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

DEPLOY Deliverable D2

D14.04 Electronic Newsletter

Thierry Lecomte (ClearSy)

Public Document

31th January 2010

http://www.deploy-project.eu

DEPLOY PROJECT

Newsletter N° 4

Contents

Partners

Introduction

In the last edition of this newsletter, Michael Butler gave an update on the Rodin tools (platform and plugins); in this edition we provide a quick summary of developments on the methodology front. (More detail can be found by looking at the public deliverables at www.deploy-project.eu/html/deliverables.html or by reading papers in the extensive repository at deploy-eprints.ecs.soton.ac.uk).

The base methodology of the DEPLOY project is to use Event-B. It is described in a book that will soon be published by Jean-Raymond Abrial (The Event-B Book, Cambridge University Press, 2010). Jean-Raymond has kindly made some of the material available at www.event-b.org/abook.html. The Event-B notation is a development of Abrial's earlier work on B that is published –also by CUP- in his famous 1996 book. The Event-B method is, however, much more than a notation and the new material describes the use of abstraction and refinement as ways of both arriving at specifications and developing them through to implementations that satisfy the their specifications. Considerable emphasis is placed on getting well-structured natural language descriptions. The whole emphasis on modeling systems is central to our current methods. Of course, it is key to our view of deploying formal methods (for dependable systems) that tool support is mandatory.

Not surprisingly, some applications fit the Event-B approach better than others and it is the role of the methodology research activity within the DEPLOY project to look at areas where more work is required. There is often a way of formulating a version of a system in Event-B and this has the advantage of maximizing the use of the existing tools. In many cases, we need to plan further plug-ins.

One area where our deployment partners need extensions is to interface their existing tools and methods with our ideas. One partner has made use of Michael Jackson's Problem Frame Approach to organize their analysis of requirements. We are currently investigating the relationship between refinements in Problem Frame Diagrams and those in Event-B.

An area that is present in several of the deployments is the transition from continuous time to discrete systems. Many of the phenomena of moving objects inevitably involve continuously varying quantities. This is an issue that is not difficult to circumvent by fixing some sampling interval. In fact, in most applications, such a "discretization" is part of the general engineering approach. It is however technically interesting to look at ways of analyzing, say, rates of change and computing safe sampling frequencies. One approach to this is described in papers by Hayes/Jackson/Jones (a recent paper was published in Springer's LNCS 4700).

The realization that we would need to look at extensions to methods was clear in the DEPLOY project proposal. One such topic that was mentioned there was the need to handle probabilities in system specification and design. This is particularly important for dependability where nothing is certain and what one wants is to specify a required probability of the overall system achieving its optimal behaviour when the possibility of component failure must be accepted. Now working in our Zürich team, Thai Son Hoang's thesis was on this topic and we now have another PhD student in Newcastle looking at stochastic specifications: Zoe Andrews recently escaped the European freeze and spent a spell with Carroll Morgan and Annabelle McIver in Sydney.

Another interesting methodological development related to dependability is the work with our SAP deployment partner on establishing service levels with differing assumptions about message delivery in the underlying middleware (see papers on the repository by Bryans et al.).

As Michael Butler indicated in the preceding newsletter, code generation is an area whose importance has increased since our initial DEPLOY proposal. Although this can be seen at one level as a tools requirement, we are investigating how it fits into an overall development method.

Many more topics are discussed in the public deliverable D6.1. For further developments in the Methodology Work Packages (WP6 Requirements led by Michael Butler, WP7 Reuse led by Kaisa

Sere, Dependability led by John Fitzgerald), please continue to watch the Deploy e-prints server.

If I could close with a plug for one further paper, that by Woodcock et al. from the recent ACM Surveys (also visible on our e-repository) provides an up-to-date view of formal methods use in industry.

Cliff Jones, Newcastle University,
Methods Coordinator of the DEPLOY project

Preamble to evidence

Transferring formal engineering methods to the industry has turned out to be difficult process for a variety of technical and non technical reasons. Introducing formal methods in industrial process generally involves a deep change of the development process because these methods have very different properties than traditional engineering methods. For instance, they can provide proof of correctness at the cost of increased modeling effort in the design phase so that testing can be spared later in the development. In order to achieve an effective transfer, it is crucial to assist industrial organizations in their decision making process about the opportunity to adopt such methods and the extent of this adoption. This is precisely one of the goals of the DEPLOY project.

The DEPLOY project is currently intensively deploying a set of methods and tools (Event-B, RODIN) in real industrial settings covering four key industrial sectors, namely: automotive, transportation, space, and business information systems. Those close-to-reality deployments are continuously monitored and information is being collected both qualitatively and quantitatively about the benefits, traps and pitfalls of such methods. So far, some of the identified benefits include the improvement in requirements quality, the early detection of design problems, the increased automation of tasks such as tests, code generation, etc. The way to conduct changes to the development ecosystem are also gathered notably about the kind and amount of training required, the integration in the development process and the tool chains.

This activity also helps in enriching and tuning the applied methods and tool chains in order to adapt them to the needs of the industrial partners according to the continuous feedback from their experience.

The experience collection will also not be limited to four industrial deployments but it will also be enriched by experience of a second wave of partners engaging into more focused deployments. Previous experiments done by others are also taken into account to some extent. Eventually a comprehensive guide for managers embodying lessons learned will be produced and made publicly available.

Newsletter 5 will provide more details about evidence gathered within the DEPLOY project.

Christophe Ponsard, Renaud De Landtsheer, CETIC

Update on the RODIN Platform

Rodin Release Policy

A policy for releasing new versions of the Rodin platform has been defined and is now enforced. The main points of this policy are:

- A new version of the Rodin platform is released every three months.
- The code is frozen during the two weeks preceding each release, so that plug-in developers can check that their respective plug-in work with the coming platform release and take any appropriate corrective action.
- The Eclipse versioning policy is enforced.
- A wiki page is dedicated to each release.

For more information about releases of the Rodin platform, please have a look to wiki page : http://wiki.event-b.org/index.php/Rodin_Platform_Releases.

The release notes highlight the new major features. More details are provided in the distributions of the platform.

Rodin Release 1.1 (15 October 2009)

See http://wiki.event-b.org/index.php/Rodin_Platform_1.1_Release_Notes. The major changes are summarized below:

- The Rodin platform now provides more ways to enter mathematical symbols. You can now either type the ASCII shortcut (as in previous releases), or type the LaTeX command (as defined in style bsymb), or click in the Symbol Table view which displays the symbols graphically, or directly enter the Unicode value of the symbol (for advanced users).
- The provers have been improved, to reduce proving time and effort (new proof rules), to reflect the corrections on provers (reasoner versioning), to reduce the proof storage space (proof purging, proof simplifying, non-textual database storage), to facilitate the manual proof review or reuse (proof skeleton view, copy / paste from skeleton to edited proof), to evolve the prover API (new tactic provider API)
- When a theorem guard is repeated in a concrete event, no THM proof obligation is generated any longer if the theorem occurs after the same guards as in the abstraction.
- The contexts and machines may be marked as being generated. The generated elements cannot be edited through the Event-B editor.

Rodin Release 1.2 (29 January 2010)

See http://wiki.event-b.org/index.php/Rodin_Platform_1.2_Release_Notes. In particular:

- This release is based on Eclipse 3.5 (Galileo).
- The Rodin platform provides an extension for predicate variables. More precisely, it is possible to define proof rules for the rule based provers using meta-variables as predicate placeholders.
- Any internal element may be tagged as generated.

Wiki, Mailing-lists and Trackers

The Event-B wiki remains the reference for all documentation (for both developers and end-users) about the Rodin platform.

In particular, you can find information about the current developments, the Rodin related mailing lists, the SourceForge trackers to report bugs or request new features. Do not hesitate to contribute:

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

Laurent Voisin, Systerel

Update on the RODIN Plug-ins

Several new plug-ins have been developed and are available in prototype form for the Rodin platform:

- **Decomposition plug-in.** This plug-in allows an Event-B model M to be decomposed into several separate sub-models M1,...,Mn. Two methods, namely the shared variable decomposition and the shared event decomposition, are supported.

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

- **Renaming plug-in.** This plug-in allows renaming of variables, parameters, carrier sets, constants, or any labeled elements (axioms, invariants, events, guards, actions). The renaming is factored through refinement chains and proofs.

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

- **Modularisation plug-in.** This plug-in provides facilities to structure Event-B developments into logical units of modelling, called modules. An integration of a module into a main development is accomplished by referring operations from Event-B machine actions using an intuitive procedure call notation.

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

- **Pattern plug-in.** This plug-in allows reuse of existing Event-B developments, referred to as patterns, in a new development.

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

- **Rule-based prover.** This plug-in provides a mechanism by which the prover can be extended with new rewrite rules. Rules may be interactive and automatic. The soundness of rules is verified through proof obligations when rules are added.

  See http://wiki.event-b.org/index.php/Rule-based_Prover_Plug-in

Moreover, the following plug-ins have been improved:

- **ProB plug-in.** Some new features have been added (multi-level animation and validation, B-Motion Studio, first steps towards test-case generation) and some improvements have been performed on existing features (scalability improvements, using proof information to improve model checking).

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

- **UML-B plug-in.** This plug-ins now supports the refinement of state-machines. Moreover, it now includes an UML-B State-machine Animation feature.

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

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
Carine Pascal, Systerel

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.

The DIG is currently composed of 48 members.

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

DEPLOY Associates

The DEPLOY Associates is a group created late 2009, gathering privileged industrial experimenters of the DEPLOY tools and methodology. The main goal of this group is to ensure broad dissemination of the results of the project (tools, methodology, documents, etc.) by:

- experimenting on new case-studies, possibly fromdomains not yet addressed by the DEPLOY project
- ensuring that adequate training is delivered to the DA personnel in charge of the case-study,

in order to obtain comparable results among DAs
- collecting feedback (metrics, models, conclusions, etc.) from DA, in order to improve project deliverables and to demonstrate the extent to which they are applicable to industry.

The DEPLOY Associates receive specific and dedicated help from the DEPLOY project (training, consultancy, etc.).

Two DEPLOY Associates have been selected so far:

- Automação E Systémas – Sao Paulo (Brazil)
- Critical Software Technologies – Southampton (U.K.)

If you would like to join the DEPLOY Associates, send your application to the Dissemination & Exploitation Manager (thierry.lecomte@clearsy.com).

Future Events

B Dissemination Day workshop (Tokyo, 17 March 2010)

This workshop, satellite event of the GRACE International Symposium on Advanced Software Engineering, held in Tokyo, aims at providing a clear picture of B/Event-B current status of development and exploitation, focusing on the support tools as well as the industrial applications. The workshop includes a large scope of presentations given by the DEPLOY project members or associated to the project results. Target audience is software/system engineers and project managers, as well as researchers in the domain.

Related link: http://events.grace-center.jp/symposium/2010en/workshop_tclearsy

Workshop on B Dissemination (Natal, Brazil, 8-9 November 2010)

This workshop, satellite event of SBMF 2010 conference, held in Natal (Brazil), is organized within the framework of the DEPLOY project. Its objectives are to present current status, ongoing research and development related to B and event B languages, as well as applications to industry size problems. Topics addressed by the workshop are many:

- Tool development (language extensions, external provers, code generation, etc.)
- Modeling challenges (real time properties, probabilistic refinement,high order logic, etc.)
- Deployment (methodology, cases-studies, return of experience, scaling up, etc.)

The workshop is intended to last 2 days:

The first day is be devoted to DEPLOY speakers. General presentation of the project and tools are accompanied by focused talks on scientific/technical matters (modeling time, code generation, model animation, model checking, etc.) that are being researched in DEPLOY. Reports on industrial applications (space, railways, automotive, information systems, etc.) complete the day.

The second day is open to any presenter, through an international call for papers, to appear. Expected contributions would range from theoretical research to practical applications of B/event B.