Translating Proprietary Protection Setting Data into Standardised IEC 61850 Format for Protection Setting Validation

Q.Hong*, A.Dyśko*, C.D. Booth*, V.M. Catterson*, S.M. Blair*, T. Rahman[†]

* *University of Strathclyde, UK, q.hong@strath.ac.uk*
[†] *National Grid, UK, tahasin.rahman@nationalgrid.com*

Keywords: IEC 61850, power system protection, standard, interoperability, substation automation

Abstract

For smart grid development, one of the key expectations is that the data should be accessible to and readily interpreted by different applications. Presently, protection settings are represented using proprietary parameters and stored in various file formats. This makes it very difficult for computer applications to manipulate such data directly. This paper introduces a process that translates the proprietary protection setting data into IEC 61850 standardised format and saves the data as System Configuration description Language (SCL) files. A code generation process that allows rapid implementation of the translation process is proposed. Among various applications, the paper demonstrates how such a translation process and generated SCL files can facilitate the development of an intelligent system for protection setting error detection and validation.

1 Introduction

For smart grid development, one of the key expectations is that data should be accessible to and readily interpreted by different applications. While power system model data has been represented in open standards such as the Common Information Model (CIM) [1] (used for data exchange and storage), protection setting data is still represented in various proprietary formats. This makes it very difficult to manipulate such data directly: the data stored in proprietary file formats cannot be easily accessed, e.g. many modern IEDs' binary file formats can only be accessed through vendor-specific tools; moreover, the content of the files cannot be easily interpreted, e.g. different settings parameters and naming conventions are often used to describe the same protection functions. Some of the difficulties that proprietary settings brought to the development of automatic applications are reported in [2, 3].

The work in [4, 5] suggests using the IEC 61850 standardised data model to represent protection settings and the System Configuration description Language (SCL) [6] as a common configuration format for IEDs. However, there are still challenges for the industry to formally agree to and adopt these approaches [5]. Therefore, manipulating the protection setting data is still problematic. Even when the standardised approach is widely adopted in future devices, the problem will remain due to the large numbers of legacy devices used in the system.

This paper presents a process that automatically translates the proprietary protection setting data into the IEC 61850 standardised data format, which can be saved as SCL files. The high-level design of the process was originally proposed in [5]. The process has been refined by the authors and implemented within a prototype tool. A code generation process that automatically generates the Java [7] software needed to implement the translation process is also developed. The code generation has been designed to require as little manual input as possible and to allow rapid implementation of the translation process.

Among various applications of the proposed translation and code generation processes, the paper will demonstrate how such processes can facilitate the development of a protection setting validation system as introduced in [8]. Due to increasing network complexity, there are cases where incorrect protection settings remain undetected until a mal-operation is experienced [9]. The protection setting validation system reads the protection setting data and automatically detects if there are any errors contained in the settings. The translation process proposed in this paper simplifies the development task and makes it easier to maintain such a system.

The paper is organised as follows: Section 2 presents background information relating to the IEC 61850 data model and the SCL file format. Section 3 introduces the overall translation process that converts proprietary protection settings into SCL files and the code generation process that facilitates rapid development of the aforementioned translation process. Section 4 introduces the protection setting validation system and demonstrates how the proposed data translation and code generation processes can facilitate the development of such a system. Section 5 presents future research and a number of other potential applications of the work presented in this paper.

2 The IEC 61850 data model

IEC 61850 provides a standardised data model for describing substation automation functions [10, 11]. A physical device (e.g. a physical IED) is decomposed into logical devices. Each logical device represents a virtual device that can
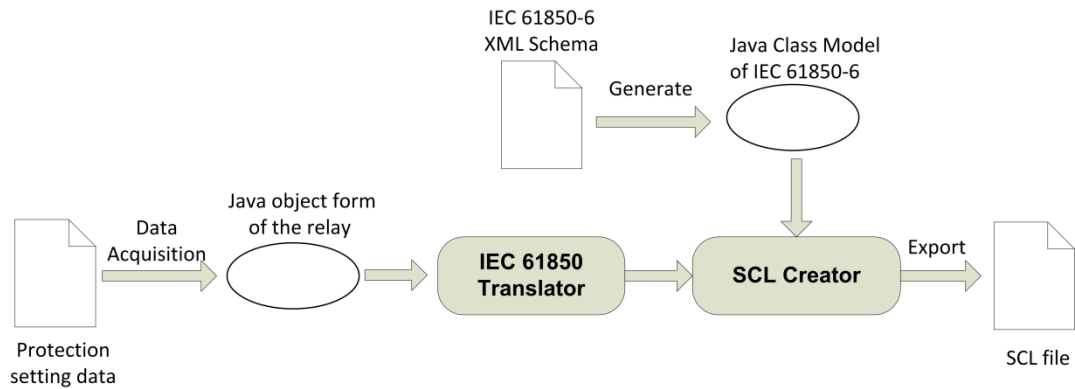
Fig. 1. Translation process that converts proprietary settings to SCL files

perform a specific function, e.g. protection, measurement and/or control. The logical device comprises a set of functional entities termed logical nodes, the smallest single components that can be represented in the model, e.g. PDIS logical node for distance protection zones [11].

Each logical node contains a set of data objects, which are instances of common data classes (CDCs) [10], to describe the function it represents. Within the data objects, there are data attributes that specify relevant details, e.g. value and unit.

SCL is a standardised file format defined in IEC 61850-6 for information exchange and function description [6]. It is based on the XML syntax [12], and can therefore be easily interpreted by computer programs. The use of the standardised data model and SCL file format facilitates interoperability across different vendors' devices.

In the current version of the standard there are 30 predefined logical nodes for describing various protection functions. However, presently the data model is not widely used for representing and storing protection settings. A number of challenges of using the standardised data model for protection settings are reported in [5]. There are also on-going activities that propose changes and extensions to the existing data model as reported in [3, 4].

3 Converting the proprietary protection setting data into SCL files

This section presents the process that automatically translates the proprietary settings into IEC 61850 standardised data and saves these as an SCL file. The implementation of the process is facilitated by the code generation process proposed by the authors. The overall translation process is illustrated in Fig. 1. There are three main stages involved in the process: data acquisition, translation, and SCL export. The process has been implemented using the Java programming language [7].

3.1 Acquisition of protection data from proprietary files

Existing protection settings are stored in various formats, such as text files and databases. For modern IEDs, vendor-

specific binary files are also used to store the settings. At the protection data acquisition stage, the setting data is extracted from the original data source for further manipulation and processing.

Various software parsers have been developed to extract setting data from text files as described in [5]. The key requirement for successful data parsing is that the data presented in the text file should be in a relatively consistent format.

Binary files cannot be accessed directly without vendor specific tools. However, these files can normally be exported by the vendor's tools as text, CSV, or XML files, which can be parsed automatically afterwards. The authors also developed several parsers capable of extracting information from exported CSV files.

For setting data stored in a database, the parsing stage will not be necessary because data can be retrieved directly using database software.

If setting data is represented in a manner which cannot be easily parsed, the developed process also supports manual data entry. The output of the data acquisition stage is a Java object representation of the relay with setting values.

3.2 Mapping proprietary setting data to the IEC 61850 data model

The mapping stage is performed by the IEC 61850 Translator (as illustrated in Fig. 1), which contains the mapping relationships between the proprietary setting data and the IEC 61850 data objects. It is important to make sure that all the necessary information is mapped to related objects and that all the data objects in IEC 61850 model have been populated with appropriate information. Transferring all of the setting data to IEC 61850 data objects may not be sufficient to represent all characteristics, because some of the features are not configurable and visible to the users, and must be obtained manually.

Compared with the original process proposed in [5], which creates the mapping directly from proprietary settings to the

IEC 61850 data objects, the mapping process is refined by adding a stage of modelling the original IED, referred as "initial-modelling" to distinguish it from modelling using the IEC 61850 data model. The initial-modelling regroups the setting parameters into function blocks that better correspond with the IEC 61850 modelling approach and also simplifies the task of creating the mappings during implementation. The initial-modelling is performed during the code generation process that will be introduced in Section 3.4.

3.3 Saving protection setting data as SCL files

IEC 61850-6 provides an XML schema for its data model [6]. The XML schema defines the structure, contents and semantics of an XML document [13]. As illustrated in Fig. 1, the XML schema provided by IEC 61850-6 is used by the open source software Eclipse Modelling Framework (EMF) [14] to automatically generate a Java class form of the SCL model. A similar application of EMF is described in [15]. The mapping results from the IEC 61850 Translator are used by the SCL Creator to create an instance of the model. EMF also supports the automatic conversion between XML files and Java objects, and this feature is used by the SCL Creator to export the instance of the model as an SCL file.

By design, IEC 61850 does not define the internals of logical nodes [5]. Creating an instance of the SCL model generated from the XML Schema can be time consuming. Therefore, in the original design in [5], the authors implemented CDCs and logical nodes with internal definitions as Java classes to represent IEC 61850-defined data, rather than using an instance of an SCL model directly. The disadvantage of this approach is that the data cannot be automatically exported into SCL files using EMF.

The refined process in this paper addresses this problem with the help of the code generation process that will be introduced in the following section. Initialising the instance of the model becomes automated and the instance of the model can thereby be exported to SCL files using EMF.

3.4 Automatic code generation

Manual implementation of the translation process requires significant programming effort which can be time-consuming. This is due to the large variety of existing relay types and protective functions. It is also time-consuming to create an instance of the SCL model, which involves building the internals of logical nodes and assigning values to relevant data attributes. Therefore, an automatic code generation process, which is designed to use as much automation as possible, has been developed.

The code generation process, illustrated in Fig. 2, starts with the initial modelling of the original IED, which regroups the settings parameters to better match the IEC 61850 modelling approach. The result of the initial modelling is used to initialise the RelayDetail Class, which contains the detailed information about the relay, e.g. functions and parameters available. The relay's information is then used for generating the Relay Object Class and IEC 61850 Translator Class. The

Relay Object Class is the Java representation of the relay and is initialised using the data extracted from the data acquisition process. The IEC 61850 Translator Class contains the mappings between the proprietary settings and IEC 61850 standardised data. The generated translator class is only a template class tailored to a specific relay type without detailed implementation. The mapping relationship need to be built manually. The DataTypeTemplate file, which is a placeholder for the DataTypeTemplate section in the exported SCL file, is an XML file that contains the definitions of all data types that are needed in the SCL files, i.e. the logical node types, data object types, and data attribute types. The information provided by the RelayDetail object and the DataTypeTemplate file is used to generate the SCL Creator class that creates an instance of the SCL model and exports it as an SCL file.
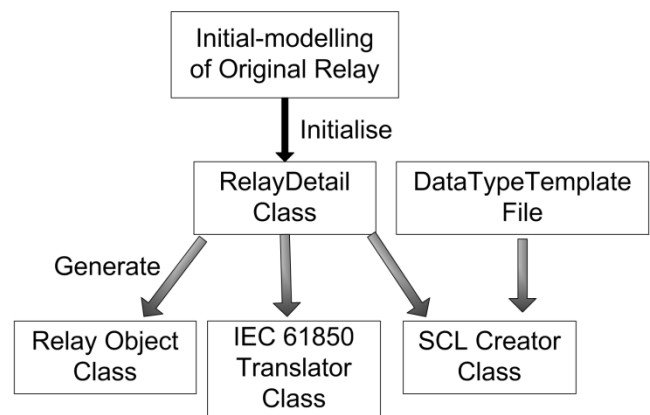

Fig. 2. Code generation process

4 Protection setting verification system using the proposed translation and code generation processes

Due to the growth of network complexity, it has been recognised (and experienced) by some in the industry that relying solely on personnel and manual procedures to assess the validity of the protection relay settings may occasionally result in errors in settings remaining undetected until a failure is experienced, which may lead to power outages or other severe consequences [8]. Therefore, an intelligent system for automatically detecting the hidden errors in protection settings files has been proposed [8]. During the design and implementation of such a system, it has been realised that the protection setting data represented using proprietary parameters and file formats causes significant difficulties for development and maintenance. This section will demonstrate how the proposed translation and code generation process can be used to facilitate the development of such a system.

4.1 Design of the protection validation tool using proprietary protection settings

The protection setting validation system involves two stages: rule-based reasoning and model-based reasoning. The rule-based stage takes the protection setting data, matches it

against relevant pre-defined rules and makes the decision as to whether settings violate any of the rules or are compliant. The model-based stage verifies the settings using dynamic protection system and power system models. Faults are simulated at various locations to verify that the applied settings lead to the desired protection action (or inaction).
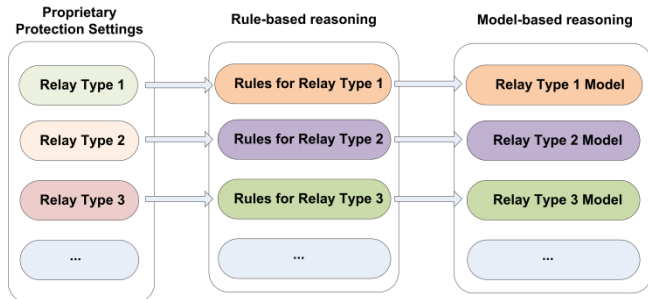

Fig. 3. Design of protection validation system using proprietary protection settings

Fig. 3 illustrates the design of the protection validation system using proprietary settings. For the rule-based stage, while the knowledge for verifying the settings may remain similar, it has to be translated into a specific set of rules for each relay type. For example, for distance protection, one of the setting criteria is that the Zone 1 positive impedance reach has to be set at 80% of the positive sequence impedance of the protected line. For the Alstom P443 relay [16], the parameter "Z1 Ph. Reach" is used to represent the Phase Zone 1 positive sequence impedance reach. In the ABB REL670 relay [17], the positive sequence impedance reach is represented using "X1" and "R1", where X1 is the reactance reach while R1 is the resistive component of the set impedance reach. For the former relay [16], the setting criterion is translated into the rule condition as: "*Z1 Ph. Reach* $==80\% \times Z_l$" where Z_l is the positive sequence impedance of the protected line and "$==$" means "equal to". When the rule condition is met, it means the setting criterion is fulfilled and no warning will be given and vice versa. For the latter relay [17], the criterion has to be represented using two rules with the conditions: "$R1==80\% \times Re(Z_l)$" and "$X1 == 80\% \times Im(Z_l)$", where $Re(Z_l)$ denotes the real part of Z_l and $Im(Z_l)$ represents the imaginary part of Z_l. Therefore, the same setting criteria can be translated into quite different rules to verify the proprietary settings.

The disadvantages of this approach are:

1) Developing rules is a significant burden: the setting criteria have to be translated into a specific set of rules for each relay type. Significant effort is required to cater for the large number of relay types in use.

2) Difficulties in maintenance: the rule base has to be updated whenever a new relay type's rules are included. Frequent updates of the rule base are not desirable because it involves changes to the internals of the software. Whenever changes are made, the rules have to be fully tested to make sure they are correctly defined. Furthermore, when there are changes to the setting criteria, relevant rules for each type of relay have to be updated. Even for a small change in the setting criteria, a large amount of time and effort are needed to update the rule base. It also increases the risk of errors in defining the rules.

3) Difficulties in function development: for a protection setting validation system, it is helpful to display the protection characteristics graphically, e.g. distance protection zones in R-X diagrams [18]. The use of proprietary setting parameters makes it very difficult to develop such a graphical tool because each relay has its own way of defining the characteristics.

If there is no error identified by the rule-based stage, the proprietary setting data are forwarded to the model-based stages for further validation. Presently, the relay models used by the authors only support proprietary settings. The key difficulty is the development of a generic methodology and software implementation that can take the setting data and set up the relay model automatically.

4.2 Design of the protection validation tool using the proposed translation and code generation process
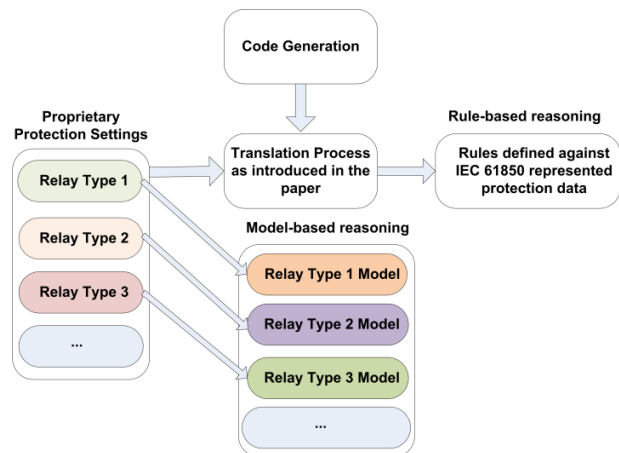

Fig. 4. Design of the protection setting validation tool using the proposed translation and code generation processes
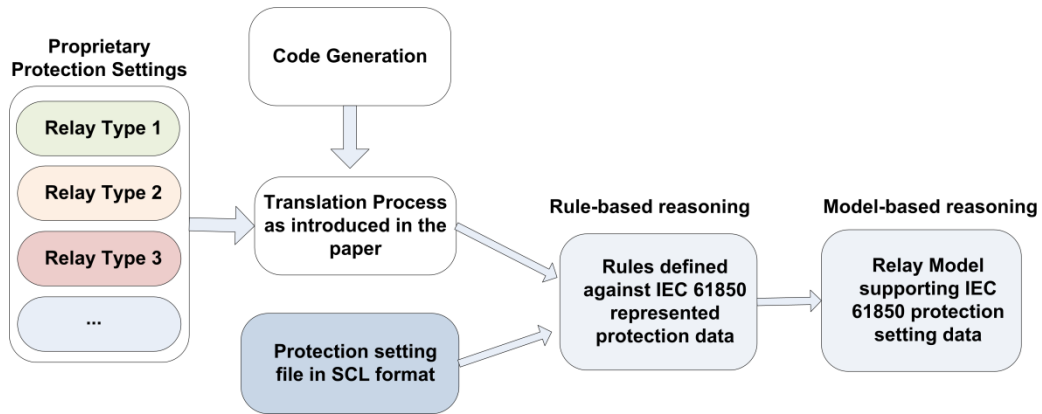
Fig. 5.   Design of the validation tool using dynamic models that support standardised protection settings

Fig. 4 illustrates the design of the protection setting validation tool using the proposed translation process. Instead of verifying settings using parameters represented using proprietary terms, the settings data are first translated into the IEC 61850 representation format using the translation process introduced in this paper, and then inserted into the rule base for inference. The rules are defined based on the common representation of data rather than proprietary parameters, so the number of rules is significantly reduced, and it is easier to update if there are any changes to the setting criteria. When including a new relay type, only the translation process needs to be built, avoiding changes to the rule base. The development of the translation process can be facilitated by the code generation process proposed in the paper. Therefore, the development and maintenance effort for the rule-based stage is significantly reduced. Furthermore, with protection functions described using the common data model, it is much easier to graphically display the protection characteristics.

Since the existing dynamic relay models used by the authors only support proprietary settings, the proposed processes cannot simplify the development of the model-based stage. However, if models that support the IEC 61850 representation of protection settings are developed, the design of the validation system can be further improved as illustrated in Fig. 5. The proprietary setting data are converted into SCL files using the translation process introduced earlier. If the protection setting files are originally stored in SCL format, it can be imported directly.   Rule-based and model-based reasoning are then carried out on the protection setting data (represented in the standard form). Since the protection data is presented in a common format, it becomes easier to develop programs to automatically utilise these relay models.

It is sensible to expect that future relay models will support protection settings in a common data format. There are already on-going activities that propose a common format for configuration of actual protection devices [4]. Presently, a protection simulation system usually requires the setting data to be input manually, which is time-consuming. The process could be automatic if the common representation of protection settings was widely adopted.

5   Future work and other applications

As mentioned earlier, the key challenge for representing proprietary data using the IEC 61850 data model is that the data model is often not sufficient to describe existing protection functions [5, 19]. There are on-going activities that propose the extension of and changes to the existing data model [3, 4]. More work is needed in order to develop a comprehensive data model for a wide range of protection functions.

Future work will involve developing relay models that support protection settings in IEC 61850 data format. This will involve an investigation of the feasibility of future commercial relays using this standard for protection settings.

There are other potential applications of the proposed translation process and the common representation of protection settings:

1)   Protection setting data for other software applications

The proposed process can be used for any software application that needs to manipulate existing proprietary protection setting data, such as protection function testing and protection coordination tools.   When the approach of representing protection setting using SCL files is widely adopted in the future, the proposed process still provides a way to integrate and manipulate legacy devices' proprietary data.

2)   Protection setting interoperability and simplifying the existing IED configuration process

The common representation of protection settings data can be converted back to proprietary settings. Therefore it is possible to convert one relay's settings to the common representation, and then convert back to another relay's format, thus achieving interoperability of protection settings even across different relay vendors' devices. The translation process and the common representation of protection setting data can also lead to a simplification of the existing IED configuration process [5].

6 Conclusions

This paper has presented a translation process that can automatically translate proprietary protection settings into the IEC 61850 standardised data format, which can then be saved as SCL files. The process has been implemented as a prototype tool, and a code generation process that allows rapid development of the translation process has also been developed. It has been demonstrated how the translation and code generation processes can facilitate the development and maintenance of a protection setting validation tool. The standardised protection settings and proposed processes can also be used for other software applications that need to manipulate proprietary protection settings, achieving protection interoperability which simplifies the existing IED configuration process. Future work involves the development of a comprehensive data model for protection settings, and the creation of relay models that support the IEC 61850-based common representation of protection settings.

Acknowledgements

References

[1]   A. W. McMorran, G. W. Ault, I. M. Elders, C. E. T. Foote, G. M. Burt, and J. R. McDonald, "Translating CIM XML power system data to a proprietary format for system simulation," *Power Systems, IEEE Transactions on,* vol. 19, pp. 229-235, 2004.

[2]   A. Apostolov. (2013, June) Disturbance Analysis for the 21st century. *Pacworld*. pp.4.

[3]   A. Apostolov and B. Vandiver, "On the standardization of distance characteristics," in *Power Systems Conference: Advanced Metering, Protection, Control, Communication, and Distributed Resources, 2007. PSC 2007*, 2007, pp. 209-212.

[4]   Working Group H5, "Common Format for IED Configuration Data " IEEE Power System Relaying Committee, 2013.

[5]   Q. Hong, S. M. Blair, V. M. Catterson, A. Dysko, C. D. Booth, and T. Rahman, "Standardization of power system protection settings using IEC 61850 for improved interoperability," in *Power and Energy Society General Meeting (PES), IEEE*, 2013, pp. 1-5.

[6]   IEC, "IEC 61850-6: Configuration description language for communication in electrical substations related to IEDs," 2 ed, 2010.

[7]   Java. (Jan 06, 2014). Available: http://java.com/en/

[8]   Q. Hong, A. Dysko, and C. Booth, "Intelligent system for detecting 'hidden' errors in protection settings," in *Universities Power Engineering Conference (UPEC), 2012,47th International*, 2012, pp. 1-6.

[9]   SIEMENS. (Jan 14, 2014). *Global Blackouts – Lessons Learned*. Available: http://www.energy.siemens.com/mx/pool/hq/power-transmission/HVDC/Global_Blackouts.pdf

[10]   IEC, "IEC 61850-7-3: Basic communication structure –Common data classes," 2 ed, 2011.

[11]   IEC, "IEC 61850 7-4: Basic communication structure —Compatible logical node classes and data object classes," 2 ed, 2010.

[12]   N. Walsh. (Jan 15, 2014). *A Technical Introduction to XML*. Available: http://www.xml.com/pub/a/98/10/guide0.html

[13]   W3C. (Jan 15, 2014). *XML Schema*. Available: http://www.w3.org/XML/Schema

[14]   Eclipse. (Jan 14, 2014). *Eclipse Modeling - EMF - Home*. Available: http://www.eclipse.org/modeling/emf/

[15]   S. M. Blair, F. Coffele, C. D. Booth, and G. M. Burt, "An Open Platform for Rapid-Prototyping Protection and Control Schemes With IEC 61850," *Power Delivery, IEEE Transactions on,* vol. 28, pp. 1103-1110, 2013.

[16]   AlstomGrid. (Dec 19, 2013). *Alstom MiCOMho P443 Technical Manual*.

[17]   ABB. (Dec 19, 2013). *Line distance protection REL670 technical reference manual*. Available: http://search-ext.abb.com/LibraryDownloadManager/Default.aspx?resource=http://www05.abb.com/global/scot/scot387.nsf/veritydisplay/0fd403f4c81c5a9cc1257b0c0044665b/$file/1MRK506312-UEN_C_en_Technical_reference_manual__REL670_1.2.pdf

[18]   GE Energy. (Jan 14, 2014). *Use of the R-X Diagram in Relay Work*. Available: http://store.gedigitalenergy.com/FAQ/Documents/General/GET-2230B.pdf

[19]   ENTSOE. (Jan 04, 2014). *IEC61850 TaskForce - Punch List*. Available: http://www.entsoe.eu/fileadmin/user_upload/_library/news/IEC61850_standard/121018_PunchListTF61850_15102012V4.pdf