Rapid-prototyping protection schemes with IEC 61850

Steven Blair, Campbell Booth, Graeme Burt

University of Strathclyde
30th June 2011, Dublin, Ireland
Contents

• Introduction
  – implementing protection schemes with communications
  – automatically providing IEC 61850 Sampled Values and GOOSE

• Using the Eclipse Modeling Framework

• Demo

• Features and benefits

• Further applications

• Conclusions
Introduction

• Rapid-prototype novel protection systems – with **communications** requirements
Introduction

• Rapid-prototype novel protection systems – with **communications** requirements
Introduction

• Rapid-prototype novel protection systems
  – with **communications** requirements
• Substation Configuration Language (SCL)
  – describes IEDs and their comms
Introduction

• Rapid-prototype novel protection systems – with **communications** requirements
• Substation Configuration Language (SCL) – describes IEDs and their comms
• Transform XML description to C implementation:

![Diagram showing the transformation process from SCD.xml to C data structures, SV and GSE encoding/decoding functions, and platform-specific C files.](image)
Eclipse Modeling Framework (EMF)

• Manipulate **structured models**
• The SCL is a structured model
  – defined by XML Schema
  – can import into EMF
• Code generation tools
Eclipse Modeling Framework (EMF)
Eclipse Modeling Framework (EMF)

- Manipulate **structured models**
- The SCL is a structured model
  - defined by XML Schema
  - can import into EMF
- Code generation tools
Transformation process

Generate Model
(using EMF)

IEC 61850-6
XML Schema

Class model of IEC 61850-6, and helper classes

Import XML

Model instance (Java object hierarchy)

Transform

SCD.xml
(substation description)

C implementation files
Transformation process (2)

• Hierarchical data-type mapping:

```xml
<DAType id="simpleVector">
  <BDA name="mag" bType="FLOAT32/>
  <BDA name="ang" bType="FLOAT32"/>
</DAType>

<DOType id="simpleCMV" cdc="CMV">
  <DA name="voltage" fc="MX" bType="Struct" type="simpleVector"/>
  <DA name="current" fc="MX" bType="Struct" type="simpleVector"/>
</DOType>
```

```c
#define CTYe_FLOAT32 float

struct simpleVector {
  CTYe_FLOAT32 mag;
  CTYe_FLOAT32 ang;
};

struct simpleCMV {
  struct simpleVector voltage;
  struct simpleVector current;
};
```

• C code reads or writes full Ethernet packets
Demo

Microcontroller (IED 1)

Three-phase Sampled Values, 16 samples per cycle

GOOSE events

PC, with Wireshark (IED 2)
Features and benefits

- **GOOSE and Sampled Values**
- Flexible: *any* valid SCD file
- Any CPU architecture/endianness
- Efficient run-time C/C++ code
- Multiple Sampled Value ASDUs per ADPU
Further applications

• Simple, inexpensive **Merging Unit**
• Link with simulation tools
  – pure comms simulation (OMNeT++)
  – power system simulation (RTDS)
• EMF applications
  – SCD editing and validation
  – Visualisation and monitoring (HMI)
  – Model conversions
Conclusions

• Lowers barrier for rapid-prototyping, with IEC 61850 communications
  – flexible and practical
  – low-cost embedded devices
  – “properly” centred around the SCL/SCD
  – focus on getting stuff done; not comms

• One example of leveraging EMF for IEC 61850

steven.blair@eee.strath.ac.uk