Project ADVANCE
Grant Agreement 287563
"Advanced Design and Verification Environment for Cyber-physical System Engineering"

ADVANCE Deliverable D.2.3

Technical Report on Assessment of Methods

Public Document

April 28th, 2014

http://www.advance-ict.eu

TECHNICAL REPORT ON ASSESSMENT OF METHODS

ADVANCE

Approval			
Name	**Function**	**Signature**	**Date**
John Colley	Project Co-Ordinator		
Luke Walsh	Project Manager		

Authors and Contributors			
Name	**Contact**	**Description**	**Date**
Brett Bicknell	bbicknell@critical-software.co.uk	Author	28/04/2014
Karim Kanso	kkanso@critical-software.co.uk	Author	28/04/2014
José Reis	jreis@critical-software.co.uk	Contributor	25/04/2014
Daniel Mcleod	Daniel.Mcleod@selex-es.com	Contributor	25/04/2014
Michael Butler	mjb@ecs.soton.ac.uk	Contributor	24/04/2014

Access List

Internal Access

Project Team, Engineering Department

External Access

Public Document

Revision History			
Issue	**Date**	**Description**	**Author**
0.1	02-12-2013	Initial draft	Brett Bicknell
1	19-12-2013	Issue 1 released to EU commission	Brett Bicknell, Karim Kanso
1.1	17-04-2014	Updated in response to reviewers comments and added progress update. Release for consortium review	Brett Bicknell, Karim Kanso
2	28-04-2014	Issue 2 released to EU commission	Brett Bicknell, Karim Kanso

CRITICAL SOFTWARE TECHNOLOGIES LTD
4 BENHAM ROAD
SOUTHAMPTON SCIENCE PARK – CHILWORTH
SOUTHAMPTON - SO16 7QJ – UNITED KINGDOM

CRITICAL SOFTWARE, S.A.
PARQUE INDUSTRIAL DE TAVEIRO, LOTE 48,
3045-504 COIMBRA
PORTUGAL

TABLE OF CONTENTS

CRITICAL SOFTWARE TECHNOLOGIES LTD
4 BENHAM ROAD
SOUTHAMPTON SCIENCE PARK – CHILWORTH
SOUTHAMPTON - SO16 7QJ – UNITED KINGDOM

CRITICAL SOFTWARE, S.A.
PARQUE INDUSTRIAL DE TAVEIRO, LOTE 48,
3045-504 COIMBRA
PORTUGAL

1. Executive Summary

This document is an update of D2.3 (issued December 2013) to address concerns raised by the European Commission (EC) during their official review in April 2014. A detailed, three phase plan is described in Section 6, of the effort that has been committed by the ADVANCE consortium to Work Package (WP) 2, totalling 24 person months in all.

To aid with clarity and understanding of the work being performed under WP2 a detailed description (see Annex B) has been added to this document covering phase I. It is the Consortium's belief that the work illustrated in this annex indicates that WP2 is on track to meet the objectives of the project.

The overall objective of WP2 is to demonstrate the advantage of using the methods of the ADVANCE project in the smart grid domain:

- First, the ADVANCE methods and tools should ensure that the performance of the voltage control algorithm is systematically validated in order that it provides smooth and stable voltage control within regulator imposed constraints (see section B.2.6).

- Second, the ADVANCE methods and tools should ensure the safety of the algorithm by proving that the controller does not issue unsafe commands (see Section B.2). Using the STPA method, safety constraints are identified revolving around how and when the algorithm sets a new target voltage for the tap changer. These are modelled in Event-B at the system level and verified in the presence of failures. Formal refinement and decomposition are used to represent the system at the component level, and multi-simulation of the discrete and continuous components allows automated testing with functional coverage analysis of the voltage control algorithm implementation. Event-B verification is used to ensure that the safety properties, as represented by invariants, are preserved by the system design. Multi-simulation is used to ensure that the Event-B model is a valid model of the voltage controller.

The case study is defined to ensure that the overall DoW objectives can be met:

- The case study is modelled to a sufficient level of detail to ensure that the ADVANCE toolset can be used to verify the voltage controller algorithm, providing automatic generation of the tests needed to ensure full coverage of the voltage controller algorithm. It is planned that a realistic period is simulated using real data from the field to ensure that the system can be verified over realistic scenarios.

- Safety analysis begins with a system-level Event-B model, where the system-level safety constraints are represented as Event-B invariants. Formal, Event-B refinement and decomposition are used to separate the discrete and continuous components and enable co-simulation. The discrete part is then further refined to define the model of the voltage controller and a behavioural model of the sensors and communications, which models communication failures in an abstract, non-deterministic manner.

Since the submission of this deliverable in December 2013 considerable progress has been made in WP2. The below list provides a summary of the achievements made:

1. Requirements modelling and consistency analysis completed and validated with Selex ES.

2. STPA safety analysis has been completed and a hazard mitigation strategy identified. This led to safety properties being clearly identified.

3. The Tap Changer algorithm has been modelled, and verification of its safety properties has identified several issues with algorithm design, as reported in section B.2.6. These issues with the algorithm design were uncovered through attempting to verify Event-B invariants and the results were fed back to Selex ES.

4. An abstract model of the communications network has been defined for the interface between the Substation Installation Unit (SIU) and the Tap Changer, including the level of detail around the protocols and topology that is required for the preliminary verification.

5. The discrete/continuous model refinement and decomposition strategy has been defined, and the FMU components have been identified and generated.

6. The preliminary continuous models of the power grid, user demand and the tap changer have been completed.

7. The first FMI multi-simulations have been performed and evaluated in Rodin. The results show that the algorithm behaves as expected in the scenarios that were run, and the results assisted in verifying the multi-simulation framework.

8. The UML-B state machines, Theory Plug-in and BMotion Studio tools have been used and evaluated during the recent development.

The key activities to be performed in the next cycle are:

1. Integration of the communications model with multi-simulation. Introducing further details around:

   - The level of network traffic over each of the potential links.
   - End-to-end delays encountered through packet transmission and routing.
   - High-level changes to routes by the routing algorithms.
   - The reaction of the network to the loss of communication from SIUs.

   These represent the properties of the network that Selex ES is interested is exploring when verifying the algorithm behaviour. The communications elements do not need to be modelled in great detail to achieve the aims of the case study; what is important is the ability to analyse the effect of communications delay or loss. This is explained in Section B.3. The main reason for including this in the critical path is because there are risks associated with communication outages but their impact and the conditions that lead to the risk occurrence are difficult to specify. Using traditional testing techniques it takes a long time to create scenarios that lead to communications outages and the pre-conditions that trigger those scenarios may not occur during testing. Also even in expensive long-duration real world testing where undesirable conditions are noticed, it may be impossible to determine the events which lead to the undesirable conditions unlike using the ADVANCE toolset.

2. Refinement of the continuous model of PV power generation and distribution.

3. Ascertain simulation coverage using ADVANCE automatic test case generation.

Section 4 and 5 have been updated to address comments raised, but due to the detail provided in Annex B they have not been modified significantly as they report on the work developed previously (January 2013 – December 2013). Section 6 provides a revised plan of work to address concerns raised by the reviewers and to focus on the critical path to ensure delivery of the WP2 objectives.

2. Introduction

2.1 Objective

This document provides a report on the assessment of the ADVANCE methods during the work completed so far in the smart energy case study. The document is composed of a description of the revised smart energy case study, examples of models, and conclusions drawn from this stage of work. The case study is part of Work Package 2 (WP2) of ADVANCE.

2.2 Audience

Those involved or interested in the case study, including the ADVANCE consortium.

2.3 Definitions and acronyms

Table 1 presents the list of acronyms used throughout the present document.

Acronyms	Description
AC	Alternating Current
AD	Applicable Document
ADVANCE	Advanced Design and Verification Environment for Cyber-physical System Engineering
CPU	Central Processing Unit
CREST	Centre for Renewable Energy Systems Technology
CS	Co-Simulation
CSWT	Critical Software Technologies, Ltd
DNO	Distribution Network Operator
EMF	Electro Motive Force
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
GOOSE	Generic Object Oriented Substation Events
JFMI	Java FMI
LV	Low Voltage
ME	Model Exchange
MV	Medium Voltage
OLTC	On Load Tap Changer
OMC	Open Modelica Compiler
PM	Project Management
PV	Photo Voltaic
RD	Reference Document
RMS	Root Mean Square
SCADA	Supervisory Control And Data Acquisition
SDK	Software Development Kit
SIU	Sensor Interface Unit
UML	Unified Modelling Language
UDUS	University of Dusseldorf
UOS	University of Southampton
WP	Work Package

Table 1: Table of acronyms

2.4 Document structure

Section 1 (Executive Summary) clarifies the scope of the work, updates made, achievements and future work.

Section 2 (Introduction) introduces the document.

Section 3 (Documents) presents the list of applicable documents.

Section 4 (Reviewed Case Study) provides details on the review and amendment of the case study and description of work.

Section 5 (Report on progress) details the progress made for the period January 2013 – December 2013, including examples of models and techniques, alongside any changes that have been required to the tools as a result of the work.

Section 6 (Revised Plan of Work) presents the work plan for the case study in the future, including how the case study will provide a suitable assessment for each of the tools in the ADVANCE framework.

Annex A provides a description of the technical architecture of the solution developed by Selex ES.

Annex B reports on the work undertaken since December 2013 through to April 2014.

3. Documents

This section presents the list applicable and reference documents.

3.1 Applicable documents

Table 2 presents the list of the documents that are applicable to the present document. A document is considered applicable if it contains provisions that through reference in this document incorporate additional provisions to this document [ECSS-P-001B].

Applicable document		Document number	Issue
[AD-1]	Smart Grid Case Study Definition, Critical Software Technologies, December 19th 2011	CSWT-EUADV-2011-SPC-00621	1
[AD-2]	Proof of Concept Application in Smart Energy Domain, Critical Software Technologies, September 27th 2012	CSWT-EUADV-2012-TNR-00180	1
[AD-3]	Shared Event Composition/Decomposition in Event-B, *University of Southampton*, November 2010	Lecture Notes in Computer Science, 2012, Volume 6957/2012, 122-141, DOI: 10.1007/978-3-642-25271-6_7	
[AD-4]	*Integrated high-resolution modelling of domestic electricity demand and low voltage electricity distribution networks*, PhD Thesis, Ian Richardson, Loughborough University, UK, 2010	N/A	N/A
[AD-5]	Functional Mock-up Interface for Model Exchange, Modelisar, 2010	N/A	1
[AD-6]	Functional Mock-up Interface for Co-Simulation, Modelisar, 2010	N/A	1
[AD-7]	*Voltage Regulator TAPCON® 230 expert: Operating Instructions*, Maschinenfabrik Reinhausen, 2014	3552133/00	N/A
[AD-8]	*Voltage characteristics of electricity supplied by public electricity networks*, BSi, 2010	BS EN 50160	2010
[AD-9]	*Families and Households*, Office for National Statistics, 2013 *Available: http://www.ons.gov.uk/ons/rel/family-demography/families-and-households/2013/rft-tables.xls*	NA	NA

Table 2: Applicable documents

4. Reviewed Case Study

4.1 Summary of work completed from D2.2

The case study for WP2, provided by Selex ES, originally focused on a Sensor Interface Unit (SIU), which can be installed on an electricity substation and used to wirelessly provide sensor readings from the feeders. A phased modelling approach was followed during the proof of concept, modelling the SIU, network and data centre entity to which each of the SIUs report back to. The focus was placed mainly on exploring the events associated with memory management, and the validation of the data exchange between the different entities.

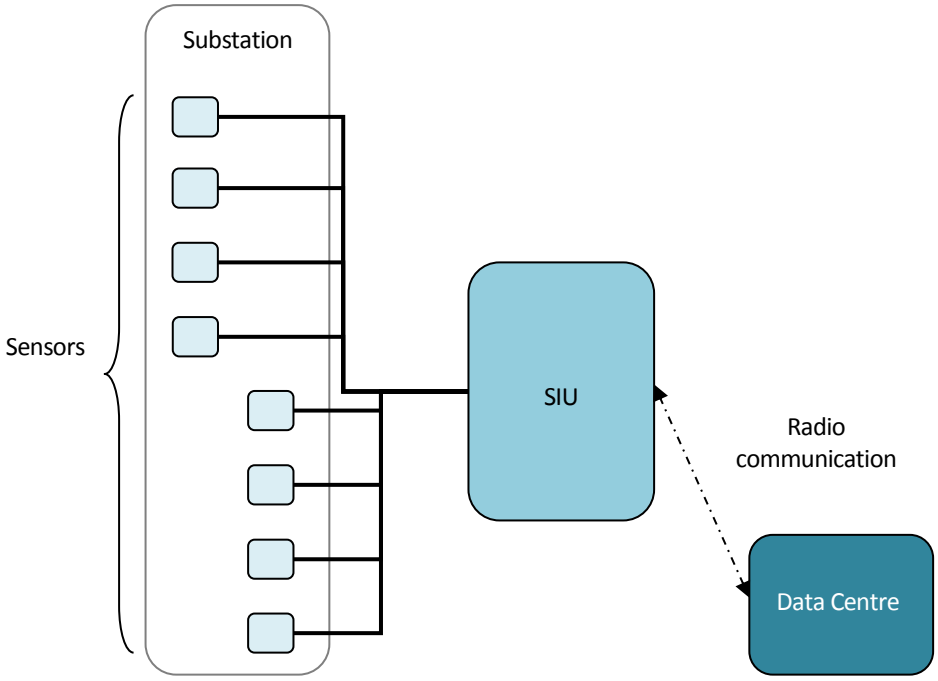


Figure 1 : A diagrammatic representation of the SIU system

Following the proof of concept, the case study and the feedback provided by the European Commission reviewers the scope of the case study has evolved to focus on the aspects that are most critical to Selex ES. Both the evolution in scope of the case study, and the work completed since the proof of concept that is aligned with this change, is reported in this deliverable.

4.2 Renewed case study scope

In the proof of concept phase of ADVANCE, WP2 focused on analysing the SIU and communications between the SIU and the Data Centre. In the second phase, the case study has focused on analysing a voltage control mechanism that relies on a distributed network of sensors. It concerns a low voltage network and the method used to regulate the power on the network. The solution in question utilises multiple SIUs for the sensor network. The generic SIU and communications models produced in the proof of concept are being reused and further developed in the current phase and being combined with newly developed models of the voltage control.

Selex ES, the provider of the case study, have been contracted to identify a solution for automating the control of the voltage on low voltage networks. Currently the voltage on low voltage networks is controlled manually, and foremost is still reliant on the concept that energy flows in one particular direction. The issues faced by Selex ES when implementing this solution include:

- Patterns of usage of the network are continuing to change more frequently; making it more difficult to predict the power required at any particular time. This means that adjustments on the low voltage network may have to be done at any time during the day or night. This is a challenge considering that a manual approach to changing the voltage – i.e. sending an engineer to the substation to make a physical change – is expensive and not a practical solution for dynamic control.
- Due to the increased use of distributed micro-generation solutions such as photovoltaic cells or wind turbines, energy flows in different directions within the energy network. This can also change dynamically depending on environmental factors. Traditional methods of planning and voltage control are no longer reliable at the low voltage level because there are now several aspects that could change the voltage on the network between the consumer and the generators.
- Despite the lack of a network usage pattern and because of details of the bidirectional flow of energy, the owner of the network needs to keep voltage levels within regulatory extremes. It is also important to be able to control the voltage as it can reduce consumption and system energy losses (technical losses). Distributed generation increases voltages towards the end of the feeder (and in the case of photovoltaic generation, only during the day), while increasing demand from heat pumps and electric vehicles decreases voltage towards the end of the feeders (see Figure 2).

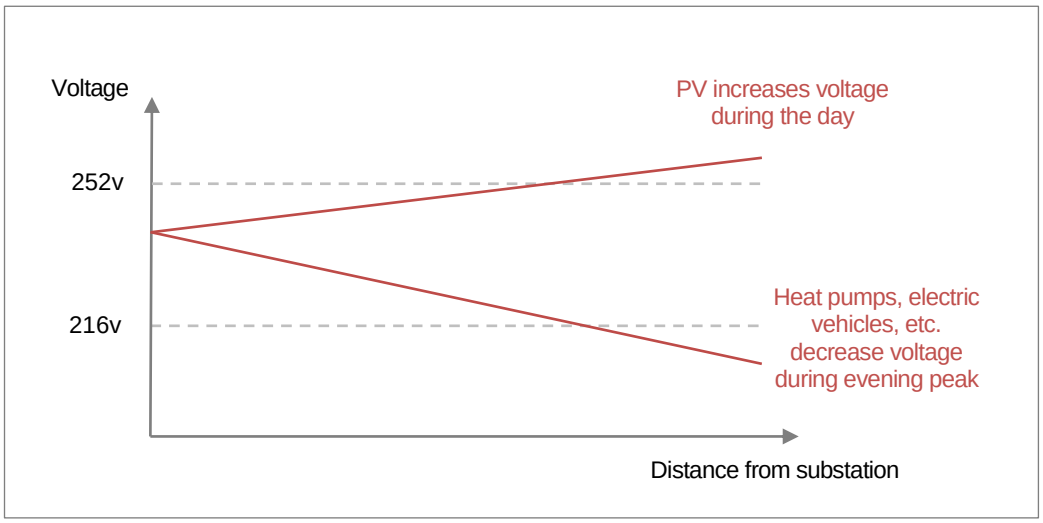


Figure 2: Exceeding voltage limits

The solution is aimed towards increasing the level of automation in the network. The sub-system that achieves this is the combination of a control system which hosts an algorithm which determines the optimum voltage set point for the secondary substation. This optimum voltage set point is used to control an 'On Load Tap Change' transformer at the secondary substation (top of the low voltage network). A number of SIUs are deployed to monitor the voltage at various stages in the network. These are fitted at the Mid Points (half way) and

End Points (at the end) of each feeder connected to the secondary substation. A control system is fitted in the substation, which hosts the voltage control algorithm. The SIUs provide reports detailing measurements of voltage and current from their locations along the feeders to the control system. The voltage control algorithm uses this information to automatically control the tap changer.

An additional issue which needs to be considered during the implementation of the solution is that there are a limited number of changes that can be made within the tap changers lifetime. Therefore the algorithm must not only regulate the voltage so that the levels on the network are always within limits and minimise the amount of power waste, but also consider the number of tap changes made in order to maximise the lifetime of the tap changer. A diagram depicting the solution is shown in Figure 3, and a deeper, technical overview is in Annex B.

Critical Software is using the framework developed in ADVANCE to support Selex ES in the early validation of the solution, system architecture and assumptions prior to actual implementation. This will include an assessment of the architecture and protocols that have been proposed, and the identification of any counterexamples where the following system properties are violated:

- the controller never issues an unsafe command which lowers the voltage when it is already too low, and
- the controller never issues an unsafe command which raises the voltage when it is already too high, and
- the controller avoids unnecessary tap changes.

This early validation is critical for Selex ES as it will provide the means to increase the confidence on the solution before it is implemented on real electricity networks involving actual customers, reducing engineering costs by identifying issues early in the life cycle. Selex ES has a particular interest in this methodology as they have not used it before, hence this is seen by Selex ES as an innovative approach for the systems engineering of smart grids.

The work within WP2 has refocused to be in line with this change in scope. A methodology for developing and integrating models of communication protocols has been produced, to allow for the protocols chosen for possible implementation in the system by Selex ES to be modelled and assessed in a dynamic fashion. There has been development of continuous models of power generation and end-user demand, which is being used as the input for simulations and verification of the low voltage network and tap changer solution. This work is detailed in the following sections.

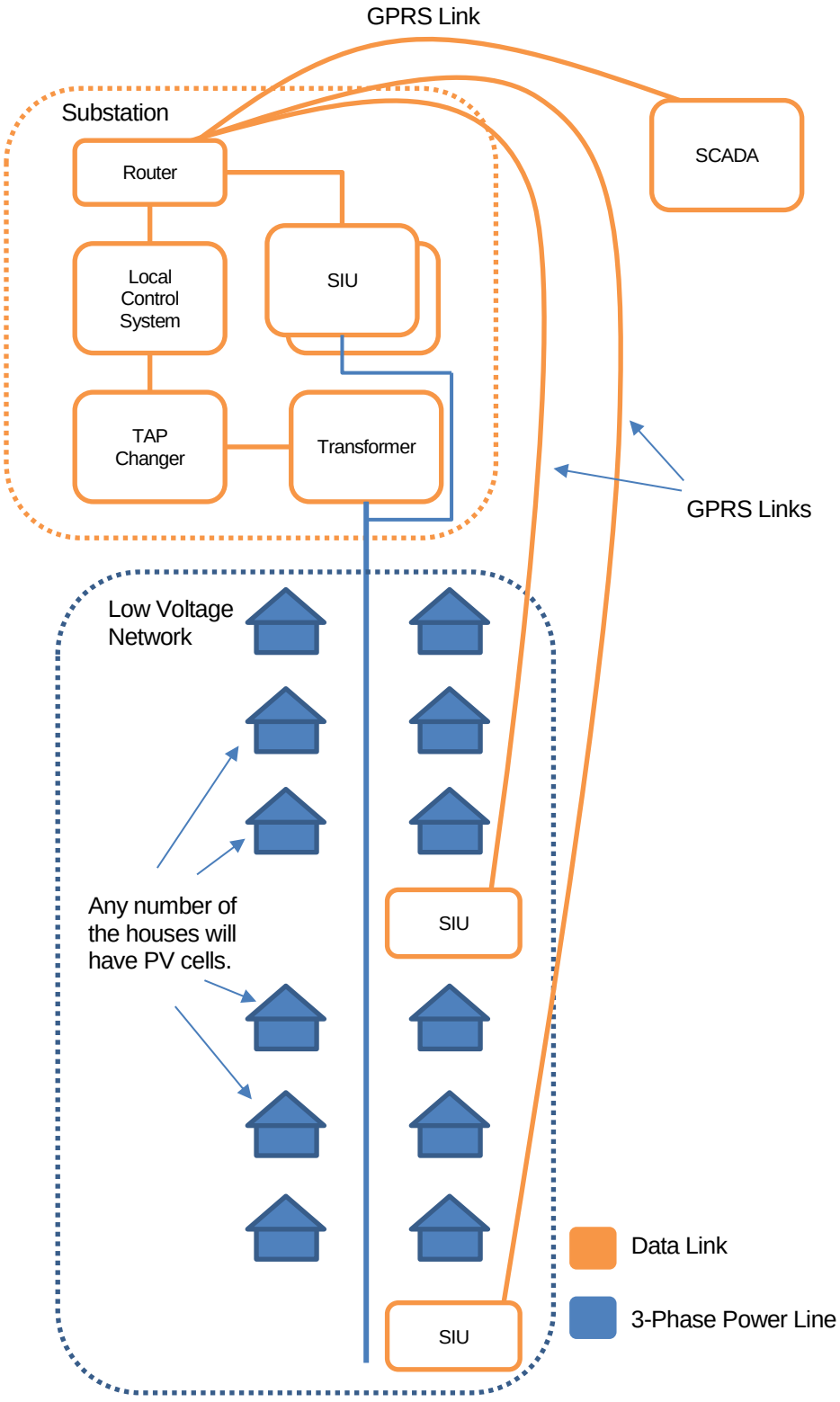


Figure 3: Selex ES Scenario Architecture

5. Report on progress (January 2013-December 2013)

This section reports on the progress made during the period of January 2013 to December 2013.

5.1 Modelling communication protocols

During the work on the revised case study, it is necessary to model and assess several different communication protocols. A generic protocol model and the methodology behind creating the models of these protocols have been put in place and have undergone testing. The results and experiences are detailed in this section.

5.1.1 Modelling strategy

A generic protocol model was initially produced to align with the notion of reusability through refinement. As indicated in Figure 4, the idea is that this generic protocol model is used as the starting point to produce models of the specific protocols of interest for the system; each specific protocol model is a refinement of the generic protocol model. This not only saves effort by promoting reusability, but also ensures that each specific protocol model aligns to the basic rules and behaviour modelled and proven in the generic protocol model.

The specific protocol model is used as the starting point for the system model, and composed with other models – and protocols – to represent the unified system. Thus it is possible to create a standard 'library' of protocol models which can be maintained and reused during the modelling of any system.

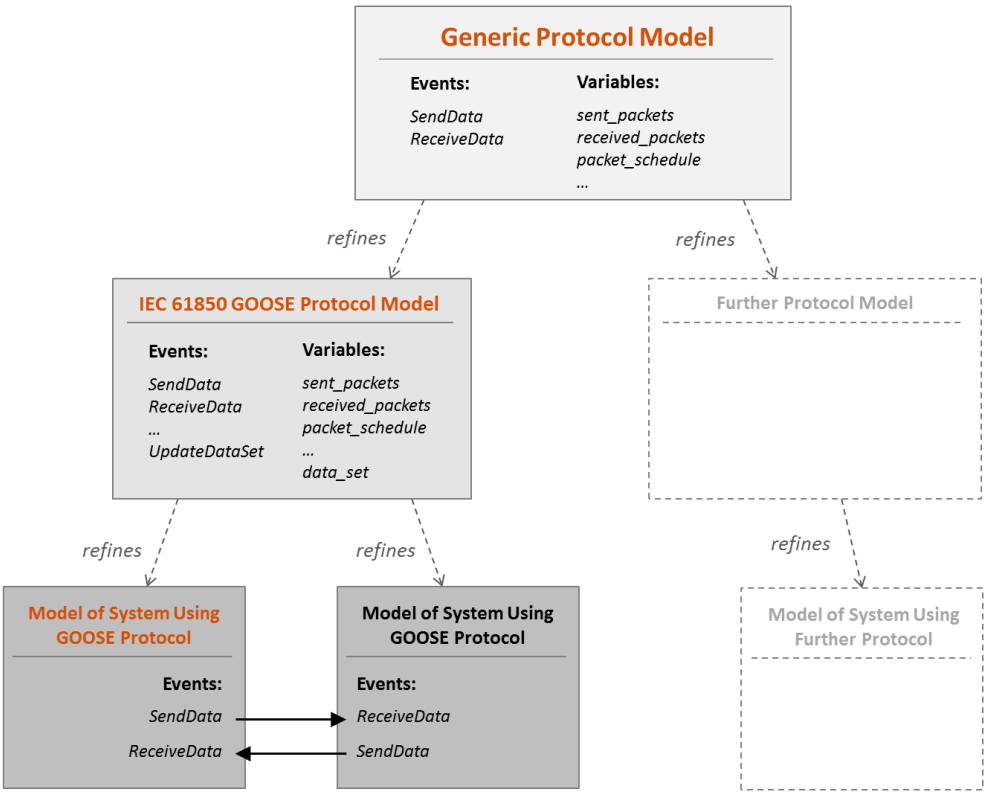


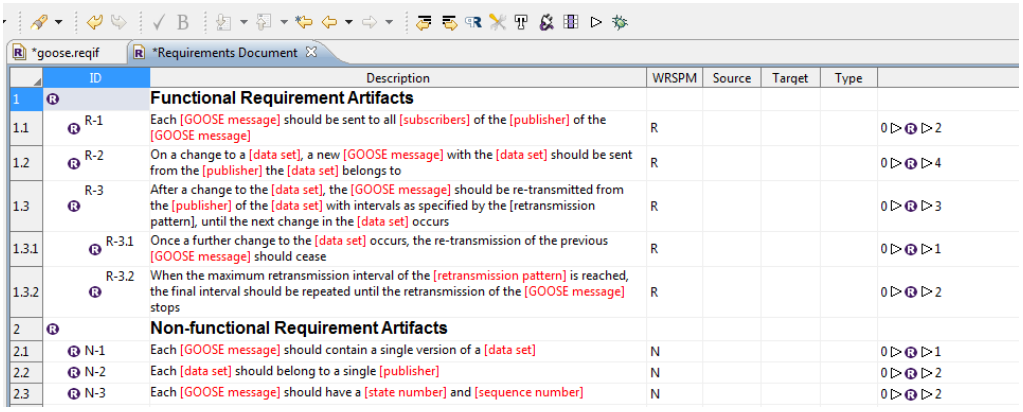
Figure 4: Refinement pattern and reusability implemented in protocol model

In Figure 4 the items with orange text represent the models that have been produced so far as part of the initial feasibility test. The dotted items represent future models that will be

produced via the same technique, i.e. the protocols that are to be assessed during the revised case study in order to match the communications solution being adopted by Selex ES. See Annex B for information about the modelled protocols.

It is recognised that the IEC 61850 GOOSE protocol will likely not be used in the solution represented in the revised case study. The protocol was modelled during this stage of work to validate the methodology and toolset using a realistic example that is relevant to the smart grid domain; indeed tool changes were required for this to be successful (see Section 5.1.2). With these tool changes in place, and the methodology tested, it is now possible to model the relevant protocols for the revised case study effectively using the same technique. Regardless of the protocol chosen, the generic model and modelling strategy is valid and reusable. See Annex B for more information.

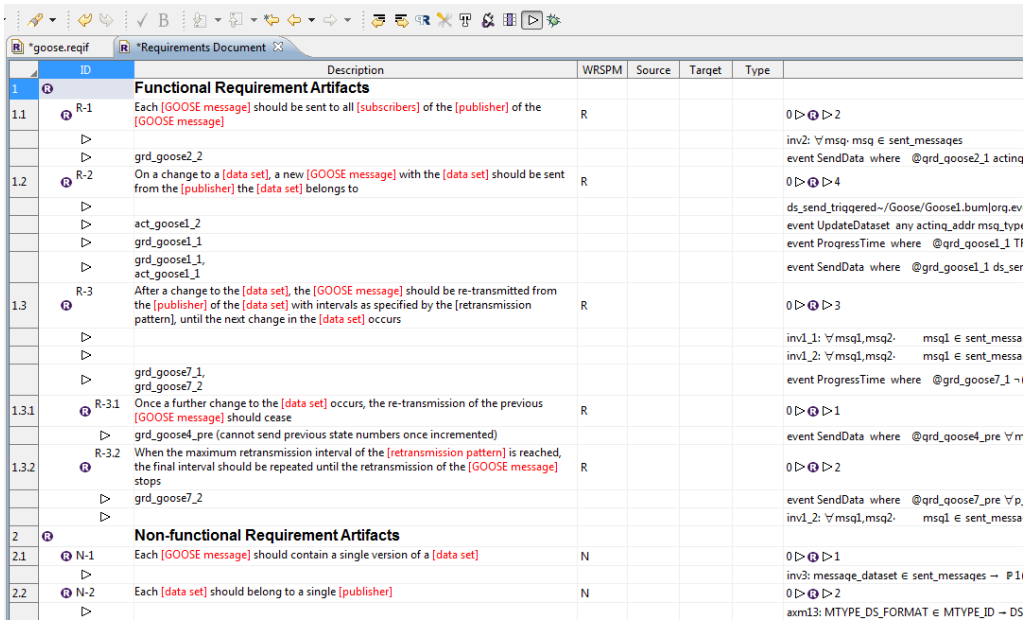
The requirements of IEC 61850 GOOSE were traced to the model using ProR; an excerpt of this is shown in Figure 5 below. The traceability of the requirements to the model is indicated in the column on the right hand side; the same requirements with the trace elements visible are displayed in Figure 6. A simple example of a system using this protocol was also produced by refining the GOOSE model, corresponding to the lowest level in Figure 4 above.



*goose.reqif *Requirements Document

	ID	Description	WRSPM	Source	Target	Type	
1		**Functional Requirement Artifacts**					
1.1	R-1	Each [GOOSE message] should be sent to all [subscribers] of the [publisher] of the [GOOSE message]	R				0 ▷ R ▷ 2
1.2	R-2	On a change to a [data set], a new [GOOSE message] with the [data set] should be sent from the [publisher] the [data set] belongs to	R				0 ▷ R ▷ 4
1.3	R-3	After a change to the [data set], the [GOOSE message] should be re-transmitted from the [publisher] of the [data set] with intervals as specified by the [retransmission pattern], until the next change in the [data set] occurs	R				0 ▷ R ▷ 3
1.3.1	R-3.1	Once a further change to the [data set] occurs, the re-transmission of the previous [GOOSE message] should cease	R				0 ▷ R ▷ 1
1.3.2	R-3.2	When the maximum retransmission interval of the [retransmission pattern] is reached, the final interval should be repeated until the retransmission of the [GOOSE message] stops	R				0 ▷ R ▷ 2
2		**Non-functional Requirement Artifacts**					
2.1	N-1	Each [GOOSE message] should contain a single version of a [data set]	N				0 ▷ R ▷ 1
2.2	N-2	Each [data set] should belong to a single [publisher]	N				0 ▷ R ▷ 2
2.3	N-3	Each [GOOSE message] should have a [state number] and [sequence number]	N				0 ▷ R ▷ 2

Figure 5: Requirements from IEC 61850 in ProR



*goose.reqif *Requirements Document

	ID	Description	WRSPM	Source	Target	Type	
1		**Functional Requirement Artifacts**					
1.1	R-1	Each [GOOSE message] should be sent to all [subscribers] of the [publisher] of the [GOOSE message]	R				0 ▷ R ▷ 2
	▷						inv2: ∀ msq· msq ∈ sent_messages
	▷	grd_goose2_2					event SendData where @grd_goose2_1 actinq
1.2	R-2	On a change to a [data set], a new [GOOSE message] with the [data set] should be sent from the [publisher] the [data set] belongs to	R				0 ▷ R ▷ 4
	▷						ds_send_triqqered~/Goose/Goose1.bum|orq.ev
	▷	act_goose1_2					event UpdateDataset any actinq_addr msq_typ
	▷	grd_goose1_1					event ProgressTime where @grd_goose1_1 T
	▷	grd_goose1_1, act_goose1_1					event SendData where @grd_goose1_1 ds_ser
1.3	R-3	After a change to the [data set], the [GOOSE message] should be re-transmitted from the [publisher] of the [data set] with intervals as specified by the [retransmission pattern], until the next change in the [data set] occurs	R				0 ▷ R ▷ 3
	▷						inv1_1: ∀ msq1,msq2· msq1 ∈ sent_messa
	▷						inv1_2: ∀ msq1,msq2· msq1 ∈ sent_messa
	▷	grd_goose7_1, grd_goose7_2					event ProgressTime where @grd_goose7_1 ¬
1.3.1	R-3.1	Once a further change to the [data set] occurs, the re-transmission of the previous [GOOSE message] should cease	R				0 ▷ R ▷ 1
	▷	grd_goose4_pre (cannot send previous state numbers once incremented)					event SendData where @qrd_goose4_pre ∀ m
1.3.2	R-3.2	When the maximum retransmission interval of the [retransmission pattern] is reached, the final interval should be repeated until the retransmission of the [GOOSE message] stops	R				0 ▷ R ▷ 2
	▷	grd_goose7_2					event SendData where @qrd_goose7_pre ∀ p,
	▷						inv1_2: ∀ msq1,msq2· msq1 ∈ sent_messa
2		**Non-functional Requirement Artifacts**					
2.1	N-1	Each [GOOSE message] should contain a single version of a [data set]	N				0 ▷ R ▷ 1
	▷						inv3: messaqe_dataset ∈ sent_messages → ℙ 1
2.2	N-2	Each [data set] should belong to a single [publisher]	N				0 ▷ R ▷ 2
	▷						axm13: MTYPE_DS_FORMAT ∈ MTYPE_ID → DS

Figure 6: Requirements traceability to Event-B elements in ProR

5.1.2 Required tool changes

From creating the generic protocol model and the example of a specific protocol implementation, several changes were required and requested to the tools in the ADVANCE framework:

- **Decomposition plug-in:**

  The decomposition plug-in was used as a means to refine models between different Event-B projects; without this it is only possible to refine models within the same Event-B project. It was deemed necessary to be able to refine between projects to achieve the approach described above; as it allows the generic protocol model and each specific protocol model to be contained in their own Event-B projects. Not only is this more intuitive – as each model and the proofs can be worked on separately and kept self-contained – but it is also the only practical way to re-use the generic protocol model; otherwise every specific protocol model has to be contained in the same Event-B project, which would quickly become unmanageable. To achieve this, it is required to refine an Event-B machine which exists in a separate project; this is not possible using the standard Rodin installation, as Event-B machines can only refine machines within the same project. The decomposition plug-in was identified as a solution to this. It is important to realise that the process described in this section is not strictly decomposition – as there is only one decomposed element – rather it is the propagation of a refinement chain to other Event-B projects, so that the refinement chain can be reused as the starting point for Event-B models in multiple instances.

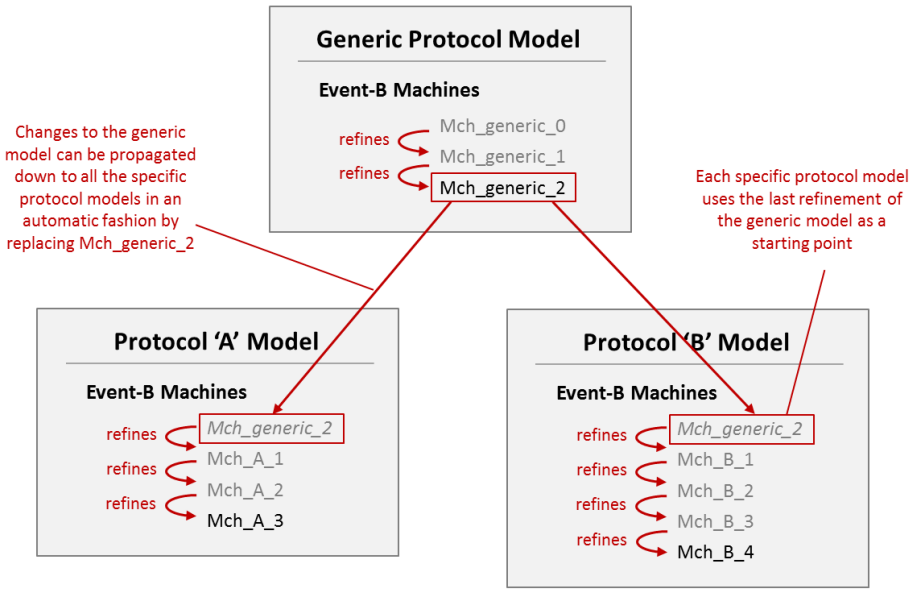


Figure 7: Example of model re-use

To achieve this, the decomposition plug-in had to be amended to allow for the ability to insert specific machines into existing Event-B projects as part of the decomposition process. Beforehand the decomposition approach only accommodated the creation of new projects through a decomposition run, not the addition or amendment of decomposition elements within existing projects. This change means it is now possible, for example, to modify the generic protocol model, and then propagate these changes automatically down to any specific protocol models which use the generic model as a starting point. Figure 7 illustrates this point. This significantly mitigates the amount of re-work required if changes have to be made higher up in the refinement chain. The same approach is also relevant if changes are made to a specific protocol model which is

used as a starting point or component in several system models; any changes can be propagated down to models which the use the protocol model by re-running the decomposition process.

- **Composition plug-in:**

  It was recognised that for the modelling approach described above to be successful, some changes are also required to the composition capabilities within Rodin. To successfully compose one or more of the protocol models with other elements - for instance, when more than one protocol is to be used in a distributed system model such as that in the revised case study – it is almost critical to be able to compose refinement chains, rather than just individual machines. Otherwise, the refinement chains have to be manually flattened before composition, which is a time consuming and potentially error prone task. In other words, involving a machine in a composition should automatically include all earlier refinements of this machine; without this, the composed model is missing essential information. For this to happen, some improvements had to be made to the composition plug-in. These improvements were communicated to the consortium and a solution was implemented.

- **ProR:**

  ProR was used to trace the relevant requirements in IEC 61850 to the model of the GOOSE protocol, to help evaluate the suitability of the method in similar cases. One of the main limitations found whilst following this process was the inability to link to requirements in another Event-B project. This causes an issue if, for instance, some requirements in the system model below the GOOSE model need to be traced to requirements linked to the GOOSE protocol. In this case, this is required if any requirements of the system are based on requirements within IEC 61850. This limitation has been fed back to consortium and is currently under consideration.

  Other desirable features identified during the work include:

  - The ability to automatically generate a report from ProR which gives completeness statistics (the number of requirements currently covered by the model) so that this can be presented to a customer during progress meetings.

  - An automatic check on the ProR document which gives feedback on any problems or inconsistencies, including any misspelt, undefined, or removed references to phenomena in the requirements.

  - A feature whereby clicking on a linked element (guard, invariant, etc.) in the ProR document takes you directly to the corresponding element in the Event-B model.

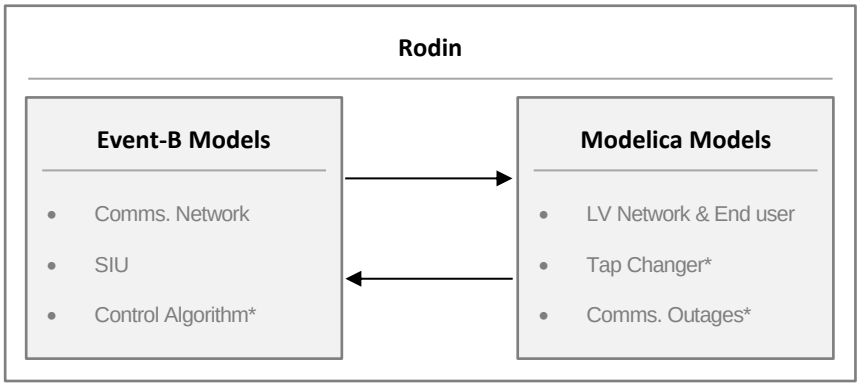
5.2 Towards Co-Simulation within Smart Grids

The formal models produced during previous phases of this work package include the SIU, datacentre and communications between them. However, determining whether these models have the required and intended semantics is a non-trivial procedure, and is known as validation. Traditionally, validation requires applying specific domain knowledge to check whether the modelled systems are faithful, and whether the formulae used for the invariants have the intended semantics.

Within the framework developed during the ADVANCE project, not only are the formal aspects of the modelling and verification of a cyber-physical system considered, but also the validation of the system against domain knowledge is considered. The solution chosen was

to incorporate a technique known as co-simulation into the framework. Co-simulation is defined as part of the functional mock-up interface (FMI) standard, and allows for models written across a variety of languages and tools to be simulated in parallel. More precisely, the models are arranged as a network where the outputs of one model feed into the inputs of other models.

Using the co-simulation methodology, it is possible to augment the existing simulation framework in ADVANCE to not only simulate discrete models, but to also simulate continuous models that represent the system's context, and hence domain knowledge. That is, the situation in Figure 8.



* Work not yet started on these models.

Figure 8: Co-Simulation Overview

The advantage of this methodology is that the continuous models are created using a modelling language that is easier to embed domain knowledge into, and further reduce the validation step. For instance, by definition, cyber-physical systems interact with the physical world, thus, within the ADVANCE methodology the "cyber" portion becomes a detailed discrete model, whereas the "physical" portion becomes a continuous model defined using a language that adequately captures the required domain knowledge. In most cases, as the continuous models are of physical systems, which are often represented (and understood) by a system of differential equations, a suitable modelling language would allow differential equations.

5.2.1 Modelica

Within this work package, the creation of the continuous models is by a language called Modelica. Modelica is an object-oriented, equation based language to conveniently model complex physical systems containing, e.g., mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents. Importantly Modelica is a non-proprietary, open language governed by the Modelica association. For this reason there are a plethora of tools that implement the language, however, only the well-developed tools discussed below have been considered within the scope of this work package.

Apart from Modelica being an open standard funded by European research projects, it was chosen as it is a proven technology with an existing industrial user base, mainly within the automotive industry and power plant providers. As a result, there is a standard Modelica library consisting of components (Modelica models). With respect to this project, the standard library includes components such as, transformers, power distribution lines, electrical generators, and electrical loads, which can easily be assembled into complicated models representing the context of the formal models. Importantly, all these components are governed by their own system of differential equations, which are combined to form the whole model.

The following paragraphs detail the tools explored in depth:

OpenModelica

The most developed, freely available Modelica application suite. As the name suggests, it is an open source implementation of Modelica and includes numerous applications built around the Open Modelica Compiler (OMC). Within this work package, the OMEdit tool was explored. OMEdit is a graphical and textual tool for the creation and simulation of Modelica models. The graphical component of OMEdit allows the user to quickly drag-and-drop existing Modelica models onto the canvas, which are then connected together by a point-and-click interface. The result of carrying out the above procedure is a new simulatable Modelica model. Further, within OMEdit it is possible to directly access the underlying Modelica code and tweak the equations, or even textually craft new models with exactly the desired system of differential equations.

Figure 9 shows an example (taken from the standard library) of how Modelica models are graphically displayed to the user. Each individual component of the figure represents a sub model, which in-turn could be further composed of other models, this is akin to object-oriented development.

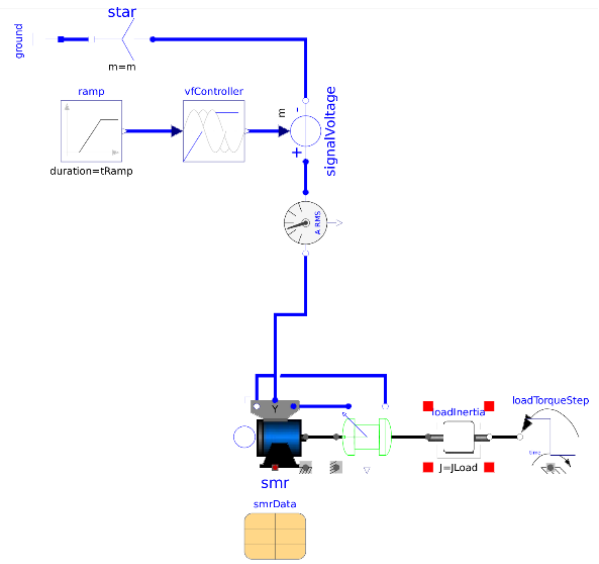


Figure 9: Example OMEdit Representation of a Modelica Model of an Electrical Generator

The OMEdit also provides support for simulation of the models. The user selects a time range of the model they are interested in, and then, the output of the simulation is a plot of the relevant variables against time. For instance, when the example model in Figure 9 is simulated, and one variable is projected, the resulting plot in Figure 10 is obtained.

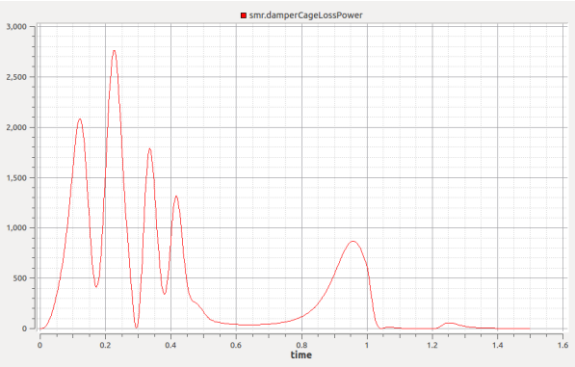


Figure 10: Example plot generated from Figure 9

JModelica

JModelica is another open source implementation of a Modelica compiler, albeit less developed than the OpenModelica suite, and lacks a graphical user interface. Nonetheless it is a powerful tool, and at time of writing under active development. The key features of JModelica are that it is Java based, and provides a Python interface. Although it was explored in this work, the lack of a graphical user interface limited its use.

5.2.2 Continuous models of power generation

To undertake validation of the SIUs using the co-simulation framework, it is first required to understand the context that the actual SIUs are designed to operate within. That is, the situation in Figure 11.

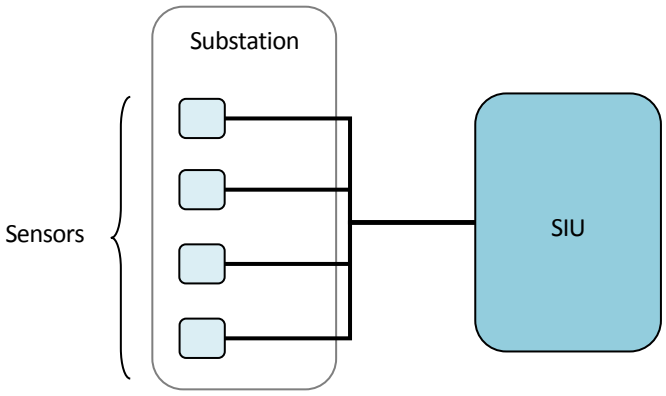


Figure 11: SIU Context

Here, the SIU connects to the sensors, which are placed at choice points on the feeders at the substation. The sensors are capable of reading various continuous attributes from the feeders that include: Current (Amperes), Voltage, Active power, and Reactive power. Modelling the context, with the aim to validate the SIU, encompasses providing realistic inputs for these attributes. To achieve this, a simple sinusoid could be generated of a constant frequency and amplitude, (e.g. Standard European values would be 50 Hertz and RMS amplitude of 230v). However, this view omits the influences that the wider power generation, transportation and distribution network, and importantly the consumer demand takes upon the inputs, and thus would not be a useful validation of the SIU. See Figure 12 for a simplified view of the wider power network, where the SIU is located at the substation. See Annex A for a detailed overview of the architecture. All parts of this network are

described by various well known systems of differential equations, thus, Modelica is highly applicable to the modelling of these types of networks.

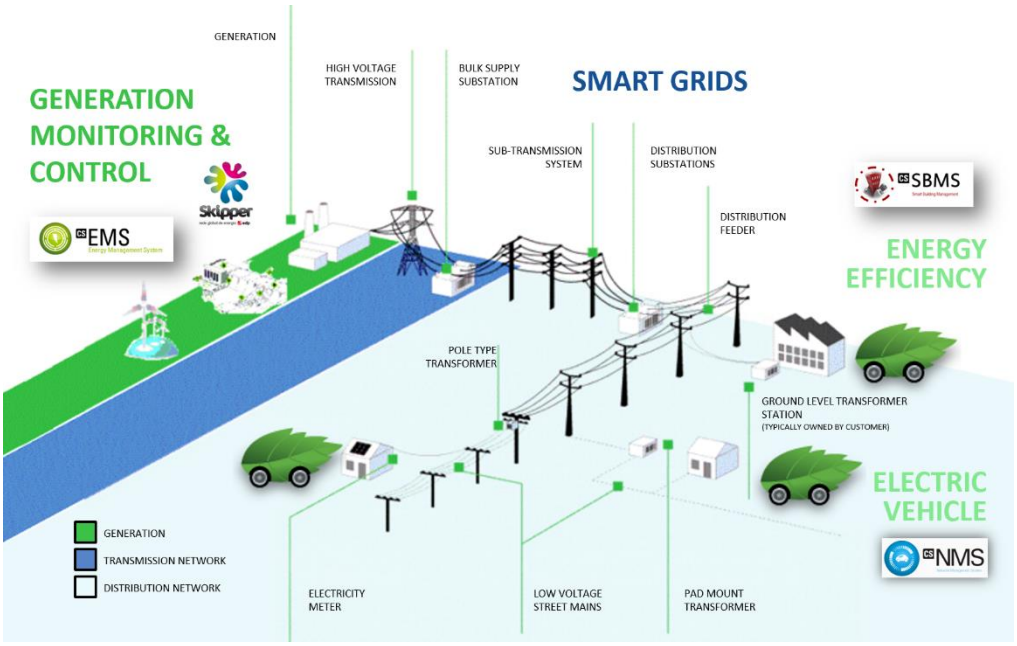


Figure 12: Power Generation, Transportation, Distribution and Consumption

Using OMEdit, a slightly simplified situation to Figure 12 was modelled. One further advantage of using Modelica for this is that the standard library contains most of the required models in a directly reusable form. By simplified, it is meant that the local distribution was omitted as the sensors are placed at the distribution substations. However, in future work the model can be expanded to represent the planned system more faithfully. The produced model is graphically depicted in Figure 13.

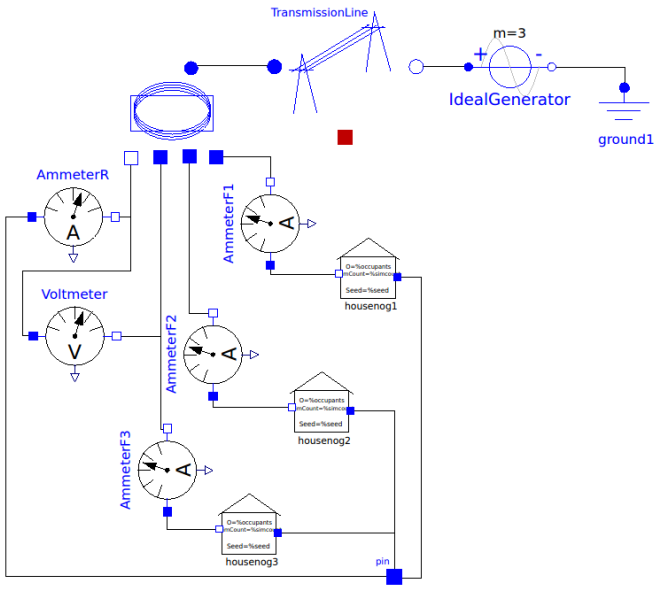


Figure 13: SIU Power Network Context

In Figure 13, the flow begins with the IdealGenerator at the top right. This is 3 phase AC, and generates a constant voltage that is independent of the load, i.e. electro motive force compensation and phase synchronisation are not considered. This decision was taken as this case study focuses on the distribution and end-user demand, and hence, it assumes that the power generation is sufficient. One of the terminals of the generator is grounded, and forms the return path. The other terminal is connected to the power transmission network, here represented by a transmission line; this line is 3 phase and defined by equations. The defining equations of the transmission line consider resistivity of the material, temperature, mutual inductance, magnetic flux, and inductive reactance; all these attributes are related by well-known differential equations. The underlying transmission line is already part of the Modelica standard library, however, it was required to lift the existing model to a 3 phase model.

The transmission line is then connected to the sub-station (step 4 from Figure 12), which steps the voltage down to 230V RMS, and then splits the 3 phase line into 3 separate feeders. Further, to complete the return path to the generator the substation internally links the neutral bar to the ground. These are the feeders which provide inputs to the system, and in the diagram the inputs are represented by the ammeters and voltmeters.

Each of the live feeders is connected to a load; this is required to drive the simulation. If each of the loads is a fixed number of Ohms, then a constant amount of current is drawn from the generator, which does not consider EMF compensation, hence, the produced simulations are non-interesting and are essentially constant. To improve the accuracy, it is required to vary the loads on each feeder. It is not simple in Modelica to produce a load curve of an average house, or group of houses, as it would be required to define these curves by equations – which would be an entire project on its own.

To provide interesting simulations to help validate the SIUs, the next section explores the creation of end user load curve that are based on probability distributions.

5.2.3 Models of end-user demand

To improve the usefulness of the Modelica models described in the previous sub-section, it is required that the loads on each feeder follow the standard load curves of an average dwelling. This allows for the loads on the feeders to vary, and provide realistic inputs to the SIU's. The inputs to the SIU are not only dependent upon the load, but also the system of differential equations in the power transportation network model. This is because when the loads rise, more power is drawn through the power transportation network, and hence, the resistance increases. This increase of resistance can be measured by the SIU as a slight drop in voltage. Interestingly, following on with the work with Selex ES, the tap changer could be introduced here to restore the voltage by changing the coil ratio at the transformer, however, this is left to the next phase as the tap changer had not yet been modelled.

NB: The basis of the models described in this sub-section, and the underlying probability distributions, were lifted from work undertaken by Ian Richardson within the CREST project at Loughborough University (UK) [AD-4].

The model developed by Ian Richardson is parameterised by the following:

- Number of occupants – Between 1 and 5 occupants are considered, this covers the majority of residential buildings and the limit of 5 occupants is not expected to be an issue.

- Weekday or Weekend – The load curve of the average dwelling varies depending on whether it is a weekday or weekend, as during the week occupants spend more time away from the dwelling.

- Day of year – An integer that ranges between 1 and 365, this is used to determine weather conditions and light levels.

- Position of house – The longitude and latitude are required. When these are combined with the day of year they define how much sun light hits the building.

Then, using various probability distributions, the model builds the following sub-models:

- **Occupancy Model:** Based on the number of occupants (1…5) and whether it is a weekday or weekend, the occupancy model is generated. The models detail when, and how many, occupants are active. This is performed on a one minute resolution over a 24 hour period. The occupancy model is directed by probability distributions that are based on actual studies. Thus, the occupancy model follows actual human behaviour, e.g. on a weekday most of the occupants will be up and active before the working day starts, and then the house becomes empty during the day, and active again during the evening.

- **Appliances Model:** A selection of appliance profiles (e.g. televisions, washing machines, water heaters and refrigerators) are generated. Each appliance profile contains important information regarding its idle and operational power consumption (broken down into active and reactive components), and how regularly it is used (e.g. consider the difference between the usage patterns of a washing machine and refrigerator).

- **Cloud Coverage Model:** Based on the day of the year and probability distribution the cloud coverage is computed. The coverage is at a one minute resolution, and takes the form of a percentage of coverage. The minimum and maximum values of the coverage are constrained, so that the result forms a bounded curve during daylight hours.

- **Lighting Model:** The lighting model is a similar to the appliance model in that it selects a number of light bulb profiles. This is depended upon the number of occupants in the house.

Finally, the energy consumption is computed for a 24 hour period at a one minute resolution. The result of the computation is in terms of active and reactive power usage. The computation steps through the day one minute at a time, and decides based on a probability distribution and number of active occupants whether to turn on an appliance (or light). Further, the computation decides how long an appliance (or light) is turned on for. There are special cases for heating devices that are dependent upon the time of year, and not on the occupancy model. Further information regarding the computation, or more generally the model is found in [AD-4].

The methodology of implementing the end-user simulation described below has changed, and an improved technique is discussed in Annex B. The improved technique does not rely on integrating external models into Modelica.

The model is available on the internet in Microsoft Excel workbook format with macros; however, Excel is not compatible with OpenModelica or the Co-Simulation interface used in ADVANCE, so the model was re-implemented using the programming language Haskell. Haskell was chosen as it allows for fast development and provides good support for recursion over the tabular structure of the probability distributions. Further, when this decision was taken, the Modelica environment available, OpenModelica, was unstable, and resulted in loss of work on many occasions. Since this time, OpenModelica as the environment has been improved.

An example of the output of the Haskell program (rendered using gnuplot) is in Figure 14.

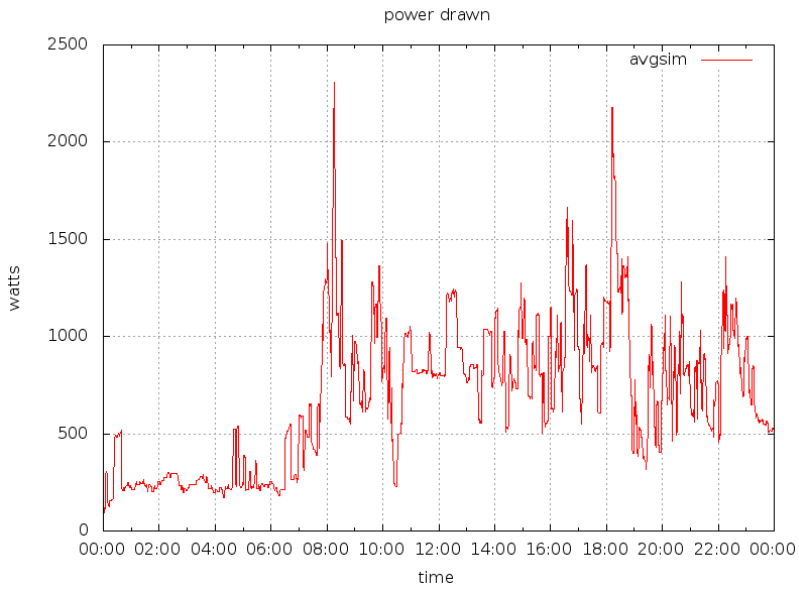


Figure 14: Consumer Demand Model Output (Active Power)

The resulting Haskell program was then integrated into the Modelica models of the previous section by using the foreign function interface within OpenModelica and creating a simple wrapper program. However, as the output of the model described above is discrete to the nearest minute of the day, it is required to convert this information into a continuous form. This transformation is performed by applying cubic spline interpolation within the wrapper program, before the output of the Haskell program is passed to OpenModelica.

This allowed for the loads to vary depending on the time of day, and hence, provide realistic inputs to the SIU. An example of the output produced after integrating OMEdit with the Haskell program is in Figure 15. This shows the total power consumed by the house (W) and a breakdown of the power consumption into active and reactive components.

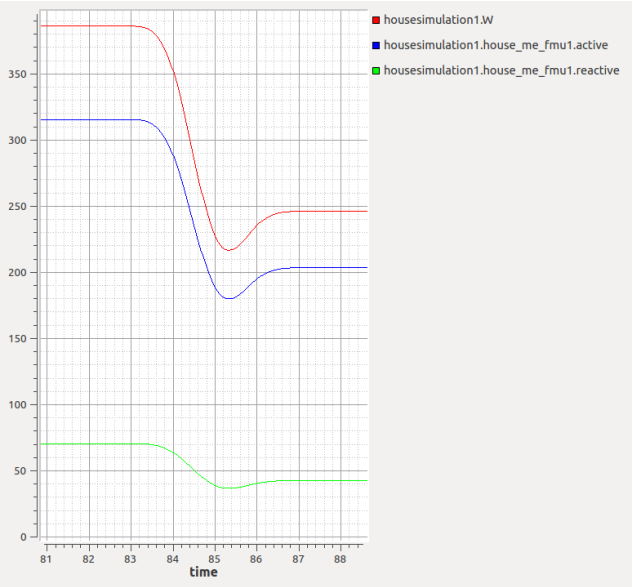


Figure 15: OMEdit House Plot

Taking full advantage of the end-user demand model, the Haskell program was further parameterised by the number of houses to simulate, and hence, allowed for neighbourhoods of houses to be simulated. In this case, all the houses share the same weather conditions, ensuring a coherent simulation; also, the number of occupants within each of the houses is selected at random. It has been left as future work to select a realistic distribution that is based on actual statistics for the number of occupants in a given house.

5.2.4 Simulating Event-B and continuous models using ProB2

The underlying technique applied within ADVANCE of simulating Event-B models with continuous models is known as co-simulation, and defined within the Functional Mock-up Interface (FMI) standard. FMI is a tool independent standard that supports both model exchange (ME) [AD-4] and co-simulation (CS) [AD-5] of dynamic models. In both cases, the FMI defines Functional Mock-up Units (FMU) that are independent software modules which contain the model and required meta-data to execute the simulation. The two variants of the FMI are as follows:

- **Model Exchange**:

  ME is the simpler of the two flavours, the model defines a set of variables, which range over Boolean, Integer and Real types. Canonically, the Boolean and Integer variables are discrete, however, the Real variables can be discrete or continuous, in the case that they are continuous the models also make available the associated derivatives (also of type Real). Some of these variables are denoted as input variables, and have read/write access, where others are only output variables and have read-only access. Internal to the model are formulae that describe the relationships between the variables. Finally, the time/clock of the model is externalised.

  Thus, the model exposes all the required information to integrate with a numerical engine that is capable of solving ordinary differential equations, e.g. forward Euler method.

- **Co-Simulation**:

  CS builds upon the model exchange standard by also internalising the numerical engine into the model. Thus, all that is required is to request the model to compute the values of all the variables for a given time *t'* by starting at time *t*, where $t < t'$. Note, all required inputs to the model should be set at time t. Then, the values of all variables are directly readable from the model without further processing.

As mentioned above, the framework developed within ADVANCE relies upon the Co-Simulation variant of the FMI. This is for technical reasons regarding integration of the FMUs into ProB2, i.e. using the off-the-shelf FMI wrapper JFMI (Java FMI). However, the tools that were explored thus far through the project did produce compatible FMUs.

- **OpenModelica**

  The OpenModelica tool set, at time of writing, only supports export of Modelica models using the Model Exchange FMI standard. However, support for CS export is planned in the future.

- **JModelica**

  Although JModelica supports exporting Modelica models using both the ME and CS variants, the resulting FMUs do not always perform as expected. That is, the values of the variables do not coincide with expected values during tests.

This limitation was reported to the ADVANCE consortium during the May 2013 plenary workshop. During the September 2013 plenary workshop it was suggested by the consortium that non-free tool sets such as Dymola (or Simulink with FMI Toolbox) would produce compatible CS FMUs, and during tests conducted at Southampton University had promising results. WP3 and WP4 continue to investigate and develop a feasible tool set for use within WP2.

5.3 Algorithm optimisation using ProB

Tests have been performed to assess the feasibility of using ProB to determine the optimisation of the algorithm to be deployed in the revised case study. The optimisation in this case concerns minimising the number of tap changes performed over a particular time. There are a limited number of changes the tap changer can complete before its lifetime expires; therefore it is imperative that the algorithm used takes this into account to prevent the possibility of frequent (and expensive) replacements of the tap changer. The idea behind this is that two models are produced:

1. *A model representing the behaviour of the actual algorithm.* This can be as detailed or as abstract as required, depending on the level of validation desired.

2. *A more abstract model of the events available to the algorithm, and the input it receives from the low voltage network.* In this case the events available will consist of the different options for the tap changer. There is no implementation of the algorithm in this model.

The first model – taking into account the algorithm – is simulated in ProB over a particular time frame. The input to the algorithm, i.e. the data provided by the SIUs, will be sourced from a continuous model of the energy supply and demand. Once the simulation is complete, the number of changes made via the tap changer over the time frame will be recorded, as well as the input that was produced via the continuous simulation. The recorded input from the simulation of the first model will be used as input for the second model. The second model will therefore have the same input and available events (tap changes) as the first. The second model will also be run over exactly the same time frame.

The second model is then run through either the model checker (using a breadth first approach) or the constraint solver in ProB, depending on the specifics of the model. The result of this will provide the shortest possible path from the start to end of the time frame used in the original simulation. In other words, it will give the path which uses the least number of events over the time frame. As the events in this case represent tap changes, this corresponds to the minimum number of tap changes that are required within the timeframe. By comparing this path to the actual path taken by the algorithm, the optimisation of the algorithm can be measured. If the number of tap changes performed by the algorithm is significantly greater than the shortest path found by ProB, then it will be clear that further improvements are required to the algorithm.

The approach described above only considers one particular set of input data – it does not cover all possibilities, rather provides an indication of the optimisation of the algorithm in that particular scenario. Multiple different simulation runs would be performed to get an idea of the performance of the algorithm in different scenarios.

As mentioned, some initial models have been created and tested. At the moment these models correspond to the second, more abstract, model mentioned above; there is currently no implementation of the algorithm. However, by modelling the possible tap changes and taking into account the range of voltages allowed for each setting on the tap changer, it is already possible at this stage to find the optimum path for a particular set of input data. For the initial tests the input data is static, although this will be replaced by the input from the

continuous models in the future. This model can be used as the starting point for the model of the algorithm; i.e. the model including the algorithm will be a refinement of this.

5.4 Additional Required Tool Changes and Feedback

During this phase of work (January 2013- December 2013) the following changes and improvements to the tools in the ADVANCE framework have been fed back to the consortium, in addition to those detailed in the sections above. A number of reported bugs and other smaller problems were also reported, although not detailed here.

- **ProB Simulation:**

  The specific protocol model and other models produced during this stage were partially verified through simulation with the Rodin integration of ProB. The quality of the simulation was hindered in some cases through the inability to concisely select the parameters of an event execution. This is particularly the case when the number of event parameters are large - for instance, when the input to the event is a data set. ProB in Rodin provides a set of possible parameter combinations which can be selected from, but when the range of the parameters, and the number of parameters, is high, it cannot possibly provide all combinations. Therefore, it may not be possible to investigate a particular path of events of interest if the parameters corresponding to that path are not included in the combinations provided by ProB. In these cases it is necessary to be able to manually enter each of the parameters for an event. This feedback was provided to the consortium and the required functionality was added to the Rodin integration of ProB.

- **Integration of ProB with the Theory plug-in:**

  The integration of the Theory plug-in with ProB was already under development, but the importance of this to the case study was stressed to the consortium. The ability to simulate models using theory elements will become essential as work progresses on the revised case study, as concepts such as sigma and sequences are required to deal with the proofs when large data sets need to be considered. This is certainty the case in the system in the revised case study; as large volumes of data need to be modelled as they are transferred between the SIU and tap changer elements.

- **Integration of Event-B and UML-B models:**

  Some of the modelling has been performed using the UML-like diagrams provided by UML-B. One limitation which has been found during this work is the inability to refine UML-B models using Event-B and vice-versa. This means that models generally have to be either composed of just UML-B models, or just Event-B code, even though it was found during the work that particular refinements of the model are best modelled using UML-B, whereas others require the use of standard Event-B. This has been raised, and will be addressed in the successor of UML-B, allowing for UML-B and Event-B elements to be mixed between refinements and within individual machines.

6. Revised Plan of Work

To ensure that the objectives of WP2 are met in the remaining period of ADVANCE, making best use of the available resources, we produced a revised plan of work that focuses effort on the priority for Selex ES which is validation of their voltage control strategy using the ADVANCE tools with respect the key properties identified through the hazard analysis. Besides the effort being contributed by Critical and Selex ES, Southampton and Düsseldorf are contributing significant effort to support the hazard analysis, modelling and co-simulation work for WP2. The work for the final year consists of three phases (Phase I is already complete and Phase II is partially complete at the time of writing).

6.1 Phase I: Jan 2014 – March 2014

6.1.1 Case Study Objectives

1. Perform requirements analysis based on Selex ES requirements input.
2. Model tap changer algorithm in Event-B.
3. Perform STPA safety analysis to identify the hazards, the hazard mitigation strategy and key safety properties.
4. Design and refine the discrete elements in Event-B.
5. Develop a preliminary model of the continuous domain voltage network.
6. Perform an initial FMI multi-simulation to provide early feedback to WP4.

6.1.2 Meeting the ADVANCE DoW Objectives

1. Evaluate ADVANCE component view and multi-simulation capabilities. (DoW 1a, 1b, 1i)
2. Evaluate the new ADVANCE iUML-B state machine facility. (DoW 1b)
3. Evaluate the ADVANCE Theory Plug-in capability. (DoW 1)
4. Evaluate the new B Motion Studio and other ProB 2 facilities. (DoW 1b)
5. Evaluate ADVANCE STPA methodology. (DoW 1c)

6.2 Phase II Plan and Achievements, April - June 2014

6.2.1 Case Study Objectives

1. Integrate a refined communications model with multi-simulation to represent properties of the communications solutions being adopted by Selex ES.
2. Refine continuous model of power generation and distribution to cover fluctuations due to micro-generation and end user consumption.
3. Analyse safety and performance of the voltage control algorithm and provide further feedback to Selex ES on the tap changer algorithm.
4. Define the strategy for simulations to ensure we get suitable coverage and test a suitable range of scenarios.

5. Demonstrate ADVANCE multi-simulation at the Rodin Workshop, Toulouse, in June 2014.

6.2.2 ADVANCE DoW Objectives

1. Evaluate ADVANCE multi-simulation driven by manual tests. (DoW 1a)

2. Evaluate B Motion Studio animation for Workshop demonstration. (DoW 1b)

3. Evaluate Model Testing input to WP4. (DoW 1a)

6.3 Phase III Plan, July - November 2014

6.3.1 Case Study Objectives

1. Evaluate use of automated ADVANCE model testing and test coverage metrics as a way of ensuring validity of the test set used to test the implementation of the algorithm.

2. Provide input to Selex ES on the validity of the test set for the tap changer algorithm.

3. Reflect on framework applicability from an industry perspective and business benefits.

4. Demonstrate ADVANCE multi-simulation of Smart Grid Solutions at the ADVANCE Industry Days.

6.3.2 ADVANCE DoW Objectives

1. ADVANCE multi-simulation driven by automated tests. (DoW 1a)

2. Validate ADVANCE coverage metrics. (DoW 1a)

6.4 Effort planning

Please note this is a private section available to members of the ADVANCE consortium and the EU and it is not available to the public domain.

Plan Phase	CSWT	Southampton	UDUS	Selex ES	Total
I (Jan-Mar)	3.89	1.5	1	1.5	7.89
II (Apr-Jun)	2	3	2	1	8
III (Jul-Nov)	3	3	2	0.5	8.89
Total	**8.89**	**7.5**	**5**	**3**	**24.39**

Annex A. Technical Architecture

This annex details technical architectures of the problem outlined in Section 4.2.

The following image is of the SIU modelled during WP2, it is a Metrology and Communication Unit (MCU) 520 produced by Selex ES within the GridKey project.

Figure 16: MCU

The image below details how the MCU fits into the wider scenario. On the left-hand side the substation, detailing three 3-phase power lines monitored by the MCU. On the right-hand side the control centre and SCADA system.

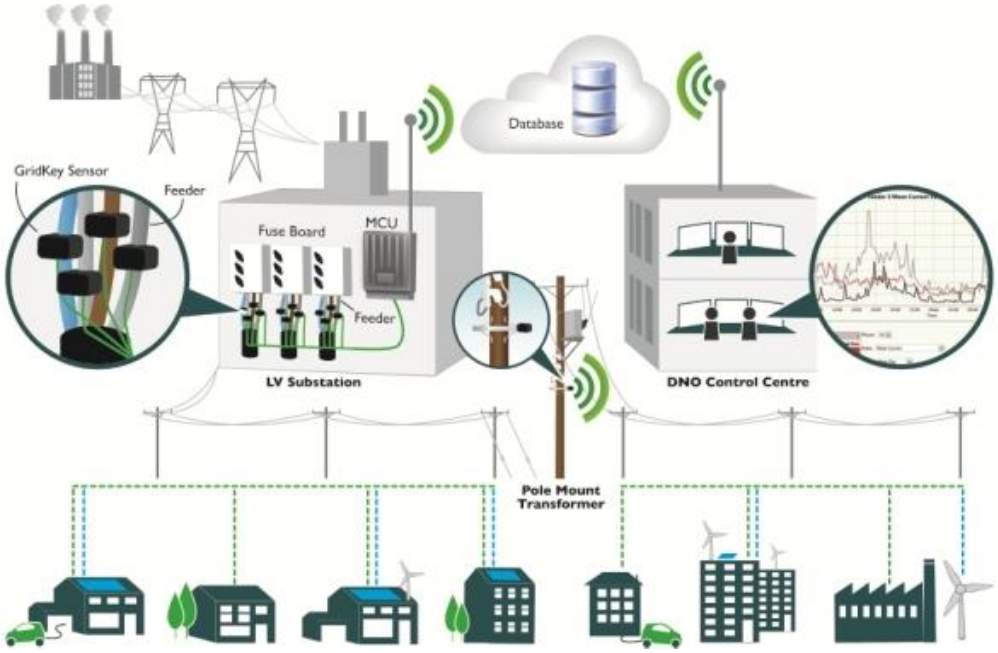


Figure 17: MCU integration

The centre of the above image details a mid/end point MCU mounted on a low voltage power distribution pole, and the lower portion of the above image details how the power lines are connected to the end users, which include domestic micro-generation and commercial renewables.

The three images below show the actual current sensors that are non-invasively connected to the feeders. Left: sensor connected to cable. Middle: Gridhound sensor opened. Right: Alternative current sensor, used when Gridhound not possible.

Figure 18: Current sensors

The following image details a typical 6 feeder substation installation, and where the MCU520's are located:

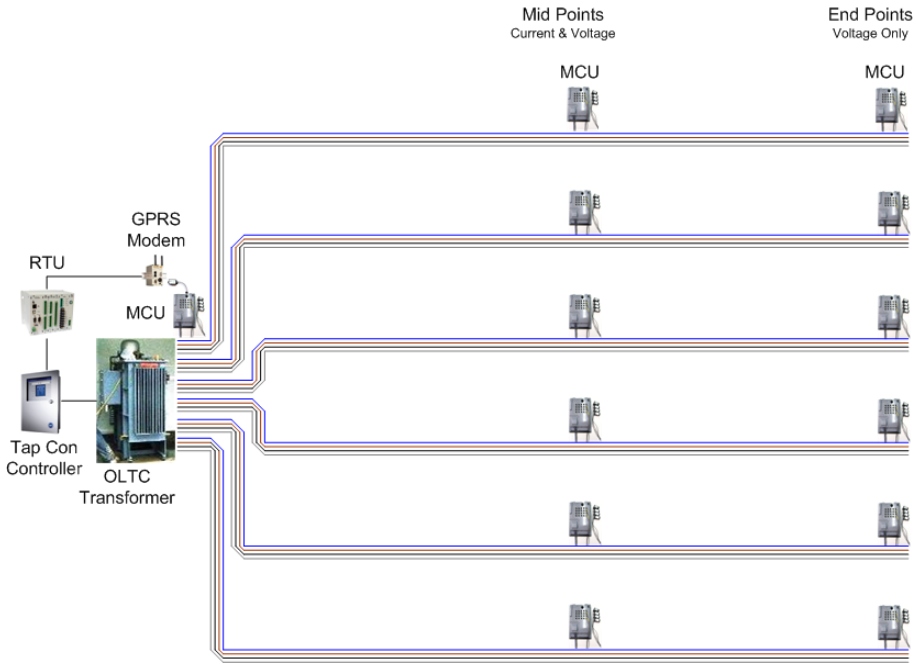


Figure 19: Feeder installation

The above image clearly shows the MCU located at the substation (on the left), and the mid/end point MCUs. On the left of the image the RTU, Tap Con and OLTC transformer are introduced. The RTU will house the voltage control algorithm, receive inputs from the MCUs, and command the Tap Con to a given voltage point.

Annex B. Progress Update (April 2014)

As explained in section 4.2, the revised case study for WP2 focuses on an algorithm designed to control the power distribution within a low voltage network. Such an algorithm is necessary due to the volatile nature of the micro-generation increasingly being fed into the network by end users; photo voltaic, wind, etc. The introduction of this micro-generation means that the power flow within the network can no longer be considered using a simple top-down approach; and that the power flow at different points in the network has to be considered when adjusting the voltage at the transformer.

The algorithm system consists of several sensor units, located at different points of the network that send periodic reports of the voltage and current levels. These reports are collated at the transformer and fed into the algorithm. The algorithm makes a decision based on these values and feeds a new target voltage into a tap changer connected to the transformer. The sensor unit used is that which was modelled during earlier work in WP2 (SIU) – the models are reused in this implementation.

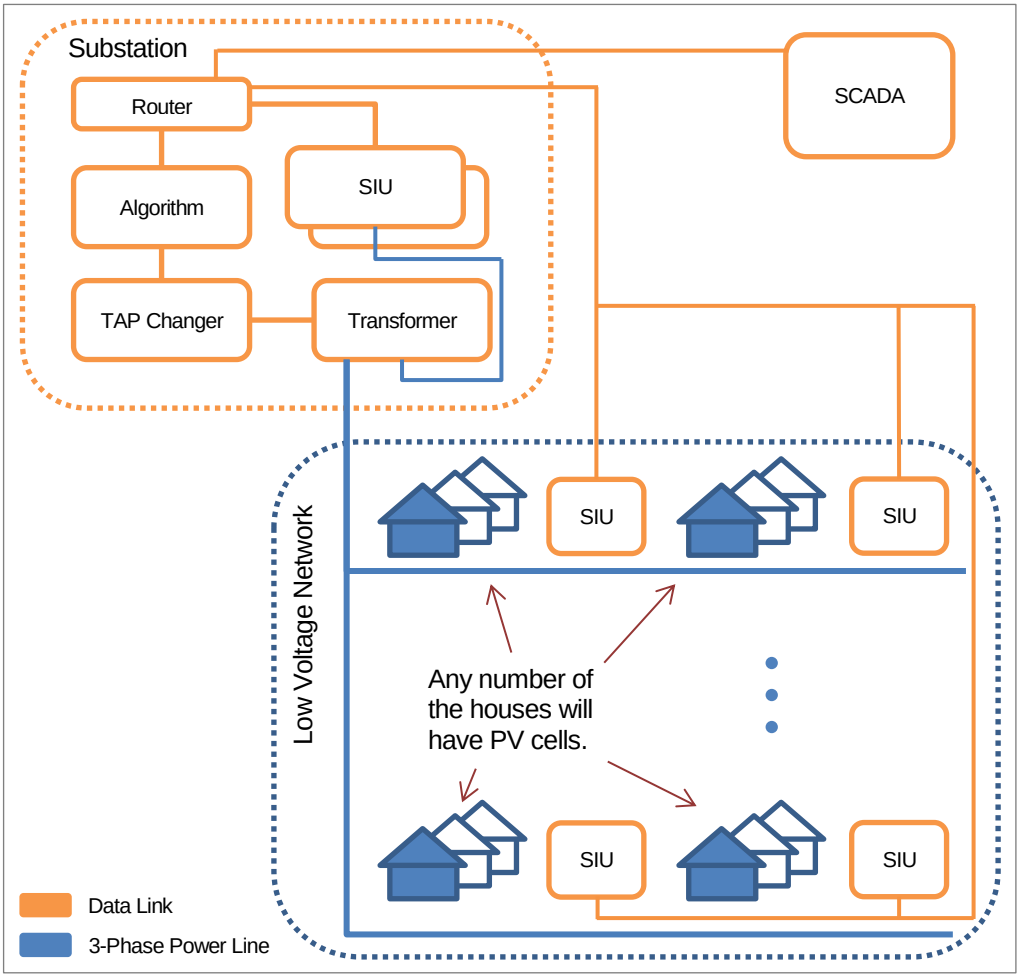


Figure 20: Architecture

Since work has begun on the revised case study, a suitable modelling strategy has emerged. This consists of splitting the modelling into several distinct parts:

Case study element	Modelling Language
Control Algorithm	Event-B
SIU	Event-B
Communication network	Event-B (with continuous input to dictate transmission failure rates)
Tap changer	Continuous (Modelica)
LV network and end user demands	Continuous (Modelica)
Communications outage occurrence	Stochastic (Modelica)

Table 3: Modelling elements

These separate models are then combined in the multi-simulation framework to be run side-by-side. The interaction between the different elements in the multi-simulation is detailed in Figure 21. The progress made on each of these model elements is detailed in the following sections.

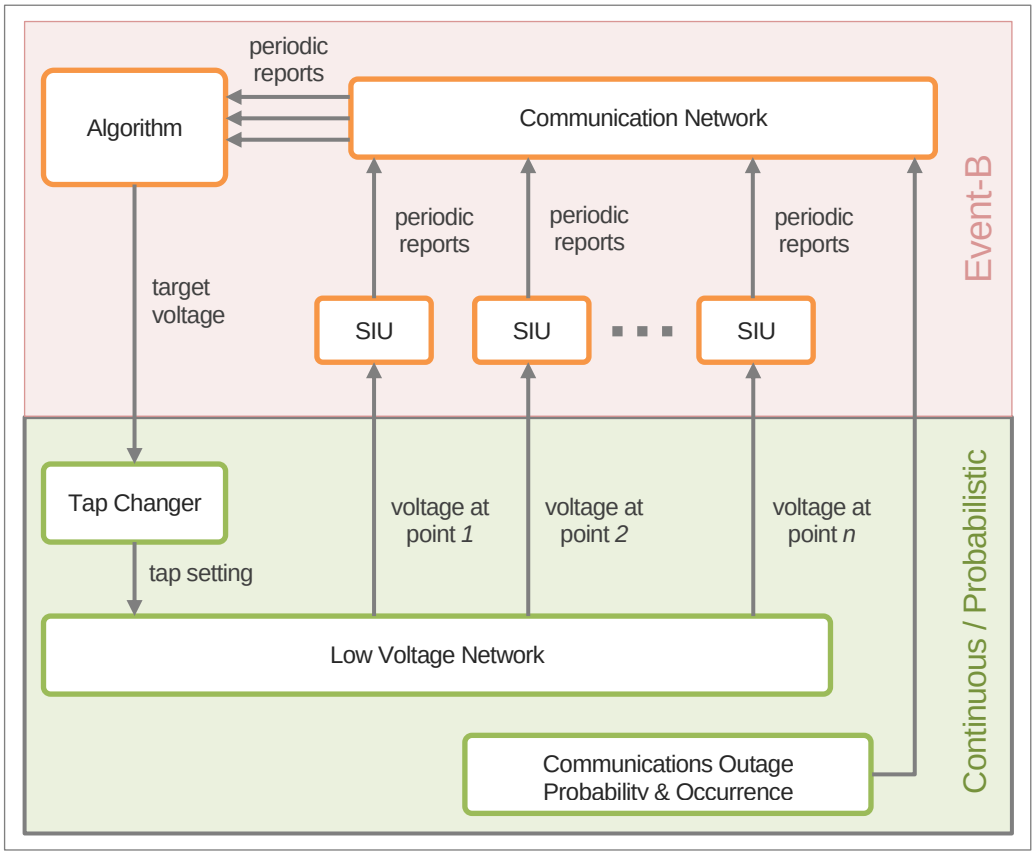


Figure 21: Multi-simulation interaction

The behaviour of the tap changer is not to be formally verified in Event-B as, within the scope of this case study, it is considered to be part of the environment. Therefore the behaviour of the tap changer will be placed in a continuous model and the behaviour of the other system elements (algorithm and SIU operation) will be verified against it.

B.1. Current Work

Models corresponding to each of the elements in Figure 21 have been created, where the communications, algorithm and low voltage models have been the most actively developed. These models are based on information provided by Selex ES. Feedback obtained during the formalisation of the models has been provided to Selex ES.

Work is ongoing to combine these models within the multi-simulation framework to test it against an industrial scale problem. This entails not only importing them into the framework, but also tweaking the models to improve the accuracy of the interaction between the different elements and the performance of the simulation.

B.2. Algorithm Model

B.2.1. STPA

The starting point for the model of the algorithm was to perform Systems-Theoretic Process Analysis (STPA) on the description of the algorithm and surrounding system. This strategy – using STPA as the starting point – has been defined in WP5. This informal analysis allowed for the main risks to be identified; these are presented in Table 4.

	Increase voltage target when shouldn't	**Fail to increase voltage target when should**	**Increase voltage target too early**	**Increase voltage target too late**
Risk / undesired behaviour	Voltage too high; increased the voltage *Too many tap changes*	Voltage too low; remains too low	No hazard *Too many tap changes*	Voltage too low; voltage potentially gets lower

	Decrease voltage target when shouldn't	**Fail to decrease voltage target when should**	**Decrease voltage target too early**	**Decrease voltage target too late**
Risk / undesired behaviour	Voltage too low; decreased the voltage *Too many tap changes*	Voltage too high; remains too high	No hazard *Too many tap changes*	Voltage too high; voltage potentially gets higher

Table 4 : Identified risks from STPA

As seen in Table 4, the potential risks (highlighted in red) revolve around how and when the algorithm sets a new target voltage for the tap changer. The hazards that have been identified for this case study materialise when the accessible voltage from the grid lies outside of the specified range from the DNO (+10% / -6% of the declared voltage)[1]. These hazards impact both from a safety perspective and a business perspective. In terms of

[1] A further restriction of the statutory range defined in [AD-8].

safety, appliances designed to operate within these bands could become dangerous if a significantly higher voltage is supplied. In terms of a business perspective, should equipment not work as expected or break due to extended periods of voltage levels outside of the identified bands, the DNO could encounter significant costs or loss of reputation. These hazards can occur if the target voltage given to the tap changer is set incorrectly, not set when a mitigating action is required, or set at the wrong time.

Setting the target voltage too early doesn't result in a risk, but could have the effect of increasing the number of tap changes; a variable that Selex ES has expressed needs to be minimised. This is because the tap changer mechanism has a limited lifetime, resulting in the need for replacement or expensive maintenance, and the potential for an increase in failure (and hence temporary loss of electricity supply to the customer). Therefore although this doesn't present a significant hazard, this scenario has economic and resource implications, and still needs to be taken into account during the subsequent verification.

The fact that STPA provided a strong starting point for the models demonstrates the important role that safety analysis plays in the ADVANCE framework.

B.2.2. Abstract states

Whilst performing the STPA process, it was necessary to consider the most abstract states of the overall system. These states formed the initial, most abstract, level of the Event-B model. The states were modelled using the iUML-B tool (the new development of UML-B) – as shown in Figure 22 – from which the corresponding Event-B was generated automatically.

The intertwined use of state machine diagrams during the development of the algorithm model assesses the suitability and benefits of using diagrammatic techniques in the ADVANCE workflow.

The transitions in Figure 22 correspond to events in the Event-B model, which have been further expanded in the proceeding refinement steps.

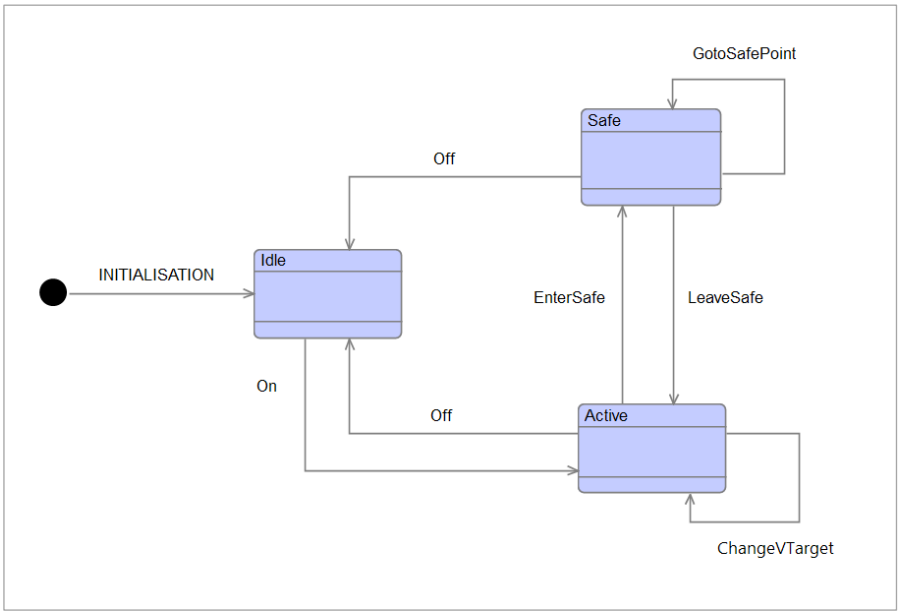


Figure 22: Abstract state machine of the algorithm and surrounding system

Idle represents the state where the other elements of the power grid are still operating, but the algorithm is disabled. Once the algorithm is active, it can change the voltage target for the tap changer (*ChangeVTarget* in Figure 22). A *Safe* mode is defined, which the algorithm will enter in pre-defined scenarios. For example, it has been decided that if the level of communication from the SIUs drops below a selectable threshold, the algorithm will move into this safe mode, as it is no longer considered to have a complete enough picture of the low voltage network to make an informed decision as to what the target voltage should be set. The only action that can be taken in this safe mode is to set the target voltage to a safe pre-determined value (*GoToSafePoint* in Figure 22).

B.2.3. Algorithm control cycle

This abstract state machine was refined over a further two steps, which introduces the trigger conditions for the safe mode and the input received by the algorithm respectively. The refinement was achieved by adding Event-B guards and actions to the events specified in the abstract state machine, along with the corresponding variables and invariants. The input to the algorithm corresponds to the periodic reports supplied by the SIUs, although this is still defined in an abstract manner in the algorithm model. This input corresponds to the combined outputs of each of the SIU models; therefore the input becomes concrete when the algorithm and SIU models are linked together in the multi-simulation.

The use of refinement during the development of the algorithm model, (a total of 9 refinement steps) shows that separating out the complexity of the model allows for the detail to be introduced in a manageable and consistent manner.

Once this detail had been put in place, the control cycle of the algorithm – dictating at which points the algorithm is run – could be added as an additional refinement. Again the specification of this control cycle was achieved using iUML-B, which is shown in Figure 23.

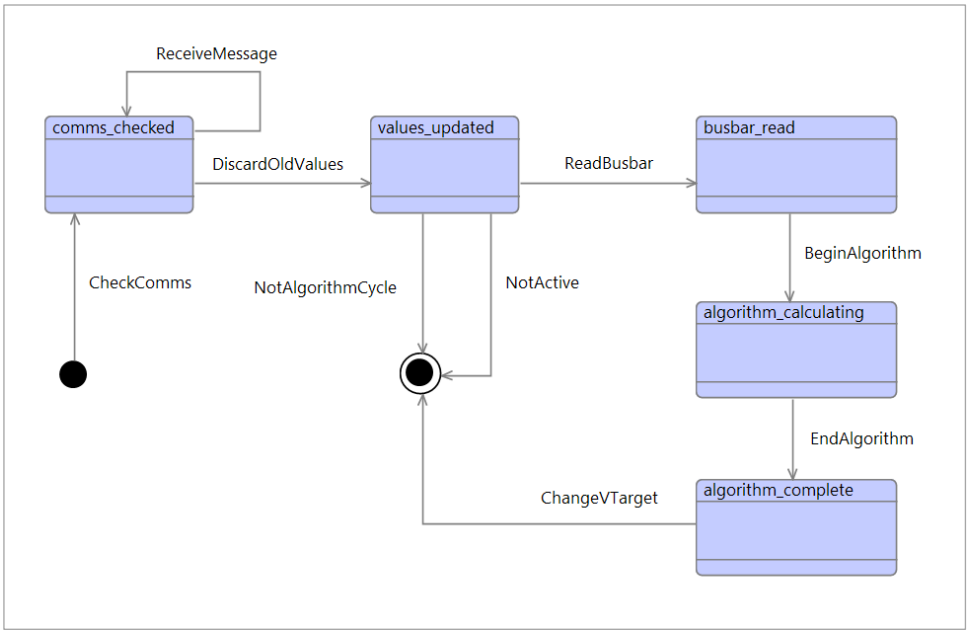


Figure 23: Algorithm control cycle

This state machine represents the events that occur each time the algorithm model is called during a multi-simulation step. The communication link to each SIU is checked (*CheckComms* in Figure 23), to decide if the algorithm should enter the safe mode, and any messages are read from the buffer (*ReceiveMessage*). As the algorithm only considers SIU

messages within a certain timeframe, any stored values older than this have to be discarded (*DiscardOldValues*). The busbar voltage – i.e. the voltage at the LV side of the transformer – is read at the start of the algorithm execution (*ReadBusbar*), as this is used during the calculation of the new target voltage. An interval is defined at which the algorithm is executed (and the target changed); therefore if the current time does not correspond to this interval, only the *NotAlgorithmCycle* event will be available at this step. Alternatively if the algorithm is disabled, only the *NotActive* event will be available during each iteration.

This state machine does not replace the behaviour in the abstract state machine shown in Figure 22, nor does it refine it directly. It introduces additional behaviour which occurs alongside – and restricts – the behaviour in the abstract state machine. For instance, if the result of the *CheckComms* event is that the SIU communication is below the defined threshold, the *EnterSafe* event in Figure 22 will become available.

B.2.4. Algorithm decision flow

The next refinement step in the model introduces the decision flow within the algorithm. This was modelled using iUML-B due to ease of translation from the description of the algorithm flow supplied by Selex ES (which was also presented in a state machine format). The resulting iUML-B diagram is shown in Figure 24 and Figure 25.

This state machine represents a refinement of the *algorithm_calculating* state in Figure 23. *BeginAlgorithm* represents the start of the process, which completes when *EndAlgorithm* is triggered.

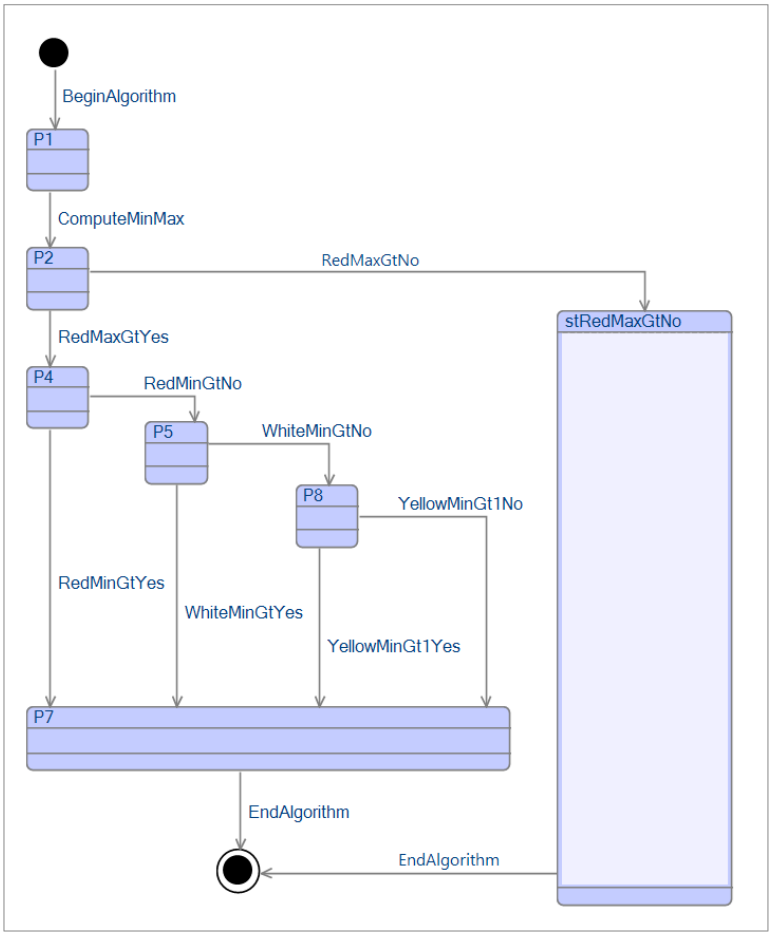


Figure 24: Algorithm flow diagram

Figure 24 represents a subset of the diagram; the state *stRedMaxGtNo* contains a nested state machine, which in itself contains further nested state machines, to separate out the complexity. An overview to indicate the size of the entire state machine with these nested state machines expanded is shown in Figure 25 below.

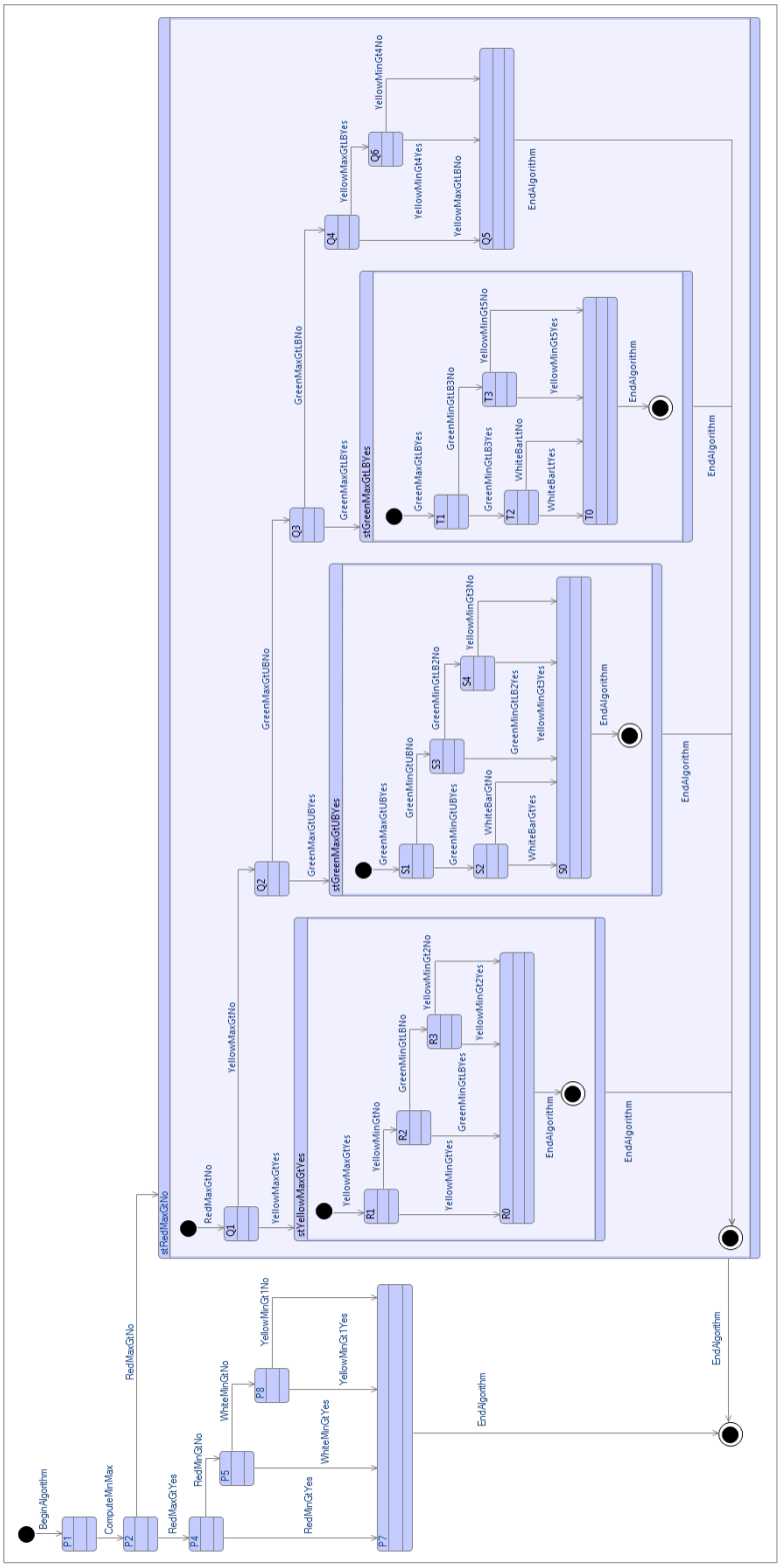


Figure 25: Expanded algorithm flow diagram

B.2.5. Further refinement and STPA invariants

A further four refinement steps were implemented in the model to add the required detail behind the events added in the state machine shown in Figure 25; the events were refined directly with Event-B. It is clear that this ability to interchange between iUML-B and Event-B, which was fed back to the consortium as a necessary improvement to the tool during earlier work, is extremely valuable.

Specified within these refinements are the invariants which ensure that, if they remain true, the risks identified during the STPA analysis (Table 4) do not occur. Therefore any violation of these invariants found through proof or model-checking highlight hazardous or undesired scenarios, and indicate that the behaviour of the algorithm needs to be amended or corrected.

> The invariants in the model guarantee that any occurrences of the risks identified during the STPA analysis are highlighted, ensuring that the algorithm is more robust and trustworthy.

These invariants represent the following two statements:

1. *If the voltage is above the defined maximum at any of the SIU points, the target voltage should be decreased at the transformer.*

2. *If the voltage is below the defined minimum at any of the SIU points, the target voltage should be increased at the transformer.*

These are the main invariants which the algorithm is to be verified against during the remaining work. It was most appropriate to add these invariants at this stage in the model, because, in order to make sense they require the algorithm decision flow and variables such as the target voltage to already be defined.

B.2.6. Results of Algorithm Verification

The verification of the algorithm has been achieved by working directly with the proof tools in a more abstract version of the model. This version of the model is still separate to the continuous models (and other Event-B models such as the SIU), so does not consider elements such as realistic communication delay or SIU operation. However, as the model is more abstract, all possible inputs to the algorithm can be considered whilst allowing for proving to preserve the invariants. Even though the model is more abstract, some important issues were still highlighted through attempting to prove the invariants in the model – these were fed back to Selex ES, and are detailed below. The results of this section provide examples of the successful use of the ADVANCE framework in finding limitations of the algorithm.

B.2.6.1. Violation of busbar voltage bounds

Invariants were specified in the model to limit the target voltage given to the tap changer to a valid range of values. This range was calculated by considering the voltage limits close to the transformer. These invariants could not be discharged in several instances, due to the fact that there are no limitations within the algorithm design to prevent the target voltage being set outside of the valid bounds at the transformer. The algorithm makes a decision based on the minimum and maximum reported voltages from all the SIUs, with no regard to where the current busbar voltage lies. For example, if the minimum reported voltage is sufficiently low, then the new target given to the tap changer can increase the voltage above the maximum limit at the busbar, regardless of how high the busbar voltage is measured at during the start of the algorithm.

It was fed back that a decision needs to be made as to the preferred behaviour in these cases, as it has been demonstrated that this is not well enough defined – and could therefore lead to unexpected behaviour. It could be that it is preferred for the voltage at the end/mid-point in the network to stay below minimum so that the voltage at the busbar can be kept below maximum, to avoid any damage to equipment connected close to the transformer. If this is the case, then the algorithm needs to be modified. The different choices regarding the preferred behaviour will be tested during the next stage of work, to help inform Selex ES on the most suitable choice. If Selex ES identify a particular solution that is preferred, this can be validated within the toolset.

Several important issues around the algorithm behaviour were interpreted as a direct result of utilising the various proving tools in Rodin over a relatively short duration of time.

B.2.6.2. Simultaneous minimum and maximum voltages

As the algorithm makes a decision based on the minimum and maximum reported voltages from all the SIUs, it is possible for a voltage below the defined minimum and a voltage above the defined maximum to be reported simultaneously. In this case the target voltage cannot be both increased and decreased; therefore one of the invariants specified in section B.2.5 is always violated regardless of how the new target voltage is calculated.

The problem occurs when there is significant imbalance between the feeders – for instance, a feeder with heavy consumption compared to a feeder with significant PV generation. Although this was already recognised as a limitation by Selex ES before the models were developed, the fact the models cannot be proven with the supplied description of the algorithm and system shows that the desired behaviour in this case has not been defined. Details of how the algorithm is expected to react in such cases (even if the desired behaviour is to do nothing) need to be established. Again, the next stage of work will help to inform this decision.

B.2.6.3. Significant differences in busbar and target voltages

The algorithm uses the actual busbar voltage as a reference, which is increased or decreased accordingly to create the new target voltage. Due to the way the tap changer operates, the busbar voltage can differ from the currently set target voltage (see section B.5.2). The tap changer allows a certain bandwidth either side of the target voltage before a remedial action is taken, and even in the case that the busbar voltage exceeds this bandwidth, a certain delay will be allowed before the action takes place. Certain proofs in the model could not be discharged until the following guards were added to the event which changes the target voltage:

$$V_{busbar} < V_{target} + V_{unit}$$

$$V_{busbar} > V_{target} - V_{unit}$$

Where V_{busbar} is the current busbar voltage, V_{target} is the current target voltage value, and V_{unit} is the discrete unit used to increment the increase and decrease of the target voltage. If these conditions do not hold, the target voltage may not change as expected. For example, if the difference between the busbar and current target voltage is large enough, the target voltage could decrease even though the result of the algorithm is to increase the voltage. In this case the new target is larger than the busbar voltage, but is still smaller than the current target voltage.

The point here is that an important question about the algorithm behaviour is raised through the process of developing and proving the model; namely, should the target voltage or

busbar voltage be used as a reference when calculating the new target? The initial choice – using the busbar voltage – appeared to be the correct decision before this was raised by the model, as this is more intuitive. The busbar voltage represents the most up-to-date value for the voltage at the transformer, and therefore it seems sensible that it should be used as a reference when increasing or decreasing the target voltage for the system. However, after considering the two options in more detail – something that we were forced to do as a result of modelling the system in a formal manner – the decision is not so clear cut. Advantages and disadvantages were found for both approaches, which will help inform Selex ES on the best decision and aid them in better understanding the limitations of each choice.

B.3. Communications Network

During discussions with Selex ES, it was made clear that another key area of interest in terms of validation is the choice of communications structure. Selex ES has identified two candidate solutions for communication structure as described below. Using the ADVANCE framework Selex ES can make an informed decision as to which solution has a more positive impact on the algorithm performance:

1. A direct point-to-point structure, where each SIU communicates to the algorithm system separately, through a GPRS link or similar (see Figure 26).

2. A mesh network structure, where each SIU is considered as a node in the network, and communications may be routed through one or more SIUs to reach the system housing the algorithm (see Figure 27).

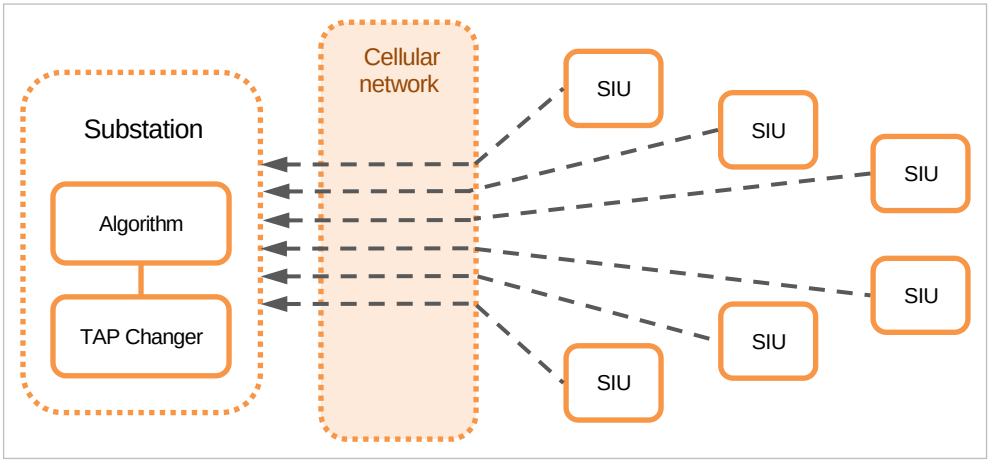


Figure 26: Point-to-point network structure

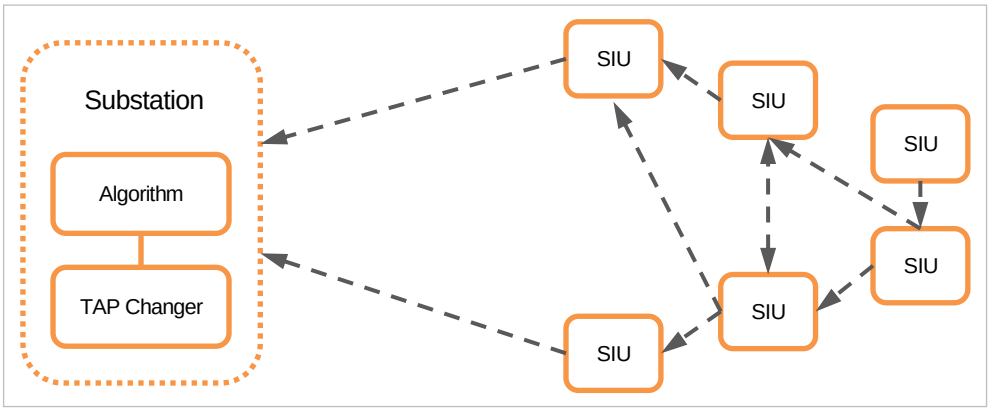


Figure 27: Mesh network structure

Each of these network structures was modelled using the strategy defined in Section 5.1.1 for modelling communication protocols. The generic protocol model described in this strategy was used as the starting point for both of the models. This strategy was originally used to model aspects of the IEC 61850 GOOSE protocol. It should be made clear that this protocol is not used in either of the solutions from Selex ES identified above. The intended purpose of modelling aspects of the GOOSE protocol was to validate and test the methodology set out in Section 5.1.1, and identify the tool features required to achieve this methodology. The GOOSE protocol was chosen due to its relevance to the smart grid domain. As detailed in Section 5.1.1 the process of working through this methodology with a specific example revealed a number of changes that were required to the toolset. With these changes in place, and the strategy tested, it was possible to model the two network structures identified above effectively and efficiently.

Selex ES commented on the speed at which the communication models had been developed, indicative of a more efficient approach and reduced-time-to-market than traditional engineering methods.

At the current stage of work, the two network structures and underlying protocols have been modelled to include only the critical elements that are required when verifying properties about the algorithm. These elements have been identified as:

- The level of network traffic over each of the potential links.
- End-to-end delays encountered through packet transmission and routing.
- High-level changes to routes by the routing algorithms.
- The reaction of the network to the loss of communication from SIUs.

This list represents the properties of the network that have been identified as having a potential effect on the performance of the algorithm. They represent the properties of the network that Selex ES is interested is exploring when verifying the algorithm behaviour. These elements do not need to be modelled in great detail to achieve the aims of the case study; one of the main benefits of Event-B is the ability to create abstracted models which nevertheless still capture the key elements that need to be considered in the system. So, even though there are elements of the network which are abstracted away in the Event-B models, this does not prevent the verification of the algorithm and surrounding system. In fact, more detailed models of the protocols – including information that is not directly relevant or required for the specific case study, or that could be abstracted and represented by more generic properties of the network – will more likely be detrimental to the verification efforts. Additional details around the network protocols and topology can be added at later stages or if additional issues are raised which Selex ES want to explore; but this additional detail is not required on the critical path to achieve the aims of the case study.

In terms of development, the mesh network model has seen the most focus as the routing mechanisms are more complex. In terms of the routing protocol used by the mesh network, the properties of the routing protocol applicable to the elements above have been modelled to a sufficient level of detail to allow for the verification of the algorithm when composed with the other models in the system. This takes into account:

- The routes seen by each node.
- The elements which influence the choice of route for a packet.
- The recalculation of routes when a particular link or set of links goes down.

By simulating traffic flowing through the network, how the network copes in the event that one or more of the SIUs lose communication can be investigated. For instance, network traffic could significantly increase down one or more links as a result of route recalculation, to the point where packets are dropped. This type of event has the potential to impact the operation of the algorithm as information from the SIUs could be lost or received late. Therefore the intention is to run these models in parallel with the algorithm and continuous elements in the multi-simulation framework, to investigate the impact of this type of scenario. By running simulations for each of the network structures identified above, a comparison can be made to help inform Selex ES of the best way forward. As mentioned in section B.4, stochastic models have been developed to ensure realistic occurrences of communication outages during the tests.

By running the communication models in parallel with the algorithm and continuous elements in the multi-simulation framework it is possible to investigate the impact of losing one or more SIUs on the performance of the algorithm.

In order to visualise the status of the network during simulation, a BMotion Studio visualisation was produced; an example is shown in Figure 28. The width of each link within the visualisation is updated depending on the amount of traffic passing through at that particular time in the model. Individual links or SIUs can also be removed by interacting directly with the visualisation elements, to explore the effect of the removal on the traffic in the network. Selex ES consider this visualisation to be valuable when analysing potential issues in the network. In particular, they would want the option to record the visualisation over the simulation period, so that it could be played back at later times and viewed over a larger number of simulation steps (that is not feasible when stepping through the simulation manually). For this reason, development of the visualisations has been moved to the new version of BMotion Studio currently in development, as this new implementation offers this functionality.

The visualisation capabilities of BMotion Studio would be of particular use during testing activities, according to Selex ES, indicating that the use of this tool plays an important role when applying Rodin and Event-B within industry.

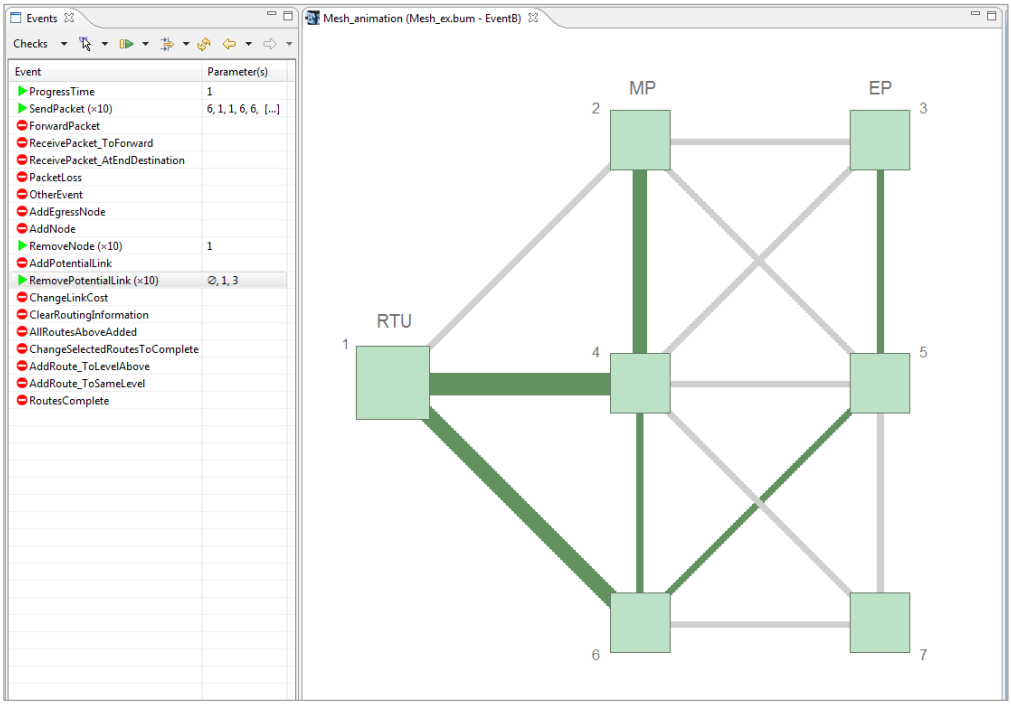


Figure 28: BMotion Studio mesh network visualisation

B.4. Stochastic Communications Model

One aim of this case study is to investigate the impact of using unreliable wireless communication channels between the algorithm and sensor units. Selex ES have provided information that defines the frequencies and durations of expected communications outages. Using this information, a Modelica model has been created that generates a discrete signal which obeys these probability distributions. The intention is that this model will be incorporated into the whole simulation to determine when a given SIU has suffered communications outage, and thus, the effect that the failure has on the algorithm can be explored. The model is parameterised by a seed for the pseudo-random generator, thus generating different plots that obey the same distributions only requires changing the seeds.

An example of the output of the model is plotted in Figure 29. Three communications channels are plotted over the course of a 24 hour period. The red and blue both fail intermittently for short periods, whereas the green permanently fails.

> The integration of housing load profiles and expected communication outages illustrate how real-world data can be used to aid verification within the ADVANCE framework.

Identifying undesirable effects due to temporary communication failures is of particular value to Selex ES. Modelling aspects of the entire system in a real world context helps to quantify the risk of communication outages. Emergent effects from communication outages are difficult to predict using traditional techniques, and are difficult to test empirically during real world tests due to the low probability of certain precursor conditions and the need for impracticably long test durations. Therefore the simulation capabilities of the ADVANCE framework offer a level of confidence that is not available to Selex ES from traditional testing techniques. The framework also provides visibility on the sequence of events leading up to undesirable conditions; something which is not necessarily available during real world testing, even if undesirable scenarios are encountered. This allows not only for the identification of undesirable scenarios, but also identification of their cause – and therefore

indication of the potential solutions. The effect of communication outages on the system is something Selex ES has identified as a risk, but have not been able to quantify. The ADVANCE framework provides the means to achieve this before the solution is implemented.

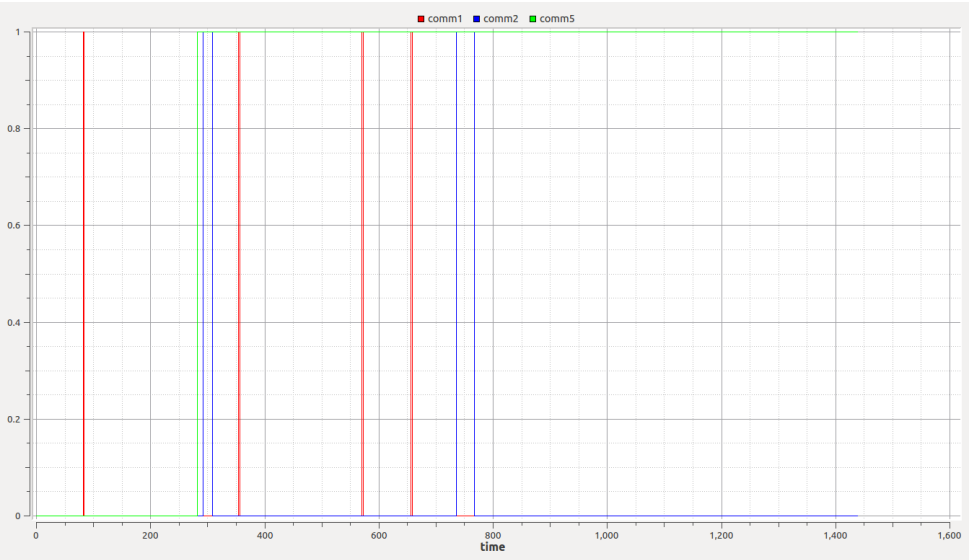


Figure 29: Communications Plot

B.5. Low Voltage Continuous Model

Following the required information furnished by Selex ES, work on modelling the specific networks for the initial deployment of the solution has started.

The models produced have been split into 4 groups,

- **Network** – Models the low voltage network distribution,
- **Tap Changer** – Models the automatic control process,
- **Medium Voltage** – Medium voltage inputs into the low voltage network,
- **End user** – Models the load and micro-generation of the end users.

These models are described in the following subsections.

B.5.1. Network

The first group, Network, is a top level view of the environment of the algorithm, and encapsulates the other models, see Figure 30. There is a voltage source at the top, which is connected to the OLTC, and then to three feeders. The loop at the OLTC that goes through the TapCon block represents the tap changer functionality, this is described in Section B.5.2. This loop automatically performs tap changes when the voltage is outside a valid range, which is set by the algorithm. The MVGen block is used to simulate the medium voltage network, and is described in Section B.5.3.

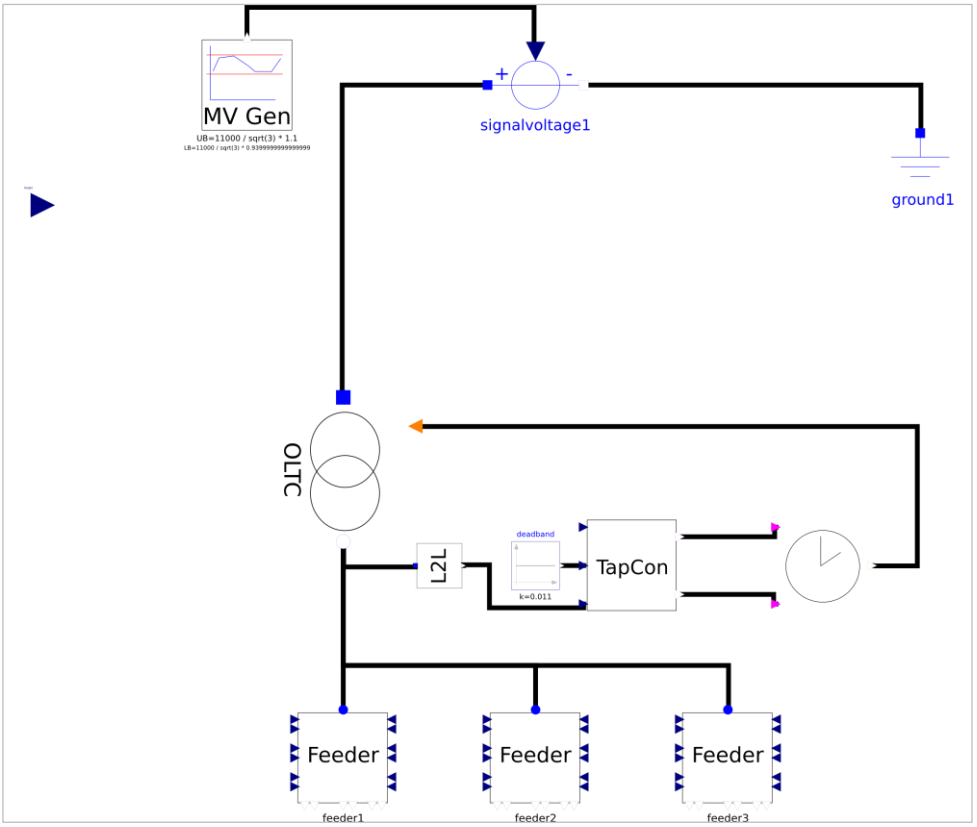


Figure 30: Birds eye view of network

The inputs and outputs of the continuous model are also defined in Figure 30, the blue triangle on the left identifies the input, *target*, computed by the algorithm. The outputs are less clear, but consist of 18 real values. These are the 6 white triangles at the bottom of each of the feeder blocks. Each feeder contains three lines, and each line has a mid- and end-point voltmeter, see Figure 31. These outputs have not been graphically linked (but are linked textually) to output ports, this is to keep the diagram uncluttered and comprehensible.

The remaining 12 blue triangles on each of the feeders are also inputs, but for the end user simulation. Each of the lines in the feeder takes 4 real valued inputs, two represent the end user demand of a block of houses and two represent the micro generation of the block of houses. This is depicted in Figure 31, where a block of houses are represented by a variable load and current source. That is, blocks LoadHouseA and GenHouseA represent the first group of houses situated between the transformer and the mid-point voltage, and blocks LoadHouseB and GenHouseB represent the second group of houses situated between the mid-point voltage and end of the line.

Figure 31 also includes models of transition lines to improve the accuracy of the models, and shows where the mid- and end-point voltmeters are located.

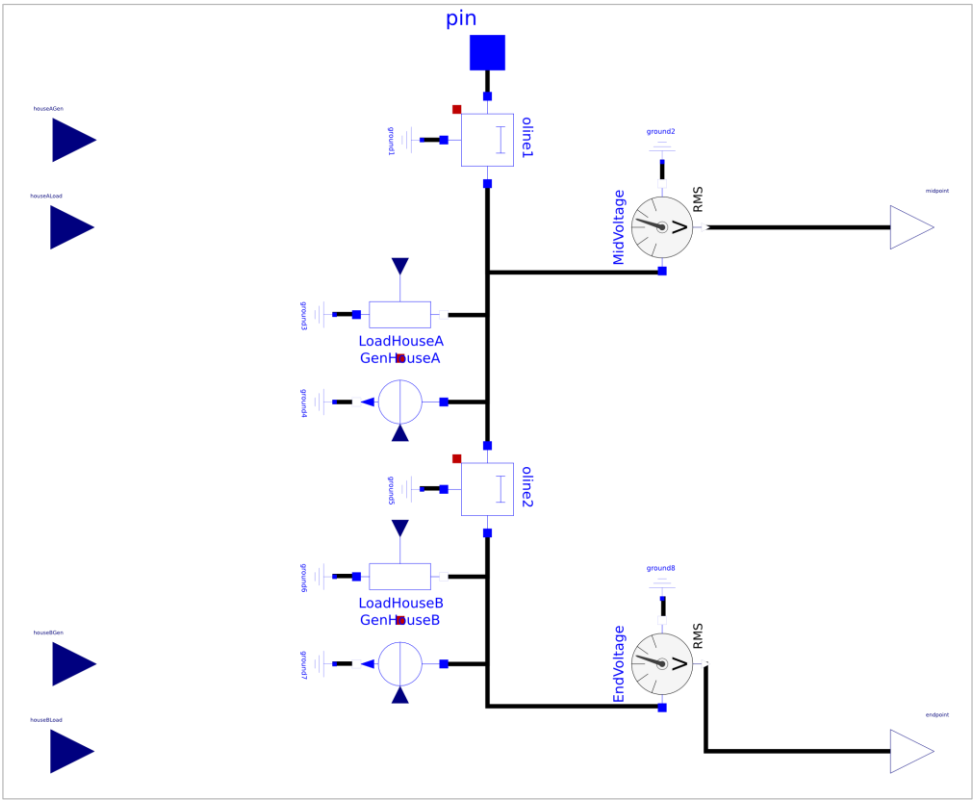


Figure 31: Feeder Line Model

B.5.2. Tap Changer

The tap changer is an automatic control device that attempts to maintain a stable power supply on the low voltage side of the transformer. It monitors the voltage level of the busbar, and compares it to a reference voltage, then following a number of rules, it determines whether to issue a tap change command to the OLTC transformer.

Internally the tap changer uses two timers (T, Ti) and a dead band (B). When the busbar voltage leaves the dead band the timer T is started, when it elapses, a tap change command is issued to the transformer. If the voltage returns within the dead band before the timer elapses, then a tap change command is not issued. Timer Ti determines how long the tap changer output to the transformer is active, and is used to drive the motor, it also represents any mechanical delay induced by the hardware. For a full description of the tap changer see [AD-7].

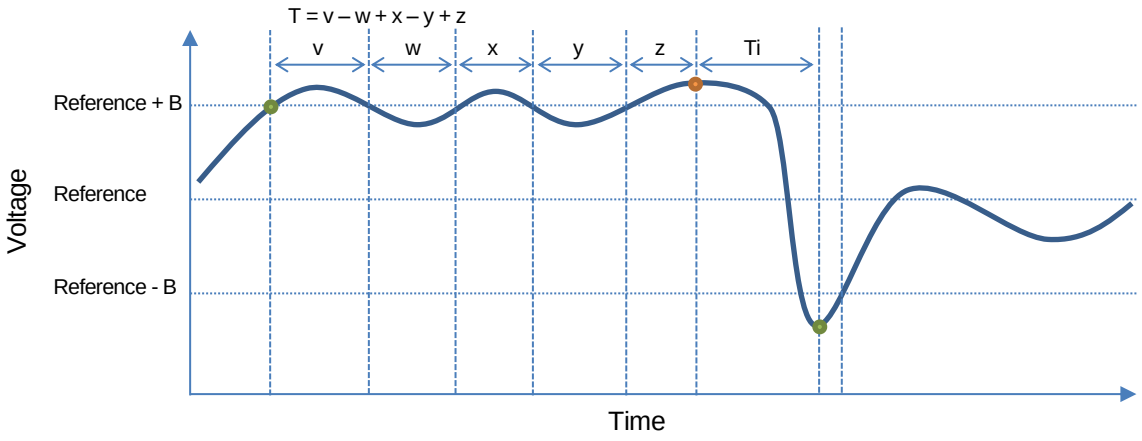


Figure 32: Tap Changer Behaviour

Figure 32 depicts the behaviour of the tap changer. The green circles indicate where timer T is started. Timer T counts in both directions, when the voltage is outside of the dead band it counts up and when the voltage is inside the dead band it counts down to a minimum of 0. The orange circle indicates when timer T elapsed and a tap change is issued to the transformer, also at the same time timer Ti is started. The tap change is not instantaneous, and the tap changer blocks until Ti has elapsed. At which point normal operation continues, and timer T is started again as the voltage is below the dead band. The intention is that the reference voltage will be calculated by the algorithm discussed previously.

For the purposes of this case study, the tap changer is considered part of the environment and has thus been modelled using state machines within Modelica. The tap changer has been split into two models, the first monitors the busbar voltage and decides whether there is a need to tap change up or down, and the second model counts the up and down requests. This counting is required as its value is used to lookup the transformer ratio information from a table. That is, the OLTC model is parameterised by a table of ratios between the primary and secondary coils, then using the value of the counter the exact ratio is selected.

The models of the voltage network and tap changer demonstrate how continuous models of the environment are used within the multi-simulation framework to validate the implementation of the algorithm and supporting system.

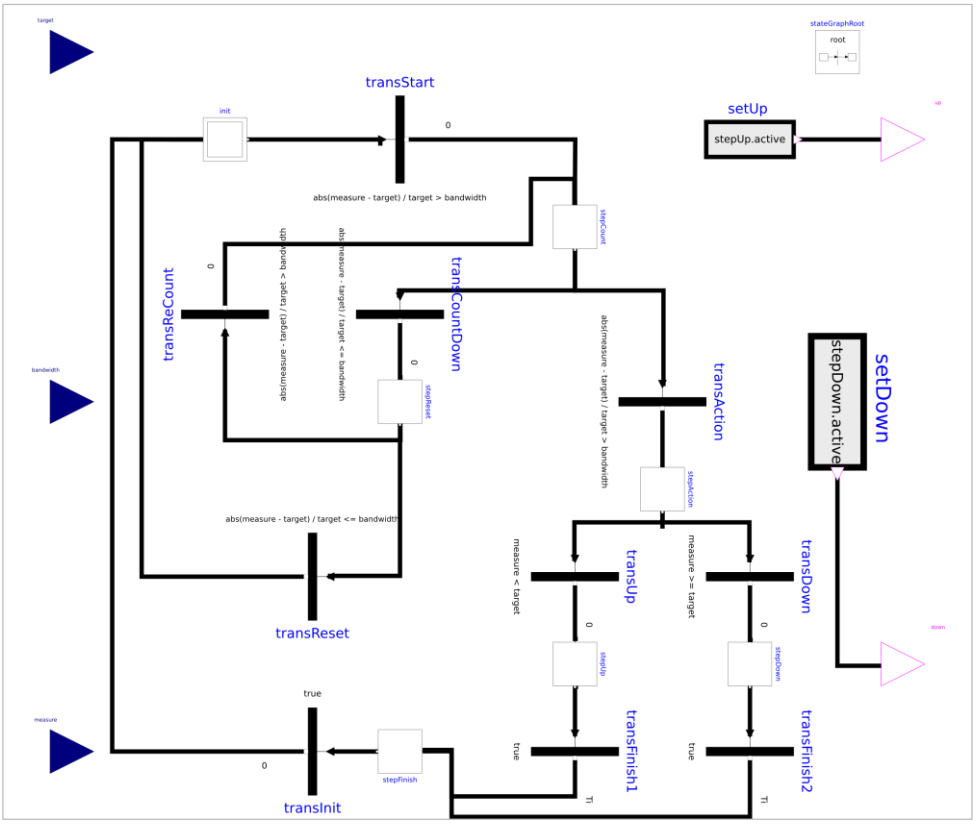


Figure 33: Model of Tap Changer

Figure 33 is the Modelica model of the tap changer, it has 3 real valued inputs (blue triangles on the left) and 2 Boolean valued outputs (pink triangles on the right). The inputs are the reference value, measured value and dead band, and the outputs are two signals, one for tap changing up and one for tap changing down. The creation of the above was mostly created using models from the standard library, however, to correctly implement the counting up and down of timer T, two of the transition models had to be customised. This was because the duration of the transition needed to vary, which the standard library models did not allow.

The second part of the tap changer model, see Figure 34, interfaces with the two Boolean valued outputs of Figure 33 and calculates an integer between a maximum and minimum. It is also implemented using a Modelica state machine, however not all the functionality was possible to do graphically and instead some textual modelling was required to specify the discrete method that the counter is calculated. This is depicted in Figure 35.

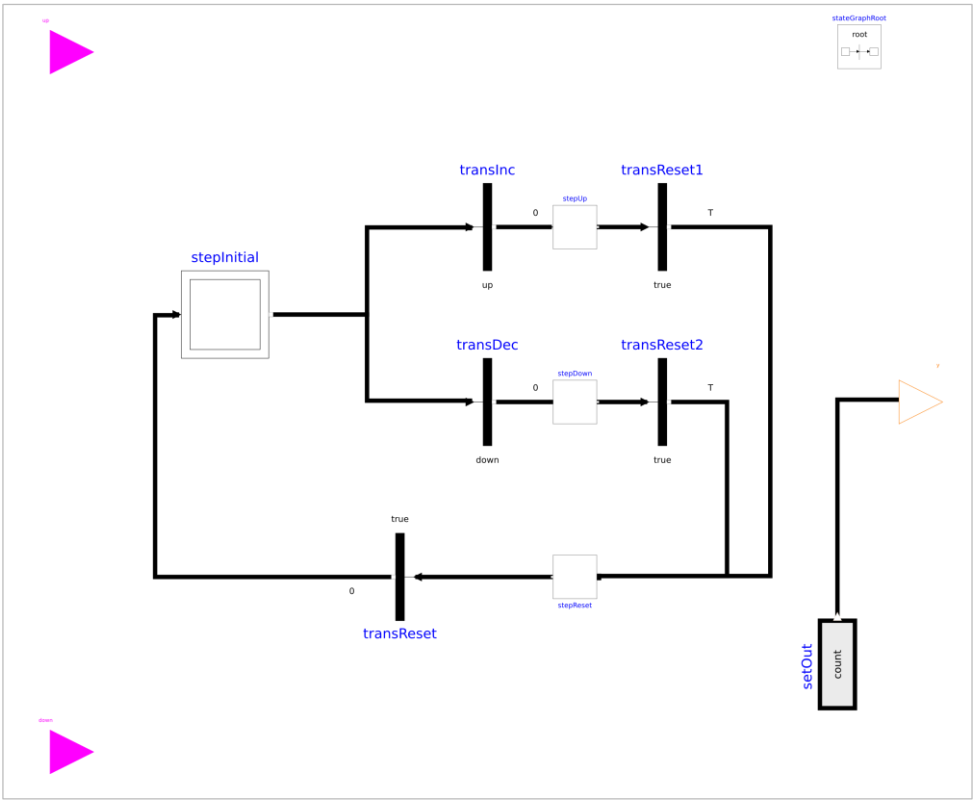


Figure 34: Counter Model Graphic

```
when stepUp.active or stepDown.active then
  count = min(max(pre(count) + (if stepUp.active then 1 else -1), LB), UB);
end when;
```

Figure 35: Counter Model Text

B.5.3. Medium Voltage Simulation

During the initial modelling of the power network, it was assumed that the input power was exactly 11KV (line-to-line). However, Selex ES expressed an interest in investigating non-ideal situations where there are permitted variations in the medium voltage network. According to EN 50160, the medium voltage must be within ±10% of the declared value (i.e. 11KV) 95% of the time [AD-8], however, as described in Section B.2.1, the sub-range of +10% to -6% is required.

Changing the voltage for the simulation to the upper bound, or lower bound is trivial. Instead, a more varied solution was sought, whereby sine waves are combined to produce a waveform that varies over the acceptable band. This was further improved by pseudo-randomly skewing the frequencies of the waves; the result of performing these operations is in Figure 36, with the original waves in Figure 37. By modifying parameters of the model, it is possible to move the peaks and troughs of the simulation to different times of the day, thus enabling the evaluation of the impact of voltage variations of the medium network.

In addition, a second solution was attempted where a Markov process was defined to model the medium voltage derivative. Small scale tests were carried out that proved the concept was technically viable within Modelica. Although this is a substantially more powerful approach, the time required to correctly define the transition criteria and subsequent

validation was not deemed worthwhile as the focus of the case study is on the low voltage network.

Ideally, it would be possible to replace this light-weight simulation by a model of the medium voltage transmission lines, and generation. However, this is beyond the scope of the current case study.

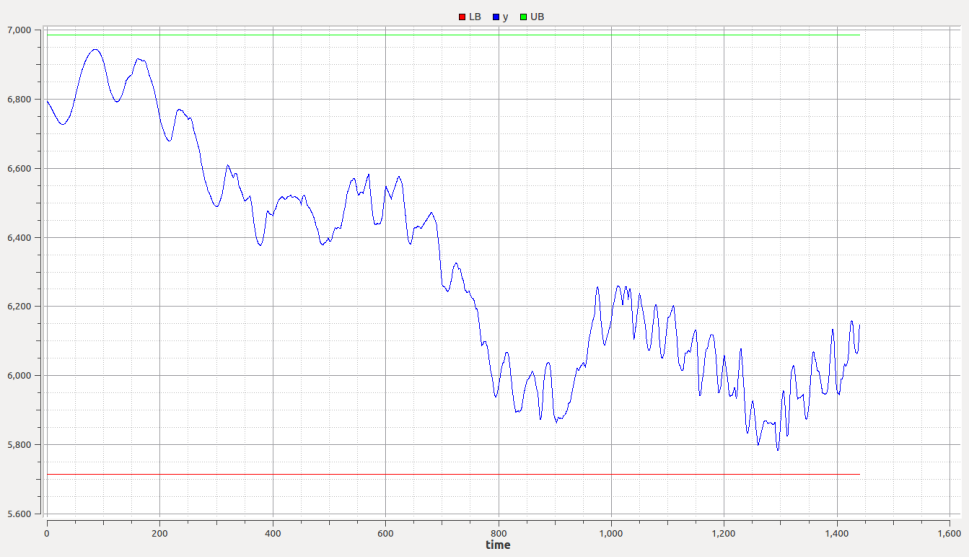


Figure 36: MV Simulation

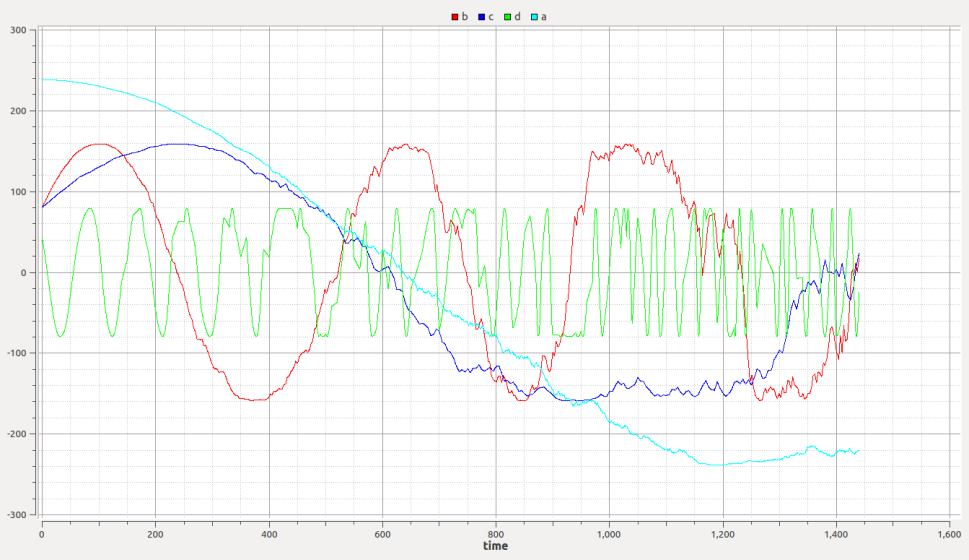


Figure 37: Individual Waves

B.5.4. End user demand and micro-generation

The end-user demand and micro-generation simulations were described in Section 5.2.3. They have not changed substantially since and are still based upon the models produced within the CREST project [AD-4]. However, following experiments with the ADVANCE tool set, invocation of the end-user models within the multi-simulation framework has changed.

Previously, the intention was to construct an FMU for a Haskell program that simulated the end-users, and then integrate this FMU using the framework developed in WP4. However, for technical reasons it was problematic to seamlessly integrate with the Modelica and Event-B components. The following issues were encountered:

- First, there is an limitation of the FMI standard, which means that the initial state of the FMU cannot be defined from the outside of the FMU. This problem is exhibited when considering the FMU's input variables. Before the initial step the variables are initialised using internal values (typically 0), and do not receive any values from outside the model until after the initial step has completed.

  This meant that the FMU produced for the network model could not depend upon values from the end user simulation (contained in a different FMU) during the initial step. This was troublesome as the initial state of the continuous network model did not draw any power and produced incorrect data for the discrete models.

- Secondly, the time taken for the whole simulation is substantially increased when performing the end-user simulation. Typically each house takes 1 minute to generate the underlying dataset, and there will be numerous houses in each group (currently 18 groups, see Section B.5). In addition to this constant overhead, and due to the FMI induced boundaries between the models, further inefficiencies were encountered that prevented the differential equation solver from taking advantage of the underlying structure directly – it had to query the external model at a very high granularity. Furthermore, it is often the case that these end user simulations do not need to be changed for each simulation, and it is possible to cache the results in Modelica of the end-user simulation between each simulation.

To fix these issues, the underlying Haskell program was modified to output a Modelica model representing the simulation of a house (or total of group of houses) over the course of a day. The simulation produces three tables, each with a real value for each minute of the day (i.e. 1440 values). The three tables represent the active power consumed, reactive power consumed, and the photo voltaic produced. This solution fixed both the identified issues. There is now an encompassing Modelica model of the whole low-voltage simulation, and the resources required to run the simulations are reduced.

To further automate this process, a script was created that took the specification of the groups of the end users and generated a Modelica package containing each of the individual models. This allows for a simple method to vary the end user simulations and low voltage network model independently. See Figure 38 for an example of the specification of the end users models, which the script takes as input. The model name column identifies the Modelica model name, and should not be changed to ensure compatibility between different end user simulations. The PV distribution column determines what percentage of houses in that group have PV installed. The final column defines the number of houses in that group.

```
# Model Name | PV distribution | Number of houses
  F1A          0.5               10
  F1B          0.2               8
  F1C          0.2               6
  F1D          0.2               7
  F1E          0.2               9
  F1F          0.2               18

  F2A          0.5               10
  F2B          0.2               21
  F2C          0.3               6
  F2D          0.2               8
  F2E          0.7               12
  F2F          0.1               18

  F3A          0.5               10
  F3B          0.2               5
  F3C          0.7               14
  F3D          0.2               7
  F3E          0.5               9
  F3F          0.2               15
```

Figure 38: End User Specification

The script that runs the end-user simulations is also responsible for ensuring that each simulation contains the same weather conditions, which is achieved by first generating a model of cloud coverage and then sharing it with each of the subsequent simulations.

The simulations also determine the number of occupants living in each house. This is achieved by basing the distribution of occupants on statistics obtained from [AD-9].

B.6. Multi-Simulation

In the previous sections, progress on the individual models has been reviewed. In this section information relating experiences of using the whole multi-simulation framework (i.e. simulating Modelica and Event-B models) are presented.

Since the last progress report, the issue of integrating Modelica models into Rodin has been explored. Following guidance from the University of Southampton the closed-source Modelica tool Dymola has been selected to produce FMUs compliant with Rodin. Simulink was investigated, but was not selected as it does not directly support Modelica and the Simulink models that Selex ES already had, that were used during the development of the SIU, were not relevant for the simulation that will be performed in this case study.

The simulation was specified using the component diagram view of the multi-simulation framework, see Figure 39. Currently the stochastic behaviour of the communications is not incorporated as the focus is on getting the basic simulation (i.e. the algorithm and low voltage network) running for the whole day, the communications models shall be incorporated subsequently. The orange boxes are Event-B machines and the green box is an FMU produced by Dymola. The grey boxes are output ports, and the light grey boxes are input ports. The lines link input and output ports together via connectors (bullet marks).

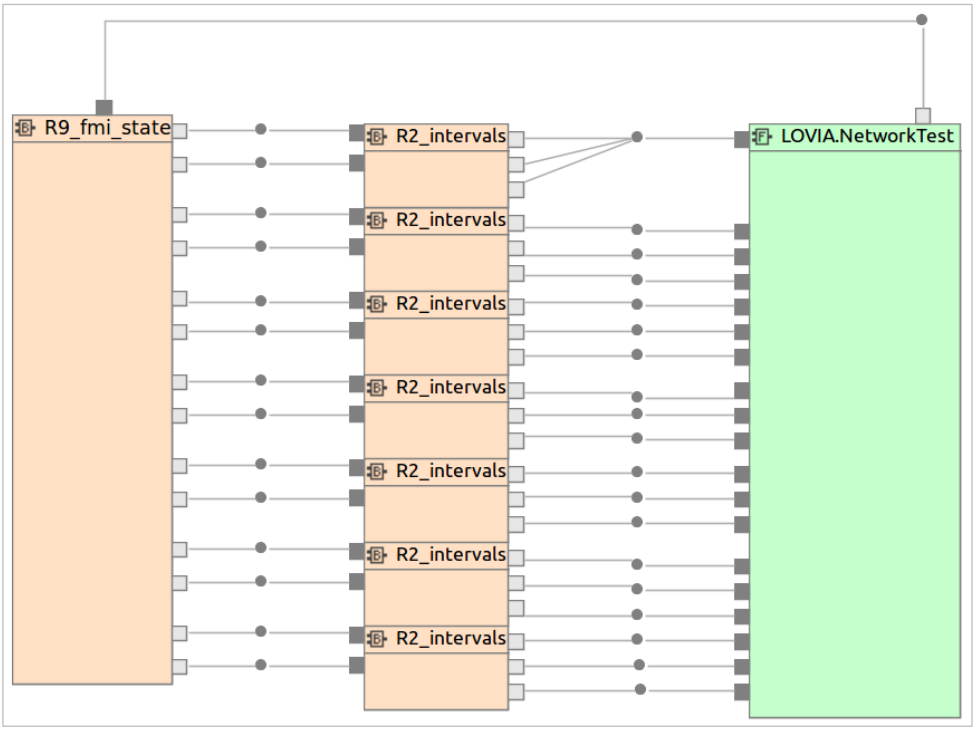


Figure 39: Component Diagram in Rodin

In Figure 39, the machine R9_fmi_state is a model of the algorithm, and the R2_intervals machines are abstract models of the SIUs and communications. The SIUs here take inputs from the continuous model every 0.5 time steps (i.e. 30 seconds) and average them over 1 time step (i.e. 1 minute), then report these averaged values to the algorithm. The algorithm takes these averaged values every 5 time steps (i.e. 5 minutes) and performs some execution on them to determine a new target, which is fed back to the FMU.

B.6.1. Simulation Results

Currently it has not been possible to run the simulation as defined in Figure 39 for the full time span of a single day, i.e. 0 to 1439. This is because of inefficiencies in the underlying framework that are exasperated by the model designs.

It has been required to simplify both the Modelica and Event-B models. This has allowed the simulation up to 500 time steps.

Modelica

Previously, the Modelica models contained faithful sine waves of the power grid (i.e. 50Hz at 230V RMS line-2-neutral and 11kV RMS line-2-line). However, the algorithm only requires RMS values of the voltage, which were computed from the sine waves. Performing this computation in Modelica is costly. Instead these sine waves have been abstracted away, such that the RMS values are used in their place.

As a knock on effect, the transformer models did not function correctly and instead had to be replaced by idealised transformer models. This means that properties such as transformer inductance are not considered.

Without performing this simplification the simulation of the Modelica models within the Modelica editors (OpenModelica and Dymola) took many hours.

Thus, the resulting Modelica models are essentially built around basic equations. The feeder model uses the V=IR and P=IV equations, and the transformer uses a=Vp/Vs=Is/Ip (where a is the turns ratio).

Event-B

The number of events in the Event-B machines had to be reduced. Previously, the models had many events that represented partial computations. In order to simplify the validation these partial computations were collapsed into single events, which improved the performance. Further, non-determinism in the Event-B machines needed to be reduced to help the master simulation algorithm in selecting which events it should pick next.

> A significant amount of feedback has been provided to the consortium in terms of the usability and current limitations of the multi-simulation framework when considering realistic industry-based models.

To improve the performance, other experiments were tried. This included combining the 7 R2_interval machines in Figure 39 into a single machine. However, this also suffered from inefficiencies within the underlying ProB interpreter, but relieved some of the burden on the master. A final experiment was that the R2_interval machines were replaced by simple Modelica models. This produced the best performance, and allowed the whole simulation to run. However, as this removed the Event-B specification of the behaviour of the SUI's, this cannot completely fulfil the validation objectives. The goal is to validate the algorithm, SIU's and communications networks in conjunction, not only the algorithm as is the case in this experiment. But this did allow initial results to be gathered while waiting for the performance issues to be fixed by UOS and UDUS. The first 500 minutes of the simulation are presented in Figure 40, where the red line is the busbar voltage (measured as line-to-line), the blue line is the target as set by the algorithm detailed in Section B.2.4, and the yellow and green lines represent the mid- and end-point voltages, respectively, of line 2 of feeder 3. It is clear from these results the framework is functional, and the models are behaving as expected.

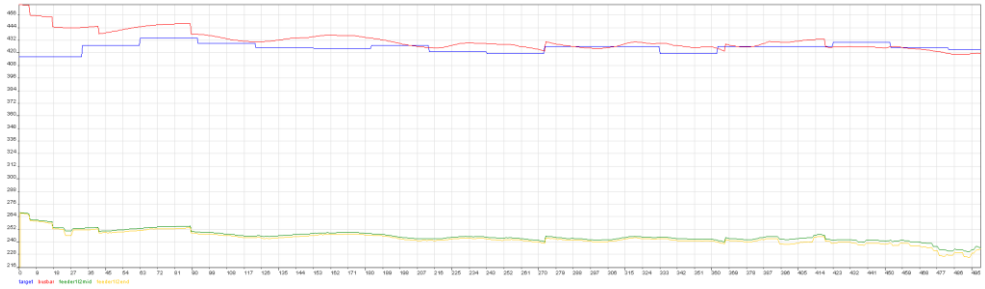


Figure 40: First Results

Complexity Analysis

After carrying out numerous experiments, it appears that the performance of the underlying simulation framework is quadratic, see Figure 41 for a plot. The blue line represents the time taken for the simulation to run, and the red line is $y=(x/100)^2$.

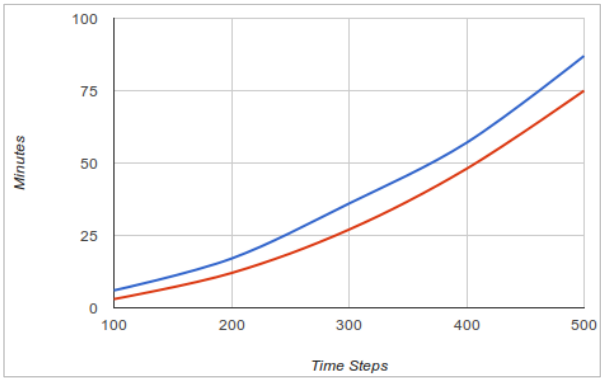


Figure 41: Time Complexity

Further, the memory consumption of the simulation, although apparently linear, is vast. For the same experiments in Figure 41, the memory consumption at 500 time steps was approximately 14GB. The relevant members of the ADVANCE consortium have been made aware of these limiting issues, and work is on-going to improve the situation.

B.6.2. Assessment of Advance tools

B.6.2.1. Planned Use of Advance tools

Elements of the ADVANCE framework are being utilised in the remaining work in the following way:

- *Composition/decomposition*: this forms an essential part of the current work, as several components of the system in the revised case study were previously modelled; these are incorporated into the system model via composition.
- *Theory plug-in*: we use concepts such as sequences and generalised summation in the Event-B models and will use the support provided by the theory plug-in and its integration with ProB for this.
- *Multi-simulation*: The distributed supply and demand on the low voltage network is being modelled continuously. Discrete sensor readings are taken from these continuous models as input to the Event-B SIU models, and further used as input to the models of the tap changer and algorithm. Therefore, the co-simulation between the continuous and discrete Event-B models is essential when validating the algorithm and other aspects of the solution.
- *Model-checking/constraint solving in ProB*: As mentioned in section 5.3, the model-checking and constraint solving abilities in ProB will be used to gain an evaluation of the optimisation of the algorithm. ProB will also be used for simulation and validation purposes, similar to the previous work, in particular to find any scenarios where the algorithm places the system in an undesired state.

In addition, the UML-B and ProR tools will continue to be used throughout the modelling process, as they have been during the work so far. Any requirements supplied by Selex ES will be recorded and maintained in ProR.

B.6.2.2. Tool Feedback

CSWT has provided feedback to the consortium on several issues, the main ones are identified below:

1. Multi-simulation

    a. Multi-simulation performance issues: The first multi-simulation could only be run for half of the required time before running out of memory (the testing system had 16GB, which was mainly consumed by ProB2).

    b. Feature suggestions for the component view in Rodin: The suggestions are mainly related to improving the user experience of the plugin, however, the following important issues were also identified:

        i. If an Event-B machine becomes deadlocked during the simulation, then the simulation is to non-terminate.

        ii. Traces of the Event-B models should be accessible via the user interface. Currently, only the plots of variables over time are provided.

        iii. Allow the models to be parameterised. This means that multiple instances of the same Event-B machine or FMU can be incorporated in the same diagram and instantiated differently, such as the SIUs. Also, the Modelica models could be parameterised with simulation settings that prevent the whole model being recompiled every time a different setting is required. E.g. for changing the MV input between simulations.

2. FMU creation and import, UML-B, and BMotion Studio.

3. Integration of ProB2 with the rest of the framework is causing issues, not only with respect to the performance issues identified in 1.a above, but also the lack of a release version. Currently ProB2 is only available as source, or a compiled nightly binary. The lack of a stable version has been mentioned to the consortium on a number of occasions, and continues to hamper the multi-simulation and the creation of animations with BMotion Studio (which is part of the ProB2 distribution).

Recommendations for next phases of work:

1. Continuous modelling solely in Modelica.

2. Move to 64-bit platform for simulation to take advantage of higher memory limits and further investigate performance issues.