USING OPC TECHNOLOGY FOR CONNECTION WITH SUBSTATION SCADA SYSTEM.

J.Aleksic, PD Elektrodistribucija Beograd, Serbia and Montenegro
S.Milovanovic, PD Elektrodistibucija, Beograd, Serbia and Montenegro

OPC OVERVIEW

OPC is abbreviation for **O**LE for **P**rocess **C**ontrol and stands as **O**penness, **P**roductivity, and **C**onnectivity. It is a specification that has been developed over the last 8 years by a team of 150 companies and represents a shining example of how industry specification organizations can be effective.

OPC appeared on the scene when application software developers and users saw the need to begin to standardize how their applications worked together to reduce integration and total life cycle costs for software used in the manufacturing automation/process control industry.

The specification is based upon the current Microsoft (Distributed) Component Object Model (COM/DCOM) and addresses the specific needs of the automation industry such as data access (i.e. drivers), alarming, historical data access, trending, batch… It doesn't suffer from the performance and scalability issues that DDE suffers today. OPC specification defined standard set of objects, interfaces and methods to facilitate interoperability. It takes the concepts of properties, methods, and events from ActiveX controls, and goes a step further to address automation specific issues.

Before OPC, each software application vendor provided their own custom ways to connect third party applications into theirs. So to connect vendor application "A" to vendor application "B", it was mandatory to have an expert C++ programmer who could quickly come up to speed and become an expert on the custom programming interfaces for both application "A" and application "B". Furthermore, users usually had to purchase a toolkit from each of the two application vendors at a cost of several thousand dollars each, long before they even cut the first check to the programmer who was supposed to integrate the two.

With the OPC initiative, developers in the manufacturing automation/process control industry began to adopt the specification and build "doors" into their applications that met the OPC specifications. Accordingly, if Application A and Application B were both written to the same OPC specification, they could connect with a mere fraction of the integration time and cost and without custom code.
OPC software interfaces are clearly defined by a public standards body, the OPC Foundation, with over 270 companies using the methodology. It is used in almost every part of world as USA, China, Argentina, Sweden, Brazil… and in all fields where there is a need for turning real-time data into actionable information.

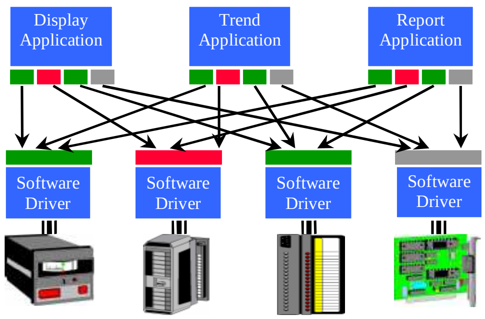

Figure 1 - Before OPC

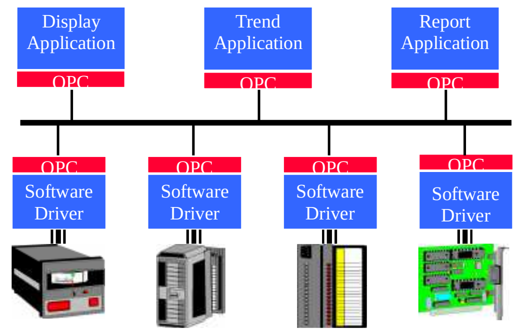

Figure 2 - With the OPC initiative

Difference between an OPC client and an OPC server

An OPC Client is a software application that needs the data from the process control systems and is able to speak the "language" defined by the OPC specifications in order to be able to get the data from an OPC Server. The OPC server is the "driver" software that talks to *specific* hardware and reads/writes data and "serves it up" to the OPC client application.

The long-term benefits for the user of OPC come from the ability to pick best-of-breed application modules such as trending, alarming, graphics, etc. from various sources and bring them together with common interfaces. For example, graphics environment (GUI) could function as an OPC client, getting its data from an OPC Data Server connected to PLCs, SCADA or RTUs. The trending package you choose could function as an OPC client to the graphics package to get its data, which also happens to be an OPC server! Or it could go direct to the OPC data server.

Although this concept may sound like the old DDE server model that has been around since the late eighties, but it is not. When DDE was invented, COM and the whole 32-bit world did not exist. The OPC specification is built upon COM and uses Windows technologies to provide superior performance and robustness that will never be found when using DDE.

OPC Server-types of data. Server provides all sorts of data such as status, control and measurements.

- Real time sensor data - temp, pressure, flow
- Control parameters - open, close, run, stop
- Status information:
  - Status of the hardware connection
  - Status of the local software and subsystem

OPC Server is not system configuration; it provides a window into existing process data.

OPC Client could be easily added as a module for application interested in:

- Subset of the Data Items (Tags) available within the underlying Control sub-system.
- Many different subsets of Data Items at different times and may have variable requirements for response and resolution.
- Independency of the data structures (or objects) used by the sub-systems. (I.e. symbolic data access wanted).

Architecture of OPC connections

Architecture consists of Application with integrated OPC Client in communication with OPC Server. OPC Server is integrated part of Data Source (SCADA, RTU, PLC…).

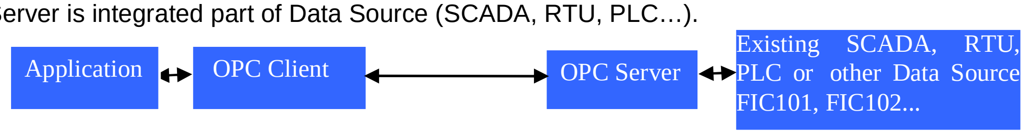

Figure 3 - Architecture

Typical Server Design

Design of Server consists of few modules such as:

- OPC/COM Interfaces
- OPC Group & Item Management
- Item Data Optimization and Monitoring
- Device Specific Protocol Logic
- Hardware Connection Management

The OPC Interfaces

Any software application that wants to provide data to other applications can be an OPC server and use the standard OPC interfaces to present the data to other software programs, regardless of the root source of the data. OPC interface products are built once and reused many times; hence, they undergo continuous quality control and improvement. Benefits from standardized interface are:

- Allow applications to easily access subsystem data
- Support polled or exception based access
- Are optimized for use over a network
- Are designed to be vendor neutral
- Are exceptionally flexible and efficient

The Logical Object Model

OPC Server provides accessibility to data from OPC Client:

- Data is accessed by Name (string) which will generally be vendor or hardware specific.
- Data for lists of items can be read explicitly (polled) or subscriptions can be created.

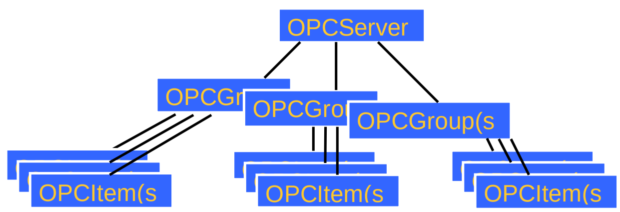

Figure 4 – Server logical object model

OPC SPECIFICATIONS:

OPC is a series of standards specifications. The original specification standardized the acquisition of process data, originally called simply the OPC Specification, is now only one of eight standards and is called *Data Access Specification*. Total number of specifications is not final. Work on new specifications is already lunched as well as work on upgraded versions of existing specifications.

1. OPC Data Access. Used to move real-time data from PLCs, DCSs, and other control devices to HMIs and other display clients.

2. OPC Alarms & Events. Provides alarm and event notifications on demand (in contrast to the continuous data flow of Data Access). These include process alarms, operator actions, informational messages, and tracking/auditing messages.

3. OPC Historical Data Access. Where OPC Data Access provides access to real-time, continually changing data, OPC Historical Data Access provides access to data already stored. From a simple serial data logging system to a complex SCADA system, historical archives can be retrieved in a uniform manner.

4. OPC XML-DA. Provides flexible, consistent rules and formats for exposing data using XML, leveraging the work done by Microsoft and others on SOAP and Web Services.

5. OPC Batch. This spec is covering the specialized needs of batch processes. It provides interfaces for the exchange of equipment capabilities (corresponding to the S88.01 Physical Model).

6. OPC Data eXchange. This specification moves from client/server to server-to-server with communication across Ethernet fieldbus networks. This provides multi-vendor interoperability! In addition, it adds remote configuration, diagnostic and monitoring/management services.

7. OPC Security. All the OPC servers provide information that is valuable to the enterprise and, if improperly updated, could have significant consequences to processes. OPC Security specifies how to control client access to these servers in order to protect this sensitive information and to guard against unauthorized modification of process parameters.

8. OPC Complex Data. A companion specification to Data Access and XML-DA that allows servers to expose and describe more complicated data types such as binary structures and XML documents.

OPC Compliance

The vision of interoperability in multi vendor systems has become a reality via the OPC standards. Certification is the process of ensuring that applications meet the standards. Certification can be accomplished in multiple manners, but most require extensive human intervention. The OPC foundation has produced automated tools to simplify this task; these tools are collectively known as the OPC Compliance tests.

OPC APLICATION IN POWER UTILITIES

Power utilities are in great demand for data acquisition. This demand is expanding every day, especially with replacing old mechanical protection devices in transformer substations with a new numerical devices and installing remote acquisition in non-observable transfer substations. Concept of control and data acquisition is totally changed. Instead of thousands of wires, there is one optical cable. Instead of old RTU-s, there is a station computer. Computer with mini SCADA system playing multiple roles: old function of RTU, local archive of data and a new HMI for control and supervision. Safety for control and maintenance personnel is raised. They don’t have to operate breakers manually and expose them self to danger of breaker malfunction. Now they are doing it from station computer, in different room.

Reconstruction of protection and remote control is usually done by public tenders. One reconstruction is done by one vendor of equipment and reconstruction in another transformer station is done by another vendor. In this case, there are no standardized solutions for station computer. Every vendor has his own proprietary version of archive and HMI, provided for additional amount of money.

World is running toward standardization, and for power utilities. In worst case scenario in every transformer substation there will be a different HMI and data archive. Also there should be trained personnel to use it and also to do maintenance of station computer. Trained personnel for all kind of different equipment and GUI are hard to find. OPC provides elegant solution for this problem of unstandardisation. Almost every vendor of protection and control substation equipment provides station computer with OPC server.

OPC Client in reconstructed transformer substation 35/10kV “Vinca”

After reconstruction of protection and control equipment, there was a big problem in lack of telecommunication way to “power utility Belgrade” control center. Brand new equipment in substation and not a single data archived or transferred to control center. Modern equipment was not utilized, as it should be. First step in solving of the problem was archiving of data from substation on station computer. This is how OPC client was started.

OPC Client arose from the need for archiving of various signals in transformer substation 35/10kV “Vinča”. Since there is no transmission way for transferring data, only solution was to write OPC Client that would collect all available data, acquired by OPC Server, and then archiving them in database. In regular period, database file would be transferred via USB-flash drive for further analyzing. Expansion of basic OPC Client by implementing some control features is due in some future versions of OPC Client. Full mini SCADA system is considered in future.

Operation modes of OPC Client are:

- synchronous,
- asynchronous,
- basic.

In basic mode of operation, simple read operation is performed on OPC Server, and by that operation acquired data consists of three values: timestamp (time and data), quality and value of measured data itself. This mode of operation has certain drawbacks. Main drawback is in providing software interrupt in which data is read, and if there is high number of data to be read, significant amount of time is required, thus minimal sampling period is large. In transformer station were **549** measured values, thus minimal time of acquiring all data was around few seconds. Another drawback is possible data loss, if change of data occurred between sampling interval. For example, problem occurred in detecting entrance of maintenance workers in transformer station. Sampling period was set to 15 seconds, which was too long time to capture change on door switch.

Second version of OPC Client

Another solution was required, so first version of OPC Client has to be upgraded for asynchronous mode of operation. In asynchronous mode, data provided by OPC Server is contained in local buffer, and then transferred to OPC Client after predefined amount of time. Only last change of measured data is sent. For example, if measuring of current is changing rapidly over time, only last value is recorded, if it is different from last transferred value. There is same drawback in asynchronous mode as in basic read mode, in recording entrance in transformer station. This was somewhat corrected by lowering sample time to 5 seconds. As mentioned before, in basic read mode, it is necessary for OPC Client to read every value and then compare it with previous value for writing into database, while in asynchronous mode, only certain data is forwarded to OPC Client by raising an event, thus lowering necessary time for reading.

Requirements of data from transformer substation are satisfied if there is measurement of certain values every 15 minutes. Since sampling interval is set to 5 seconds, there is more than enough time for acquisition of all needed data.

Database structure

Database is written in Microsoft Access 97. Detailed description of tables can be seen in Fig. 5.

Table: Merenja	
Field name	**Data Type**
Id	AutoNumber (Long Integer)
TagId	Number (Long Integer)
Kljuc	Text (12)
Vrednost	Number (Double)
Vreme	Date/Time (General Date)
Kvalitet	Number (Integer)

Table: Vrednosti	
Field name	**Data Type**
Id	AutoNumber (Long Integer)
TagId	Number (Long Integer)
Vrednost	Number (Double)
Vreme	Date/Time (General Date)
Kvalitet	Number (Integer)

Table: Tagovi	
Field name	**Data Type**
Id	AutoNumber (Long Integer)
Tag	Text (255)

Figure 5. Database fields and data types

Internal relationship of database is shown on Fig. 6.

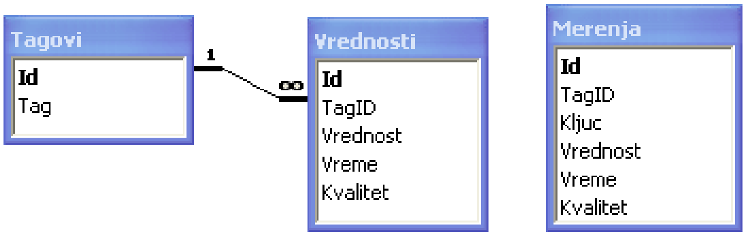

Figure 6. Internal relationship of database

User interface is shown on Fig. 7. It consists of one screen with few push buttons and a slider bar. Since application is designed to work unattended, there was no need for more detailed user interface. By pressing "Connect" button, OPC Client opens connection to OPC Server followed by creating group of items that are measured. Update rate can be set manually ranging from zero to 300 ms. When update rate is set to 0 ms, any change in measured items is sent immediately to database.

While OPC Client gathers data, they are shown on list box below. Example of received data is shown of Fig. 8. In this list box, columns are id number of measured item, measured value, quality value, date and time, respectively. List box is cleared every two hours in case of large amount of data that can slow down process of writing to database.

User interface has few textboxes that shows status of OPC client, and server. One of these textboxes shows status of OPC items, and the other is dedicated for OPC server status. There is also a colored circle that changes color when data is read. This can be very helpful in process of troubleshooting so the operator can easily see where the problem is.

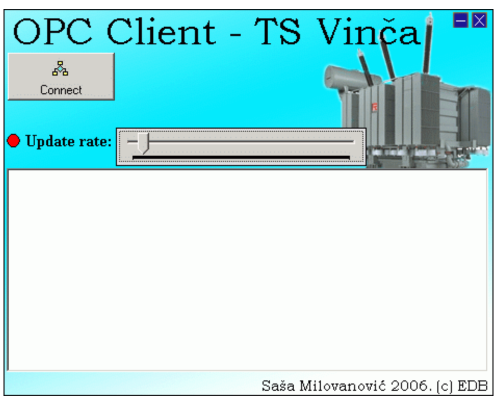

Figure. 7. User Interface

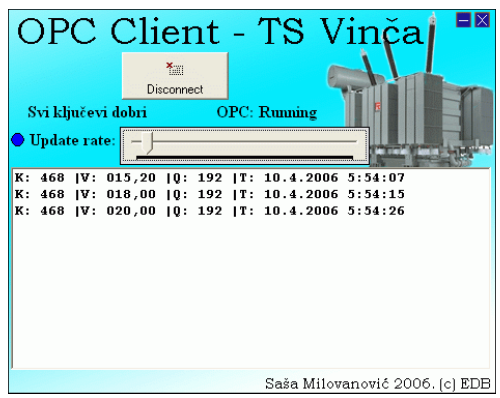

Figure 8. Measured data

Occasionally, operator is scheduled to collect all measured data by transferring database. This operation is shown on Fig. 5. When there is data collected into database, and OPC Client is disconnected from OPC Server, two push buttons are shown: "Copy data" and "Clear data". By pressing copy data, save file dialog box is shown for operator to save database on removable media such as flash drive. When data is copied, operator has choice of deleting and compacting previously recorded data to database, or to continue with filling database with new data. Usually, database is cleared after copy has been completed.

Protection of recorded data and detection of hard drive capacity will be added in one of the future versions of the OPC client.

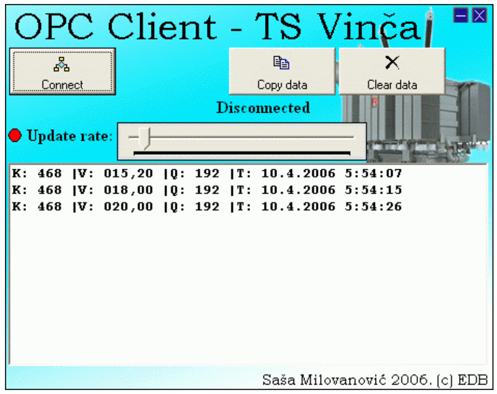

Fig. 5. “Copy and clear” database

FUTURE APLICATION OF OPC IN POWER UTILITIE OF BELGRADE

Adding the OPC specification to Microsoft's OLE technology in Windows allowed standardization. Now the industrial devices' manufacturers could write the OPC DA Servers and the customers and users could make software (like HMIs) could become OPC Clients.

As described in a paper first step in implementing OPC technology was connecting to existing OPC Server and data acquisition, second step was making the archiver. Time for completing these two-steps was about one week. One week from first steps to installing OPC client on station computer including reading OPC specification, download free drivers from OPC foundation site and writing a code in VB.

Next step is to make a HMI console with substation design and all appropriate measurements, status of switch equipment and command. Once made, this HMI and archiver could be easily adjusted for any another transformer substation. This will mean that all HMI-s in every transformer substation will be unified and standardized, easy for maintenance and not depending on vendor of equipment.