

# An Open Platform for Rapid-Prototyping Protection and Control Schemes with IEC 61850

Steven M. Blair, *Student Member, IEEE*, Federico Coffele, Campbell D. Booth, and Graeme M. Burt, *Member, IEEE*

**Abstract**—Communications is becoming increasingly important to the operation of protection and control schemes. Although offering many benefits, using standards-based communications, particularly IEC 61850, in the course of the research and development of novel schemes can be complex. This paper describes an open source platform which enables the rapid-prototyping of communications-enhanced schemes. The platform automatically generates the data model and communications code required for an Intelligent Electronic Device (IED) to implement publisher-subscriber Generic Object-Oriented Substation Event (GOOSE) and Sampled Value (SV) messaging. The generated code is tailored to a particular System Configuration Description (SCD) file, and is therefore extremely efficient at run-time. It is shown how a model-centric tool, such as the open source Eclipse Modeling Framework, can be used to manage the complexity of the IEC 61850 standard, by providing a framework for validating SCD files and by automating parts of the code generation process.

The flexibility and convenience of the platform is demonstrated through a prototype of a real-time, fast-acting load shedding scheme for a low-voltage microgrid network. The platform is the first open source implementation of IEC 61850 which is suitable for real-time applications such as protection, and is therefore readily available for research and education.

**Index Terms**—Automation and control, code generation, communications, IEC 61850, microgrids, power system protection, rapid-prototyping.

## I. INTRODUCTION

COMMUNICATIONS is becoming increasingly important to the operation of modern and emerging protection and control schemes, particularly for managing the impact of distributed generation (DG) [1] and low-voltage microgrids [2], [3], for enabling fast-acting protection and restoration [4]–[6], and for ensuring wide-area integrity of a power system [7]. IEC 61850 offers several benefits to these schemes, such as: high-speed Ethernet communications, a standardized data model, a formal configuration language, reduced life-cycle costs, and interoperability [8], [9]. Nevertheless, using IEC 61850 may involve a time-consuming “ground-up” approach, the purchase of a relatively expensive—although fully-featured—software library and compatible hardware, or modification of an existing Intelligent Electronic Device (IED). The IEC 61850 standard itself is large and complex [10], [11]. For these reasons, it can be impractical to use IEC 61850 in the development of prototype systems.

This work was supported by the Engineering and Physical Sciences Research Council.

The authors are with the Institute for Energy and Environment, Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, G1 1XW, UK (e-mail: steven.m.blair@strath.ac.uk).

This paper describes a different approach, which involves automatically generating the data model and low-level communications code required for one or more IEDs, directly from their configuration description. This approach therefore eliminates a significant engineering burden during the development and testing of prototype schemes which require communications.

The platform implements publisher-subscriber communications using Generic Object-Oriented Substation Event (GOOSE) and Sampled Value (SV) messaging. Hence, by supporting communications protocols defined in the IEC 61850 standard, rather than an arbitrary communications protocol (which may be a tempting approach for a prototype system), the software generated by this process is interoperable with other IEC 61850 IEDs. The platform, which is available at [12], is the first open source implementation of IEC 61850 which is suitable for real-time applications such as protection, and is therefore readily available for research and education purposes.

This paper builds on the contributions of [13]. Section II provides the relevant background on IEC 61850, and describes the advantages of the proposed platform. The implementation is explained in detail in Section III, and Section IV describes the use of the platform to implement prototype microgrid load shedding and real-time monitoring systems. Section V describes potential opportunities for the existing platform, further work which could extend its capabilities, and other types of applications—relating to IEC 61850—which could benefit from the approach described in this paper.

## II. BACKGROUND AND PLATFORM OVERVIEW

### A. IEC 61850 Background

IEDs, such as the devices illustrated in Fig. 1, contain a number of logical nodes, where each logical node implements a particular protection or control function. Logical nodes often require data inputs, known as data sets, from other IEDs. GOOSE and SV messages perform the role of transferring these data sets over a communications network. Parts 8-1 [14] and 9-2 [15] of IEC 61850 define the mapping of IED data to GOOSE and SV message formats, respectively, using Ethernet as the communications protocol.

A System Configuration Description (SCD) file is an Extensible Markup Language (XML) document which defines all IEDs, data sets and communications within a power system, using the XML syntax known as the System Configuration description Language (SCL) [16].

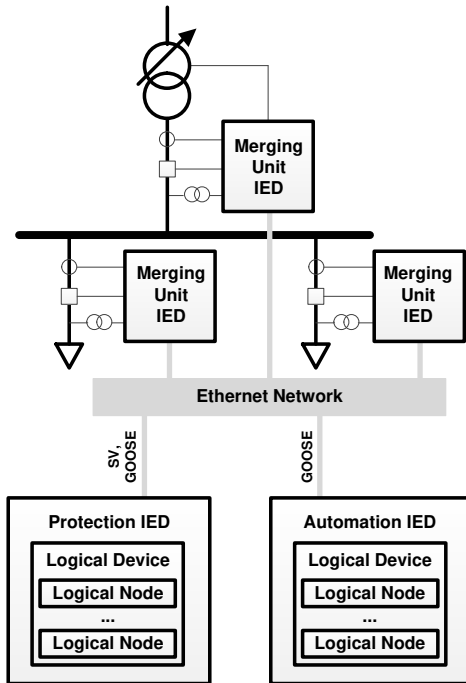


Fig. 1. Examples of typical IED types

### B. Benefits of Proposed Platform

The platform described in this paper uses an SCD file to automatically generate code, in the C programming language, which implements IEDs. There are three main reasons for this approach:

- 1) The generated code is tailored to the data types and other requirements of a specific SCD. Therefore, the generated code is very fast at run-time because the communications stack is “hard-coded”, rather than generic [17]. The IED does not need to 1) interpret the SCD, in a generic manner, at run-time, 2) maintain an internal model of the SCD, and 3) query the internal model whenever a GOOSE or SV packet is received. This approach facilitates the use of relatively low-cost embedded devices, and is especially suitable for applications where deterministic performance is critical, such as for fast-acting protection schemes. The generated code is also significantly simpler than a generic IED implementation.
- 2) The code generation process is designed with as much automation as possible, as described in Section III, to facilitate the rapid-prototyping of experimental systems.
- 3) The entire process, and the resulting configuration of each IED, is centered around the SCD, and hence is faithful to the configuration methodology defined in IEC 61850-6 [16].

In addition to generating C code to implement communications, the platform also automatically generates Python or Java libraries from the C code. This is beneficial because it provides the option to implement logical nodes in a high-level programming language.

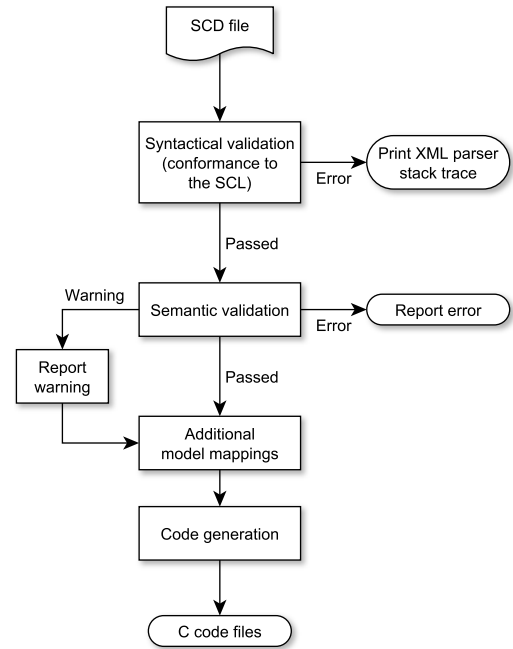


Fig. 2. Overall validation and code generation process

## III. IMPLEMENTATION

### A. Leveraging a Model-Centric Framework

The overall code generation process is illustrated in Fig. 2. The first step is to import an SCD file into a programming language environment. There are many tools, implemented in many programming languages, for parsing XML documents. However, most of these tools deal with XML in a generic manner without understanding the structure, rules and semantics that are specific to the SCL. The SCL model is defined by an XML Schema in IEC 61850-6; therefore it is appropriate to use a tool that can import this XML Schema, as well as importing an instance of the model (i.e., an SCD file). The Eclipse Modeling Framework (EMF), built on the open source Eclipse platform and implemented in the Java programming language, is designed to assist with the development of software that is based on a structured model [18].

Fig. 3 summarizes the role of EMF within the platform described in this paper. EMF automatically generates a Java code representation of a model from, in this case, the XML Schema defined in IEC 61850-6. It also generates an XML parser for SCD files, which is tailored to this model. The output of the SCD parser is a model instance—a hierarchy of Java objects—which can be queried and manipulated in software.

The benefits of this approach are that 1) the class hierarchy is generated automatically, 2) objects are automatically created and populated with data from an SCD file, and 3) EMF fully automates the “syntactical validation” stage in Fig. 2. Although possible, it would be time-consuming to manually create the class hierarchy defined in IEC 61850-6 and to write functions to instantiate the model. Furthermore, using a model-centric tool such as EMF ensures that the platform can more

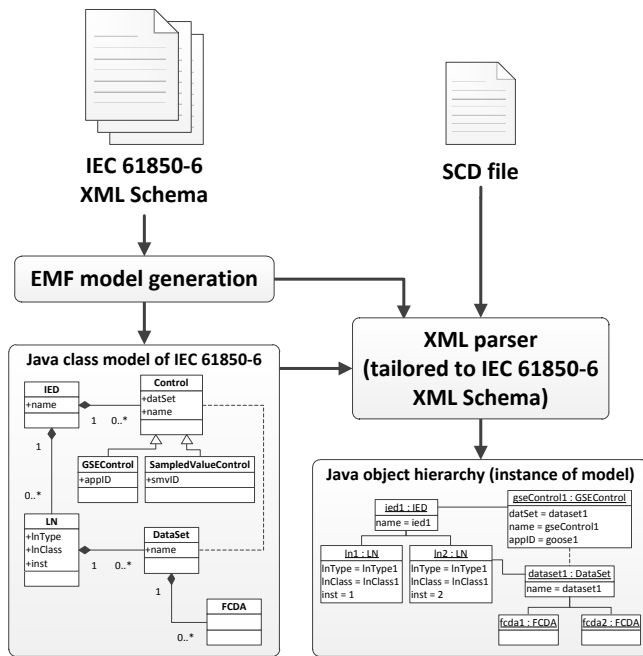


Fig. 3. Process of importing IEC 61850-6 model into EMF (for brevity, illustrative UML class and object diagrams are shown)

readily adapt to future changes to IEC 61850-6.

### B. Semantic Validation of SCD Files

EMF provides a framework for querying the SCD contents for further validation, beyond the basic syntactical conformance described in Section III-A. The platform presently checks that the following constraints are met, as required by IEC 61850-6 [16]:

- Where necessary, the names of IEDs, logical nodes, data sets and data types are unique.
- Each Control instance has matching DataSet and Control-Block instances.
- Each logical node “Input” has a corresponding source in a data set (typically in another IED).
- No circular sub-data object (SDO) references occur.
- Data attributes, basic data attributes and SDOs must map to valid types that exist in the SCD file.

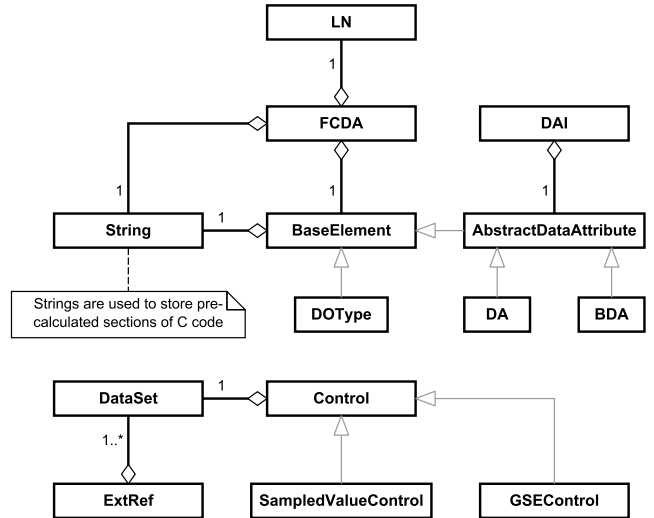
Semantic validation is necessary because any error in an SCD file, such as a duplicate data type name, may be difficult to identify manually and could result in generated C code that will not compile. The validation process can be used independently or as part of another software tool, if code generation is not required.

### C. Augmenting the SCL Model

Many components within the SCL are implicitly related to each other. For example, logical nodes and data objects must map to valid data type definitions. Throughout the SCL, this mapping is achieved by matching the text of XML attribute values [19], as illustrated in Fig. 4a, rather than with semantically stronger links, e.g., by using Uniform Resource

```
<IED name="IED1">
  <LN lnType="savContainer" lnClass="GGIO" inst="1">
</IED>
...
<LN lnType="savContainer" lnClass="GGIO">
  <DO name="sav" type="mySAV"/>
</LN lnType>
```

(a) Example of existing “text-matching” mapping in SCD files (simplified for brevity)



(b) UML class diagram of additional model mappings (existing relationships in the SCL are grayed-out)

Fig. 4. Comparison of existing implicit and proposed explicit SCD mapping methods

Identifiers (URIs) to reference other items in the XML document, or by using the Resource Description Framework (RDF) to allocate each item with a unique identifier and then defining the relationships between items [20].

The SCL model has been augmented to make these links explicit. This avoids cumbersome text-matching and thereby significantly simplifies the code generation process. Fig. 4b summarizes the necessary additional associations to the SCL, as a unified modeling language (UML) class diagram. The augmentation has been implemented by populating additional Java hash map data structures, rather than changing the definition of the SCL itself. This was chosen to avoid the need to maintain two models (the original SCL model and the non-standard augmented model), and to avoid needing to transform between those models.

Therefore, this stage creates a layer of abstraction between the Java classes which implement the SCL model (and were automatically generated by EMF), and any client application which uses the data from an SCD file. In this case, the client application is the C code generation process.

### D. Code Generation

To ensure that the platform is flexible for many types of applications, the generated code should be able to be compiled and executed on a variety of devices, including microcontrollers and desktop computers. This sub-section explains how this is

achieved, while ensuring that the code remains very fast at run-time.

1) *Data Types*: An SCD file includes definitions of all data types used within the file, such as logical node types, data object types and data attribute types. In the generated C code, each type is mapped directly to a C data structure, resulting in a hierarchy of C data structures. Primitive types, such as integer and floating-point numbers, are mapped to generic primitive types, which are then mapped to device-specific C primitive types of appropriate byte-length and sign. Hence, there is an interface layer between the automatically generated code and the device-specific code; the data type model is thereby inherently device-independent. This also keeps all device-specific code together, which is therefore easier to maintain.

An example of this mapping is given in Fig. 5, for the Common Data Class “SAV” [21]. Similarly, IEDs are defined as a C structure containing logical node instances (within logical device structures); this therefore captures the object-oriented composition modeled in the IEC 61850-6 data type definitions.

2) *Encoding and Decoding GOOSE and SV Packets*: Each GOOSE and SV packet contains a data set, which may consist of a number of data objects and data attributes. Therefore, each data object type and data attribute type has corresponding functions for serialization into a data set, and de-serialization from a data set. GOOSE data set values are encoded using Basic Encoding Rules (BER) [22], but SV data set values use fixed-length fields, as defined in IEC 61850-9-2 [15].

The encoding and decoding functions are independent of the endianness (the byte order of multi-byte data types) of the target hardware device, until compile time. This ensures 1) consistency with IEC 61850, regardless of the target hardware’s native endianness, 2) that the code for encoding and decoding data sets is device-independent, and 3) that there is no impact on the run-time performance of the code, because the choice of endianness is selected by the C pre-processor.

#### IV. CASE STUDY: PROTOTYPING MICROGRID LOAD SHEDDING AND MONITORING SYSTEMS

##### A. Microgrid Protection and Control

The low-voltage microgrid network shown in Fig. 6a has been used to demonstrate the use of the platform described in this paper. There are several challenges involved with protecting and controlling microgrids, such as detecting faults when grid-connected and when under islanded operation (particularly when there is a high presence of converter-interfaced DG) [23], and maintaining system stability during unplanned islanding events [2]. Communications can help to resolve these challenges. Therefore—along with their relatively compact size, which facilitates communications—microgrids are good candidates for the use of IEC 61850 for protection and control.

##### B. Overview of Demonstration

The microgrid illustrated in Fig. 6a has three interconnected areas, each with local loads and DG, and a single connection to the grid. The microgrid can be operated in grid-connected or

islanded mode to guarantee continuity of power supply during faults on the electrical distribution network. During the change from grid-connected to islanded operation, it is critical to have a control system to ensure the stability of the islanded power system [24]. For example, it may be necessary to trip loads from the microgrid if the total load on the system exceeds the combined rating of all operational DG. A prototype centralized load shedding scheme, with real-time monitoring, has been developed to manage this process.

This example, while functionally straightforward, demonstrates the flexibility and convenience of the platform described in this paper. Once the SCD file is specified, only the logical node implementations are needed. This example also demonstrates the generated code performing two different applications (a load shedding IED and a monitoring IED) on different hardware devices, and demonstrates interoperability with a third-party IED. Furthermore, a real-time controller-in-the-loop demonstration is not only more convincing than a purely simulation-based demonstration, but it can also readily be extended to, for example: include communications delays, investigate different communications network topologies, or test more complex load shedding algorithms.

##### C. Power System Simulation

As shown in Fig. 6b, the microgrid network has been simulated using a Real-Time Digital Simulator (RTDS) [25]. Each DG unit has been modeled as a diesel generator, but any type of generation could be used. The RTDS uses a GTNET card to communicate with other IEDs using, in this case, GOOSE messaging. Each of the three load/generation areas transmits a GOOSE message containing the status of the local DG unit (whether the DG unit is operational and connected to the microgrid or not) and the real power consumed by each local load. The loss of mains (LOM) controller, located at the grid connection point, sends a GOOSE message indicating whether the system is grid-connected or islanded (due to a remote fault or other event). All circuit breakers in the simulation open at the first current zero-crossing after a 60 ms delay following a trip command.

##### D. Rapid-Prototyping IEDs

The configuration of all IEDs has been specified in an SCD file. Therefore, the IEC 61850 communications code, used by both the load shedding and monitoring IEDs, has been generated automatically using the process described in this paper.

The load shedding IED receives the GOOSE messages sent from the IEDs simulated in the RTDS. When the network changes from grid-connected to islanded operation, this IED is responsible for determining which loads to shed to help the DG units to maintain voltage and frequency within the regulatory limits. The control logic for the load shedding algorithm—i.e., the implementation of the logical node—has been implemented in Simulink, which allows the control scheme to be developed and simulated at a high-level, and allows the re-use of existing Simulink blocks. Once complete, C code which implements the scheme, and is optimized for embedded

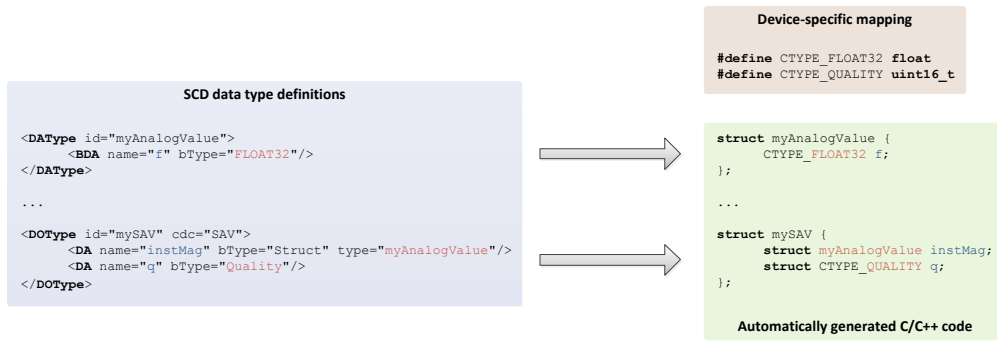


Fig. 5. Example of mapping between SCD data types and C data types

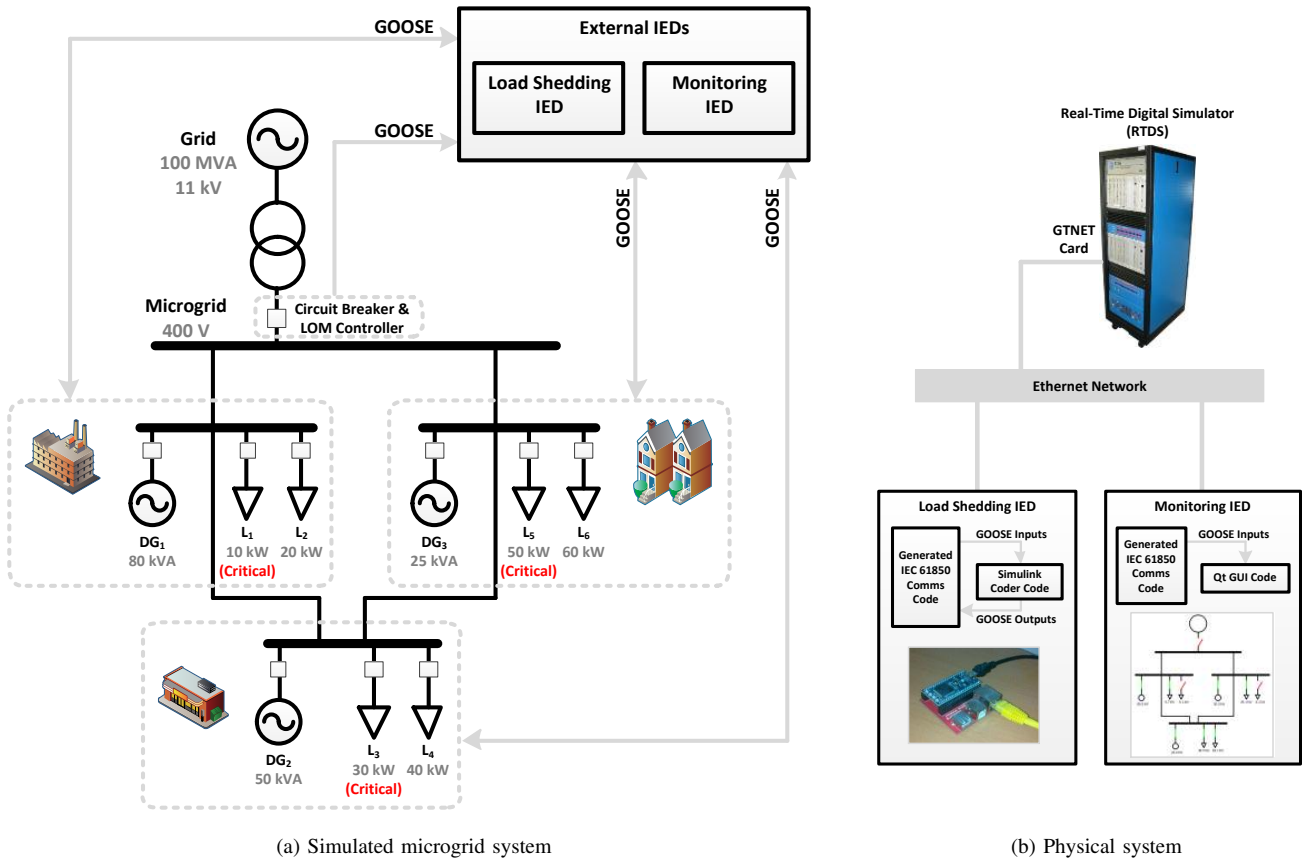


Fig. 6. Microgrid system and prototype IEDs

use, is generated automatically using Simulink Coder [26]. The communications code and the Simulink Coder code have been combined and used to create a rapid-prototype of the load shedding IED, by installing and executing the C code on a microcontroller—an “mbed” [27], with a 96 MHz ARM Cortex-M3 processor—as illustrated in Fig. 6b.

The monitoring IED subscribes to all GOOSE communications to visualize the state of the microgrid power system in real-time. For example, it graphically illustrates which loads have been tripped by the load shedding IED. It uses the Qt graphical user interface framework [28] and executes on a desktop computer.

### E. Results from Real-time Testing

Fig. 7 illustrates the effect of islanded operation of the microgrid, without the load shedding scheme (Fig. 7a), and with (Fig. 7b). Clearly, without load shedding, the total load power exceeds the available generation capacity, and the system frequency falls rapidly. Depending on the local control and protection policy, this will likely trigger under-frequency protection for each DG unit and will cut supply to all loads. For example, in the UK, Engineering Recommendation G59/2 [29] requires that DG is tripped if the frequency drops below 47 Hz (0.94 p.u.) for 0.5 s, as is illustrated in Fig. 7a. With the fast-acting load shedding scheme, however, two low-priority loads,  $L_2$  and  $L_4$ , are tripped—with relatively low impact on the supply of power to the remaining loads. The

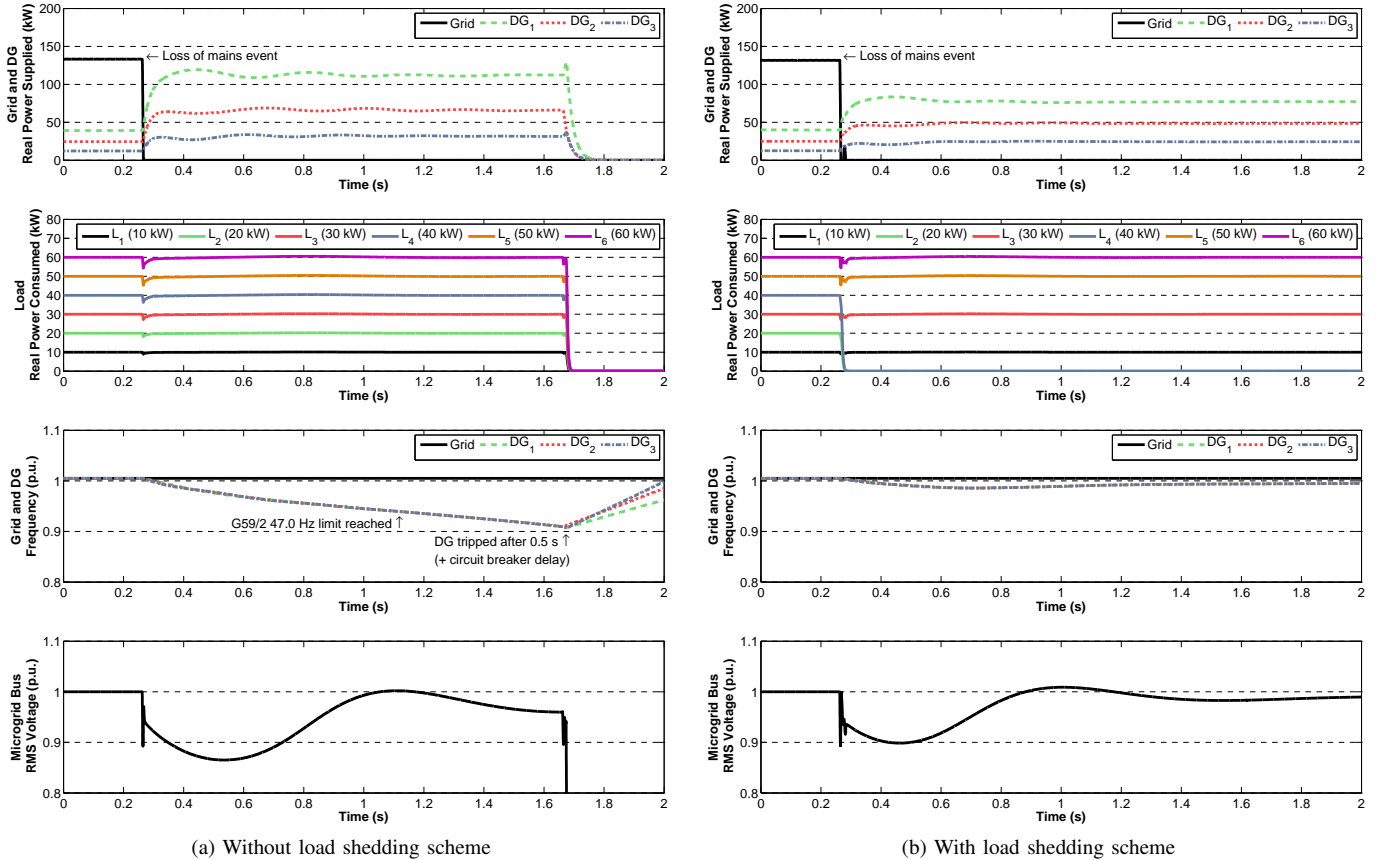


Fig. 7. Effect of fast-acting load shedding scheme

 TABLE I  
 COMPARISON OF DATA SET ENCODING AND DECODING PERFORMANCE

	SV	GOOSE	Fixed-length GOOSE
<b>Packet length</b>	126 bytes	~221 bytes (depending on data values)	260 bytes
<b>Encoding time</b>	24.2 $\mu$ s	111.1 $\mu$ s	48.8 $\mu$ s
<b>Decoding time</b>	18.5 $\mu$ s	44.7 $\mu$ s	45.6 $\mu$ s

load shedding IED is therefore essential to allow switching between grid-connected and islanded operation, whilst keeping the microgrid stable. Note that the communications and load shedding decision occurs during the grid connection circuit breaker opening delay. The load circuit breakers are thereby tripped—but not opened, due to their own operation delay—before the grid connection circuit breaker opens.

#### F. Real-time Performance Analysis

Real-time performance is critical for protection applications. Table I compares the typical performance of the generated code for encoding and decoding SV and GOOSE packets. This has been measured on the 96 MHz microcontroller described in Section IV-D, by toggling a digital output pin before and

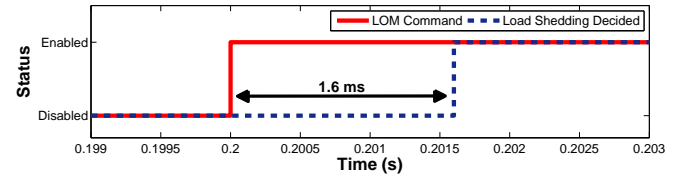


Fig. 8. GOOSE round trip timing for load shedding scheme

after encoding or decoding the packet, and monitoring this with a digital oscilloscope. The same data set, based on the “9-2LE” implementation guideline [30], has been used for both SV and GOOSE. It contains eight 32-bit floating-point values and the corresponding eight Quality values. The 802.1Q header has been included. For SV, one Application Service Data Unit has been used per Application Protocol Data Unit [15] to ensure a fair comparison with GOOSE.

As shown in Table I, GOOSE packet encoding times are typically reduced by approximately 50% by using fixed-length encoding [14]. Generally, there is increased benefit from using fixed-length GOOSE encoding if the data set contains several nested data object or data attribute structures.

Therefore, sub-millisecond transfer times between logical nodes (in different IEDs) can be achieved, which satisfies the 3 ms requirement for Type 1A trip messages (of Performance Class P2/3), as defined in IEC 61850-5 [31]. For the load shedding system demonstrated in Section IV-E, the total GOOSE

round trip, which includes two GOOSE transfer times and all control logic processing, is approximately 1.6 ms. This is illustrated in Fig. 8 by the time of the load trip commands in the RTDS simulation, relative to the command to open the grid connection circuit breaker.

## V. FURTHER APPLICATIONS

### A. Protection Testing

The platform can be used to multicast mock Sampled Value data. The data could be generated in real-time, or “replayed” from memory. In either case, the platform would provide the stimulus for other protection devices (which may be prototypes themselves, and may use the same IEC 61850 software). This does not replace the rigorous approaches adopted by industry for validating protection systems, but it does offer an inexpensive method for testing experimental schemes.

### B. Multi-IED Simulation

The platform proposed in this paper can be used to generate code that implements a particular IED per hardware device, as demonstrated in Section IV, or multiple IEDs on one hardware device [13]. The latter can be used to verify the operation of GOOSE and SV communications for an entire system, without using any networking hardware, hence de-risking the design of a novel protection or control scheme. Later in the system development process, the same software can be executed on the actual target hardware devices.

### C. Combine with Predefined Logical Node Implementations

The software described in this paper, like IEC 61850, does not define the implementation of logical nodes; instead, it provides a platform for implementing logical nodes, without the burden of also implementing the data model and communications code. This process could be extended to offer automatic code generation of a complete IED, if a library of predefined, commonly used logical nodes has been developed.

### D. Communications Simulation

The generated C code could be used to implement software entities which represent IEDs within a communications simulator, such as OMNeT++ [32]. Although OMNeT++ does not model a power system (unless OMNeT++ is embedded into a power system simulation), it could be used to monitor the communications between IEDs to, for example, assess various communications network topologies. The platform described in this paper would ensure that all packets used in the OMNeT++ simulation are valid GOOSE or SV packets.

### E. Potential Applications for EMF with IEC 61850

The platform described in this paper is just one example of taking advantage of the fact that the SCL is a machine-readable model (albeit with the caveats noted in Section III-C). The model-centric approach offered by EMF has several other features which can be used to automate other tasks relating to IEC 61850 [10], [13], such as:

1) *Substation Visualization and Monitoring Tools*: EMF could be used as a platform for visualizing the power system electrical topology—which can be described in an SCD file—along with, for example, the location of IEDs and their communications services. The “coordinates” SCL syntax extension [16] facilitates such applications. Furthermore, the visualization could be linked to real-time data from IEDs, thereby automating the generation of a generic power system monitoring tool. Therefore, in contrast to the monitoring IED presented in Section IV which—apart from the communications code—was implemented manually, EMF offers the potential to generate power system monitoring applications automatically.

2) *SCD Editor*: As described earlier in this paper, EMF automates the process of importing instances of a model, e.g., existing SCD files, but it also provides a framework for creating and editing these model instances. For example, a graphical “tree”-based SCD editor can be generated automatically from the IEC 61850-6 XML Schema. The editor automatically provides functionality such as undo and redo, and enforces the syntactical constraints in the XML Schema, such as allowing only one LNO instance per logical device. More advanced graphical editors are also supported within the framework.

EMF can also be used for implementing changes to the SCL model itself, i.e., changes to the XML Schema, and for automatically generating the resulting schema files.

## VI. CONCLUSIONS

This paper has presented an approach for automatically generating an IEC 61850-compliant implementation of IEDs. The platform is generic and it supports a wide variety of protection and control applications, yet it generates code which is extremely fast due to being tailored for a specific SCD file. It is therefore suitable for embedded platforms, and for protection applications where run-time performance is critical.

A real-time microgrid load shedding and monitoring system has been demonstrated, which highlights the convenience and practicality of the platform for rapid-prototyping protection and control schemes. The platform offers power engineers a simple to use and openly-available method for developing and testing new types of communications-enhanced schemes.

The Eclipse Modeling Framework offers a powerful framework for developing standards-based tools for power system research and operation. EMF understands the SCL—the model used to define the system configuration—and thereby can help automate the generation of any application which is based around the SCL.

## REFERENCES

- [1] A. Timbus, M. Larsson, and C. Yuen, “Active Management of Distributed Energy Resources Using Standardized Communications and Modern Information Technologies,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4029–4037, Oct. 2009.
- [2] H. J. Laaksonen, “Protection Principles for Future Microgrids,” *IEEE Trans. Power Electron.*, vol. 25, no. 12, pp. 2910–2918, Dec. 2010.
- [3] A. Colet-Subirachs, A. Ruiz-Alvarez, O. Gomis-Bellmunt, F. Alvarez-Cuevas-Figueroa, and A. Sudria-Andreu, “Centralized and Distributed Active and Reactive Power Control of a Utility Connected Microgrid Using IEC61850,” *IEEE Systems Journal*, vol. 6, no. 1, pp. 58–67, Mar. 2012.

- [4] M. Yalla, M. Adamiak, A. Apostolov, J. Beatty, S. Borlase, J. Bright, J. Burger, S. Dickson, G. Gresco, W. Hartman, J. Hohn, D. Holstein, A. Kazemi, G. Michael, C. Sufana, J. Tengdin, M. Thompson, and E. Udren, "Application of peer-to-peer communication for protective relaying," *IEEE Trans. Power Del.*, vol. 17, no. 2, pp. 446–451, Apr. 2002.
- [5] C. D. Booth, I. M. Elders, J. D. Schuddebeurs, J. R. McDonald, and S. Loddick, "Power system protection for more and full electric marine systems," *Proceedings of IMarEST - Part B - Journal of Marine Design and Operations*, pp. 37–45, 2008.
- [6] B. Pickett, T. Sidhu, S. Anderson, A. Apostolov, B. Bentert, K. Boers, O. Bolado, P. Carroll, S. Chano, A. Chaudhary, F. Cobelo, K. Cooley, M. Cooper, R. Cornelison, P. Elkin, A. Elneweihi, R. Garcia, K. Gardner, T. Kern, B. Kasztenny, G. Michel, J. Niemira, T. Nissen, S. Charles, D. Ware, and F. Plumptre, "Reducing Outage Durations Through Improved Protection and Auto restoration in Distribution Substations," *IEEE Trans. Power Del.*, vol. 26, no. 3, pp. 1554–1562, Jul. 2011.
- [7] M. Adamiak, A. Apostolov, M. Begovic, C. Henville, K. Martin, G. Michel, A. Phadke, and J. Thorp, "Wide Area Protection-Technology and Infrastructures," *IEEE Trans. Power Del.*, vol. 21, no. 2, pp. 601–609, Apr. 2006.
- [8] K.-P. Brand, "The standard IEC 61850 as prerequisite for intelligent applications in substations," in *IEEE Power and Energy Soc. General Meeting*, vol. 2. IEEE, 2004, pp. 714–718.
- [9] R. Mackiewicz, "Overview of IEC 61850 and Benefits," in *IEEE PES Transmission and Distribution Conference and Exhibition*. IEEE, 2006, pp. 376–383.
- [10] T. Kostic, O. Preiss, and C. Frei, "Understanding and using the IEC 61850: a case for meta-modelling," *Computer Standards & Interfaces*, vol. 27, no. 6, pp. 679–695, Jun. 2005.
- [11] M. Shulim, "Experience With a Standard Protocol in Substation Automation Projects," *EnergyPulse*, 2008. [Online]. Available: [http://www.energypulse.net/centers/article/article\\_print.cfm?a\\_id=1792](http://www.energypulse.net/centers/article/article_print.cfm?a_id=1792)
- [12] S. M. Blair, "rapid61850," 2012. [Online]. Available: <https://github.com/stevenblair/rapid61850>
- [13] S. M. Blair, C. D. Booth, and G. M. Burt, "Architecture for automatically generating an efficient IEC61850-based communications platform for rapid prototyping of protection schemes," in *PAC World Conference*, 2011.
- [14] IEC TC 57, "Communication networks and systems in substations Part 8-1: Specific Communication Service Mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3 (IEC 61850-8-1:2011)," 2011.
- [15] —, "Communication networks and systems in substations Part 9-2: Specific Communication Service Mapping (SCSM) - Sampled values over ISO/IEC 8802-3 (IEC 61850-9-2:2011)," 2011.
- [16] —, "Communication networks and systems for power utility automation - Part 6: Configuration description language for communication in electrical substations related to IEDs (IEC 61850-6:2009)," 2009.
- [17] J. Starck and S. A. Kunsman, "Pushing the limits," *ABB review*, 2010.
- [18] Eclipse, "Eclipse Modeling - EMF - Home," 2011. [Online]. Available: <http://www.eclipse.org/modeling/emf/>
- [19] T. Saxton and H. Falk, "Harmonizing the International Electrotechnical Commission Common Information Model (CIM) and 61850," EPRI, Tech. Rep., 2010.
- [20] A. W. McMorran, R. W. Lincoln, G. A. Taylor, and E. M. Stewart, "Addressing misconceptions about the Common Information Model (CIM)," in *IEEE PES General Meeting*. IEEE, Jul. 2011.
- [21] IEC TC 57, "Communication networks and systems for power utility automation - Part 7-3: Basic communication structure - Common data classes (IEC 61850-7-3:2010)," 2010.
- [22] ITU, "ITU-T X.690, ISO/IEC 8825-1: ASN.1 encoding rules," 2002.
- [23] E. Sortomme, S. S. Venkata, and J. Mitra, "Microgrid Protection Using Communication-Assisted Digital Relays," *IEEE Trans. Power Del.*, vol. 25, no. 4, pp. 2789–2796, Oct. 2010.
- [24] A. H. Kasem Alaboudy, H. H. Zeineldin, and J. Kirtley, "Microgrid Stability Characterization Subsequent to Fault-Triggered Islanding Incidents," *IEEE Trans. Power Del.*, vol. 27, no. 2, pp. 658–669, Apr. 2012.
- [25] RTDS, "Real Time Power System Simulation - RTDS Technologies," 2011. [Online]. Available: <http://www.rtds.com>
- [26] A. J. Roscoe, S. M. Blair, and G. M. Burt, "Benchmarking and optimisation of Simulink code using Real-Time Workshop and Embedded Coder for inverter and microgrid control applications," in *Universities Power Engineering Conference (UPEC)*, 2009, pp. 1–5.
- [27] Mbed, "Rapid Prototyping for Microcontrollers - mbed," 2011. [Online]. Available: <http://mbed.org/>
- [28] Nokia, "Qt - Cross-platform application and UI framework," 2012. [Online]. Available: <http://qt.nokia.com/>
- [29] ENA, "Engineering Recommendation G59 Issue 2 - Recommendations for the Connection of Generating Plant to the Distribution Systems of Licensed Distribution Network Operators," Tech. Rep., 2010.
- [30] UCA International Users Group, "Implementation Guideline for Digital Interface to Instrument Transformers Using IEC 61850-9-2," Tech. Rep., 2004.
- [31] IEC TC 57, "Communication networks and systems in substations - Part 5: Communication requirements for functions and device models," 2003.
- [32] O. Community, "OMNeT++ Network Simulation Framework," 2011. [Online]. Available: <http://www.omnetpp.org/>



**Steven M. Blair** (S'09) received the M.Eng. degree (with distinction) in computer and electronic systems in 2008 from the University of Strathclyde, Glasgow, U.K., where he is currently pursuing the Ph.D. degree in electrical engineering.

His research interests include power system protection, fault current limitation, marine electrical systems, communications, and real-time simulation.



**Federico Coffele** received the M.Eng. degree in electrical engineering from the University of Padova, Italy, in 2007, and the Ph.D. degree in electrical and electronic engineering from the University of Strathclyde, Glasgow, U.K., in 2012.

He is presently the Research and Development Manager of the Power Networks Demonstration Centre at the University of Strathclyde, Glasgow, U.K. His main research interests include power system modelling and real time simulation, power system protection, and power system control.



**Campbell D. Booth** received the B.Eng. and Ph.D. degrees in electrical and electronic engineering from the University of Strathclyde, Glasgow, U.K., in 1991 and 1996, respectively.

He is currently a Senior Lecturer with the Institute for Energy and Environment, Department of Electronic and Electrical Engineering, University of Strathclyde. His research interests include power system protection; plant condition monitoring and intelligent asset management; applications of intelligent system techniques to power system monitoring, protection, and control; knowledge management; and decision support systems.



**Graeme M. Burt** (M'95) received the B.Eng. and Ph.D. degrees in electrical and electronic engineering from the University of Strathclyde, Glasgow, U.K., in 1988 and 1992, respectively.

He is currently the Director of the Institute for Energy and Environment, University of Strathclyde, where he also directs the University Technology Center in Electrical Power Systems sponsored by the Rolls-Royce Group plc. He is a professor of electrical power engineering, and has particular research interests in the areas of: integration of distributed

generation; power system modelling and real-time simulation; power system protection and control; microgrids; and more-electric systems.