RODIN

Scope

Developing computer systems that can be justifiably trusted, i.e., dependable systems, is one of the major challenges that the ICT community is now facing. The complexity of modern systems is currently perceived as the main dependability threat. A major mechanism for han-dling complexity is the deployment of abstract modelling and incremental development. These are the main leverages that formal engineering methods offer us to master complexity and achieve high degrees of dependability. Moreover, formal modelling and design have the potential to lower the development cost by dramatically reducing the cost of testing and redevelopment. However, development of modern software-intensive systems is unfeasible without powerful automation of the modelling and design process. The toolsets enabling automation should be flexible to deal with domain-specific problems yet persistent in enforcing rigorous development methods. The development methods should be based on a sound mathematical basis and be scalable to deal with realistic industry-based applications. The Rodin (Rigorous Open Development Environment for Complex Systems) project has systematically addressed these issues and created an advanced open environment for rigorous development of complex software-intensive systems.

Advances

Rodin has made substantial advances in our understanding of how modern software-intensive systems should be built. The project has resulted in producing:

- *an open tool kernel supporting extensibility of the underlying formalism and integration of tool plug-ins.*
- *a collection of plug-in tools for model construction and verification, model checking, behaviour simulation, testing and code generation.*
- *a set of guidelines on using a systems approach in the rigorous development of complex systems, including design abstractions for fault tolerance, architectural guidelines, and techniques for model decomposition.*
- *a collection of reusable domain-specific development artefacts such as models, architectures, proofs, components, etc. created and validated by the industrial case studies.*

These results will be used by system and software engineers, software architects and system integrators to ensure system dependability and quality, and by the technology providers to extend the platform with the new tools.

Positioning in global context

Model-driven development has become a leading paradigm in developing complex software-intensive system. However, the existing methods and tools are either unable to provide rigorous assurance in correctness of system behaviour or to scale for industrial development. Rodin has advanced the formal development methods and created the supporting toolset to embrace the challenge of *formal modeldriven development that scales to large industrial systems*. The outstanding academics consolidated their efforts in creating a viable automated formal design method and opensource extendable toolset. This is a unique effort that paves a path towards integration of various modelling approaches into a powerful versatile platform supporting design of dependable systems.

Contribution to standardization and interoperability issues

The Rodin platform is an Eclipse-based IDE and as such follows all conventions for developing Eclipse tools ensuring their high interoperability.

Target users / sectors in business and society

The Rodin results will be used by

- *software and application developers*, who will be able to build dependable software systems in a rigorous way focusing on producing models that can be easily traced to the requirements and on ensuring system dependability starting from earlier development steps.
- *system integrators*, who will benefit from better understanding of the individual system components accompanied by their abstract formal models and from the improved quality of the component developed formally.
- *technology providers*, including SMEs, that will be able to develop new tools and smoothly integrate them into the Rodin extendable environment.

Overall benefits for business and society

With our growing reliance on software, the total societal costs of its failures are hard to underestimate. Yet the analysis of recent software-caused accidents has shown that the current development process is inadequate for achieving high dependability of software-intensive systems. This is mainly caused by the fact that testing – traditionally used to assure software quality – is unable to provide us with the desired degree of quality assurance, even though it often takes up to 70% of the development time. Rodin has advanced the state-of-art in development of de-pendable systems by putting forward a systems approach to software development, enabling automated formal modelling and verification, as well as facilitating automatic model-based test-ing. Hence the results of the project will allow software developers to reduce the amount of testing, yet empower them to design systems, which are more robust, better structured and analysed. This should have long-term benefits not only for the European business environment but also for society overall.

Examples of use

The Rodin development environment has been used in a number of industrial case studies within and outside of the project. Systems modelled and analysed include an air-traffic information display system, a railway interconnect system, an engine failure management system, an ambient information system and mobile telecoms protocols. Detailed discussions of the system requirements, models and patterns applied during their developments and the experience in using the tools can be found in the project deliverables at *rodin.cs.ncl.ac.uk*.

Achievements

- **The Rodin open platform** for stepwise development of complex software intensive sys-tems. The platform is designed to support the use of formal modelling from early stages of system development. It is developed as an open source and can be freely downloaded from www.event-b.org.
- **A palette of plug-ins** extending the platform functionality. All the plug-ins are developed as an open source and can be freely downloaded from www.event-b.org.
- **Fault tolerance guidelines and patterns** for formal stepwise development of systems capable of tolerating various types of faults and errors. These guidelines are described in the project deliverables and can be downloaded from the project site *rodin.cs.ncl.ac.uk*. The overall summary is provided in deliverable D29 *(Final report on methodology)*.

RODIN

title
Rigorous open development environment for complex systems

contract number
511599

type of project
Specific Targeted Research Project

contact point
Alexander Romanovsky
UNIVERSITY OF NEWCASTLE
UPON TYNE, UK

project website and partner list
http://rodin.cs.ncl.ac.uk/

EC contribution
3 171 000 €

start date
01/09/2004

duration
38