DEPLOY PROJECT

Newsletter N°8

Contents

Partners

Completion

Started in February 2008, the DEPLOY project, probably the biggest Formal Method project ever funded by the European Community, is finishing in April 2012. During more than 4 years, 86 researchers and engineers from 14 partners have collaborated to the development, improvement and assessment of the Rodin platform, plugins, and educational materials, while sharing the common objective of easing its adoption by Industry.

On the tool side, the Rodin platform 2.4 is out, the next release is expected in April 2012. A company, Rodin Tools Ltd (http://www.rodintools.org), has been created recently as a Not for Profit Company taking over responsibility for the Rodin toolset at the end of DEPLOY. A community of developers has emerged over the last years, contributing to a number of plugins.

On the documentation side, several resources are available:

- DEPLOY publications, reports, tutorials and projects/models, developed by the project members, for a total of 280 items (http://deploy-eprints.ecs.soton.ac.uk).
- DEPLOY technology transfer wiki containing all user's documentation for the Rodin platform (http://wiki.event-b.org).
- Rodin User's Handbook (http://handbook.event-b.org).

Concerning the industrial assessment of Rodin and Event-B, several reports have been produced by our four industrial partners (http://www.deploy-project.eu/html/deliverables.html). Their opinion is summarized later on in this newsletter. In parallel, an Open repository of evidence for adopting Formal Methods in industry has been set up (http://www.fm4industry.org), collecting feedback from an audience larger than DEPLOY.

After 7 years in Rodin and DEPLOY ICT projects, we can ensure that Rodin has never been so close to industry-ready state. Rodin is attracting a lot of interest regarding audience captured by technical workshops, conferences, and dissemination events organized worldwide. In a word, Rodin is ready to survive the end of DEPLOY.

Thierry Lecomte
ClearSy

Deployment in the Automotive Sector

Robert Bosch GmbH is one of the world's largest automotive suppliers and develops embedded control systems for the automotive domain among many other products. Reliability, dependability and safety of these embedded control systems are essential and have to meet highest standards. Today, testing is used to achieve dependability, reliability and safety. However, the increase in system complexity means that the effort of testing will grow and therefore will become uneconomical in the future.

The goal for Bosch in DEPLOY is to evaluate whether formal methods in general and Event-B in particular can be applied for the development and verification of automotive systems. This requires that the formal method is embedded into the existing development process including requirements development, architectural design, implementation as well as adequate tool support for embedded automotive software systems of industrial size. At the beginning of the project, Bosch had several people involved in the DEPLOY project with a background in software engineering but no particular knowledge in Event-B and Rodin. The big challenge was to evaluate whether formal methods can be applied in general and how formal methods and tools could be integrated into existing automotive development processes which in turn required to find answers to the following questions:

1. *Which classes of automotive systems can be formally specified with Event-B and Rodin?*

2. *What level of tool and methodological support can we expect for the different phases in the development process?*

3. *What types of properties of automotive systems can be verified with Event-B and Rodin?*

During the last four years Bosch has been working on these questions by applying the Event-B methodology and the Rodin toolset to two pilot applications (cruise control and start/stop software) taken from the automotive domain. The first priority of Bosch was to identify to which classes of automotive systems the Event-B method and Rodin could be applied. After evaluating Event-B on the cruise control and the start/stop software we found out that Event-B is in general suitable for formal specification and verification of discrete parts of automotive systems, i.e., the parts dealing with the control of the application which are typically implemented by state machines. However, there are still some open issues which prevent the Rodin tool to be used in industrial contexts (e.g. support for team development, scalability of the tools for large projects, and stability of the tools). We also experimented with the formal specification of continuous parts and time but found Event-B not ideally suited for these aspects.

A second priority of Bosch was to understand how formal methods in general and especially Event-B and Rodin could be integrated into the existing development process in the automotive industry. During both pilots Bosch realized that there is a large gap between the requirements of a system and a formal specification. Thus, evaluating methods and tools to close this gap has been an important task for Bosch.

During the first pilot application natural language requirements were modelled using the Problem Frames approach, a semi-formal diagrammatic requirements notation that allows the problem to be decomposed into simpler sub-problems which can be analysed in isolation before looking at the composition of these sub-problems. Applying the Problem Frames approach significantly reduced the gap between

natural language requirements and a formal model in Event-B. However, the mapping between Problem Frame diagrams and Event-B machines turned out to be not as simple as expected. Especially describing state machines – which are contained in many automotive applications – as textual requirements turned out to be difficult to maintain. Furthermore, the problem structure resulting from the Problem Frame diagrams did not always fit the structure of the solution which made it more difficult to directly map Problem Frame elements to Event-B elements. Therefore, Bosch decided to introduce another step between requirements development and formal modelling, namely specification and design in the second pilot application. The idea of this step is to use RSML, a notation for specifying state machines. Using this notation the effort for formal modelling in Event-B could be reduced from 3 months to 6 weeks. This reduction has been made possible because the complexity of the formal modelling task was reduced to a task of simply translating the state machines described in RSML into Event-B events.

The third question to be evaluated by Bosch was to check which types of properties of automotive systems can be verified with Event-B and the Rodin toolset. Properties of interest to Bosch typically include safety properties that a system must ensure, e.g.,

1. "*The cruise control is only allowed to operate in a defined speed range*".

2. "*When the driver hits the brake the cruise control must be deactivated*".

Properties of the first class could be easily modelled as invariants and proven by the Rodin theorem provers. Properties of the second class required the events in the Event-B model to be ordered and the introduction of additional invariants to be proven by the Rodin theorem provers. For ordering the events the Rodin Flow plugin developed by the University of Newcastle has been applied. Another type of property that has been verified was the check for deadlocks in the model. Proving deadlock freedom in Rodin turned out to be very difficult and pushed the Rodin toolset at its limit. Therefore, Bosch applied ProB, a model checker for Event-B, to check that the model did not contain deadlocks. An interesting observation Bosch made during proving was that the percentage of automated proofs using Rodin was more than 90%. Only 10% of the proofs required manual interaction.

Taking our experiences of the application of formal methods to automotive applications into account, we think that Event-B in particular and formal methods in general are promising methods for use in the automotive domain especially for the modelling and verification of discrete parts of embedded control systems. However, a number of open issues (e.g. user experience, scalability, and tool performance) have to be solved before these methods can be fully deployed in the automotive industry.

Felix Lösch
Robert Bosch GmbH

Deployment in the Business Information Systems Sector

SAP is the world's leading provider of enterprise software. Business software integrates data and services of various organizational units across an entire company, and is therefore very large and complex. Correct functioning of business software is very important because failures could incur great financial losses. Formal methods have gathered broad attentions for their capabilities to prove correctness rigorously. Thanks to the rich repository of models already available to us from former modelling efforts at SAP, it becomes less difficult and more meaningful to apply formal methods to the development of business software, since we could leverage existing assets.

One important principle during our deployment of formal methods was to achieve a high degree of automation. Ideally, the application of formal methods should be hidden completely from designers and developers, because the mastering of formal methods among them is nothing to be expected in our industry. To achieve this goal, we often needed to either make reasonable trade-offs (e.g. by sacrificing expressiveness of modelling languages in favour of verifiability) or enhance existing formal methods with improved automation. Our deployment came in three stages: modelling, verification, and model-based testing (MBT).

There were two challenges in the modelling phase. First, the size and complexity of a software model could still be overwhelming for formal analysis. To overcome this issue, we either made compromise by leaving out certain modelling features that are either deemed unessential or too expensive to be analysed, or tried to break down monolithic model structures into smaller components/layers in order to reduce difficulties in verification. Second, most industrial modelling languages lack formal semantics. Even for models whose semantics is more or less clear, it should be discouraged to apply formal methods directly on them, or we would get into an unpleasant situation where we need to re-implement same formal methods every time applied on a different modelling language. Therefore, we took the approach to translate all models into Event-B on which any future formal analysis would be performed. This has brought an additional advantage to formally capture and verify the relations between different models. The choice of Event-B also allowed us to enjoy the powerful tool support by the Rodin platform.

In the verification stage, we concentrated on two kinds of properties:

1. *consistencies among different modelling layers, and*

2. *a selected set of invariants that a model must preserve during runtime*.

Examples are the absence of deadlocks and data consistency in business processes. We applied both theorem proving and model checking. All domain specific models were first translated into Event-B, and we used automated provers and the ProB plug-in of the Rodin platform to conduct verification. We could not achieve full automation of theorem proving even after several enhancements through various static analysis techniques (such as automatic invariant discovery) and proof strategy optimizations. On the contrary, model checking did not require much human intervention. However, we often ran into the state explosion problem. In such cases, we had to manually reduce the explored state space by e.g. setting bounds on model variables, etc. Nevertheless, we could still manage to obtain meaningful results by combining both theorem proving and model checking. For instance, we usually applied model checking first in hope to find potential errors. We fixed the model accordingly, and repeated model checking until no more bugs could be found. Then, we started the more difficult and time-consuming proving procedure. Such a

strategy could save a lot of time, and is quite efficient to find model errors.

Full confidence in the correctness of software is hard to achieve. Therefore, we focussed on finding bugs with help of software models in the last phase of deployment. Software models are very useful in guiding test designs, because they capture the essence of how software is supposed to behave. For instance, message choreographies models were used to automatically derive conceptual test cases, which can be easily mapped to actual test cases that run on the system under test. According to the pilot users, the automatically generated test suites were covering all tests that had been created manually before. Another advantage of model-based testing is its complete automation. Test designers only need to create an initial test model which is most of time very intuitive, because we designed test models to be similar to domain specific models that test designers are familiar with. MBT proves to be non-intrusive and very productive by replacing usually very tedious manual tasks of designing and creating test cases.

Formal modelling and verification brings many quality assurance advantages for the development of business software. Design documents are complemented with software models that are accurate, executable, analysable, and can be used in deriving test cases directly linked to requirements. Formal validation and verification is not only used to prove correctness, but also to effectively find bugs, which is a nice alternative to traditional testing. However, given the assumption that formal methods are hidden behind existing domain-specific modelling abstractions, their success relies on the degree of automation. This is attested by the fact that model-based testing received a larger acceptance of developers than formal verification. For formal verification, we are still in progress to increase the degree of automation, to make tool usage and feedback more user friendly, and to improve tool efficiency when dealing with large software models. Nevertheless, our pilot deployment is very promising and welcomed by software architects and designers. We will continue to build a seamless experience of using formal methods in business software development processes.

Wei Wei, Sebastian Wieczorek, Andreas Roth
SAP

Deployment in the Space Sector

Space Systems Finland Ltd (SSF) is a Finnish SME which focuses on the development of mission and safety critical systems. SSF sets the ambitious goal of evaluating whether a fully tool supported formal model driven development methodology could be achieved within DEPLOY. This requires that all stages of a normal development process are supported: requirements engineering, architectural design, implementation, integration and validation. In addition, it would be desirable if support for dependability and safety analysis such as Failure Modes and Effects Analysis (FMEA) and traceability could be achieved.

In DEPLOY, SSF has experimented with formal modelling and development of several typical space applications. These developments were based on (a subset of) requirements for

- BepiColombo Solar Intensity X-ray and Particle Spectrometer (SIXS) / Mercury Imaging X-ray Spectrometer (MIXS) on-board software (OBSW), and
- a reusable Attitude and Orbit Control System software (AOCS).

The formal development of OBSW has demonstrated that, for some systems, modelling of their dynamic behaviour poses the main challenge. For instance, OBSW did not have complex safety constraints that could have been represented by the corresponding Event-B invariants. This was not surprising for the SSF engineers because a typical software requirement expresses what the software should do in a given situation; Event-B events alone are enough for modelling such requirements. Formal modelling of the systems similar to OBSW usually need to be more focused on reasoning about liveness oriented properties. Indeed, it has proved to be more interesting to analyse the dynamic behaviour of OBSW using the UPPAAL model-checker.

The AOCS case study has inspired work on mode logic. The goal of this work was to propose a generic method for modelling mode-rich systems in Event-B. In particular, the system specification was structured according to architectural layers and consistency conditions between mode logic at different layers were defined. In our work on ensuring mode consistency of AOCS we have relied on the mode logic that has been defined a priory. However, often mode logic has to be defined by the system developers. To facilitate this process we proposed a structured approach to defining fault tolerance part of mode logic, e.g. backward mode transitions. New work on using FMEA to defined mode logic is one of key contributions of the project.

DEPLOY is so far the only project at SSF where Event-B has been used, even though SSF has had prior to DEPLOY other similar also academic projects. Consequently, the tradition of using formal methods exists to some extent, but it is not utilized in any commercial way. DEPLOY showed us approximate costs and risks involved in training personnel without any prior experience in formal methods, but also revealed the most common limitations to wide adoption of FMs.

We feel that formal methods are of long-term strategic importance. SSF will continue to investigate the use of formal methods within DEPLOY and beyond.

Timo Latvala
Space Systems Finland Ltd.

Deployment in the Railways Sector

In the DEPLOY project context, in the first time, Siemens IC-MOL tried to define a

process using event-B in modelling Transportation Systems. As Siemens IC-MOL has considerable experience of applying formal methods to software components of railway systems, for DEPLOY the challenge is to raise this to the level of overall systems in order to address system safety. Siemens IC-MOL has been using B method for more than 15 years, and a considerable investment has been made in tools and methods. In particular, an automatic refinement tool has been developed to allow the (almost) automatic production of the concrete B model from the abstract B model. It is therefore important that the use of event-B at system level does not impose new investments at software level.

For this purpose, Siemens IC-MOL defined a process including event-B for the Transportation Systems Development. This process was applied to carry-out mini-pilot and pilot prototypes of the CBTC "manage operating modes" function. By developing mini-pilot and pilot prototypes, it appeared quite quickly that probabilities had to be added in the model. An experiment has been performed on the mini-pilot to add probabilities, with success. The realisation of mini-pilot and pilot gave us a confidence that a large scale event-B development is feasible with the proposed process.

Another achievement related to the integration of ProB in the data validation of CBTC controller software component which is still problematic for every deployment of CBTC systems on site. The old process based on Atelier B revealed several drawbacks, in particular with huge data properties. The motivation is therefore to automate the proof on huge data properties with alternative technologies. Siemens IC-MOL was interested in ProB because this tool provides services to deal with B properties in order to animate and model check B models. The success story with ProB improves significantly the data validation process at Siemens IC-MOL. It does not require B experts to carry out data validation. Indeed, the B experts are required only in case of problem, whereas in the former process, B experts were required in any case, for long and fastidious tasks. In addition, using ProB significantly reduces the time checking data properties: from 2 or 3 days with Atelier-B to 2 or 3 hours per project.

In order to integrate ProB in our validation process, a tool called RDV (Railway Data Validator) was developed. This tool automatically generates the B projects (containing assertions machines), run ProB on created B projects, and collect the result in a synthesis report. It works well not only for zone controller data validation but also for all other equipment controller (Carbonne Controller, Line Controller, IOC Controller). RDV has been used with a great satisfaction on all on-going projects carried out by Siemens IC-MOL : Paris metro line 1 (France, Commissioning on 2011), Barcelona metro line 9 (Spain, Commissioning on 2009), Alger metro line 1 (Algeria, Commissioning on 2011), Sao Paolo metro line 4 (Brazil, commissioning on 2010), Charles de Gaulle Airport Shuttle (France, Commissioning for the extension part on 2012).

Hung Le Dang
Siemens IC-MOL

Testimony of DEPLOY Associates

XMOS

Between October 2010 and October 2011, Stephen Wright and Kerstin Eder of Bristol University were seconded to XMOS Ltd of Bristol (http://www.xmos.com), working with Dr Henk Muller and Prof David May, both formerly of Bristol University and now at XMOS. XMOS is a ”fabless” microprocessor design company, developing embedded processors for a variety of markets including audio, display, communications, robotics and motor control. As part of a Bristol University Knowledge Transfer Secondment (KTS) (Grant EP/H500316/1), a formal model of the complete XCore Instruction Set Architecture (ISA) was constructed in Event-B using Rodin. This project applied and extended the techniques for Event-B ISA analysis developed by Dr Wright during his doctoral research to an industrial setting. This included construction of a model with all POs discharged, and automatically translated to a Virtual Machine capable of executing binaries compiled using XMOS's own tool chain.

The model encompassed all 209 of the XCore's instructions, yielding a model with 690 events and 4783 POs. Construction of the model pushed the boundaries of Rodin's scaling capabilities, and various new procedures were developed for partitioning of events and discharging of POs. The formal analysis uncovered several issues in the published ISA specification, ranging from straightforward errors through to subtle ambiguities of meaning. These were logged in XMOS's own bug reporting database.

With XMOS's consent, the complete output of the project has been published, including covering documentation and the project's final report, on the Deploy website (http://deploy-eprints.ecs.soton.ac.uk/346/).

Critical Software Technologies

Critical Software Technologies (CSWT) has concluded the work in DEPLOY where it used the avionics subsystem Integrated Secondary Flight Display (ISFD) as case study. This system is used to provide attitude, air and navigation information in the event of a primary display failure, supporting pilot's decisions in terms of:

• Determining the correct aircraft attitude, and exact altitude and airspeed.

• Determining the correct glide slope approach and localiser angles in relation to the runway

• Determining whether there is a fault in the ISFD unit

The project started with a series of half-day Event-B training sessions held between CSWT engineering team composed by three engineers and the University of Southampton. These sessions were quite useful because they helped with learning the Event-B language and foremost they provided a set of real case studies where formal methods have been used successfully. With the support from Southampton University, CSWT developed a set of abstract models covering Display Modes, Segments, Display Data Values, Status and events to address updates on values, range checking, attitude alignment.

The experience of CSWT in DEPLOY shows that this approach has a positive impact on the requirements engineering phase. Rodin and the automatic proof generation

mechanism highlight when there is a problem with the system requirements. Thinking about the problem in terms of sets and invariants pushes the system engineer to further think about the problem and the solution and to come up with a more complete set of requirements that can be verified using proof generation. Formal methods allow for the verification of decisions earlier in the development cycle through controlled experiments, system behaviour can be made more predictable and the chances of overspending are reduced because the characteristics of the system are formalised and thoroughly tested early in the lifecycle.

CSWT would like to thank the DEPLOY consortium for the opportunity that was given to participate in the project and would like to give a special thanks to professors Michael Butler and Abdolbaghi Rezazadeh from Southampton University.

AeS

AeS Group is a Brazilian company, created in 1991. At that time AeS was working in the building automation field. In 1998, AeS became involved in the railway field, when the first Brazilian General Door Control System (GDC) for Rolling stock doors was developed. The aim of this equipment was to be an interface between train requests (operator request, Signaling requests, etc...) and the door system itself. This system was safety critical, as the major operation of this equipment consists in sending an open command to the doors in each cabin.

During the B 2007 Conference, AeS had its first contact with Rodin tool and some members of the Rodin project. When DEPLOY project began, AeS was invited for a deeper participation in the project as a DEPLOY Associate. During the last 4 years, AeS has tried to apply Event-B formal method in different phases of the development process, such as requirements, software coding, validation and testing to evaluate the possible benefits. All this attempts received plain support from DEPLOY partners.

We can conclude that the amount of information and learning received during this period of collaboration work was definitely and surprisingly huge, but manageable, and we achieved much more than we expected. At AeS, we are currently not using formal methods in all phases of our development process, but FM are helping us to save money when applied. This has been verified for example with the substential decrease of the testing phases (time and effort). Moreover the quality of the resulting development documentation has increased a lot as well.

Update on the Rodin Platform

The current version of the platform is Rodin 2.4. The Rodin development team is now preparing Rodin 2.5, scheduled for end of April 2012.

Rodin Release 2.3 (released in March 2011)

See http://wiki.event-b.org/index.php/Rodin_Platform_2.3_Release_Notes.
This version of the Rodin contains some improvements and new features, such as:

- **Memory footprint**: Projects now require much less memory to build (some projects that required 1200 Mo now build with less than 512 Mo).
- **Advanced prover customisation**: It is now possible to fully customize the automated prover and the post-tactic which is run after every command in the interactive prover. The new mechanism allows to define, from the GUI,

various parameters of tactics, such as the timeout, and to combine them in several ways.

– **Prover efficiency:** The automated prover discharges more proof obligations without user intervention, thanks to additional proof rules and reasoners.

Rodin Release 2.4 (released in January 2012)

See http://wiki.event-b.org/index.php/Rodin_Platform_2.4_Release_Notes

Rodin 2.4 is based on Eclipse 3.7.1. The platform is now released and maintained for **64-bit Linux and Windows**. The 32-bit versions are still supported to allow a smooth upgrade path.

The new **Rodin Editor** is now the default editor for machines and contexts (see http://wiki.event-b.org/index.php/Rodin_Editor). The original structured editor is still available though.

This release also features the new **Type Environment** view that displays all free identifiers (together with their Event-B type) that are available in an interactive proof. This view can be activated by clicking *Window > Show View > Type Environment*.

P**roof simplification**, which had been de-activated for several releases, is now back with an improved algorithm. The purpose of this mechanism is to reduce the memory and disk footprint of completed proofs and make them more legible for inspection. This is done by removing unneeded steps (mostly unused inferences generated by the post tactic). This feature is disabled by default and can be simply turned on by clicking *Window > Preferences > Event-B > Sequent Prover > Simplify complete proofs when saving*.

Finally, a **new tactic combinator** *Attempt after Lasso* is now available. This allows any user to implement fancy proofs by attempt without writing any line of Java code.

Rodin Release 2.5 (expected in April 2012)

See http://wiki.event-b.org/index.php/Rodin_Platform_2.5_Release_Notes.

Version 2.5 of Rodin will be the ultimate version developed within the frame of the DEPLOY project. As such, consolidation of the toolset is the main objective. No major new feature is expected.

Thomas Muller
Systerel

Update on the Rodin Plug-ins

Existing plug-ins continue to be maintained and developed. Details may be found at http://wiki.event-b.org

Here we highlight a number of recent plug-in developments.

New Release of Multitasking Code Generator: The new release of the multitasking code generator provides big improvements in tool usability and configurability. Tasking Event-B is now integrated with the Event-B Editors. The plug-in provides the ability to translate to C, Java, etc. in addition to Ada source code. The theory plug-in is used as a mechanism for defining new programming data types, and to define translations to target data types. The translator is extensible; allowing addition of translations for new structural code features and addition of translation rules for

mathematical operators.

See http://wiki.event-b.org/index.php/Code_Generation_Activity

MBT (Model-Based Testing) for Event-B Plug-in: MBT is an approach from software engineering that uses formal models as basis for automatic generation of test cases. A test case is defined as a sequence of actions (or events, or triggers) together with corresponding test data that can be executed against a System Under Test (SUT). In DEPLOY, we support a version of MBT using Event-B models as test models from which test cases are generated. We provide an MBT method together with a Rodin plug-in that allows generation of test cases satisfying different coverage criteria (e.g. covering of all events in a model or covering paths to a set of target global states). This includes the generation of appropriate test data that satisfy the guards of the single test steps.

See http://wiki.event-b.org/index.php/MBT_plugin

Improvements to ProB: Driven by a case study from the space sector (a protocol modelled by SSF), where memory consumption was an issue, we have investigated ways to reduce ProB's memory consumption. A first step was to implement a first version of state compression, whereby we simplify stored states so that they require less memory. This was achieved without compromising speed and is now always activated. Furthermore, if the preference COMPRESSION is set to true, then ProB will also detect common (sub-)expressions in states and store the common expressions only once. E.g., when several states have the same value for a given variable x then its value will only be stored just once. This is particularly useful when complicated variables only change infrequently.

See http://www.stups.uni-duesseldorf.de/ProB/index.php5/Main_Page

Theory Plug-in: The Theory plug-in enables the definition of mathematical and prover extensions. It provides a high-level interface to the Rodin Core capabilities with regards to mathematical extensions. The Rule-based Prover was originally devised to provide an usable mechanism for user-defined rewrite rules through theories. Theories were, then, deemed a natural choice for defining mathematical extensions as well as proof rules to reason about such extensions. In essence, the Theory plug-in provides a systematic platform for defining and validating extensions through a familiar technique: proof obligations.

http://wiki.event-b.org/index.php/Theory_Plug-in

Mode/FT Views Plug-in: The Mode/FT Views plug-in is a modelling environment for constructing modal and fault tolerance features in a concise manner and formally linking them to Event-B models. Fault tolerance part adds additional structural checks and reserves a place to trace FT requirements. A Mode/FT view is a graph diagram containing modes and transitions which provide additional information necessary for establishing a formal connection with the model. The tool statically checks the views and generates a number of proof obligations.

See http://wiki.event-b.org/index.php/Mode/FT_Views

Flows Plug-in: The Flow plug-in is a Rodin Platform extension assisting in the formulation of verification conditions related to the dynamic properties of a model, i.e., event ordering and enabledness conditions. A visual notation is used to define graph-like structures that are translated into Event-B theorems. The tool simplifies a number of routine tasks such as writing and maintaining deadlock-freedom and relative deadlock freedom proof obligations. Its core functionality is concerned with the verification of the feasibility of use case scenarious: checking that certain event paths are feasible for a given model.

See http://wiki.event-b.org/index.php/Flows

The most up to date information on all plug-in developments can be found on the Event-B wiki:

See http://wiki.event-b.org/index.php/Current_Developments

Michael Butler
University of Southampton

DEPLOY Federated Event Fontainebleau, 27 February-1 March 2012

The DEPLOY Federated event, hosted by IUT Sénart-Fontainebleau with the help of the LACL laboratory, took place in the south of Paris. It was the occasion to draw a complete picture of the current status of the Rodin platform, the on-going research and industrial use, inside and outside the DEPLOY project.

The first day was devoted to tutorials on theory and proof extensions as well as ProB integration.

The second and third days were devoted to workshop-style talks on experiences with Rodin usage and on new plug-ins for Rodin. There were 30 presentations some from DEPLOY partners and some from outside of DEPLOY. There was plenty of lively interaction between participants during and between the talks. The workshop proceedings are available online.

The fourth day, the Industry Day, was devoted to report on DEPLOY achievements and to discuss industrial use of Rodin and Event-B. The slides are available online.

See http://www.bmethod.com/php/federated-event-2012-en.php and http://wiki.event-b.org/index.php/Rodin_Workshop_2012.

DEPLOY after DEPLOY

Our project is coming to its end. The project has clearly achieved its major scientific and technological objectives and has made substantial advances in developing advanced engineering methods for constructing dependable systems.

The project legacy includes the main DEPLOY web site at http://www.deploy-project.eu - this site will be maintained after the project end, but no new information will be added. The site includes all public project deliverables and Newsletters (see http://www.deploy-project.eu/html/deliverables.html).

The home of Event-B and the Rodin platform at http://www.event-b.org will be actively used after the project end; all people involved in tool development will use it for dissemination of their results, this includes activities conducted in various public (both national and European) and industrial projects. This site provides downloads of all freely available tools and plugins and the up-to-date Event-B and Rodin documentation wiki at http://wiki.event-b.org.

The Rodin handbook developed in DEPLOY is made publicly available at http://handbook.event-b.org. The Rodin platform development will continue at SourceForge: http://sourceforge.net/projects/rodin-b-sharp.

All DEPLOY publications, reports, tutorials, training materials, presentations, papers, and models can be openly downloaded from http://deploy-eprints.ecs.soton.ac.uk. This site will be used and maintained by the follow-up FP7 ADVANCE STREP on Advanced Design and Verification Environment for Cyber-physical System Engineering (http://www.advance-ict.eu).

The open repository of evidence for adopting formal methods in industry created by the DEPLOY team is made available at http://www.fm4industry.org. It will be maintained for the foreseeable future and will be used by a wider community for collecting evidence on formal methods uses and impact on industry.

As part of the project we created a non-for-profit company called Rodin Tools Ltd, that will take over the responsibility for the Rodin toolset at the end of DEPLOY - http://www.rodintools.org. The company consists of

- a Strategy Committee of external advisers to look at the development strategy,
- a Platform Development and Maintenance partner to carry out the wishes of the Strategy Committee and Company members, and
- a Coordination partner to manage the Company, run workshops and training, etc.

It has been a rewarding experience to work in the DEPLOY team and to lead this work. I would like to express my deepest gratitude to all people who have been involved in preparation and implementation of the project since the idea of this project was first conceived in April-June 2006 in discussions with Kaisa Sere, Cliff Jones, Michael Butler and Jean-Raymond Abrial. I sincerely hope that all of us find these efforts worthy.

Alexander Romanovsky
DEPLOY Coordinator