RODIN (Rigorous Open Development Environment for Complex Systems)

Project Number: IST 2004-511599

Joey Coleman[1], Cliff Jones[1], Ian Oliver[2], Alexander Romanovsky[1], and Elena Troubitsyna[3]

[1] University of Newcastle upon Tyne, UK
[2] NOKIA Corporation, Finland
[3] Åbo Akademi, Turku, Finland

Partners:
University of Newcastle upon Tyne, *UK* (Coordinating Site)
Åbo Akademi, Turku, *Finland*
ClearSy System Engineering, *France*
Federal Institute of Technology (ETH), Zurich, *Switzerland*
NOKIA Corporation, *Finland*
Praxis High Integrity Systems Ltd, *UK*
University of Southampton, *UK*
VT Engine Controls Ltd, *UK*
Industrial Interest Group:
Adelard, *UK*; Alstom Transportation, *France*; CETIC, *Belgium*; DGA, *France*; Escher Technologies, *UK*; Gemplus, *France*; IBM *UK*; I.C.C.C. Group, *Czech Republic*; QinetiQ, *UK*; RATP, *France*; STMicroelectronics, *France*; SYSTEREL, *France*; VTT, *Finland*
Duration: 36 months from September 1, 2004
Contact: Alexander Romanovsky
School of Computing Science
University of Newcastle upon Tyne
Newcastle upon Tyne, NE1 7RU, UK
Email: `Alexander.Romanovsky@ncl.ac.uk`
Website: `http://rodin.cs.ncl.ac.uk/`

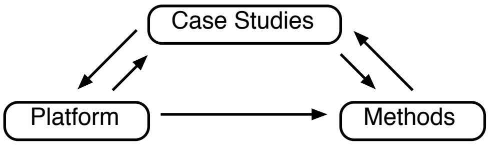
1 Introduction

The RODIN STREP project's sharply focused objectives fall squarely within the remit of the IST strategic objective 2.3.2.3 Open Development Platforms for software and services of the FP6 second call[4].

Members of the RODIN consortium have been involved in the inception of earlier formal methods (C.B. Jones' on VDM and J.-R. Abrial's on B) but these methods did not explicitly cover fault-tolerance. One of the main strands of work in RODIN is development of methods (and platform) that support the design of systems that are tolerant of faults and unpredictable changes in the environment.

[4] `http://www.cordis.lu/ist/so/sw-platforms/home.html`

The interplay between methods, tools and case studies is at the heart of the RODIN plan. One of the main criteria of our work is a proper validation and assessment of the methods and platform through industrial case studies. Earlier methods developed by the project partners have had tool support, a feature crucial to their adoption by industry. The arrows in the following diagram show the influences which we expect to utilise within the project in order to achieve the most applicable outcomes. (The detailed RODIN plan expands on these points.)



The RODIN partners form a strong coalition capable of achieving the challenging project objectives. VTEC, Nokia and Praxis have solid experience in using rigorous methods and bring case studies into the project. ClearSy have successfully developed high quality industrial software. Recently Nokia and Southampton have gained considerable experience in tool development and integration. In the consortium we have partners with excellent research background in various formal methods which complement each another (Åbo, Southampton, Newcastle, ETH). By including several partners with outstanding backgrounds in dependability, we can ensure that high-level research on the fault tolerance techniques will be applied at the application level (Newcastle, Southampton, Åbo).

2 Main objectives of the project

The overall objective of RODIN is to create a methodology and supporting open tool platform for cost-effective, rigorous development of complex, dependable software systems and services. The project focuses on tackling complexity caused by the environment in which the software is to operate; and that which comes from inadequate architectural structure. Mastering complexity requires design techniques that support clear thinking and rigorous validation and verification; formal design methods do so. This also requires architectures that are tolerant of faults and unpredictable changes in environment.

The project will develop a *unified methodology* combining formal methods with fault-tolerance design principles by using a systems approach, where both software and environment are modelled together. We will tackle complex architectures: the systems approach will support the construction of useful abstractions and provide techniques for structured refinement and decomposition.

To maximise cost-effectiveness, the methods and platform will support reuse of existing software. We will thus extend existing formal methods with generic mechanisms to support *component reuse and composition.*

Tool support for construction and analysis of models is crucial and we will concentrate on a comprehensive *tool platform* which is *openly available* and *openly extendable*. The methods and platform will be assessed through case studies.

Novel aspects in this project include the pursuit of a systems approach; the combination of formal methods with fault-tolerance techniques; the development of formal methods support for component reuse; and the provision of an open, extensible tools platform for formal development. In particular, we believe that the open tools platform will have a significant impact on future research in formal methods and will encourage greater industrial uptake; it has the potential to set a European standard for industrial formal method tools.

3 Workpackages

The planned work of the project is organised as four major research themes which correspond to four workpackages listed in this document. In addition to these, there are workpackages devoted to dissemination and exploitation; project management; and project review and assessment.

Workpackage 1 — Research drivers

The case studies in RODIN are from a wide cross section of industry and they represent a diverse selection of concerns. The fault tolerance aspects are critical: all case studies have dependability requirements and fault-tolerance measures incorporated.

Case study 1 — Formal Approaches to Protocol Engineering (Nokia) concentrates on the development of a 3GPP Positioning System that is capable of calculating the physical position of a mobile device. This case study will examine the kinds of information required to construct useful specifications. One of the major results of this work will be a better grasp of how such techniques can be integrated into, and enhance, current methodologies.

Case study 2 — Engine Failure Management System (VTEC) is based on an engine management system used in aerospace. The primary function of the system is to manage the failure modes of an engine control system. This case study will investigate methods of using formal methods efficiently via reuse, resulting in a (domain specific) open software support package

Case study 3 — Formal Techniques within an MDA Context (Nokia) is related to the construction and composition of systems utilising OO and "model-driven" techniques. As systems become more complex it is necessary to fit them into a framework to control their complexity; examples of this appear in end-to-end mobile environments. The focus of the case study is to investigate the role of formal specification and methodology in this context.

Case study 4 — CDIS Air Traffic Control Display System (Praxis) is a safety-related system that is responsible for displaying flight-related information to air traffic controllers. This case study provides initial material for our theoretical work by offering specification, design and verification material from a completed industrial-scale project by Praxis. This study aims to reconsider challenging issues that arose during that development.

Case study 5 — Ambient Campus (Newcastle) aims to explore Ambient Intelligence (AmI) as an emerging field which is lacking well-defined engineering methods. In this case study we will investigate how to use formal methods combined with fault tolerance techniques to develop highly dependable AmI applications. Moreover, we will develop modelling and design templates for adaptable and reconfigurable software.

Workpackage 2 — Methodology

Methodology consists of a number of parts: process, method and language (i.e. semantics and syntax). We plan to build on our experience with earlier formal methods, recent unpublished work by J.-R. Abrial on fault-tolerance, and on current work about determining system specifications from their environment.

RODIN's focus includes all three components of methodology, but it will concentrate on method. The project will produce a set of methodological components that can be chosen as appropriate to their application. Of course, the stricter the usage, the more formal the development and results but the less generally applicable that particular incarnation of the method will be. This choice is left to the developer but leaves a number of questions surrounding how rigorous the specification is. We take the view that some formal specification is better than none at all largely because subsequent review processes can identify if and where more detail is required.

Workpackage 3 — Open tool kernel

The objective of this workpackage is to develop the basic kernel tools to support the RODIN methodology. They will be implemented within the Eclipse platform to allow for interaction with the plug-ins from WP4. The tools will be designed with formal development in Event-B in mind, but they will be as general as possible. The final platform will be mostly released as open source software.

Workpackage 4 — Modelling and verification plug-ins

The tools associated with RODIN will be expected to provide theorem proving, animation, simulation and model checking capabilities associated with the formal language used. Of course, when using languages such as UML we are not limited to just one language but a mix of languages. The plug-ins will support linking UML and B, Petri net and constraint-based model checking, model-based testing and code generation. The plug-ins will be tested by use in the case studies.