Functional Integration in Substation Automation Systems: System Tools and Interoperability

Rogério Dias Paulo (EFACEC Engenharia, S.A., Portugal)

Summary

Functional integration is being seen as a technologically viable and cost effective direction to pursue in substation automation today and in the future. Efficient engineering of such distributed systems requires computer-aided design tools targeted at system design and management.

In this paper, which compiles partial findings from the INTEGRA project, system design tools are focused as an important element to consider in the engineering of future functionally integrated substation automation systems heavily based on distributed technologies. The required features of a functional design language in this domain are discussed including the need for an unambiguous formal specification, notation and exchange format. The modelling domains of such language are also identified including primary process description, functional description and secondary system description.

International standards applicable to substation automation are still lacking when functional design is concerned and the applicable features of IEC 61850 and IEC 61499 to this goal are presented. Standards-based formal functional design languages applicable to substation automation should be pursued by building on the existing IEC 61850 series as an architectural foundation for the engineering of future substation automation systems.

Keywords

Functional Integration, Substation Automation, Engineering Tools, Design Languages, IEC Standards

1 Introduction

The expected cost benefits of integrating protection, control and automation, local/remote monitoring, diagnostics or maintenance are significant by: (i) reducing the amount of hardware, (ii) eliminating unnecessary redundancy and (iii) simplifying engineering processes. Technology available today (such as multifunctional devices and digital busses) makes functional integration viable.

As the substation goes digital and integration becomes possible at many, if not all, functional levels, the substation automation system (SAS) becomes a flexible, but rather complex, networked system where logical associations replace physical connections. System components rely on each other and on the communication network for performing functions, hence, in addition to the conventional technical requirements, failure modes, network performance guarantees and maintenance states must be incorporated in system design so that acceptable technical integration is achieved.

Acceptable functional integration also requires the application of cost and time-effective engineering approaches, which, in a distributed systems environment with growing functional requirements is only straightforward with the use of system-oriented tools for editing, automating and visualizing system models. This is a fundamental requirement from both technical and management standpoints.

Interoperability in hardwired systems is not an issue but in what concerns networked systems and, in particular, their supporting engineering tools, many questions arise making this issue particularly relevant.

The adoption of IEC 61850 as a global standard in substation automation comprehends a major shift towards enhanced interoperability and facilitated functional integration by focusing on communications and distributed systems engineering. The immediate benefits are now commonly accepted when applying this standard to current functional architectures and advantages are expected when fully distributed systems are employed.

Achieving major life-cycle benefits from functional integration in multi-vendor distributed systems is not possible, however, without major implications in substation automation systems engineering.

2 Engineering

From an engineering standpoint an SAS can be seen through a three dimensional perspective encompassing: (i) engineering roles, (ii) managed artefacts and (iii) engineering processes.

An engineering role model would include (i) the system architect, (ii) the system integrator and (iii) the system operator. System artefacts include (i) physical components (such as primary system equipment, devices or communication networks), (ii) models (of functions, including decomposition, semantics, interface, distribution, association; algorithms and settings; of user interface; etc.) and (iii) data to be exchanged between tools, devices, functions and roles, corresponding to partial descriptions of the system at a given point in time.

Actors of each engineering role are responsible for performing a set of activities which require the exchange of artefacts with other actors, of the same or of distinct role, as well as with the system itself. Such activities invariably require that actors are acquainted with the managed systems, therefore unambiguous system models which accurately reflect the system in-place or under design are required.

Processes such as specification, design, commissioning, operation or maintenance impact the system throughout its life-cycle requiring different artefacts to be modified and exchanged. Engineering processes tend to become more and more iterative and evolutionary as opposed to waterfall-like when the system life-cycle is considered and invariably require intervention from multiple engineers and other personnel.

Design models, for example, are needed during initial system setup for specification, validation and transformation to specific device configuration data, but are also needed during operation and maintenance so that engineers have accurate documentation of system configuration. During system extensions or modifications, design models are updated and the whole engineering process is repeated. Many design models for the same system exist during its lifetime and must be managed. In most existing processes such models are described in conventional project documentation, which is often ambiguous and hard to keep accurate with the system in-place. There is consequently poor support for automated model transformations. For fully distributed systems a more formal approach would be required to efficiently handle design models, particularly in a dynamic workforce environment.

3 Requirements for Computer-Aided Design

In a fully integrated distributed system, device operation is interrelated making unified configuration particularly relevant since the re-configuration of one node may adversely impact the behaviour of other nodes in the system. System design tools should allow the engineer to define, modify, visualize, share and, to some point, validate integrated system configurations.

Tools should facilitate functional integration by raising the level of system abstraction from data, devices and communications to functions and their behaviour. Although integrated design is required there should be a clear separation between functional design and communications or architectural design.

3.1 A Language for Designing Automation Systems

Design tools are invariably based on a supporting design language, either explicitly or implicitly. In the domain of substation automation the definition of a design language should include: (i) a meta-model, (ii) a notation and (iii) a digital format. A (quasi-)formal meta-model (the definition of the modelling language itself) with abstract syntax and clearly defined semantics is needed to avoid ambiguity, a standard notation facilitates both model editing and knowledge management and a digital format is fundamental for automated data storage and exchange.

The meta-model definition is the key element and should include provisions for the definition of: (i) primary (or controlled) process definition, (ii) secondary system functional design, (iii) description of secondary system architecture, including devices and networking, and (iv) associations between these models.

Other general guidelines to follow in its definition would include: (i) suitability for off-line design and, at least static validation checks, (ii) incorporation of version management, (iii) orientation for re-engineering and iterative design, (iv) precise semantics for automated transformation of models to executable definitions, deployable to the actual system and (v) formal description using standard model definition languages such as UML, BNF or others. Model-driven development, meta-modelling and domain specific languages are now maturing disciplines of software engineering which can bring the necessary tools, concepts and experience needed for meta-modelling in this domain.

Primary System Definition

Primary system description is most relevant since all control, protection and automation functions are designed for application to a specific substation and its understanding is required for any activity involved in automation engineering from specification to maintenance. In the substation domain power equipment and electrical connections organized in hierarchical structures are the common elements. Other non-substation equipment may also be involved therefore the meta-model should allow: (i) the definition of equipment types (over which substation equipment models can be defined as well as equipment from any other domain) and (ii) the definition of systems and subsystems as interconnected collections of equipment of given types. The primary system meta-model should include provisions for asset management information to be included.

Functional Design

The supporting meta-model should incorporate common concepts such as (i) function decomposition in atomic units, (ii) behaviour encapsulation with interface definition via inputs and outputs, (iii) support for behaviour definition, either white-box (algorithm defined by end-user) or black-box (intellectual property protected or firmware algorithms), (iv) data typing and, necessarily, (v) function instantiation. It should also allow the definition of the allocation of functions to devices and the definition of associations between input and output data independently of allocation or communication technology employed. Moreover it should be strongly targeted at reuse of functional definitions. The mapping of function instances to primary equipment virtualized or controlled closes the loop.

Secondary System Description

Secondary system description includes the definition of devices and their association through communication networks as well as logical communication endpoints and their configuration, according to each protocol employed. Given the myriad of devices in the substation today (RTUs, gateways, modems, converters, protection devices, controllers, workstations, network switches, routers, clocks, etc.) a device type/instance meta-model may also be applicable which should also contain provisions for asset management. Regarding communications an abstract meta-model for communication services, end-points, protocols and their configuration regardless of specific protocol is required allowing the unambiguous mapping of the functional model.

Notation

A concrete syntax for model handling by users should be essentially of graphical nature, although text-based syntaxes may be employed for particular purposes. In this regard commonly used notations should be reused such as single-line diagrams for primary system design, function blocks or state charts for functional design and network maps for secondary system architecture.

Exchange Format

An XML schema being an obvious choice if interoperability is a concern, the digital exchange format for automation system models can be derived directly from the language abstract syntax according to translation rules, effectively corresponding to an alternative notation or concrete syntax targeted for automated processing.

3.2 Interoperability

Interoperability can only be achieved through international standards with industry acceptance. This has been a true fact regarding communication standards in the domain of power systems automation for many years, but not as much where tool interoperability is concerned. Efficient functional integration in future multi-vendor systems requires that interoperability concerns target not only communication interfaces but also system model integration. Existing interoperability is mainly focused on providing external standards-compliant communication views which wrap internal device or vendor-specific function models and not to manage an open functionally-oriented view of the whole system.

Distributed function models are the key common elements which, together with primary process description, bridge all engineering roles and processes; therefore the standardization of domain functions under the scope of IEC 61850 constitutes a major step towards acceptable functional integration in multi-vendor environments. IEC 61850 includes: (i) a functionally-oriented communication interface data model, (ii) a technology independent communication services model, (iii) a function domain model for substation automation and (iv) a configuration process together with a standard language for model-exchange. First implementations of IEC 61850, although recent, have been successful but are still lagging in what concerns tool integration regarding both its potential and user expectations.

Being primarily a communications standard, IEC 61850 does not include provisions for functional design. Although not directly targeted at substation automation, other IEC standards are available for automation systems functional design, namely IEC 61131-3 (programmable logic controller programming languages) and IEC 61499 (distributed function blocks), a unified architecture driven from both IEC 61131-3 and IEC 61804 (distributed control systems).

3.3 Other Design Elements

In the functional design meta-model described above, the protection, control and automation functions are the main consideration. Other system-wide functions such as user interface were excluded from the analysis but these may also benefit from integrated system design. For example the integration of IT in substation automation increases the need for security management, including access control. In the future the configuration of security management will also require mapping with the functional definition itself. Any design meta-model supporting these additional features should also follow the same principles of clear separation from abstract functional definition to the definition of concrete implementation.

4 Review of Applicable IEC Standards for Functional Design

In the following tables IEC 61850 and IEC 61499 are compared and reviewed according to the presented requirements.

Table 1. Main features of IEC 61850 and IEC 61499

	IEC 61850	**IEC 61499**
Formal Definition	No	Yes
Meta-model	Can be extracted from standard by analyzing parts 6 and 7	Yes
Notation	No	Yes
Exchange Format	Yes	Yes
Main Scope	Communications engineering	Agile automation engineering
Application Domain	Power systems automation	Distributed industrial process measurement and control systems
Primary System Description	Yes	No
Functional Description	At communication interface only	Yes
Secondary System Description	Yes	Yes
Integrated system modelling	Supported for communications only, but system tool cannot independently handle some required definitions such as input/output associations. Engineering process is mainly bottom-up.	Yes, system tool is targeted at integrated system design concerning behavioural definition and allocation of functions to processors.
Functionally-oriented	The IEC 61850 meta-model provides functionally-oriented communication interfaces but does not include provisions directly targeted at functional design.	Yes

Table 2. Specific functional design features of IEC 61850 and IEC 61499

	IEC 61850	**IEC 61499**
Complete meta-model	Yes, for communication engineering.	Yes, for system behaviour engineering.
Functional decomposition	Yes, in atomic units known as logical nodes.	Yes, in atomic units known as function blocks.
Encapsulation of behaviour	Yes, in logical nodes.	Yes, in function blocks.
Behavioural definition	No	Yes, reusing program organization units and programming languages from standards such as IEC 61131-3 and including event-based execution semantics.
Mapping of functional model and I/O interface (process or device HMI)	No	Encapsulated in special function blocks.
Mapping of functional model and communication interfaces	The IEC 61850 meta-model is the communications interface.	Encapsulated in special function blocks.
Reuse of functional definitions	Yes, of domain models only.	Yes, including object-oriented definitions (type/instance meta-model).

	IEC 61850	IEC 61499
Distribution and flexible allocation of functions to devices	Yes, with a network-centric approach: logical nodes are aggregated in logical devices which are aggregated in servers contained in devices.	Yes, with a processor-centric approach: function blocks are allocated to resources which are contained in devices.
Flexible data typing	Yes, primitive data types, structures and enumerations. Some standard definitions for domain-specific information modelling.	Yes, primitive data types, structures and enumerations.
Data object model	Rich model integrating status data, controllable data, descriptive data, configuration data and runtime settings data.	Simple input/output status and event data.
Domain models	Standard compatible function models available or under preparation for substation automation and other power system automation domains. Includes provisions for meeting performance and functionality criteria.	Architectural support only.
Primary system description	Yes, for substation equipment including connectivity and association with functional model.	No
Network infrastructure description	Yes	Yes

Although successfully merging IEC 61499 with IEC 61850 would attain many of the discussed requirements, the goals of standardization of each standard are distinct and the meta-models of both standards are not straightforwardly compatible. The same formal and system-oriented approach taken by IEC 61499 should, however, be considered if reviewing and expanding of IEC 61850 to include functional modelling is pursued.

5 Conclusions

Functional integration in substation automation systems may strongly benefit from integrated system engineering tools economically, technically and from the management point of view.

Today’s domain knowledge and available technology at hardware, real-time software, software engineering and automation engineering levels is mature enough to drive the economic development of wide-scope engineering tools in this domain.

Considering the need for interoperability, formal functional design languages applicable to the domain of substation automation should be pursed under international standardization efforts built on the existing IEC 61850 series as an architectural foundation for the engineering of future substation automation systems.

Bibliography

[1] IEC TC57. “IEC 61850-7-x - Communication networks and systems in substations - Part 7-x: Basic communication structure for substation and feeder equipment” (IEC, Geneva, 2003)

[2] IEC TC65. “IEC 61131-3 - Programmable controllers - Part 3: Programming languages” (IEC, Geneva, 2003)

[3] IEC TC57. “IEC 61850-6 - Communication networks and systems in substations - Part 6: Configuration description language for communication in electrical substations related to IEDs” (IEC, Geneva, 2004)

[4] I. De Mesmaeker *et. al.* “Substation Automation based on IEC 61850” (Cigré SC B5 6th Regional Cigré, Cairo, November, 2005)

[5] IEC TC65. “IEC 61499-1 - Function blocks - Part 1: Architecture” (IEC, Geneva, 2005)

[6] T. Kostic *et. al.* “Understanding and using the IEC 61850: a case for meta-modelling” (Elsevier Journal of Computer Standards & Interfaces, vol. 27/6, pp. 679-695, 2005)

[7] R. Paulo. “Model-driven Design of Substation Automation Systems: Proposed Approach, Drives and Impediments” (GTTSE 2005, Braga, July 2005)

[8] A. Meneses *et. al.* “INTEGRA Project – Applying the IEC 61850 technology” (Cigré 2006, Paris, August 2006)

[9] J. Curk *et. al.* “Standard IEC 61850 opens possibility to develop new more efficient architectures of substation automation and protection systems” (Cigré 2006, Paris, August 2006)

[10] D. Holstein *et. al.* “Cigré Report 307 – WG B5.09 – Remote On-line Management for Protection and Automation” (Cigré, 2006)

[11] G. Senfter. “IEC 61850 - Is it worth the trouble?” (CIRED 2007, Vienna, May 2007)

[12] K. Schwarz, “Impact of IEC 61850 on System Engineering, Tools, Peopleware and the Role of the System Integrator” (DistribuTech 2007, San Diego, February, 2007)