and Industrial Control System, http://www.aps.anl.gov/epics

[2] The OPC Data Access Custom Specification 3.0, http://www.opcfoundation.org

[3] OLE Automation Programming Reference, Microsoft Press, Redmond, WA, 1996.

[4] DCOM Distributed Component Object Model, http://www.microsoft.com/com/tech/DCOM.asp

[5] COM Component Object Model, http://www.microsoft.com/com/tech/COM.asp