Project DEPLOY
Grant Agreement 214158
"Industrial deployment of advanced system engineering methods for high productivity and dependability"

DEPLOY Deliverable D2

D14.03 Electronic Newsletter

Thierry Lecomte (ClearSy)

Public Document

10th July 2009

http://www.deploy-project.eu

DEPLOY PROJECT

Newsletter N° 3

Contents

Partners

Introduction

The advanced engineering methods used by the DEPLOY project involve the application of formal modelling and analysis of models. Modelling and analysis enable deep understanding at early stages of design and lead to clean system architectures. Key to the successful deployment of these methods is the support provided by analysis tools such as theorem provers, model-checkers and animators. The DEPLOY project makes strong use of the EventB method for modelling and exploits the Rodin Platform as the basis for analysis of EventB models. Rodin is an open-source open-architecture toolset built on the Eclipse platform and was developed as part of the FP6 RODIN project. The open architecture facilitates extension of functionality through the addition of tool plug-ins.

The pilot deployments of the project in four major sectors (automotive, transportation, space and business) have provided a valuable testing ground for the Rodin toolset. The industrial deployment partners (Bosch, Siemens Transportation, Space Systems Finland and SAP) have made strong use of the Rodin toolset within the pilots, building complex models and successfully applying the Rodin verification technology to analysis of the models. While the reaction has been generally positive, the pilots did identify several areas for improvement in the toolset.

Notable amongst these were improvements to the usability of Rodin through better editing and refactoring capabilities, improvements to the proof and model checking capabilities, facilities for enriching the mathematical theories supported by Rodin, facilities for supporting team-based developments and, not least, facilities for code generation from models. These enhancements to the Rodin toolset are being undertaken in Workpackage 9 of the DEPLOY which is concerned with tool extensions. The requirements identified by the pilots are not surprising and most had been identified in a general way in the original workplan for Workpackage 9.

The experience of the pilot deployments has helped the project gain a clearer understanding of those requirements and helped us to prioritise the many demands on tool enhancements. Excellent progress is now being made towards addressing these additional tooling requirements.

Many industrial organisations already make use of a tool chain to support development such as requirements traceability tools, UML-based modelling tools, code generation tools and testing tools. Successful deployment of a formal engineering toolset, such as Rodin, within these organisations will require integration of the formal tools with the in-house tool chain. SAP is a good example of an organisation that makes strong use of their in-house software engineering tool chain. As part of the pilot deployment in WP4, SAP have prototyped an integration of Rodin with an in-house graphical modelling framework. The result is an environment in which graphical models can be analysed through proof and model checking in a way that appears seamless to the user. This integration has been made possible because both Rodin and the SAP tool chain exploit the Eclipse framework. The use of an open-architecture tooling framework is clearly a key element of achieving tool integration. Semantic integration is also important and, in the case of the SAP work, this was facilitated by following the approach of UML-B, which cleanly integrates parts of UML with EventB.

While much of the continued development and use of Rodin takes place within the DEPLOY Project, there is a growing group of users and plug-in developers outside of DEPLOY. In July this year, DEPLOY organised a workshop at the University of Southampton to bring together existing and potential users and developers of the Rodin toolset and to foster a broader community of Rodin users and developers. For Rodin users the workshop provided an opportunity to share tool experiences and to gain an understanding of on-going tool developments. For plug-in developers the workshop provided an opportunity to showcase their tools and to achieve better coordination of tool development effort. Moving towards an open source development project will mean that features that cannot be resourced from within the project can be developed outside the project. It will also help to guarantee the longer-term future of the Rodin platform.

The nurturing of a broader community of users and developers is very much in the spirit of the

Grand Challenge in Verified Software promulgated by Tony Hoare and Jay Misra. The Grand Challenge aims at a large-scale international effort to develop verification technology that will enhance the productivity and reliability with which software is designed in demonstratable ways.

Michael Butler, University of Southampton

Tool Coordinator of the DEPLOY project

DEPLOY: Current State

DEPLOY is a four year FP7 ICT Integrated Project on industrial deployment of system engineering methods providing high dependability and productivity started in Feb 2008. The overall aim the project is to make major advances in engineering methods for dependable systems through the deployment of formal engineering methods. The work is driven by the tasks of achieving and evaluating the industrial take-up of the DEPLOY methods and tools, initially in the four sectors which are key to European industry and society.

During the last 6 months the main focus of the consortium has been on initial deployment of DEPLOY methods and tools in the four deployment partners, extensive work on the deployment pilots (medium size applications typical for the deployment partners' domains), strengthening the Rodin platform and on strategic refocusing the project.

The refocus has been chiefly driven by the needs of the deployment partners. The main areas, in which the project is now planning to invest extra resources, are code generation and model based testing. More work will be done on developing methods and tools supporting reasoning about timing properties of the systems.

In addition to these, the project is now working on creating a mechanism for technology transfer to a number of Deploy associates – the external industrial establishments with strong interest in our methods and tools. Very soon we will have an infrastructure in place for organising dedicated training and joint work on developing mini-pilots with such companies.

A very important step made recently by the project Executive Board is a substantial refocus of project work on measurement; this has been aligned with achieving our ultimate goal – real deployment of the methods and tools in industry. We are now putting more focused efforts in ensuring that the deployment partners will have strong concrete evidence of the benefits the technology brings to their engineers and managers.

The most important areas of our ongoing work are developing of the pilot applications, making substantial methodological advances ensuring rigorous and traceable transition from system requirements to the architectural models, extending the functionality of the platform by adding new plugins supporting reuse, defining specific frameworks for collecting evidence in the four deployment partners and developing approaches to structuring complex models.

The chief project aims for the remaining part of the second year are to complete (and learn the lessons) of the pilot deployment, to intensify our work on making the tool platform more usable and scalable, to prepare for the full deployment during the second part of the project, to extend the community of the external users and developers of our tools, to ensure that the refocused plans are being smoothly implemented and to produce the deployment strategy document.

Alexander Romanovsky, Newcastle University

Project Coordinator

The cruise control system as a pilot application

DEPLOY Work package 1 is lead by Bosch and deals with applying EventB to problems typical of the automotive industry. The project plan envisions two pilot applications. As a first pilot application, we have chosen the cruise control system. This section provides a short introduction to cruise control systems in general and motivates our decision for the cruise control system as a pilot application.

Introduction to the cruise control system

According to the Audi Online Glossary[1] "... the cruise control is an electronic aid that keeps the car moving at a constant speed. It reduces stress on the driver particularly where speed limits must be observed, when towing a trailer, and on long trips. The system stores and maintains the speed selected by the driver. The set speed can also be manually increased or decreased. Afterwards, if desired, the car will resume the last speed set. The cruise control can be deactivated with the "Off" switch or by pressing the brake or clutch pedal."

Cruise control systems are typical applications within the automotive environment. Formally modelling and proving automotive applications challenges us in the following ways.

Most automotive applications are designed as embedded systems to implement one or a few dedicated functions. The cruise control system is embedded in the motor management software which is in turn part of the engine control unit. The engine control unit controls the behaviour of the internal combustion engine. With the cruise control being just one of the many subsystems of the engine control unit, driver requests issued via control interface elements or the pedals must be coordinated with requests from other subsystems, sensor data, and even prioritised requests from other systems. The coordination and prioritisation of requests is an important task of embedded systems within the automotive environment. Based on a thorough analysis of requirements, we derived appropriate interfaces and communication concepts that are just as indispensable as unambiguous decision rules. All these aspects must be modelled in EventB, an activity that has already been started, see also "Current status".

As almost any embedded system, most applications typical of the automotive industry implement controlling functions. Thus, when describing requirements and when modelling in EventB, we have to consider different aspects of control theory. One of our aims during the DEPLOY project is to find appropriate ways of expressing requirements dealing with aspects of control theory and of formally modelling them, see also "Challenges".

Typical software applications within the automotive environment may never behave in a way that may endanger the driver, the vehicle, or other vehicles nearby. For that reason, safety requirements are defined and, up to now, tested rigorously with considerable effort. With EventB, there is the possibility of modelling safety requirements and proving that they are never violated. During the course of the project, we would like to verify that EventB provides ways of modelling and proving automotive safety requirements effectively.

Reacting as quickly as possible to driver requests or changing environmental conditions is another important task of most applications implemented in the automotive environment. Timing constraints with regard to the cruise control system range from relatively simple requirements such as a defined reaction time for driver requests to more complex issues such as switching off as quickly as possible if environmental conditions require this. Thus, an appropriate way to model time is also an important aspect when applying formal methods to systems typical of the automotive environment.

When it comes to automotive applications, we often need to deal with a large variety of system variants. The basic functionality of a cruise control system is always the same, no matter if it is implemented in high end or middle class vehicles. However, there is a wide variety with regard to the control interface elements the driver may use to interact with a cruise control system. Voice controlled interfaces may be used just as well as simple operating levers or buttons. We need to find a way of describing and proving system variants effectively in EventB.

Often, we can easily describe the functionality of an automotive application. However, due to the coordination effort with other applications residing in the same vehicle, the typically large variety of system variants, and the evaluation of data from many different sensors, its development and

implementation is a complex matter. With the cruise control system being a comparatively simple part of the motor management software, its complexity is still manageable. Nevertheless, the cruise control system is very typical within the automotive industry. Therefore, all important aspects of formally modelling automotive applications can be addressed while an appropriate measure of complexity is provided to get a realistic impression of the advantages and challenges of applying formal methods to automotive applications.

Current status

Formally modelling a system and its environment requires a thorough analysis and appropriate description of requirements. During the first year of the project, our most important assignment has been to develop a well-structured requirements document that includes safety requirements, timing constraints, and control theory aspects. We have chosen to apply and slightly modify the Problem Frames Approach as suggested by Michael Jackson[2]. The requirements document has been finished and is used as a reference document for modelling and correctness proofs.

During the last weeks, we have started to model the cruise control system in EventB. Following our requirements approach, we have started modelling in an abstract way. Current experience shows that the Problem Frames Approach provides an excellent way of preparing requirements so that the system and its environment can be modelled easily in EventB. Up to now, modelling the cruise control system did not cause serious problems. However, time did not allow us to extensively prove desired properties.

Challenges and further Steps

For the remaining part of the year, we plan to completely model and prove the cruise control system in EventB. On less abstract levels, requirements will become more elaborate and contain more complex safety, timing and control theory information. We expect challenges especially with regard to timing and control theory aspects.

Expressing time

We have not yet found a way of simply and elegantly expressing explicit timing properties. In particular, we need to formalize a real time clock that issues ticks (e.g. every millisecond) which can then be processed by multiple timers. We need to be able to refer to timer values in guards of events which come true when a timer has elapsed or according to special time patterns, for example. Finding a solution to this problem is one of the keys to using formal methods to the automotive development process.

Expressing control theory aspects

Considering aspects related to control theory, modelling them in EventB seems to be an unsolvable task. One solution might be to integrate specific control theory tools (e.g. ASCET-SD, Simulink, Matlab) with RODIN. This way, the control algorithm itself can be modelled using these such tools. However, we will then still need a way of defining assumptions about specific properties of control algorithms in EventB. We will need to clarify whether this is a possible and feasible solution.

Use of probabilities in EventB models of railway system

DEPLOY Work package 2 is lead by Siemens Transportation Systems and deals with applying EventB to problems typical of the railway industry. This section describes the current status of the introduction of probabilities into EventB models, for safety-critical systems.

Need of event probabilities and invariant probabilities

EventB models are well adapted to system modelling, in order to formally prove that a set of properties (including safety properties) always hold.

But EventB modelling is not yet adapted to systems where properties (including safety properties) shall hold in "most cases", and where it is possible (but very unprobable) that a combination of failure might lead to an unsafe situation.

The objective is thus to prove that, given the probability of each failure, the probability to break a property by any combination of failure is lower than an acceptable rate.

On Going work

In close cooperation with Newcastle University and Abo Academy, STS investigates two approaches:

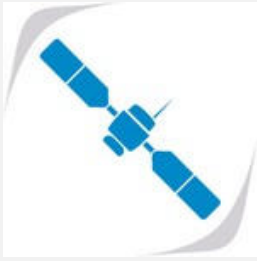
- use of probability within the EventB model :each failure is modelized in the EventB model, with its associated probability. It is proved that a safety property is either true, or the probability is below an acceptable rate.
- use of external methods (model checking) toprove probabilistic properties of the EventB model

---

Event-B applied to the design of the BepiColumbo Space Probe

DEPLOY Work package 3 is lead by Space Systems Finland and deals with applying EventB to problems typical of the space sector. This section describes the current status of the introduction of EventB into the ongoing space project - BepiColombo.

Introduction to BepiColombo



BepiColombo is a European Space Agency (ESA) mission that will explore the planet Mercury. For this purpose, it will send two orbiters, one of which is responsible for carrying two X-ray instruments: Solar Intensity X-ray and particle Spectrometer (SIXS) and Mercury Imaging X-ray Spectrometer (MIXS). The instruments are controlled by the MIXS/SIXS Data Processing Unit (DPU) – an intrinsic element of the spacecraft. It controls the power and the operating states of the instruments, monitors their operation and handles telecommand (TC) and telemetry (TM) communication. The On-Board software (OBSW) running on the DPU's CPU consists of five different components: the Core Software (CSW) and two application software (ASWs) for each instrument : - SIXS-P, SIXS-X, MIXS-T, and MIXS-C ASW. SSF is in charge of the design and development of OBSW for the two instruments on board the BepiColombo spacecraft.

BepiColombo uses a subset of standard PUS (Packet Utilization Standard) services in the form of telecommands and telemetries. TC/TM handling software has been and will be a part of many space software projects. That is why it has been given the main focus of EventB modelling activities in BepiColombo pilot so far. This fact helps explain why the modelling effort has almost solely concentrated on those requirements that directly concern TC/TM handling. In numbers, the latest BepiColombo models fully or partially modelled app. 18% of the system requirements.

All relevant concepts of TC/TM processing are introduced in a number of refinement steps, where the initial focus is on modelling generated TC/TMs and their associated statuses (with respect to their validation and execution activities) and the ways TC memory management, validation and execution are conducted. Naturally, these descriptions are incorporated into the specifications step-by-step, building up the more detailed TC validation/TC execution and TM reporting chain, and leaving some other behavioral aspects underspecified. Upon specifying a very basic system behavior regarding TC/TM handling, subsequent models introduce different standard TC services and some special services responsible for commanding the instruments. This, in turn, resulted in refining already introduced TC validation/execution while taking into account the purpose and the functionality of a TC under validation/execution.

Seemingly transparent architectural division of OBSW components becomes to some extent visible after introducing management of component SW modes. For every component, mode commands (telecommands) are executed according to specified transition diagrams (see Figure 1). The central part in managing operating modes belongs to CSW. Its operating modes and mode transitions triggered by dedicated telecommands are governed by system level mode transition diagram, meaning that all mode transitions made on the level of other components have to be synchronized with this upper-level (system) modes. The transition diagrams give an overall picture of system behaviour and are especially suited to modelled in this framework. In addition, introducing mode management seems reasonable as a next step since TC/TM processing is dependant on and, at the same time, affects the modes. In fact, TC execution is limited to certain modes and, moreover, it is TC execution that explicitly induces mode changes. Mode changes can also be induced by FDIR mechanisms, but that is currently not foreseen in the scope of this study.

Introducing mode management on the level of CSW and all the other four subcomponents requires

significant attention modelling their dependencies. Expressing them in a model and proving them is of vast importance. This way we can guarantee the synchronization between different system level modes and the instrument software modes, as illustrated by an example in Figure 1.

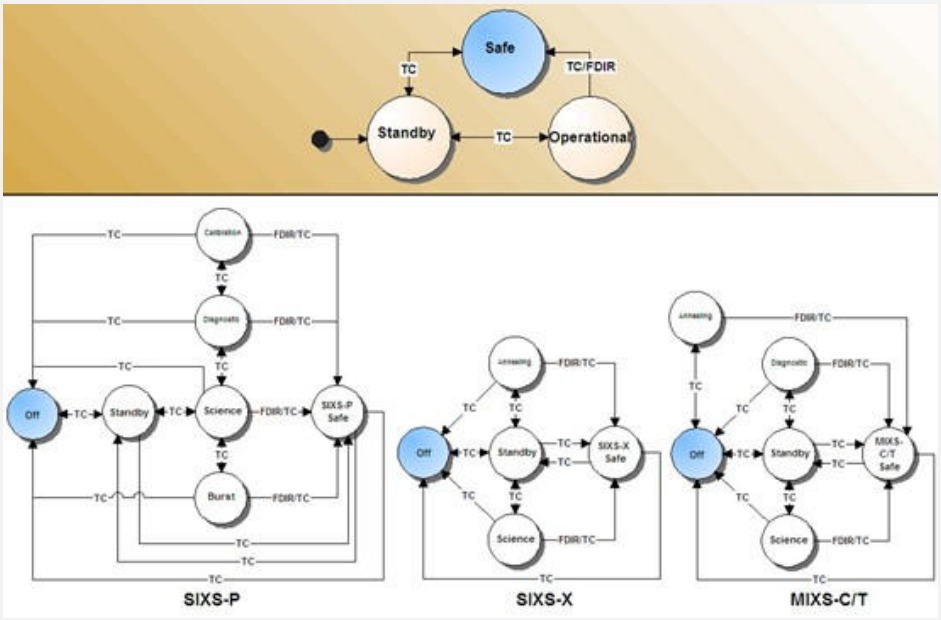


Figure1 An instance of a system level mode and the corresponding instrument SW submodes

Current Status

Yet, even though only 18% of requirements are covered, the resulting models are quite complex. Despite this, SSF's experience suggests that the required modelling efforts are not very alarming. The only really time-consuming activity in the pilot development is proofs – a considerable amount of time is needed for producing a proof, even when it is needed only to reuse an existing proof. Hence our interest has moved during the project from quantitative goals such as requirements coverage to qualitative goals such as better understanding of the EventB proof methodology. However, we expect this to change. Due to a limitation of the framework, the pilot models, for instance, do not take into account any real-time restrictions – no timing properties are introduced whatsoever. The importance of addressing these issues suggests that this is one of the main objectives for further deployment activities.

Choreography modelling

DEPLOY Work package 4 is lead by SAP and deals with applying EventB to problems typical of the business information system sector. This section describes the current status of the use of EventB for the modelling of choreography.

Introduction

Developers in the business information sector make use of diagrammatic domain specific languages and tools. These diagrammatic languages and tools are designed and implemented in the style of model-driven architecture with a strict MOF [3]-based meta-model. Each language is designed for one particular purpose, such as modelling message choreography or modelling business object behavior.

In the WP4 pilot we are focusing on choreography modelling, as one of these diagrammatic languages, which has a significant impact on the success of a system design in this area. Its importance comes from the fact that business software integrates many organizational parts and functions into one logical software system and that such software systems are typically very complex. Service-oriented architecture (SOA) is regarded as a next evolutionary step to cope with the software complexity of business information systems: independent business components provide enterprise services that can be composed individually to implement customized business processes.

In this context, formal models of service choreographies provide means to precisely specify the

functionality of a service and reason about properties of services in a very early design stage, for instance, it is possible to determine at this stage how to form the reliability parameters required from the SOA middleware for establishing a certain service interaction.

We have implemented an automated translation from service choreography models to EventB models using a flat refinement structure, similar to the UML-B approach. The resulting formal models allow for the verification of local enforceability of the choreography and other essential properties. Middleware properties play an important role in this context, so the developers can reason whether the developed choreography is fault-tolerant enough to deal for instance with message flips, and if necessary exchange the used middleware configuration. Various approaches have been explored to model different middlewares.

Current status

At the current stage it is still necessary to improve the refinement structure in order to increase the automation of the verification process. Design pattern methods and tools researched in Deploy will be explored in order to increase prover automation on the one hand side and to guide developers with the help of documented best-practice design.

Since developers in this area are accustomed to successful traditional development processes, it is impossible to make radical changes. Our approach thus aims at a smooth transition between the traditional development processes and the formal approaches. Therefore it is important that the developers get appropriate feedback from Rodin using the diagrammatic notation. We have therefore integrated interactive model simulation on the basis of ProB directly in the graphical choreography model. It is also possible to find model defects using the model checking features of ProB. Providing adequate feedback from these model-checking runs to the user in the choreography diagram is work in progress.

In order to support existing quality assurance processes we are also integrating a test derivation feature based on the ProB model checker. With it, test skeletons are generated which can be complemented to functional integration tests. Again this feature plays an important role in order to convince developers of the usefulness of formal modelling tools.

A further direction of the pilot development will be to explore the embedding of service choreographies in the overall business processes, as for instance modeled in the BPMN language.

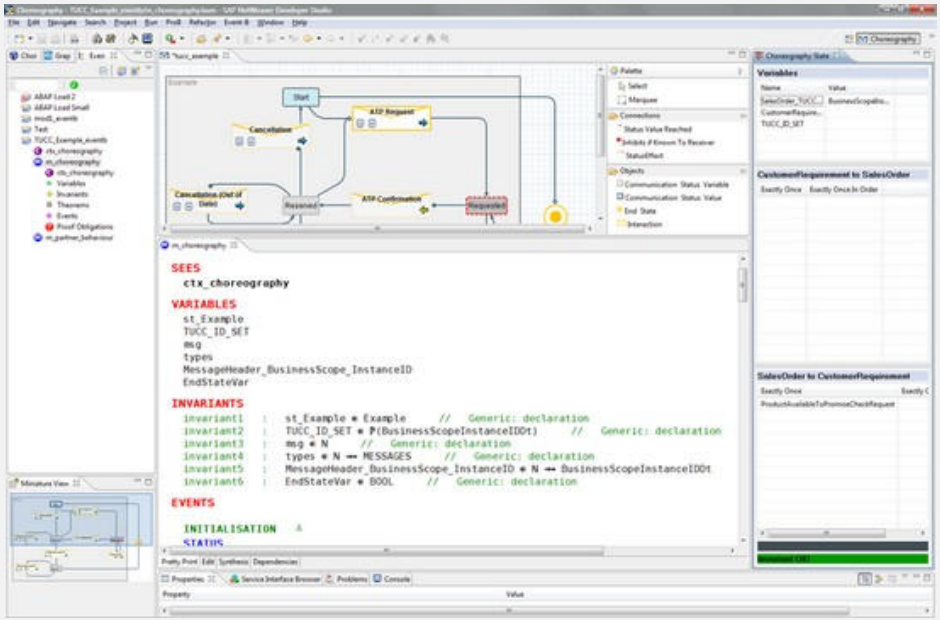


Figure2 Example of Choregraphy

Update on the RODIN Platform

Release 1.0 of the platform has shipped July 1st, 2009. We are very proud of this first full release. In the past 6 months, more than 70 issues have been resolved. The most prominent new features of the core platform are:

- Auto-completion in the editor: everywhere the user enters a predicate, expression or

assignment, the platform proposes a list of available names when requested by hitting CRTL-SPACE.
- Support of the mathematical language V2 with generic operators for the two projections and the identity relation, as well as the new partition operator.
- Better encoding of enumerated sets that allows for easy modification after the initial generation by the wizard.
- Theorems can now be freely mixed with axioms and invariants. Moreover, users can now specify that some guard is a theorem.
- A detailed log of all the changes to the Rodin platform is available on SourceForge on the download page (just click on the little notepad icon next to the software release).

Caution: There is a major change between release 1.0 and the previous ones. This means that all models opened or created with release 1.0 or later are not backward compatible with previous releases. When you plan to move to release 1.0, please backup all your workspaces before opening them with the latest Rodin platform. The procedure for upgrading one workspace to release 1.0 or later is fully explained in the release notes of the platform.

For more information on the current developments, please visit our wiki (**http://wiki.EventB.org/index.php/Current_Developments**). The wiki is also the reference site for all documentation (both user and developer) about the platform. Do not hesitate to visit it and contribute.

As a final note, monitoring the changes to the development version of the Rodin platform is now very easy: just subscribe to RSS feed http://cia.vc/stats/project/rodin-b-sharp/.rss.

Contact and Further Information

Laurent Voisin (laurent.voisin@systerel.fr) is responsible for WorkPackage 9, that is the tooling workpackage of the DEPLOY project. He is also the reference person for all technical issues regarding platform development and coordination at the technical level (API evolution, coding guidelines, etc.)

More information about the Rodin platform can be obtained from the web site: **http://www.EventB.org**

All documentation related to the Rodin platform (User Manual, Developer's Guide) is available from the wiki: **http://wiki.EventB.org**

Update on the RODIN Plug-ins

In the previous newsletter we reported on a number of plug-ins that have been developed including the AtelierB prover plug-in, the UML-B plug-in, the ProB animation and model-checking plug-in, a requirements management plug-in and a model composition plug-in. These tools have been further enhanced in the last period and are being applied in the pilot deployments.

Since the previous newsletter, a notable new plug-in for Rodin is a text-based editor. The standard Rodin editor is a form-based editor. The new plug-in provides a full concrete syntax for the EventB language along with standard text editing capabilities such as copy/paste/cut, name completion and syntax highlighting. Synchronisation between both editors is supported. Another new plug-in is a refactoring plug-in that provides facilities for systematic renaming of identifiers throughout an EventB development.

Development of several other new plug-ins has commenced including a rule-based prover plug-in and a pattern application plug-in. The most up to date information can be found on the EventB wiki: http://wiki.event-b.org/index.php/Current_Developments

Contact and Further Information

Michael Butler (mjb@ecs.soton.ac.uk) is the Tool Coordinator of the DEPLOY project. He is in charge of coordinating efforts devoting to tooling both within the DEPLOY project and outside (external contributors

---

Deploy Interest Group

This group (DIG) is composed of companies/universities/individuals interested in the DEPLOY objectives and results. The DIG has privileged access to information (bi-annual newsletter, dedicated hands-on sessions, etc.) and we are looking for further collaboration with DIG members (feedback, new case-studies, new contributing plug-ins, etc). Special attention is given to DIG: dedicated means are allocated to help DIG members getting educated / gaining experience with the Rodin tools.

Current members are:

- Marc Benveniste (STMicroelectronics - France)
- Ian Oliver (Nokia - Finland)
- O. Sami Saydjari (Cyber Defense Agency)
- Ken Robinson (University of South Wales - Australia)
- Juan Bicarregui (Formal Method Europe - United Kingdom)
- Aryldo G. Russo Jr. (Acesso e Segrurança - Brazil)
- John Brightman (AT ENGINE CONTROLS)
- Vecheslav Kharchenko (National Aerospace University - Ukraine)
- Jean Mermet (Keesda - France)
- Viktor Mashkov (University J.E.Purkyne - Czech Republic)
- Colin O'Halloran (Qinetiq - UK)
- Andreas Enbacka (Sysart Oy - Finland)
- Gao Hongjiang (Xi'an Jitotong University, China)
- Maria Teresa Llano Rodriguez (Heriot-Watt University, UK)
- Hironobu Kuruma (National Institute of Informatics, Japan)
- Hrvoje Belani (University of Zagreb, Croatia)
- Camilo Rueda (Universidad Javeriana-Cali, Colombia)
- Paul Simon (Individual - France)
- Bruno Gomes (Federal University of Rio Grande do Norte, Brazil)
- Gudmund Grov (Heriot-Watt University - United Kingdom)
- Simon Hudon (ETH Zürich - Suisse)
- Xinben Li (Zhejiang Wanli Univ. - China)
- Bo Liu (University of Southampton - UK)
- M. Sushil - Lecturer
- Merwyn Monteiro (University of New South Wale - Australia)
- Rod Chapman (Praxis - UK)
- Marcel Verhoef (Chess - NL)
- Divakar Yadav (U P Technical University - India)
- Ait-Sadoune (LISI/ENSMA - France)
- Kenyu Yamada
- Ruchika - Lecturer
- Stéphane Badreau (Capgemini - France)
- Denis Grotsev (Kazakh National University - Kazakhstan)
- Abderrahman Matoussi (LACL Paris 12 - France)
- Dave Nuttall (MBDA Systems)
- Atif Mashkoor (Nancy University - France)
- Luke Wildman (WRSA, RAMS - Australia)
- Stephen Wright (University of Bristol - UK)
- Mahdi El Masaoudi (Sherbrooke University - Canada)
- Frederic Gervais (Université Paris-Est - Paris)
- Benjamin Aziz (STFC Rutherford Appleton Laboratory, UK)

Joining the DIG is simple. Please send an electronic letter of intent to the Dissemination & Exploitation Manager (thierry.lecomte@clearsy.com)

Past Event

- Rodin User and Developer Workshop (Southampton, 15-17 July 2009)

This workshop, held at Southampton University, aims at bringing together existing and potential users and developers of the Rodin toolset and to foster a broader community of Rodin users and developers. For Rodin users the workshop will provide an opportunity to share tool experiences and to gain an understanding of on-going tool developments. For plug-in developers the workshop will provide an opportunity to showcase their tools and to achieve better coordination of tool development effort. The workshop will be preceded by a 1-day tutorial on Rodin plug-in development on 15 July 2009. The tutorial is intended for existing and prospective plug-in developers. It will give an overview of the architecture of Rodin and provide guidance on building plug-ins. The main workshop will be on 16-17 July. The format will be presentations together with plenty of time for discussion. Broadly, Day 1 will be devoted to tool usage while Day 2 will be devoted to tool developments.

Future Event

- "Recent Innovations and Applications in B" Workshop (Eindhoven, 3 November 2009)

This workshop will be held during FM'2009 conference as part of the Formal Methods Week in the Technische Universiteit Eindhoven auditorium . It aims at providing a clear picture of B/EventB current status of development and dissemination, focusing on the industrial applications (ranging from pilots to large-scale implementations) as well as the ongoing tool developments. The workshop will include a large scope of presentations given by DEPLOY project members, industrialists and researchers:

- EventB in Space – or are we still on the ground? - Dubravka Ilic, Space Systems Finland
- Formal Development of Enterprise Service Communication - Andreas Roth, SAP AG
- A proved "correct by construction" realistic digital circuit - Marc Benveniste, STMicroelectronics
- Formal Methods Outside the Mother Land - Aryldo G Russo Jr. / AeS Group & Research Institute of State of Sao Paulo (IPT)
- Probabilities in EventB for railways safety critical systems - Jérôme Falampin, Siemens Transportation Systems
- ProB for Validating Large Scale Railway Models - Michael Leuschel, University of Düsseldorf
- EventB recipes for proof-based design of distributed systems - Dominique Méry, Université Henri Poincaré Nancy 1, LORIA
- The cruise control system as a pilot application - Michael Jastram (University of Dusseldorf) - Christine Rossa (Bosch) - Rainer Gmehlich (Bosch)
- Automatic refinement and code generation: lessons learned - Thierry Lecomte, ClearSy
- Formal modelling Feedback on Train Tracking - Mathieu Clabaut, Systerel
- The Rodin platform: latest and future additions - Michael Butler, University of Southampton

---

[1] Audi. Audi Online Glossary, July 2008. http://www.audi.com/audi/com/en2/tools/ glossay.html.

[2] Michael Jackson. Problem Frames and Methods. Analysing and Structuring Software Development Problems. Addison-Wesley Longman, September 2000.
[3]The **Meta-Object Facility** (**MOF**) is an Object Management Group (OMG) standard for model-driven engineering.