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

ADVANCE Deliverable D3.1 (Issue 2)

Tool Development Roadmap

Public Document

March 28, 2013

`http://www.advance-ict.eu`

Contributors:

Michael Butler	University of Southampton
Laurent Voisin	Systerel
Michael Leuschel	University of Dusseldorf
Fernando Mejia	Alstom
Jose Reis	Critical Software

Reviewers:

Jose Reis	Critical Software
Michael Leuschel	University of Dusseldorf
John Colley	University of Southampton

Contents

Preface

This deliverable provides a detailed roadmap for the tool development work of WP3 and WP4 for the remainder of the project. As described on the ADVANCE Description of Work (DoW), the tooling work will build on the Rodin plaform for analysis of Event-B models. Rodin is an open source development with extension points for addition of new tool feature plug-ins. This roadmap describes the planned development of enhancements that improve Rodin scalability and the availability of new proof tools, language extensions and composition/decomposition capabilities. It also details the timing of the delivery of the multi-simulation framework and ProB tool enhancements.

In the roadmap, we follow the structure of the DoW where most tool development is in Workpackages 3 and 4:

- WP3 Methods and Tools for Model Construction and Proof
- WP4 Methods and Tools for Simulation and Testing

In Chapter 1 of this roadmap we identity the tooling requirements coming from the case studies being developed in WP1 and WP2. In Chapter 2 we describe the planned WP3 tasks in more detail including mapping delivery of various tool developments to the planned WP3 deliverables and allocation of partner responsibility. In Chapter 3 we do the same for the WP4 tasks. The roadmap will help maintain coordination between the tool developers in ADVANCE and help the case study workpackages (WP1 and WP2) to plan their evaluation of the ADVANCE tools.

Chapter 1

Tooling requirements coming from WP1 and WP2 case studies

The initial roadmap was developed through a road-mapping session at the ADVANCE Kick-off Meeting held in November 2011 in Southampton involving all project partners. The aim of the roadmap is to provide more detailed planning of tool development in ADVANCE than described in the DoW.

Following the proof of concept phase, the case studies in the railway domain (WP1) and the smart energy grid domain (WP2) have identified mode detailed requirements on tools. We identify those requirements in this chapter and indicate the tasks in which each requirement is being addressed in the following tables. Based on these requirements we have refined and prioritised the tasks described in Chapters 2 and 3. The priority scale is High, Medium and Low. We expect to address all of the high and medium priority tasks and will endeavour to address the low priority tasks if time permits.

1.1 WP1 Dynamic Trusted Railway Interlocking Case Study

	Requirement	Task
1.1	Composition / decomposition is required in order to be able to manage the whole model and decompose it into its natural parts : IXL, IXL-DC, trains, trackside equipment and communication network	Task 3.5
1.2	Support for theories is mandatory to keep the model simple enough and provable with reasonable cost. This is because the concepts manipulated in IXL models are quite intricate and one needs to have the appropriate level of abstraction to write, reason about and prove correct one's model. It is also important that theories required in the case study are supported by ProB.	Task 3.4 Task 4.3
1.3	Advances in automated proof are always important with respect to modelling productivity. In particular, automated proof in presence of theories is very important for WP1 and does not work well at the time of project inception.	Task 3.3
1.4	Support for animation (together with composition/decomposition and theories) is required to permit demonstrating (by tests) that the model indeed corresponds to the intended system. It is also necessary for interacting with domain experts which are not fluent with the Event-B formalism.	Task 4.3
1.5	Model testing based on model animation scenarios is required to analyse the impact of small changes to the model in terms of availability and representativity of the intended system.	Task 4.4
1.6	Model based testing is required in order to verify that the prototype IXL-DC satisfies its specification.	Task 4.4
1.7	In order to validate the hypotheses used for specifying and modelling the system, it is necessary to be able to run side-by-side a mathematical model of train (e.g., of its braking system) together with a discrete model of the interlocking. This needs co-simulation.	Task 4.2
1.8	Maintain usability of Rodin platform.	Task 3.2

1.2 WP2 Smart Energy Grid Case Study

	Requirement	Task
2.1	ProB animation, model checking and automated proof have been demonstrated to be very effective on the Smart Energy Grid proof of concept models in validating models against requirements and eliminating basic errors. It is important that the level of automation of these tools continues to increase.	Task 4.3 Task 3.3
2.2	The Smart Grid proof of concept models used some ad-hoc definitions for operators such as generalised summation. These are now supported by the Theory plug-in. Integration between the Theory plug-in and ProB is important for animation of models that require additional theories.	Task 3.4 Task 4.3
2.3	It is important to be able to reflect the target architecture of the smart grid application in the formal modelling. To this end support for decomposition of models along architectural lines is important.	Task 3.5
2.4	In the second phase of the WP2 case study, we are developing continuous models of sub-station voltage fluctuations and we would like to be able to use this together with discrete models for overall simulation. To this end, support for multi-simulation of models involving Event-B and Modelica or Simulink is important.	Task 4.2
2.5	The diagrammatic modelling notations supported by UML-B provide a modelling language that is easier to present system behaviour to working engineers. A diagrammatic representation of the architectural view of models that represents component composition, including compositions of discrete and continuous models, would greatly improve usability of decomposition and multi-simulation.	Task 3.4
2.6	Diagrammatic animation using BMotionStudio is being explored in the second period as our view is that will make it easier to demonstrate models to stakeholders.	Task 4.3
2.6	Models will evolve as a development progresses so it is important to be able to re-validate modified models as automatically as possible. Support for scenario-based model testing is important for this.	Task 4.4
2.7	We intend to evaluate the effectiveness of code generation from Event-B Models for the WP2 case study and robustness and coverage of Event-B features by code generation will be important.	Task 4.5
2.8	Maintain usability of Rodin platform.	Task 3.2

Chapter 2

Methods and Tools for Model Construction and Proof

Methods and tools for model construction and proof are being provided by WP3 of ADVANCE. The objectives of WP3 are:

- to provide the methodological and tooling means for modelling Systems-of-Systems
- to provide expert formal proof support to the industrial partners
- to improve the usability and productivity of the Rodin platform to support larger-scale developments

We outline the WP3 deliverables as specified in the ADVANCE Description of Work. We then provide details of what needs to be undertaken for each of the WP3 tasks by describing sub-tasks, who is reponsible for those sub-tasks and in which WP3 deliverable they will be reported.

2.1 WP3 Deliverables

- D3.1 Tool development Roadmap (M3). This deliverable provides a detailed roadmap for the tool development work of WP3 and WP4 for the remainder of the project.
- D3.2 Methods and tools for model construction and proof I (M10). This deliverable will describe the maintenance actions carried out on the Rodin platform together with a summary of progress on the improvement of automated proof and model checking.
- D3.3 Methods and tools for model construction and proof II (M22). This deliverable will describe the maintenance actions carried through, together with an update on the progress of automated proof and model

checking. It will also present the achievements made in the area of language extensions and composition / decomposition.

- D3.4 Methods and tools for model construction and proof III (M30). This deliverable will provide a summary of the improvements made to the Rodin platform throughout the project.

2.2 Task T3.1 Tools Roadmapping (M1-M3)

This task is now completed and the result is reported in this deliverable.

2.3 Task T3.2 Platform maintenance (M1-M36)

This task runs throughout the whole project. It ensures that the Rodin platform stays usable, especially when being applied to larger-scale models. It involves both corrective and preventive maintenance of the tools, based on bug reports and feature requests from the partners. Platform maintenance consists of the following list of sub-tasks (the partner responsible is indicated in brackets after each sub-task):

1. Answer questions from ADVANCE partners on usage of the Rodin platform on the project mailing list. **(Systerel)**
   Priority: High

2. Process feature requests and platform issues identified by ADVANCE tool users and tool developers in this and other workpackages. This will involve prioritisation of bug fixes and requests and identification of the partner responsible for undertaking the fix or request. It will also involve definition of a clear specification of the fix or feature. Management of all bug reports and feature requests is tracked using Sourceforge. **(Systerel)**
   Priority: High

3. Maintain and evolve model editors, including the text editor. This will include improving the usability of the editors as well as supporting language extensions that are developed by the ADVANCE project. **(Systerel/Dusseldorf)**
   Priority: Medium

4. Ensure Event-B handbook remains consistent with platform evolution. **(Systerel/Dusseldorf)**
   Priority: Medium

   Regular progress reports on each of these maintenance tasks will be reported in WP3 Deliverables D3.2, D3.3 and D3.4.

2.4 Task T3.3 Automated Proof and Model-checking (M4-M30)

The new modelling domains in WP1 and WP2 will require new automated proof capabilities over and above the existing capabilities of Rodin. We describe in more detail the sub-tasks being undertaken in order to achieve this. After each sub-task we list the partners responsible along with the deliverable in which the results of the sub-task will be reported.

1. Implement new or improve existing automated proof tools, linking to state of the art external tools (off-the shelf first-order theorem provers and SMT solvers).

   (a) Systerel have undertaken some experiments with using first-order automated provers but the experience was not so positive. The main difficulty here is the gap between the languages supported by off-the-shelf first-order provers and the Event-B mathematical language. Developments in first-order provers will be tracked but for now most effort in this area will be on use of SMT technology. Systerel have developed a prototype SMT plug-in that currently supports reasoning about integers. Recent developments in SMT solvers are leading to support for some structures in set theory. We will exploit these to enrich the subset of the Event-B mathematical language supported by the SMT plug-in.
   (Deliverable D3.2, D3.3, Systerel)
   Priority: Medium

   (b) Experiments will be performed to assess the effectiveness of the use of SMT solvers and comparisons with the existing Rodin provers will be made.
   (Deliverable D3.2, D3.3, Systerel)
   Priority: Medium

2. Provide expert support on how to use the provers for increasing the ratio of automated proofs.

   (a) Develop specialised prover tactics to support the case study work in WP1 and WP2.
   (Deliverable D3.2, Systerel)
   Priority: Medium

   (b) Add material to the user handbook in the form of an FAQ that provides tips on optimisation of prover tactics.
   (Deliverable D3.2, Systerel)
   Priority: Medium

3. Improve the model checking tools so that the proportion of the search space that can feasibly be covered gets increased.

   (a) Develop methods for combining SAT/SMT with ProB constraint based model checking
   (Deliverable D3.2, D3.3 Dusseldorf)
   Priority: Medium

   (b) Develop methods for state space compression and state hashing to improve efficiency of model checking
   (Deliverable D3.3, Dusseldorf)
   Priority: High

4. The existing theory plug-in for Rodin supports extension of the mathematical language and proof rules through user-defined theories.

   (a) We will develop links between the theory plug-in and the SMT plug-in so that user-defined theories can be mapped to SMT.
   (Deliverable D3.3, Southampton/Systerel)
   Priority: Low

   (b) We will develop links between the theory plug-in and the ProB model-checking plug-in so that user-defined theories can be mapped to ProB, allowing model with user-defined theories to be animated and model-checked effectively.
   (Deliverable D3.3, Southampton/Dusseldorf)
   Priority: High

   (c) We will develop links between the theory plug-in and the user tactic language. These will give users more control over which theories get applied for particular models allowing for better optimisation in automatic proof.
   (Deliverable D3.3, Southampton/Systerel)
   Priority: Medium

2.5 Task T3.4 Language extension (M13-M36)

1. Enrich the Event-B mathematical language with constructs and mathematical theories that are better suited to modelling of cyber-physical systems.

   (a) Develop a theory of real arithmetic.
   (Deliverable D3.3, Southampton/Systerel)
   Priority: Medium

   (b) Develop other domain-specific theories as the need arises on the WP1 and WP2 case studies.

(Deliverable D3.4, All partners)
Priority: High

2. Extend the Event-B language where necessary to improve expressiveness.

   (a) Currently Rodin does not automatically generate proof obligations for checking preservation of enabledness (e.g. absence of deadlock) though users sometimes add these manually. We will develop systematic, flexible generation of enabledness-preservation proof obligations along with appropriate automated proof tactics.
   (Deliverable D3.3, Southampton/Systerel/Dusseldorf)
   Priority: Low

   (b) Event-B is currently best suited to modelling discrete behaviour but many cyber-physical systems involve combinations of discrete and continuous behaviour. We will develop techniques for linking modelling of discrete behaviour with continuous behaviour.
   (Deliverable D3.3, Southampton/Systerel)
   Priority: High

   (c) Develop Event-B extensions to specify temporal properties along with associated proof and model-checking techniques.
   (Deliverable D3.3, Dusseldorf/Systerel)
   Priority: Low

   (d) Alstom and Critical Software Technologies have a strong interest in graphical modelling languages such as UML and SysUML. We will extend the existing UML-B plug-in to support additional UML and SysML features, in particular, support for architectural diagrams, and develop mappings to Event-B so they can be formalised and analysed.
   (Deliverable D3.3, Southampton)
   Priority: High

3. Formal design patterns will arise from the case studies of WP1 and WP2. WP3 will provide support for integrating these in a library of re-usable patterns to improve productivity.

   (a) Develop mechanism for management and instantiation of generic modelling and refinement patterns.
   (Deliverable D3.3, Southampton/Systerel)
   Priority: High

   (b) Develop patterns for systematic modelling and refinement of timing properties.
   (Deliverable D3.3, Southampton/Systerel)
   Priority: Low

(c) Develop other modelling and refinement patterns as they arise in the WP1 and WP2 case studies.
(Deliverable D3.3, D3.4, All partners)
Priority: Medium

2.6 Task T3.5 Composition and decomposition (M13-M36)

1. Enhance the existing Event-B composition and decomposition methods and tools in order to support the typical system architecture approaches used in cyber-physical systems particularly targeting the case study architectures.

   (a) Support for composition and decomposition of Event-B models already exists and has been applied to small-scale examples. We need to apply this to larger examples in order to understand any limitations it may have in terms of scalability and useability. Limitations will be identified and desirable features identified. These will provide requirements for further tool improvements.
   (Deliverable D3.2, Southampton/Systerel)
   Priority: High

   (b) Implement additional tool support for composition and decomposition of Event-B models specified by previous item.
   (Deliverable D3.3, Southampton/Systerel)
   Priority: High

   (c) Provide support for propagating changes in abstract models down through decomposition, including support for shared event and shared variable composition in composed machines and renaming.
   (Deliverable D3.3, Southampton)
   Priority: Medium

   (d) Ensure that the model decomposition mechanisms are supported by the multi-simulation framework of WP4.
   (Deliverable D3.3, Southampton)
   Priority: High

2. Develop a cookbook describing how to use composition and decomposition to facilitate team working on large projects.

   (a) Rodin currently has support for version control of Event-B models including merging different branches. However this does not support merging of proofs or model checking performed on separate branches. We will develop method and tools to support team working that links the compositional reasoning methods supported by Event-B composition and decomposition with version

control mechanisms.
(Deliverable D3.3, Southampton/Dusseldorf/Systerel)
Priority: Medium

Chapter 3

Methods and Tools for Simulation and Testing

Methods and tools for simulation and testing are being provided by WP4 of ADVANCE. The objectives of WP4 are

- Develop a multi-simulation framework
- Scale up ProB to be able to deal with large designs (validated in the context of high-level descriptions of other systems) and very complicated constraints
- Extend ProB's model-based testing to accommodate constrained random testing
- Enable rich code-generation from a higher-level than the current state-of-the-art

3.1 WP4 Deliverables

- D4.1 Specification of multi-simulation framework (M3). This is completed.
- D4.2 Methods and tools for simulation and testing I (M13). This deliverable will describe the prototype multi-simulation framework, as well as the result of first experiments. A roadmap for implementing the full-blown multi-simulation framework will be presented.
- D4.3 Methods and tools for simulation and testing II (M25). This deliverable will describe the final multi-simulation framework. It will also describe the first steps in model based testing and code generation.

- D4.4 Methods and tools for simulation and testing III (M30). This deliverable will cover the final version of all simulation, code generation and testing tools.

3.2 Task T4.1 Specification of multi-simulation framework (M1-M3)

This task is completed and is reported in Deliverable D4.1. This deliverable describes the various simulation tools and techniques that will have to be combined to provide an effective multi-simulation framework. The specification focuses on the structural mechanism needed to integrate a simulation tool into the framework, possibly as a plug-in, and the communication mechanisms that need to be supported to ensure the efficient transfer of data between co-operating simulation tools. The requirements for the final multi-simulation framework are expressed and an architecture is specified.

3.3 Task T4.2 Multi-simulation framework development (M4-M30)

1. Prototype multi-simulation framework.
   (a) Experimental prototypes to evaluate strategies for implementing requirements described in D4.1. The evaluation will include prototype framework implementations in C, Java and Prolog. These will be evaluated with respect to performance and ease of integration with architectural requirements identified in D4.1 and a final decision about implementation strategy will be made.
   (Deliverable D4.2, Southampton/Dusseldorf)
   Priority: High
   (b) Support co-simulation of composite system model consisting of combinations of abstract Event-B components and implementation level components in C or Java. This will require the ability to co-ordinate simulation of implementation components together with ProB simulation of the abstract components.
   (Deliverable D4.2, Southampton/Dusseldorf)
   Priority: High
   (c) Explore technologies for simulation of continuous models including Simulink and Modelica. Develop techniques for linking continuous models with discrete Event-B models and specify techniques for linking discrete simulation of Event-B with continuous simulation tools.
   (Deliverable D4.2, Southampton)
   Priority: High

(d) Elaborate the specification of the multi-simulation framework in light of the prototyping effort. A decision has been made to use the Functional Mock-up Interface (FMI) standard for co-simulation using different simulation engines. The FMI standard is supported by a growing number of simulation tools.
(Deliverable D4.2, Southampton/Dusseldorf)
Priority: High

2. Final multi-simulation framework.

   (a) Develop robust, flexible multi-simulation framework that supports combined simulation of assemblies of heterogeneous components that may be continuous or discrete or both and may be abstract models or implementations. This will use the FMI standard as the interface mechanism for different simulation executables. Support for driving FMI simulation components will be provided through the scripting environment for ProB.
   (Deliverable D4.3, Southampton/Dusseldorf)
   Priority: High

   (b) Develop guidelines on effective usage of the multi-simulation framework.
   (Deliverable D4.4, Southampton/Dusseldorf)
   Priority: High

3.4 Task T4.3 Model simulation with ProB (M4-M30)

1. Improve scalability of ProB to deal with large hardware (physical components) models.

   (a) This will require an analysis of the modelling structures that are best suited to modelling physical components and development of efficient encoding of these structures in the ProB kernel.
   (Deliverable D4.2, Dusseldorf)
   Priority: High

   (b) Identify specific performance bottlenecks arising in WP1 and WP2 case studies and tune the performance of the ProB kernel to address these issues.
   (Deliverable D4.3, Dusseldorf)
   Priority: High

2. Improve the constraint-solving kernel of ProB to enable model-based testing (Task T4.4).

(a) Improvements will be achieved through the implementation of cardinality analysis, symmetry breaking and instrumentation for coverage measurement.
(Deliverable D4.3, Dusseldorf)
Priority: High

3. Development of new visualisation techniques to aid humans understand large-scale simulations.

   (a) The BMotion Studio developed in the DEPLOY project enables tailored graphical animation of high-level Event-B models. We will develop a palette of domain-specific graphical animation structures to support the WP1 and WP2 case studies.
   (Deliverable D4.3, Dusseldorf)
   Priority: High

   (b) Understanding the state space explored in a large-scale simulation can be difficult for humans. We will explore graph-based methods for visualisation of large state-spaces and event-trace sets and develop experimental prototypes.
   (Deliverable D4.4, Dusseldorf)
   Priority: Low

3.5 Task T4.4 Model-based testing (M4-M36)

1. Extend the Model-based testing (MBT) framework to accommodate random testing.

   (a) A Model-based testing (MBT) framework for Rodin has been developed in the DEPLOY project, but was tailored to the specific needs of the industrial partners in DEPLOY (in particular SAP). Also, this framework does not support random testing. The framework will be made more general and make use, amongst others, of new random enumeration algorithms inside the ProB kernel.
   (Deliverable D4.3, Dusseldorf)
   Priority: Medium

2. Link MBT framework to multi-simulation framework.

   (a) A requirement coming from the case studies in WP1 and WP2 is the ability to perform scenario-based testing of formal models for validation purposes (see Requirements 1.5, 1.6 in Section 1.1 and Requirement 2.6 in Section 1.2). That is, model testing rather than model-based testing of implementations.
   We will address these requirements by ensuring that the MBT

framework conforms to the FMI-based architecture and communication definition of the multi-simulation framework so that large-scale testing can be driven via the multi-simulation framework. Scenarios for driving the simulation can then be defined in a number of ways as FMI components (such as scripts, UML-B statemachines, implementations, etc).
A first version of model testing will already be provided in **(Deliverable D4.3, Dusseldorf/Southampton)** and the final version in **(Deliverable D4.4, Dusseldorf/Southampton)**
Priority: High

3. Ensuring high levels of coverage of functional requirements in the test sets for a system implementation will make an important contribution to future certification. To support this, we will extend ProB's coverage detection for the multi-simulation framework and develop a constrained random testing tool.

   (a) Support specific coverage criteria on event parameters and on machine variables including data coverage, boundary value analysis and data partitioning. Extend the random testing feature of the MBT framework to enable constrained random testing that is driven by the required coverage criteria.
   (Deliverable D4.3, Dusseldorf)
   Priority: Medium

   (b) Explore the development of user annotations to support user definition of additional coverage criteria.
   (Deliverable D4.4, Dusseldorf)
   Priority: Medium

3.6 Task T4.5 Code generation (M13-M30)

1. Code generation from high-level models.

   (a) A code generation framework has been developed as part of the DEPLOY project that supports user-defined translation rules for data types and operators. We need to analyse an initial set of datatypes required in the WP1 and WP2 case studies and enrich the translation rule set to support these. Target languages include Ada and C.
   (Deliverable D4.2, Southampton)
   Priority: High

   (b) Continue to track typical data type usage in WP1 and WP2 case studies and extend translation rule set to support these.

(Deliverable D4.3, Southampton)
Priority: High

2. Link code generation to multi-simulation framework.

   (a) This will involve ensuring that the code generation framework conforms to the FMI-based architecture and communication definition of the multi-simulation framework using.
   (Deliverable D4.3, Southampton)
   Priority: Medium

3. Generation of stand-alone prototypes from high-level models using the ProB kernel.

   (a) For abstract models with nondeterministic event ordering, specific ordering strategies providing required coverage criteria can be inferred using the ProB kernel. These strategies can be translated into stand-alone executables that can be than run independently of the ProB kernel.
   (Deliverable D4.3, Dusseldorf)
   Priority: Medium