Advance

SEVENTH FRAMEWORK PROGRAMME

www.advance-ict.eu

Advanced Design and Verification Environment for Cyber-physical System Engineering

Newsletter 1, April 2013

UNIVERSITY OF Southampton ALSTOM

Systerel Safe real-time solutions

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Critical software

INTRODUCTION

Current engineering practices make the design of cyber-physical systems to high assurance levels extremely expensive. It is widely recognised in both industry and government that development costs for future systems will become prohibitive unless there are significant improvements in the methods and tools used for systems engineering. ADVANCE is an FP7 ICT project aimed at delivering affordable methods and tools for formal modelling, verification, and validation that will reduce development costs in existing engineering, while improving the quality and safety of the developed system.

ADVANCE is building on an existing formal modelling language - Event- B - and its associated tools - Rodin - providing strong support for formal verification. In ADVANCE, Rodin is being further strengthened and augmented with novel approaches to multi-simulation and testing. The ADVANCE project is unique in addressing both simulation and formal verification within a single design framework. The formal modelling and verification leads to deeper understanding and more consistent specifications and designs than informal or semi-formal methods. Meanwhile simulation helps engineers to ensure that models are accurate representations of the desired functionality and physical components. The work of ADVANCE is being driven by two major industrial case studies, one on railway interlocking and the other on smart energy grids.

RAILWAY CASE STUDY – LEAD: ALSTOM

Alstom produces, for mainlines and urban railway operators, automatic train management systems involving four main components:

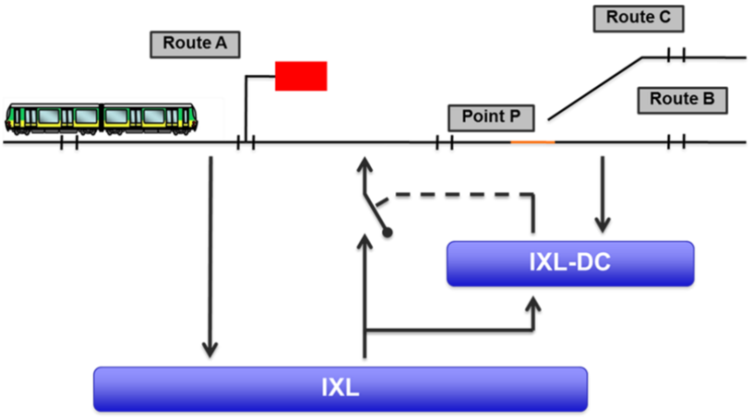
- Automatic train supervision systems (ATS), that assist railway operators to manage train movements according to predefined routes and timetables;
- Automatic train operation systems (ATO), that assist or even replace train drivers to operate trains optimally;
- Automatic train protection systems (ATP), that control the movements of trains in order to prevent accidents and
- Interlocking systems (IXL), that form, set and lock safe routes for trains within switching stations.

ATP and IXL are safety-critical systems given that a failure on their part may result in death or severe injury of people. Consequently, customers require manufacturers to provide evidence that no systematic error can result in this sort of failure. And, concomitantly, manufacturers constantly look for techniques able to eliminate such systematic errors and to provide evidence that they do so. Formal methods are part of these techniques and ATIS uses them to develop software of ATP and IXL systems.

The purpose of the ADVANCE Railway case study is to investigate an approach for the safety validation of IXL systems, based on Event-B.

THE INTERLOCKING DYNAMIC CONTROLLER (IXL-DC)

Ensuring train safety in a formal way with IXL equipment is very difficult. For that, we propose an approach that ensures train safety by introducing a software layer which filters out any IXL commands leading to a hazardous situation. This application, called the interlocking dynamic controller (IXL-DC), allows outputs of the interlocking to be controlled dynamically. The controller is positioned between the interlocking and the physical outputs. The figure below illustrates the role of the IXL-DC with respect of the IXL system. The IXL-DC will authorise delivery of IXL system outputs if and only if they satisfy IXL safety properties.



PROOF OF CONCEPT PHASE

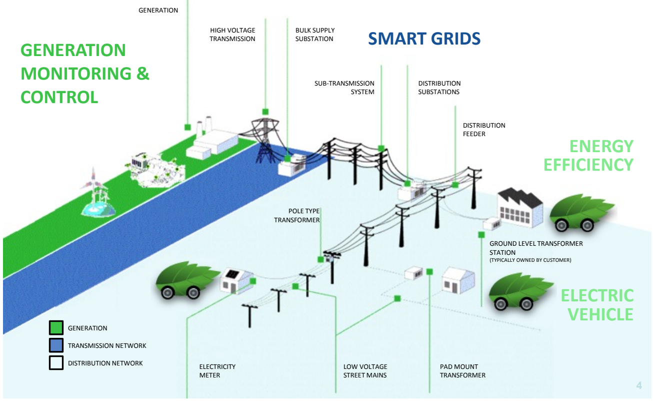
The proof of concept phase of the Railway case study started with the requirements definition. The focus then was to develop a new Interlocking Systems model whose purpose is explicitly to ensure that the safety requirements are met. We have developed an initial formal model to understand better the issues involved and to exercise the ADVANCE tools (i.e., Rodin and its plug-ins including ProB). This phase has been important because it allowed us to identify the main concepts that we had to handle in our further modelling and helped us to find good ways to model the dynamic controller.

Some key findings of this phase are that Event-B is an appropriate formalism for modelling the IXL-DC functionality at a system-level. The ProB animator and model checker was found to be invaluable in improving the accuracy of the model. We also found it useful to develop specific mathematical operators for domain-specific concepts as this eased the modelling and proof process. The Theory plug-in for Rodin, that supports the definition of new operators and proof rules, was used to good effect for this purpose.

SMART GRID CASE STUDY – LEAD: CRITICAL SOFTWARE TECHNOLOGIES

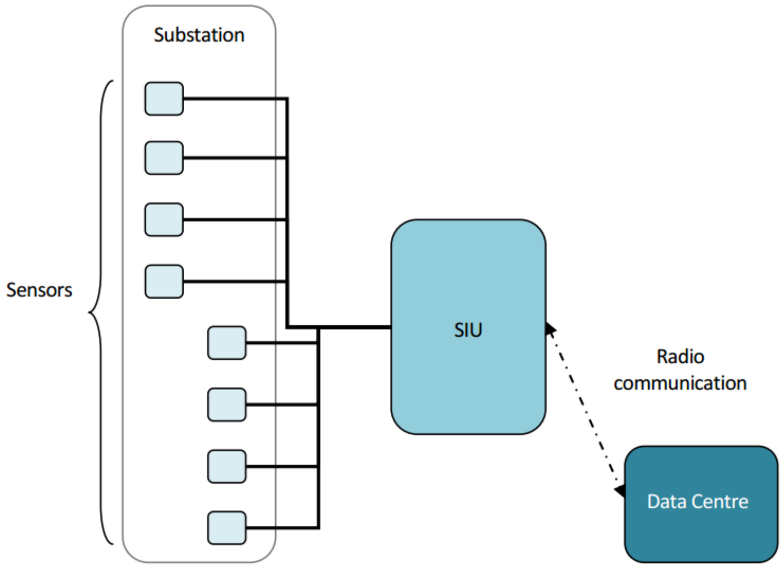
OVERVIEW

The differentiating factor of a smart grid in comparison to existing energy grids is the two-way communication between the consumer and supplier. This presents the opportunity for the consumer to have a more dynamic role in managing their energy use; by having access to up-to-date energy records and the option to switch between different rates and energy plans. It also allows for a more efficient and intelligent system on the supplier side; which can work around high demand and power outages by constantly updating the price and availability by considering the activity on the rest of the grid. The result is a reduction in the cost and waste of energy for both parties.



The proposition of the ADVANCE Smart Grid case study is to verify the security and reliability of the communication infrastructure on the distribution network of a smart grid system. In particular the case study is going to verify properties associated with low voltage substations system, data centre and smart metres. An important aspect that is being modelled is the data interchange between the consumers, suppliers and other native devices in the network. Formal methods present an ideal means of achieving this goal and establishing a best practice as the technology surrounding smart grid systems is still fairly loosely defined and attracts high development costs throughout the life cycle due to the traditional development methods used.

When in operation there will be a low voltage substation systems composed of a sensor Interface Unit (SIU) installations, each of which will take readings from multiple sensors, as indicated in the figure below. The SIUs will communicate with a single data centre. The focus of the proof of concept was on modelling the SIU, Network and Data Centre entities and the events associated with memory management, validation of data exchanged between the different entities, management of network faults, and the scaling up of SIUs. The current work is looking at incorporating in the model requirements defined by IEC 61850 and developing a continuous model of power generation and end user power demand.



FORMAL MODELLING STRATEGY

- The modelling strategy is based heavily upon the idea of decomposition. The combined system naturally lends itself to decomposition into separate models for the data centre, network, and monitoring/control units (in this case the SIU). Each of these models are developed independently and then composed again at later points to check that the composed system is a refinement of the original combined system. This is a suitable approach for distributed networks in general - and provides a strategy which can be reused - as the model for the combined system will be the same for any system of this type. It is only when work begins on the separate decomposed models that the attributes specific to each system are introduced.

- ## ACHIEVEMENTS

  The model for the data centre, SIU and network was created using the UML-B and State machine plug-in tools for Rodin, which provide the automatic translation of UML and state machine diagrams to Event-B. This was deemed an advantageous approach as the internal processes in the data centre are easiest represented using a process flow.
- Requirements Definition has been completed using the ProR Rodin plug-in using the method developed in ADVANCE. The ability to trace requirements with ProR was found to be invaluable for group working when the requirements are changed.

- Model development and refinement has used the iUML-B state machines that are under development in ADVANCE. This representation has been found to be clearer to those without much experience of formal languages.
- ProB animation was found to be particularly valuable, especially during the early stages of model development when it is important to get the fundamental behaviour of the model right.
- The model decomposition support provided by the Decomposition plug-in has also been used in the modeling to facilitate group working.

TOOL DEVELOPMENT IN ADVANCE – SOME HIGHLIGHTS

RODIN

The ADVANCE Project is building on the open source toolset for Event-B:

www.event-b.org

As well as maintaining the core Rodin platform, ADVANCE is developing new tool features as plug-ins to the Rodin platform relevant to engineering of cyberphysical systems. We highlight some of the tool developments here.

ADVANCE MULTISIMULATION FRAMEWORK

While mathematical proof is at the core of the Event-B method, simulation plays an important role in ensuring the validity of any formal model. It is not feasible to contemplate developing a single simulation language and verification environment that can meet all the requirements for cyber-physical development and verification. Legacy designs must be re-used and the specialised expertise of developers with existing tool chains leveraged. The primary objective of the ADVANCE multi-simulation framework is to address the needs for different design and verification tools, both discrete and continuous, test-based and formal to cooperate within a single development and verification framework.

While the ProB plug-in already provides a powerful tool for simulation of Event-B models, Event-B and ProB do not support continuous models as found in approaches such as Simulink and Modelica. We aim to support simulations of composition of models and implementations in other languages with Event-B models. To this end we have decided to adopt the Functional Mock-up Interface (FMI) Standard (www.fmi-standard.org) to support integration of simulation tools. We have developed a model composition framework that allows FMI compliant components to be composed with Event-B components together with a scripting environment for ProB that allows multi-simulation drivers to be defined. A first release of the ADVANCE multi-simulation framework is planned for autumn this year.

INTEGRATION OF THEORY AND PROB

Mathematical extensions have been co-developed by Systerel (for the Core Rodin Platform) and Southampton (for the Theory plug-in). The main purpose of the Theory plug-in new feature is to provide the Rodin user with a way to extend the standard Event-B mathematical language by supporting user-defined operators, basic predicates and algebraic data types. Along with these additional notations, the user can also define new proof rules (proof extensions). See wiki.event-b.org/index.php/Theory_Plug-in

Düsseldorf University is extending the ProB internal representation of predicates and expressions to support animation and model checking of new data types and operators defined using the Theory plug-in. In some cases, ProB will use the definition of new operators directly while in other cases it will provide a means to provide a more efficient implementation of an operator directly in Prolog (the implementation language of ProB). The integration of ProB and Theory is still in progress. An initial prototype has been built and tested for a few theory examples. A first release is planned for this summer.

PHYSICAL UNITS

Formal models of cyber physical systems will contain variables which represent values with physical units. Düsseldorf University is thus exploring to use the ProB model checker as a tool to infer and validate physical units usage in formal models. In particular, we want to make sure that the physical units in a model are used in a consistent way.

ProB's integrated plugin for unit analysis (4) can be used with Event-B machines from inside the Rodin platform. This includes

- Annotating variables with their physical unit,
- Infer the unit of variables without an annotation,
- Detect incorrect usage of units, like addition of differently annotated variables.

More details on the physical units feature can be found here:

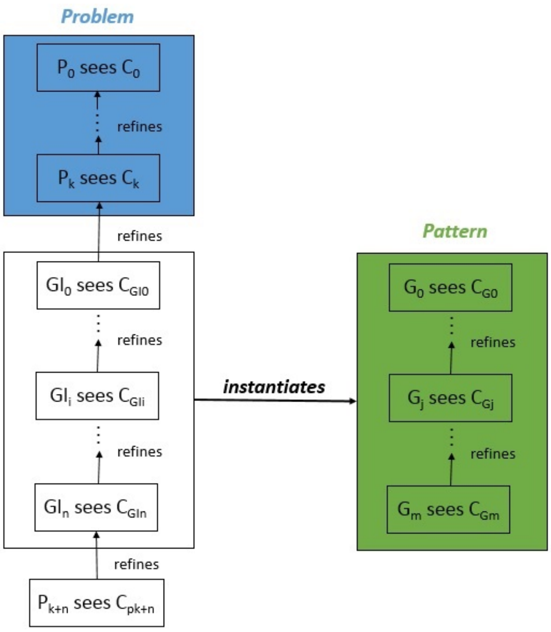
www.stups.uni-duesseldorf.de/ProB/index.php5/Tutorial_Unit_Plugin_With_Rodin

CONSTRAINT SOLVING

ProB's constraint solving capabilities are at the core of many of ProB's features: animation of high-level models with complicated predicates, model-based testing, constraint-based invariant and deadlock checking, etc. It is thus important to improve this aspect of ProB. In particular, we have continuously improved the performance of the kernel. Other improvements lie in better expansion of universal and existential quantifiers, reification for the *bool* operator and support for infinite and recursive functions. The latter is particularly important in light of the Theory plug-in work.

GENERIC INSTANTIATION

We have developed a way of treating Event-B models as generic and of instantiating generic models and chains of refinements. We define sufficient proof obligations to ensure that the proofs associated with a generic development remain valid in an instantiated development thus avoiding re-proofs.



The instances inherit properties from the generic development (pattern) and are parameterised by renaming/replacing those properties to specific instance element names (see diagram). Proof obligations are generated to ensure that assumptions used in the pattern are satisfied in the instantiation. In that sense our approach avoids re-proof of pattern proof obligations in the instantiation. The reusability of a development is expressed by instantiating a development (pattern) according to a more specific problem.

The Generic Instantiation (GI) plug-in developed by Southampton University supports instantiating and reusing generic developments in other formal developments:

wiki.event-b.org/index.php/Generic_Instantiation_Plug-in_User_Guide

Another generic instantiation tool has also developed by Hitachi Ltd. and ETH Zurich.

LINKING STPA AND EVENT-B

System-Theoretic Process Analysis (STPA) from Nancy Leveson is a technique for hazard analysis developed to identify more thoroughly the causal factors in complex safety-critical systems, including software design errors. Event-B is a proof-based modelling language and method that enables the development of specifications using a formal notion of refinement. We propose an approach to hazard analysis where system requirements are captured as monitored, controlled, mode and commanded phenomena and STPA is applied to the controlled phenomena to identify systematically the safety constraints. These are then represented formally in an Event-B specification which is amenable to formal refinement and proof. A paper on linking STPA and Event-B was presented at the 21st Safety-critical Systems Symposium in Bristol, UK (www.safety-club.org.uk/e210). In the next period, this approach will be applied to the ADVANCE case studies from WP1 and WP2.

DISSEMINATION

Southampton organised demonstration booths at two important European industrial events where the ADVANCE work was highlighted:

- 2013 Safety-critical Systems Symposium in Bristol
- DATE 2013 in Grenoble

The tools as well as the smart energy grid and railway case studies generated strong interest from participants at both these events.

CONTACT

If you are have any queries about the ADVANCE Project, please feel free to contact us:

Coordinator: Dr John Colley (J.L.Colley@ecs.soton.ac.uk)

Or visit our website:

www.advance-ict.eu