Not lost in translation

Facing the challenge of cost-effective plant engineering in subsystem integration

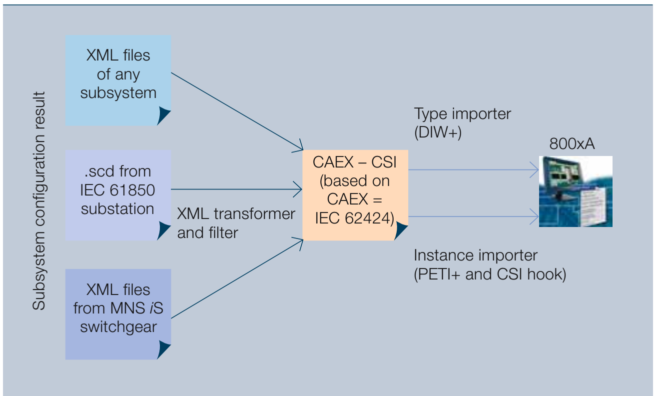
Peter Erning, Kurt Langer, Hartmut Rüdele, Dirk Schulz

Translating language is not always an easy task. Poor translations can lead to misunderstandings and even a total breakdown of communication. What is true of human language is even more true when it comes to automation systems. With different manufacturers using different standards, the integration of subsystems requires a translation of engineering data to enable communication with a distributed control system.

Distributed control systems, as are used, for example, in process or power plants, can achieve a considerable complexity and often integrate subsystems from different suppliers. For optimal integration, subsystem engineering data must be available to the overall control system. The challenge thus lies in translating this data to a format that the overlying control system understands. Of course manual translation of descriptions is possible but this is both labor intensive and error prone. ABB has developed a concept that permits the automatic translation of subsystem engineering data to ABB's own System 800xA automation system. Functional prototypes for the automatic integration of MNS *i*S low-voltage switchgear and IEC 61850 substation automation systems are available.

Early distributed control systems DCSs for process automation were homogeneous solutions provided by single suppliers. These suppliers were able to deliver almost all system components and to perform the engineering using their own specific engineering tools. Over the years, process-automation architectures have become more heterogeneous. Due to the rapid development in the microelectronics area, control and advanced functionalities have increasingly found their way to the field level of industrial plants.

1 Integrator system concept



The challenge is to provide a flexible software tool that enables an automated import of all engineering data from any previous engineering of a subsystem to a DCS with minimized manual effort.

With regard to automation systems, this means that today entire subsystems are being integrated into DCSs. Subsystems from different manufacturers are based on different architectures must therefore be able to communicate and act as one DCS. For example data from one subsystem must be accessible throughout the entire DCS. To achieve this, the engineering architecture of the subsystem must be visible to the main control system.

The main common characteristic of the subsystems discussed in this article is that they have DCS-independent but subsystem-specific engineering process. Examples are fieldbus networks of intelligent field devices, low-, medium- and high-voltage switchgear systems, and process-specific machinery and equipment.

The challenge is to provide a flexible software tool that enables an automated import of all engineering data from any previous engineering of a subsystem to a DCS with minimized manual effort. To achieve this, ABB initiated a research project to automate the import of engineering data from ABB's MNS *iS*[1] low-voltage switchgear systems and IEC 61850-based substations[2] into ABB's System 800xA automation platform.

Problem description and scope

Plant engineering projects typically involve different partners and suppliers. The complexity of the engineering process and the corresponding engineering effort depend on many factors: plant type, phase of the project-engineering lifecycle, system architectures, scope of deliveries, product specifics, tools and documentation (on paper, computer readable formats and others used by the members of separate teams in different companies), data storage and information exchange between substructures (sub-systems) of the plant.

Regarding the plant-automation part, a control system can be considered an assembly of a traditional DCS and other subsystems that also exhibit some DCS characteristics with regard to their architecture, instrumentation, control, communication and engineering tools. Examples are fieldbus systems; low-, medium-, high-voltage (LV, MV, HV) switchgear systems; and process-specific machinery and equipment.

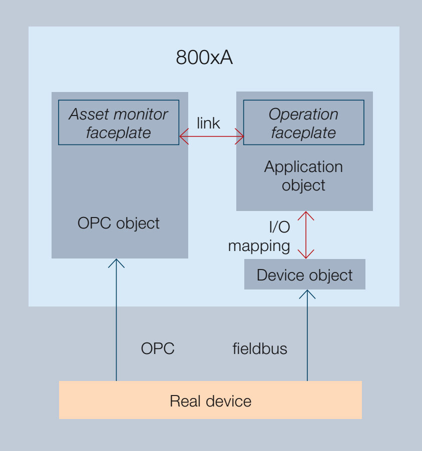
This article focuses on a method that is suitable for the automatic import of all engineering data of a subsystem that are needed on the DCS level, avoiding a subsequent manual entry of data that are already available in the subsystem tool(s).

Footnotes

[1] MNS *iS* is an ABB low-voltage switchgear platform. MNS stands for Modulare Niederspannungs-Schaltanlage (modular low-voltage switchgear).

[2] IEC 61850 is a modern standard for substation communication and integration.

2 Object model in System 800xA



Factbox Customer and ABB internal benefit

The described concept and solution has the following advantages:

- Manual engineering work for subsystem integration is reduced to the configuration of the importer tools and the enrichment of the object types. This remaining engineering work is almost independent of the size of a subsystem because object instances are created automatically. The engineering time is drastically reduced and the quality is improved.
- The concept enables a split of the import tool responsibilities between the subsystem supplier and the DCS supplier.
- The subsystem provider does not need to take into consideration the special interfaces and future changes on the System 800xA side. All required actions for integration can be modeled in an XML file (CAEX-CSI) created by a subsystem-specific transformer.
- The DCS supplier is responsible for common importer tools on the System 800xA side, also assuring a proper handling of a second upload in case of changes.
- The concept is prepared for existing and future protocols (PROFIBUS, Profinet IO, IEC 61850, OPC, etc).
- Engineering workflows with different delivery schedules can be handled in a better way (see section “Ongoing development”).

The “subsystem” in this context can be any system with the following characteristics:

1. The subsystem’s engineering data are available in computer readable format.
2. The subsystem provides at least one open communication interface.

The integration of engineering data can be achieved manually, but this leads to both a high risk of errors and unnecessary effort. The development of system-specific integration tools therefore represents an attractive alternative. However, these tools have to include expert knowledge of both the system that is to be integrated and the target DCS (which in the context of this article is ABB’s System 800xA extended automation system).

The syntax of the intermediate file is independent of the respective subsystem. This achieves a decoupling of the subsystems from the DCS.

Hence a generic concept for the import of external engineering data into System 800xA is a strongly desirable objective. The import mechanism should automatically create and configure objects in System 800xA based on the data contained in the subsystem’s specific engineering files. The mechanism must provide for objects that communicate via multiple communication paths (eg, fieldbuses, OPC[3]) but should provide a common interface to the user. System 800xA possesses an object-oriented architecture. For maximum benefit this should also be reflected in structures imported from subsystems. Subsystems often define project-specific object types, which must also be created in System 800xA and provided with a customized graphical interface. This tailoring should be done on the object-type level rather than on the instance level. Furthermore, the integration concept should provide a generic approach instead of separate integrator tools for every subsystem (the latter would lead to high maintenance costs). The import mechanism also has to take into account changes in the initial engineering data and should offer second upload capabilities (ie, intelligent change management). Finally, in order to minimize additional maintenance effort, existing tools from the System 800xA environment should be reused as far as possible.

Solution approach

In order to address these challenges, a generic importer concept was developed. The concept’s overall topology is shown in 1. The subsystem-specific data files are transformed into an intermediate file, which then serves as input for the actual import tools that create and configure the corresponding objects in the System 800xA DCS. The use of a subsystem-independent intermediate file achieves the desired decoupling of the subsystem’s knowledge domain from the DCS knowledge domain.

This concept relies on the interaction of several software components. First, a so-called transformer has to convert the subsystem-specific information. All information that exists in the specific XML files and that should be available in the DCS has to be extracted and transformed into the intermediate format. This information can, eg, comprise control and monitoring data, location information and documentation data. It should be pointed out that the engineering data do not necessarily conform to the object-oriented paradigm. Hence it can become necessary to analyze the properties of the individual objects, group similar objects, and define corresponding type and instance trees.

Footnotes

[3] OPC is a standard for real-time communication between control devices from different manufacturers. Today, OPC is officially a name and not an acronym, but originally stood for OLE for Process Control. OLE stands for Object Linking and Embedding, a document embedding and linking technology also developed by Microsoft. See also “OPC Unified Architecture” on page 56 of this edition of *ABB Review*.

[4] CAEX: Computer Aided Engineering eXchange, a neutral data exchange format for plant information defined by IEC 62424. CAEX-CSI stands for CAEX Complex device & Subsystem Integration (name of the research project)

The pivotal part of the whole concept is the intermediate data format (CAEX-CSI).[4] The syntax of the intermediate file is independent of the respective subsystem. This achieves a decoupling of the subsystems from the DCS. The file contains definitions for the utilized devices and subsystem types as well as information about the instance hierarchy. The type information can be stored explicitly in the intermediate file itself. In order to reduce transformation effort, it is also possible to specify links to the corresponding device description files (eg, GSD[5] files for PROFIBUS devices).

The import of the information contained in the intermediate files is achieved in two steps. First, the type information is analyzed and the corresponding types are created in System 800xA. After type import, manual alterations (eg, creation of graphical interfaces) can be effected on the newly created types. As a next step, the instance information is parsed, the device hierarchy is generated in the DCS and the device objects are instantiated and configured.

Typically, there is more than one instance per device 2: a device object for retrieving the raw process data and an application object to process the data and display them on an operator faceplate. In case of an additional communication path such as OPC there is even a third instance equipped with an asset monitor. The necessary interconnection of these objects is also automatically implemented by the instance importer.

Achievements

Functional prototypes have been developed for two types of subsystem, the new ABB LV switchgear system MNS *i*S and IEC 61850-based high- or medium-voltage systems. In both cases, XML-based system descriptions are available from the specific subsystem engineering tools. First transformer versions for MNS *i*S and IEC 61850 are available.

Functional prototypes have been developed for two types of subsystem, the new ABB LV switchgear system MNS *i*S and IEC 61850-based high- or medium-voltage systems.

The CAEX-CSI data importer consists of the following prototypical components:

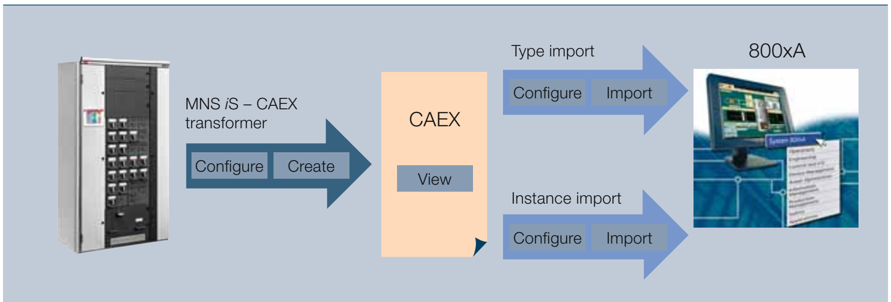
- DIW (device import wizard) that is used for the object type import is an ABB product that was extended with some functionality (see 1, prototype "DIW+").
- PETI (process engineering tool integration) is an existing ABB product that is capable of importing instances from CAEX, including a proper handling of a second upload in case of changes. The next PETI version will satisfy some additional CSI requirements (see 1, "PETI+").
- A special aspect system ("CSI hook"), which ensures that each object gets a working OPC connection and a correct link to an object specific faceplate.

The transformation and import workflow is supported by a lightweight frame application. The MNS *i*S frame application is depicted in 3. It connects the above components and guides the user through the individual integration steps.

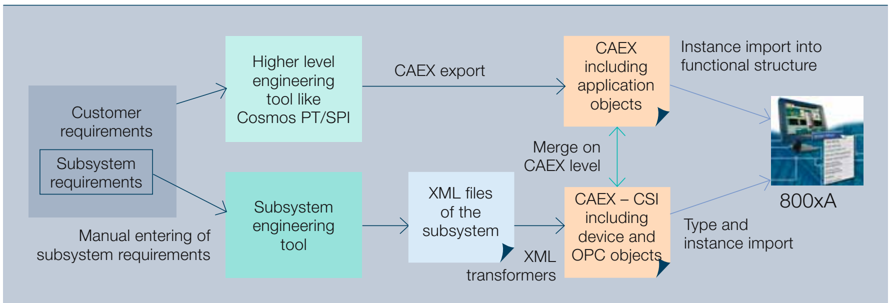
Ongoing development

Beside the activities to develop existing prototype importers into products, there will be an additional focus on overall engineering workflows. It is not uncommon in a plant-project schedule for decisions on subsystems to be taken later than the DCS engineering stage. Consequently, additional data coming from plant engineering tools (eg, Comos from Innotec or SmartPlant from Intergraph) also have to be used and will be merged with data coming from the subsystem 4.

3 Engineering data integration workflow



4 CAEX simplifies merging of multiple paths of engineering data



This article was initially published by © atp international 1/2008 (Oldenbourg Industrieverlag GmbH) with the title "Subsystem Integration – Facing the Challenge of Cost Effective Plant Engineering". It was adapted for use in *ABB Review*.

Peter Erning
Kurt Langer
Hartmut Rüdele
Dirk Schulz
ABB Corporate Research
Ladenburg, Germany
peter.erning@de.abb.com
kurt.langer@de.abb.com
hartmut.ruedele@de.abb.com
dirk.schulz@de.abb.com

Footnote

[5] GSD: Geräte-Stamm-Daten (device master data), a PROFIBUS description file