INTEROPERABILITY ISSUES IN MODELLING AND DEVELOPING SUBSTATION CONTROL SOFTWARE

Giovanna DONDOSSOLA, Francesco MAESTRI, Judit SZANTO
Cesi - Italy
dondossola@cesi.it

INTRODUCTION

The purpose of the paper is to present ideas on how to produce interoperable software for substation automation functions. The objective of the authors is to explore interoperability issues in ASME – Automation Software Modelling and development Environment.
Interoperability issues arise at different levels of the design and development process involved in the implementation of automation software modules: modelling and notation; information exchange format; compliance of the specified software interfaces with the emerging standard for communication in power systems.
This paper is mainly focused on the integration of substation automation functions designed by ASME with the substation control architectures specified in accordance with IEC 61850, with particular reference to the Substation Configuration Language.
The paper is structured into the following sections:

- *an overview of the ASME Environment,*
- *the description of a sample Substation Automation Application,*
- *a summary of the objectives of the standard IEC 61850*
- *the presentation and exemplification of the integration aspects related to ASME and IEC 61850,*
- *conclusive remarks.*

OVERVIEW OF THE ASME ENVIRONMENT

ASME supports high level design of automation logic followed by conversion of the specified functions into executable code to be distributed on heterogeneous devices from different suppliers.
In ASME automata networks are modelled using state and network diagrams derived from an extension of the UML (Unified Modelling Language) notation, the object-oriented modelling standard developed by the OMG in 1997 [1].
A UML-like Profile for ASME has been defined and a first prototype of the related user-interface has been implemented using Borland Together [2].
Definition of a data interface, based on the ASME object meta-model, consents diagrams to be converted into XML (eXtensible Markup Language) documentation [3].
The XML representation of an ASME specification provides an easily understandable interface towards external tools and a ready to use intermediate language for the tools of the ASME environment. The automatic conversion of the user input into an XML specification file is one of the main functions of the UML based Graphical User Interface and the present GUI prototype implements the preliminary steps towards the automatic generation of this file.

The specification technique

Applications are specified using finite state automata recursively grouped into nets and subnets, with subnets ordered on hierarchical priority levels inside nets.
Low and high level operational and decisional issues are expressed by automata, whilst nets are used both as designer's facility for the logical decomposition of functions and, through the attribution of subnets to hierarchical levels, as a means to influence the execution order of the specified operations. Functions are designed independently from their attribution to physical devices. One of the key features of ASME is the possibility to specify generic applications, i.e. not one specific application function but a class of applications to be later instantiated on specific switchyard installations.
The specification technique implies the knowledge of the underlying operational model: the application designer not only specifies what to do but, to some extent, how to do it or, at least in which order the specified functions should be executed.
Each automaton is first specified as an instantiable object to be later instantiated in terminal nets. Computation and communication among automata are modelled by channels, where a channel can be seen as a data object with attributes expressing both informative and connective properties.

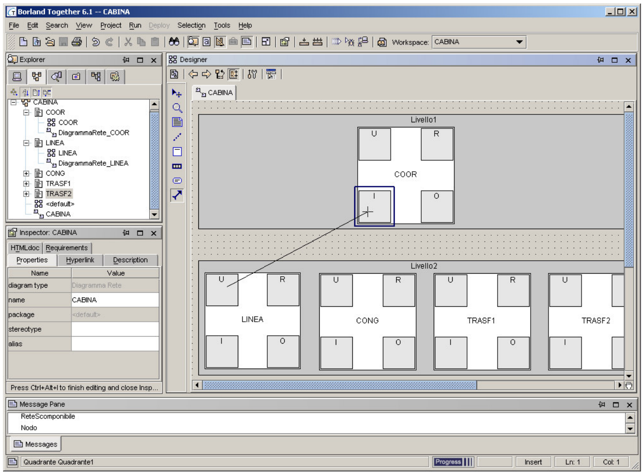

Figure 1: ASME Graphical User Interface.

One way connections are drawn inside nets to define the required data flow; at the end of the specification phase each

connection traces a routed path from automaton to automaton. Connection segments and their verse are specified at the net level, whilst informative content is assigned to the connection when the automaton owned end points are instantiated.
Figure 1 shows the network diagram capturing a first structural decomposition of an automation function into five nets attributed to two hierarchy levels: a higher level coordination function, modelled by the net COOR, and four lower level operative functions modelled by the nets enclosed in level 2. A graphical specification session may produce two intermediate formats: a native tabular format and an SCL compliant XML file.

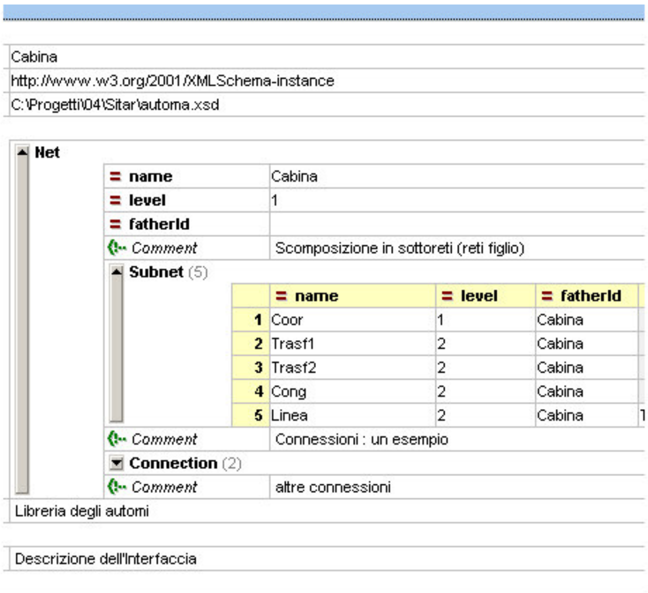

Figure 2: XML view of the ASME sample application.

Figure 2 reports the XML representation reflecting the graphical specification in Figure 1. The sample application function, modelled by the net "Cabina", is decomposed into its subnets according to the XML schema, based on the ASME object meta model. The intermediate format of the specified application is processed by the ASFA translator (consisting of a partitioning tool, a parser, a code generator and a configurator) and converted into standard ANSI C code.

Computational model and run time environment

Activity and evolution of the automation system are based on a PLC (Programmable Logic Controller) like, cyclic, execution model. At each computation cycle process inputs are sampled, stored and evaluated. Evaluation of the process input and of the system's state variables, causes the system to compute a future state and a set of outputs towards the process, to be emitted at the end of the cycle. State transition takes place atomically at the beginning of a new cycle.
In each cycle, data flow driven computations evolve through an ascending and a descending phase: from process level to coordination level functions in the upward phase (when a synthesis of low level information is conveyed to the decisional level) and from top to bottom in the downward phase, when orders are issued and actuated. The functional behaviour required by the computational model is granted by an application independent supervisor layer, called Execution Support Module. The Execution Support handles synchronization related issues, data flow control, inter process and inter site communication and exception handling, besides providing a target independent API (Application Programmable Interface) library to RTOS (Real Time Operating System) supplied facilities.
Application functions have no direct access to process data: interactions with the switchyard equipment are handled by a dedicated process access layer, which provides a uniform virtual image of the process.
This hardware abstraction layer grants the independence of the application from the technology adopted to control the process. The hardware image is modelled by a modular data area called Exchange Ram.

SAMPLE PRIMARY SUBSTATION APPLICATION

Integration aspects of ASME and IEC 61850 shall be exemplified on a sample application consisting of station- and bay- level automation sequences, specified using the ASME environment.
The primary equipment is that of a very simple MV substation: two voltage transformers connecting a high voltage busbar to two sections of a medium voltage busbar, with a sectionalizer between the two MV sections. Six MV lines are powered by the MV busbars.

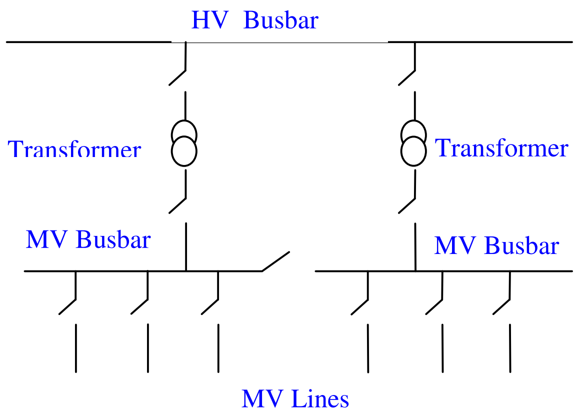

Figure 3: Sample substation topology.

The outlay of the sample substation is illustrated in Figure 3. In the case study the application implements two functions: automatic resumption of service and parallel transformers.
The first function allows automatic resumption of service when one of the voltage transformers goes down, e.g. triggered by internal protection (temperature too high, oil alarm, …). The function computes the load carried by the damaged transformer before the anomaly, and causes the remaining transformer to take over as much of the entire load as it can carry. Use of the GUI (Figure 1) and XML translation of function specification (Figure 2) are exemplified on this function. The second function consists of a sequence of automatic actions meant to assist remote operators: when an operator requests, for instance, that an

additional transformer be inserted, the function translates the request into the correct sequence of commands.

IEC 61850 AND SCL

The goal of the international standard IEC 61850 *Communication networks and systems in substations* is to provide interoperability between functions to be performed, in the same substation, by physical devices supplied by different vendors: i.e. provide interoperability between IEDs from different suppliers (interoperability meaning the ability of two or more Intelligent Electronic Devices to cooperate exchanging correct information). The IEC 61850 series uses abstract models to define information and information exchange: models and modelling methods are the core of the standard.
Part 6 of the IEC 61850 series [4] defines the Substation Configuration Language (SCL) based on XML ver. 1.1, to be used for the description of substation automation devices and their communication links. The main purpose of SCL is to provide a compatible way to exchange information between engineering tools implemented by different vendors. SCL provides a formal way to specify a complete Substation Configuration by describing

- the single line diagram of the switchyard in terms of primary apparatuses and their electrical connections,
- the related automation system in terms of Intelligent Electronic Devices and their communication infrastructure,
- the mapping between the primary and the secondary system.

INTEGRATION OF ASME AND IEC 61850

The possibility of an integrated use of the ASME specification and development environment and the IEC 61850 Substation Configuration Language was suggested by the consideration that SCL and ASME focus on two complementary aspects of Substation Automation: hw device oriented architectural description and hw independent, logical, function specification.
While defining exhaustively all communication related requirements of Substation Automation functions, the specification of individual implementations of those functions falls outside the scope of SCL. On the other hand, the main objective of ASME is to specify complex automation functions, with no assumptions about the technical aspects of the hw involved in the performance of those functions.
Though applied to different scopes, there are strong analogies in the modelling approach adopted by the two specification environments.
The general approach of the 61850 series is to decompose application functions into the smallest entities able to exchange information, where the granularity is given by a reasonable allocation of these entities, called Logical Nodes, to dedicated devices. Similarly, during the ASME specification phase, the automation functions are decomposed into small communicating entities, the granularity being given by the designer's needs to govern complexity. The decomposition stops then the entity's behaviour is simple enough to be modelled by a finite state automaton. Besides defining the communication interface, each ASME automaton defines the operational behaviour of a subfunction.
A combined use of ASME and SCL seems to offer a natural way to present a uniform view of both relevant aspects of substation automation.

Hardware virtualisation

A first aspect of ASME-SCL integration is related to hardware virtualisation. As mentioned earlier, ASME applications interact with the process through a virtual image of the data subset which is relevant to the specified application. While the definition and structuring of data to be contained in the virtual image are a prerequisite to application specification, ASME does not define in which way the hardware image, the virtual field interface, should be obtained.
As section 7-4 of IEC 61850 [5] defines standard Logical Nodes modelling the behaviour of an extensive set of primary components and the control logic of the related automation functions, a simple way to relate the contents of the ASME interface to the underlying automation equipment is to define a mapping between ASME interface data items and the attributes of data objects contained in IEC 61850-7-4 defined Logical Nodes. Structured Interface modules are obtained by grouping the Logical Nodes into Logical Devices allocated to SCL specified IED types.
Specific hw devices with the required capability can be later substituted to the IED types, when the application is instantiated to a specific switchgear installation.

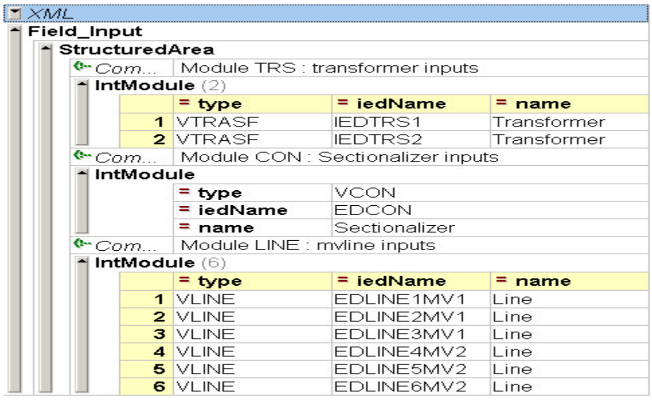

Figure 4: SCL specified ASME Process Interface.

Figure 4 shows sections of the hardware abstraction layer specified for the sample application's process interface, defined in terms of IEC 61850 compliant IEDs and their Logical Nodes.

Combined ASME-SCL description

A second aspect of ASME-SCL integration consists in specifying the exhaustive hardware and software architecture

of a substation by a combined use of SCL and ASME. First we define substation topology and the related IED architecture in an SCL configuration file, then we specify the required automation logic through ASME designed automata networks.
As automation functions are decomposed into their elementary components, each subfunction is mapped into a Logical Node, modelling its communication requirements. The behaviour of the function modelled by the Logical Node may be specified by a single ASME automaton or by a net of automata. The association is defined by mapping each communication channel of a structured automation function into a Data Object of the related LogicalNode, with Data Object Attributes expressing channel definition.
The set of standard Logical Nodes defined in IEC 61850 Part 7-4 is not sufficient to express the more complex automation functions that can be specified by ASME; but following the extension rules defined in IEC61850-7-4 some of the standard Logical Nodes have been extended and new, compatible, logical nodes have been defined whenever requested by the complexity of the functions to be modelled.

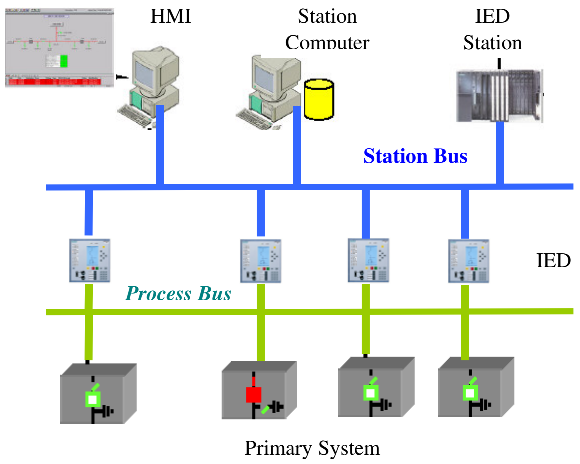

Figure 5: Substation IED architecture.

Figure 5 represents a possible substation architecture for our sample application.

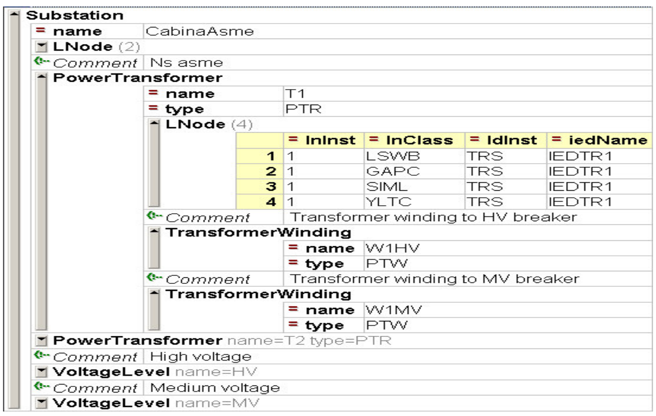

Figure 6: SCL-ASME configuration file.

Figure 6 reports excerpts of the SCL configuration file that defines the system's architecture and adds the mapping between ASME defined automation software modules to IEC 61850 Logical Nodes. The selected excerpt shows the XML element defining the structure of the IED controlling the first power transformer. Four logical nodes virtualize the IED's functionality. SIML and YLTC are IEC 61850 standard Logical Nodes modelling low level component behaviour. GAPC, the standard LogicalNode intended to model functions that are not predefined, has been modified to represent the ASME transformer function, while LSWB is an ASME extension modelling the environment's Execution Support.

CONCLUSIONS

Integration of the ASME and IEC 61850 specification models provides a uniform view of all relevant features of substation automation, with the logical decomposition of the switchyard functions offering an ideal starting point for the structured design of the related automation functions. The integration issues highlighted in the paper may be considered a first step towards the definition of standard interfaces for interoperable automation software to be performed on heterogeneous IEDs. Further integration issues have been identified: the pre-configuration of IED stereotypes performing predefined, standardised substation automation sequences; and the run time definition of the hardware image based on the SCL capability description of IEDs provided by the vendors.

ACKNOWLEDGMENTS

This work is a result of the Project *Network21, The Network for the 21st Century,* under the Public Interest Energy Research Programme founded by the Italian Ministry of Productive Activities.

REFERENCES

[1] Object Management Group, *Unified Modeling Language Specification,* Web page: http://www.omg.org

[2] Borland, *User Guide for Together ControlCenter™ and Together® Solo,* Web page: http://www.borland.com

[3] W3C Recommendation, *Extensible Markup Language (XML)* www.w3c.org/xml.

[4] IEC 61850-6, 2004, "Communication networks and systems in substations – Part 6: Configuration description language for communication in electrical substations related to IEDs", *International Standard, First Edition,* Reference Number IEC 61850-6-2004(E).

[5] IEC 61850-7-4, 2003, "Communication networks and systems in substations - Part 7-4: Basic communication structure for substation and feeder equipment - Compatible logical node classes and data classes", *International Standard, First Edition*, Reference Number IEC 61850-7-4-2003.