 THE RESEARCH LEADING TO THESE RESULTS HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S SEVENTH FRAMEWORK PROGRAM (FP7/2007-2013) FOR CRYSTAL – CRITICAL SYSTEM ENGINEERING ACCELERATION JOINT UNDERTAKING UNDER GRANT AGREEMENT N° 332830 AND FROM SPECIFIC NATIONAL PROGRAMS AND / OR FUNDING AUTHORITIES.

CRitical S**YST**em Engineering **A**cce**L**eration

System & Software Development Lifecycle

D611.011

DOCUMENT INFORMATION

Project	CRYSTAL
Grant Agreement No.	ARTEMIS-2012-1-332830
Deliverable Title	System & Software Development Lifecycle
Deliverable No.	D611.011
Dissemination Level	CO
Nature	R
Document Version	V1.0
Date	2014-01-31
Contact	Gray Bachelor
Organization	IBM UK
Phone	+44 784 1011 431
E-Mail	gray_bachelor@uk.ibm.com

AUTHORS TABLE

Name	Company	E-Mail
Hanno Schouten	TNO	hanno.schouten@tno.nl
Sytze Kalisvaart	TNO	sytze.kalisvaart@tno.nl
Jon Chard	IBM UK	jon.chard@uk.ibm.com
Gray Bachelor	IBM UK	gray_bachelor@uk.ibm.com

CHANGE HISTORY

Version	Date	Reason for Change	Pages Affected
0.1	20/1/13	Assembled from TNO and IBM materials	All
0.2	22/1/14	Updated with IBM bricks	Many
0.3	29/1/14	Updated with IBM Bricks	Many
0.4	30/1/14	Input from final review	Use-case
1.0	30/1/14	Input from final review	Multiple

CONTENT

1 Introduction

1.1 Role of deliverable

D6.11.1-1/2/3 This deliverable contains the specification, development and assessment of bricks of this corresponding work package (WP6.11). Each brick shall be represented in different chapters of this deliverable. The document will be released three times throughout the project duration whereas the first will only contain the specification part.

The related Brick application providers and owners are responsible for their own content herein and any subsequent contributions or delivery of SDLC bricks towards the Crystal Domain activities. IBM as organiser of WP6.11 accepts no responsibility for such content herein or deliveries.

This is the first iteration of the D6.11.1 deliverable and is limited to input from IBM and TNO. IBM provides an initial set of brick information at M9 prioritised by input from the Aerospace Public Use Case analysis and demonstrator.

1.2 Relationship to other CRYSTAL Documents

For the IBM Rational Solution within WP6.11 this document should be used in conjunction with D6.11.5.

Where relevant and where known today additional document references are provided below.

1.3 Structure of this document

The document is simply organised by bricks and additional commentary and analysis to support TNO's work so far, it is organised by brick within the body, with additional analysis in an Annex. The equivalent IBM analysis material is shown in D6.11.5.

1.4 Trademark acknowledgement

All trademarks and logos are acknowledged as being owned by their respective owners.

2 Crystal brick 4_9a: Lifecycle management of simulation models

2.1 Description

Name:	Lifecycle management of simulation models
Contact:	Hanno Schouten (TNO), Sytze Kalisvaart (TNO))
Dependencies	
License	
Additional information	Under development Links to Brick 4.6 and 4.9a. Work belongs to Task WP6.11.5

2.1.1 Introduction

The task WP6.11.5 Lifecycle management of simulation models is oriented toward a working method rather than a tool. The goal is to manage the evolution of Matlab models such as brick 4.9a (Performance simulation) and brick 4.6 (Hardware in the loop) and to improve the user friendliness of configurable models. These models are used throughout the model-based development process of one or more projects or the basis of architectural decisions for a complete product line. Similar to Software Configuration Management, this method will manage the evolution of these models and their metadata. The metadata will not only consist of documentation and version management data, but also allows for tracing model configurations, validation tests and results, simulation results and use-cases for certain model configurations.

2.1.2 Manual

In complete analogy with Software Configuration Management, simulation models also require tracking of versions and variants. Typically, the models are adapted and modified within one project, reused in the next project and configured for test cases and for hardware variants. Variants of models may exist at various abstraction levels, e.g. single parameters models, parametric models, and full detail models. Typically, the model level variants are developed in the course of several development cycles and keep being used in parallel. Also the purpose of a model can be different per variant, and it can change over time. Furthermore, the validation quality of a model needs to be characterized and traced. Test results on implemented systems may result in improved models (using data assimilation techniques). This results in a wealth of model versions with metadata, which need to be managed.

This method will support the engineering methods from the healthcare use cases WP403 and WP406. In WP403 and WP406 TNO will develop models in Matlab/Simulink to support Hardware-in-the-Loop testing and model-based software testing. In this task (WP6.11.5) TNO will work on a method to manage such Matlab models. The Matlab models represent dynamical behaviour of physical systems. They are continuous signal, multi physics models containing mechanical, electrical and software engineering.

TNO will focus its work on Matlab in order to minimize the number of software tools needed and as a starting point for use with other simulation models, e.g. Dymola or Rhapsody. Other partners are invited to apply and evaluate the working method for their favourite simulation tools.

In case additional tools are needed, the selection of the tool depends on the character of the development process to be supported: for a research oriented, very variable development process, free-ware tools are preferred (e.g.: simple version management using SVN). For highly incremental, industrial development and engineering methods more elaborate and structured tool environments are preferred, such as Rhapsody Designer.

It is also preferred that the resulting method or tool will be open or according to open standards, e.g. the OSLC standard. In this way the method is open to customisation for other users and can be included in other (existing) system engineering tools. This will provide interoperability for at least the metadata of the Matlab models. The method will be accessible to partners as a collection of files representing a simple example. From this example, developers are free to link to parts of the code for their own tooling.

2.1.2.1 Exemplary manual

A manual for using the simulation model lifecycle management method are as follows:

Preparation

Typical users: system engineers, system architects, simulation model developers, software engineers, demonstration staff

- Obtain Matlab license for running or developing models. This may also be a comparable simulation tool such as Dymola.
- Obtain authorization for accessing the simulation model server
- Enter preferences for default output file paths, location of test database, preferred viewer tools, etc.

Using simulation

User type	Goals
System architect	Analyse total system performance
System engineer	Simulate performance of a function
Demonstration staff	Show intended behaviour of system
Usability expert	Research appropriate behaviour of system with end users
Software engineer	Test code developed against realistic simulation of other parts of the system, e.g. hardware

- Open web browser with model access links
- Identify key model parameters such as included model parameters, function to be simulated, black box boundary (smallest detail modelled as white box), maximum simulation time, and validation status.
- Search model server with key model parameters. The model server presents the most recent versions of models by default. You may browse through model test results and model validation data.

- Select simulation model.
- A remote model interface opens. Specify model parameters such as data set and output data file path and name. Possibly, the simulation model is configurable for your desired simulation.
- Select “Start simulation”.
- The simulation will run and create a result data file. It will notify you once the simulation is ready and offer a link for opening the output file.
- The output file is automatically filled with metadata on simulation model and version, input data set and location and simulation parameters. In parallel, your simulation is added in the test database as a link.
- You can open and analyse the output file in your favourite viewer or plotting tool.
- You can add your conclusions and discussion to the test database and define follow actions.

Reproduction of field call

This is a special case of the “Use simulation”

User type	**Goals**
Product lifecycle team	Detect technical cause for defect in field call.
System engineer	Reproduce defect without need to recreate hardware and software configuration completely

- Open web browser to access field configuration database.
- Select the client that produced the field call and the specific device for which the field call was entered.
- Select the client configuration.
- Press “Search simulation”. The related historical version of the simulation that is relevant for this configuration is selected automatically, if available. Possibly, several choices are presented from which the system engineer selects the most suitable simulation.
- As a simulation database may not be complete, the system engineer may need to partially reconstruct the hardware configuration or adapt the simulation model.
- In case a log file of the defect is available, this file can be used as input for the simulation. The log file will be analysed on signals which can be used as input for the simulation. The user can then manually select which signals must be replayed in the simulation, e.g.: given user input when defect occurred.
- The system engineer configures and runs the simulation and tries to reproduce the defect.
- In case the defect is reproduced, the configuration, context, state, parameters and possible solutions are automatically added to the test case database.
- In case the defect is not reproduced, the simulation model is labelled as incomplete for the field call test case. Future model adaptation may be scheduled if deemed of sufficient priority.

Developing simulation models

User type	**Goals**
System engineer	Create virtual test environment for software code or control system

Simulation model expert	Create beautiful and elegant simulation at appropriate level of detail and scope for the intended use
System architect	Create high level simulation for product function behaviour

- Open web browser to access requirements database or system function list.
- Select requirements or system functions for which the simulation is made.
- Define model parameters such as intended use and model scope and detail of simulation.
- Create your code in the preferred modelling environment, for example in Matlab/Simulink.
- Run and test simulation in the modelling environment.
- Validate the simulation with real life measurements (also done in the modelling environment). This may only be possible once partial hardware is created or a test setup. The validation tests are referenced in the simulation model and stored in the test database.
- Publish the simulation. Model intended use, scope, detail, configuration parameters, simulation time estimator, validation status are automatically stored in the simulation model database.

Validate model

This step is a highly challenging scientific task as a good reference for validation may not be available for new systems. Incremental system development is very helpful in this case as real system behaviour may be derived from systems already in the market.

User type	**Goals**
Simulation model expert	Make sure the simulation represents the real system behaviour at sufficient accuracy.
System architect	Make sure the architectural decisions are not based on flawed or optimistic simulations.

- Design minimal validation test for verifying key performance parameters of the system that is modelled.
- Define test setup for validation.
- Perform validation test.
- Try to creatively explain the differences between simulation and test results.
- Update model to include suspected missing behaviour.
- Rerun validation test. Also identify limits of the model, or define operation area's in which the model has a high/medium/low confidence level.
- Once the simulation model is sufficiently validated, the simulation expert declares the model validated and the validation test is automatically added to the test result database. The simulation model contains a link to the validation test and the validation level is characterised as a metadata parameter of the model.

2.1.3 State of the art

Since Matlab/Simulink is a well-known tool for several years and very much suited for a model-based development approach, there is a fair chance that a tool already exists which (partly) meets the described functionality. An initial search on the internet showed that the following two tools seem to offer a large part of the desired functionality. Note that, at this point these tools are not yet further investigated, and there can exist more comparable tools.

Simulink project

Simulink project is a Matlab toolbox from the Mathworks to support the management of Matlab files within a project, including version management. It also enables a connecting to other tools for source control, version control, software configuration management (SCM), product lifecycle management (PLM), and application lifecycle management (ALM).

Source: http://www.mathworks.nl/discovery/simulink-projects.html

Teamcenter

Teamcenter is a product lifecycle management tool by Siemens PLM Software. The integration with Matlab/Simulink enables information exchange between Matlab en Teamcenter. It is therefore possible to capture and trace metadata from Matlab into Teamcenter.

Source: http://m.plm.automation.siemens.com/en_us/Images/11242_tcm1224-46604.pdf

2.2 Use Case coverage and application

Please also refer to

1. See table “Technology Brick”, column “Use Case Specific Needs” (https://projects.avl.com/11/0154/Lists/Technology_Bricks/AllItems.aspx)

Use-case	Additional notes
4.3	
4.45	

2.2.1 Use Case 4.3

In Use Case 4.3 simulation models are used to test the developed software in the software- and system- verification phase of the development process, see Figure 1. A detailed description of the needs from Use Case 4.3 for simulation model lifecycle management can be found in Annex I.

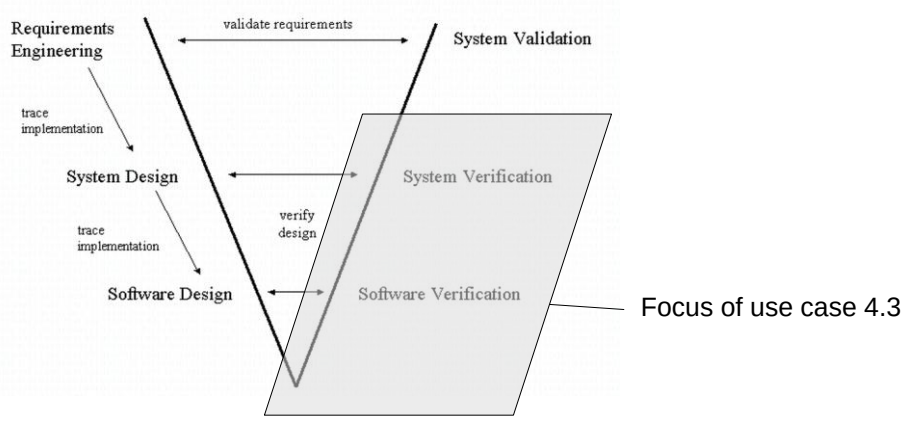

Figure 1: Focus of Use Case 4.3 shown in V-model

This use cases requires a method which is able to

- Perform version control of various models, sub-models and possible configurations
- Handle different possible levels of implementation / details of sub-systems

- Track validation test and their results for models, sub-models and different configurations
- Track possible and/or useful configurations of the models
- Documentation of model design decisions, model code explanation and model configuration options
- Use-case examples for the models or certain configurations

The goals listed above will mainly support the model developers. To improve the use of these models for a user who is not developing the models, the method should also be able to:

- Select the proper model configuration and sub-systems
- Switch sub-systems between different level of details
- See validation status of models and their different sub systems
- Select the proper test routines for a certain model
- Select the desired reporting of results

This means that a direct link to Matlab is desired. In this way the tool can be used to configure or setup your Matlab model.

The technical requirements following from Use Case 4.3 are:

ReqID	Requirement description	Stakeholder(s)	Interoperability	MoSCoW	Rationale
REQ_B_409a.1	perform version control to all models and auxiliary files	model engineer	y	M	track changes and process of model development.
REQ_B_409a.2	track configurations of models	model engineer	y	M	track possible and used configurations of sub-models to combine in working simulation models.
REQ_B_409a.3	take into account different level of detail of models and sub-models	model engineer		M	
REQ_B_409a.4	link validation tests and their results to models and its configuration	model engineer	y	M	document what is validated with which results
REQ_B_409a.5	provide overview of available models. Narrow possibilities by several choices made by the user	software test engineer		S	Help a user to select the proper simulation model
REQ_B_409a.6	Configure or build up the actual Matlab model. Direct link to Matlab.	software test engineer			Use tool to easily setup Matlab model

2.3 General Improvement

As task WP6.11.5 is currently in the specification phase, no description of the developed solution is currently available apart from the above manual and use case requirements. This will be provided in the next iterations of this document.

TI NAME:					
TI_ID	CRYSTAL_TI_x	**Kind of TI**	T, M, MM, G	**Contact email**	TBD
Description: TBD					
Link to internal working documents:					

2.4 Integration and Interoperability

1.

TI NAME:					
TI_ID	CRYSTAL_TI_x	**Kind of TI**	T, M, MM, G	**Contact email**	TBD
Description: TBD					
Link to internal working documents:					

3 Crystal Brick BX_TBD: IBM Rational Solution for Systems and Software Engineering

IBM Rational Solution for Systems and Software Engineering

3.1 Description

Name:	IBM Rational Solution for Systems and Software Engineering
Contact:	Gray Bachelor, IBM UK, Arjen van de Wetering, IBM NL
Dependencies	Made up of individual IBM Rational products and practice libraries
License	Commercial license managed via IBM® Rational® License Key Server
Additional information	Generally Available

3.1.1 Introduction

The IBM Rational Solution for Systems and Software Engineering (SSE) is an integrated solution helping teams to specify, design, implement and validate complex products and the software that powers them. It offers an integrated set of capabilities to enable you to predictably deliver competitive, high-quality products while meeting regulatory and compliance requirements. The Rational Solution for SSE is based on open standards allowing IBM and third party tooling to be integrated with the solution as required. Open integration enables traceability across the lifecycle, flexible access to engineering information and collaboration between disciplines and across the lifecycle.

Its foundation components comprise:

IBM Rational DOORS and DOORS Next Generation for requirements management

IBM Rational Rhapsody and Design Manager for visual architecture and design

IBM Rational Quality Manager for automating testing and defect management

IBM Rational Team Concert for collaboration, workflow, change and configuration management

IBM Rational Engineering Lifecycle Manager for visualization, organization and management of cross-discipline engineering information.

The Rational Solution for SSE address the key needs of systems and software development teams, enabling them to:

- Adopt requirements-driven systems design with validation and verification at each step:
  - *Build the right product* because the requirements are visible at all times
  - *Build the product right* with structural and behavioural analysis and design
  - Link requirements information throughout the development lifecycle with links to the systems and software design, development and testing environments
  - Demonstrate traceability across multiple levels of requirements
- Perform testing throughout development:
  - Simulate often to prove correctness and assess alternatives
  - Automatically create and execute tests from the design model or target platform
  - Manage test cases as part of the system design

- Extend collaboration beyond basic configuration and change management, with:
  - Integrated work items
  - Real-time planning
  - Centralized design repository
  - Automated design reviews
  - Cross-discipline regulatory reporting
- Reuse existing designs and applications:
  - Intelligent reverse-engineering and analysis
- Design efficient embedded source code:
  - Generate C, C++, Java® and Ada applications
  - Refine the application by simultaneously working with the system design, software and target platform
- Address compliance with industry or government regulations or standards:
  - Industry-specific extensions and solutions for Aerospace/Defence, Automotive, Medical Devices

More information can be found on the IBM product information web site:

Overview of the IBM Rational Solution for Systems and Software Engineering

3.1.2 Installation

The Rational Solution for Systems and Software Engineering can be installed, updated and modified using the IBM® Installation Manager.

See the detailed notes on requirements, planning and installing.

3.1.3 Manual

Information on the Rational Solution for Systems and Software Engineering capabilities, products, help and other support and learning resources is located in the IBM Rational Information Center.

3.2 Use Case coverage and application

The use case needs are being assessed and described in D6.11.5. The intention is to summarise here as they become agreed with the use-case owners at subsequent milestones.

Use-case	Additional notes
2.2	
2.3	
2.8	
4.3	
4.45	

Please indicate the use case specific needs with regard to the brick. Use a dedicated paragraph for each use case for which the brick is allocated.

Useful notes:

2. See table “Technology Brick”, column “Use Case Specific Needs”

(https://projects.avl.com/11/0154/Lists/Technology_Bricks/AllItems.aspx)

3.3 General Improvement

The Technical Items are being identified from then use-case analysis and will be described in D6.11.5.

Description of the developed solution based on Technical Items allocated for this Brick.

Useful notes:

1. General improvements are concerning enhancements to a state of the art solution, but the integration and interoperability aspects.

TI NAME:					
TI_ID	CRYSTAL_TI_x	**Kind of TI**	T, M, MM, G	**Contact email**	TBD
Description: TBD					
Link to internal working documents:					

3.4 Integration and Interoperability

The IBM Rational Solution for Systems & Software Engineering supports integrations to IBM and third party products based on the Open Services for Lifecycle Integration (OSLC) specification. Non-OSLC based interfaces to third party products are described in the brick descriptions of individual products.

.

3.4.1 IOS: OSLC integrations of IBM products.

OSLC consumer, provider, and tracked resource set specification support is shown in the following table:

Product	Current version	CM	QM	RM	AM	AU	TRS
DOORS	9.5.2	C2	C2	CP2	C2		P1
DOORS NG	4.0.5	C2	C2	CP2	C2		P1
Rhapsody*	8.0.5	C2	C2	CP2	CP2		P1
Design Manager	4.0.5	C2	C2	CP2	CP2		P1
Rational Quality Manager	4.0.5	CP2	C2	CP2		CP2	P1
Rational Team Concert	4.0.5	CP2	C1	C2			P1
Rational Engineering Lifecycle Manager	4.0.5						P1

Key

C = Consumer

P = Provider

1 = Version 1.0

2 = Version 2.0

3.4.2 Other interfaces

4 Crystal BRICKB2_10: IBM Rational Rhapsody

IBM Rational Rhapsody

4.1 Description

Name:	Rational Rhapsody v8.0.5
Contact:	Gray Bachelor, IBM UK, Arjen van de Wetering, IBM NL
Dependencies	Consult the installation planning information through the link provided. Certain capabilities (e.g. OSLC integrations) are dependent on Rhapsody Design Manager
License	Commercial license managed via IBM® Rational® License Key Server
Additional information	Generally Available

Description of the Brick – please refer to existing and/or public resources if possible

4.1.1 Introduction

The IBM® Rational® Rhapsody® product line is a component of the IBM Rational Solution for Systems and Software Engineering providing a visual development environment for systems engineers and software developers to create real-time or embedded systems and software. It enables teams to collaborate to build architectures and designs using industry-standard languages, validate functionality early in the development life cycle, perform design reviews, and automate the delivery of innovative, high-quality products. Rational Rhapsody is available in a number of editions covering the needs of:

- Systems engineers working with graphical modelling languages, industry architecture frameworks and standards
- Software engineers working with UML and coding languages including C, C++, Java and Ada.

Depending on which edition you use, Rational Rhapsody provides the following features. For specific platform, edition, and language requirements and limitations, see the release notes for Rational Rhapsody.

- Support by all editions for modelling with UML, SysML, AUTOSAR, MARTE, DDS, DoDAF*, MODAF*, UPDM*, multicore, MISRA-C, MISRA-C++; and for the creation of custom profiles for the development of domain-specific languages (DSL).
- Sharing, linking, and reviewing design information with the extended team through the web through IBM Jazz-based integration with Rational Rhapsody Design Manager
- Profiles, settings, stereotypes, tags, and APIs with which you can extend and configure the product
- Requirements modelling and traceability features including impact, and coverage analysis capabilities with integration to leading requirements management tools, such as Rational DOORS®
- Static checking to ensure that the design is consistent
- A systems engineering toolkit for automating common systems engineering functions, including Rational Harmony for Systems Engineers
- Interfaces to key engineering tooling including:
    - XMI* (XML Metadata Interchange) and Rational Rose® importing
    - Interfaces to other modelling environments including MathWorks Simulink and other IBM tools.

- Configuration Management Interface
- Documentation tool interface

- Support for safety standard development: ISO 26262, IEC 61508, DO-178B, and DO-178C

More information can be found on the IBM product information web site:
Overview of Rational Rhapsody

4.1.2 Installation

Rational Rhapsody can be installed, updated and modified using the IBM® Installation Manager.
See the detailed Rational Rhapsody installation notes.

4.1.3 Manual

Detailed information on the IBM Rational Rhapsody product, key technical topics, learning and community, and support and troubleshooting is located in the IBM Rational Rhapsody Information Center

4.2 Use Case coverage and application

The use case needs are being assessed and described in D6.11.5. The intention is to summarise here as they become agreed with the use-case owners at subsequent milestones.

Use-case	Additional notes
2.2	
2.3	Adopted to support model-based systems engineering similar to HarmonySE methodology. Design Manager is envisaged to support IOS and allow to create links with other engineering artefacts such as requirements.
2.8	
4.3	The detailed use of IBM Rhapsody within UC4.3 will be defined after M9.
4.45	

Use Case specific needs which are addressed by this Brick. Please indicate the use case specific needs with regard to the brick. Use a dedicated paragraph for each use case for which the brick is allocated.
Useful notes:

3. See table “Technology Brick”, column “Use Case Specific Needs”

(https://projects.avl.com/11/0154/Lists/Technology_Bricks/AllItems.aspx)

4.3 General Improvement

The Technical Items are being identified from then use-case analysis and will be described in D6.11.5.

Description of the developed solution based on Technical Items allocated for this Brick.
Useful notes:

2. General improvements are concerning enhancements to a state of the art solution, but the integration and interoperability aspects.

TI NAME:					
TI_ID	CRYSTAL_TI_x	**Kind of TI**	T, M, MM, G	**Contact email**	TBD
Description: TBD					
Link to internal working documents:					

4.4 Integration and Interoperability

IBM Rational Rhapsody supports integrations to IBM and third party products based on the Open Services for Lifecycle Integration (OSLC) specification.

4.4.1 IOS: OSLC integrations of IBM products.

IBM Rational Rhapsody 8.0.5 (interfaces provided via IBM Rhapsody Design Manager)		
Interface	Consumer/Provider	Specification
Change Management	Consumer	2.0
Quality Management	Consumer	2.0
Requirements Management	Consumer/Provider	2.0
Architecture Management	Consumer/Provider	2.0
Tracked Resource Set	Provider	1.0

4.4.2 Other interfaces

4.4.2.1 MathWorks Simulink interface

IBM Rational Rhapsody allows you to integrate MathWorks Simulink models into Rational Rhapsody designs. MathWorks Simulink models are represented as "Simulink blocks" in the UML model, and these blocks can interact with Rational Rhapsody objects/parts or other Simulink blocks. See Integrating Rational Rhapsody and the MathWorks Simulink

4.4.2.2 IDE interfaces

Rational Rhapsody can be integrated with MS Visual Studio, Eclipse and Tornado IDEs. See Integrating Rational Rhapsody with IDEs

4.4.2.3 XMI interfaces

Rational Rhapsody provides XMI import and export capabilities. See Exchanging model data by using XMI.

4.4.2.4 Collaboration and configuration management interfaces

Rational Rhapsody can be integrated with IBM and third party collaboration tools including PVCS Dimensions, CVS and Subversion. See Integrating Rational Rhapsody with collaboration tools.

5 Crystal Brick B2-16: IBM Rational DOORS

5.1 Description

Name:	Rational DOORS 9.5.2
Contact:	Gray Bachelor, IBM UK, Arjen van de Wetering, IBM NL
Dependencies	Consult the installation planning information through the link provided.
License	Commercial license managed via IBM® Rational® License Key Server
Additional information	Generally Available

5.1.1 Introduction

Rational® DOORS® is a component of the IBM Rational Solution for Systems and Software Engineering that provides requirements management making it easy to capture, trace, analyse, and manage changes to information.

Rational DOORS provides numerous capabilities for teams within an organization and beyond to participate in and contribute to the requirements management process, including:

- Linking requirements to design items, test plans, test cases, and other requirements for easy and powerful traceability.
- Requirements change management with either a simple predefined change proposal system or a more thorough, customizable change control workflow through integration to Rational change management solutions.
- Linking requirements to test cases using the Test Tracking Toolkit for manual test environments.
- Web access to the requirements database through Rational DOORS Web Access.
- Requirements Interchange Format support, for exchanging requirements information with suppliers and development partners in the development process.
- Open Services for Lifecycle Collaboration (OSLC) specifications for requirements management, change management, and quality management to integrate with systems and software lifecycle tools.
- Integration with other Rational tools, including Rational Requirements Composer, Rational Rhapsody®, Rational Quality Manager, Rational Focal Point™, and Rational System Architect, and also many third-party tools, providing a comprehensive traceability solution.

More information can be found on the IBM product information web site:

Overview of Rational DOORS

5.1.2 Installation

Rational DOORS can be installed, updated and modified using the IBM® Installation Manager.

Detailed installation notes including deployment planning, installation, migration and license management are covered in the Rational DOORS and DOORS Web Access 9.5.2 Installation Guide

5.1.3 Manual

Detailed information on using Rational DOORS is available in the DOORS area of the IBM Rational Information Center.

This resource cover key topics including:

- Introduction and video tours
- Tutorials
- Managing and composing requirements
- Extending and Integrating
- Administering
- Troubleshooting

5.2 Use Case coverage and application

The use case needs are being assessed and described in D6.11.5. The intention is to summarise here as they become agreed with the use-case owners at subsequent milestones.

Use-case	Additional notes
2.2	
2.3	Adopted to support requirements management. IOS compliant interfaces should allow to retrieve requirements including attributes, store updated requirements and create links with other engineering artefacts
2.8	
4.1	Adopted to support requirements management. IOS compliant interfaces should allow to retrieve requirements including attributes, store updated requirements and create links with other engineering artefacts

Use Case specific needs which are addressed by this Brick. Please indicate the use case specific needs with regard to the brick. Use a dedicated paragraph for each use case for which the brick is allocated.

Useful notes:

4. See table “Technology Brick”, column “Use Case Specific Needs”

(https://projects.avl.com/11/0154/Lists/Technology_Bricks/AllItems.aspx)

5.3 General Improvement

The Technical Items are being identified from then use-case analysis and will be described in D6.11.5.

Description of the developed solution based on Technical Items allocated for this Brick.

Useful notes:

3. General improvements are concerning enhancements to a state of the art solution, but the integration and interoperability aspects.

TI NAME:					
TI_ID	CRYSTAL_TI_x	**Kind of TI**	T, M, MM, G	**Contact email**	TBD
Description:					

TBD	
Link to internal working documents:	

5.4 Integration and Interoperability

IBM Rational DOORS supports integrations to IBM and third party products based on the Open Services for Lifecycle Integration (OSLC) specification.

5.4.1 IOS: OSLC integrations of IBM products:

IBM Rational DOORS Next Generation 4.0.5		
Interface	**Consumer/Provider**	**Specification**
Change Management	Consumer	2.0
Quality Management	Consumer	2.0
Requirements Management	Consumer/Provider	2.0
Architecture Management	Consumer	2.0
Tracked Resource Set	Provider	1.0

5.4.2 Other interfaces

5.4.2.1 HP Quality Center

In additional to OSLC-based integrations, an integration is available for HP Quality Center and Rational DOORS.

5.4.2.2 REQIF interface

Rational DOORS has a ReqIF (Requirements Interchange Format) interface that allows import and export of Requirements

pic.dhe.ibm.com/infocenter/clmhelp/v4r0m5/index.jsp?topic=%2Fcom.ibm.rational.rrm.help.doc%2Ftopics%2Fr_reqif_format.html

6 Crystal Brick B2_16: IBM Rational DOORS Next Generation

6.1 Description

Name:	Rational DOORS Next Generation v4.0.5
Contact:	Gray Bachelor, IBM UK, Arjen van de Wetering, IBM NL
Dependencies	Consult the installation planning information through the link provided.
License	Commercial license managed via IBM® Rational® License Key Server
Additional information	Generally Available

6.1.1 Introduction

IBM® Rational® DOORS® Next Generation is a requirements management tool that leverages the technology of the IBM Rational Jazz™ platform. With IBM Rational DOORS Next Generation, you can define and manage requirements and traceability in systems and software engineering projects.

Rational DOORS Next Generation provides:

- Modules to facilitate the creation of structured documents containing individually managed requirements
- A shared type system to enable the definition and reuse of artefact types and their attributes
- Built-in dashboards to provide easy visibility and analysis of information across teams
- Common administration for users, servers and project administration via IBM Jazz Team Server.

Rational DOORS Next Generation can be adopted by teams not currently employing a requirements management solution, including:

- Teams currently using documents and spreadsheets for requirements who are seeking more effective requirements management.
- Teams seeking a less formal, lighter-weight requirements process than those typically implemented with Rational DOORS.
- Teams seeking a web-based requirements tool (with optional local rich client).
- Teams seeking a single-server solution for multi-disciplinary product development teams (requirements, design, development, and test).

Rational DOORS Next Generation can also be used teams already working with Rational DOORS 9.5.

- Jump-start new projects in Rational DOORS NG: use existing information in DOORS 9.5 using the requirements interchange format (ReqIF).
- Involve your supply chain: invite your suppliers to use DOORS NG for requirements collaboration. Merge their updates into your DOORS 9.5 repository.
- Reference existing information in DOORS 9.5: use OSLC links in module traceability columns.
- Generate documents: use Rational Publishing Engine to create document templates with DOORS 9.5 and DOORS NG data sources.

More information can be found on the IBM product information web site:

Overview of IBM Rational DOORS Next Generation

6.1.2 Installation

Rational DOORS Next Generation can be installed, updated and modified using the IBM® Installation Manager.

Detailed installation notes including deployment planning, installation, migration and license management are covered in the IBM Rational Information Center Rational Solution for Systems and Software Engineering Installation Guide

6.1.3 Manual

Information on Rational DOORS Next Generation product description and key capabilities including a comparison between IBM Rational DOORS and DOORS Next Generation is located in the IBM Rational Information Center. Product update information, product downloads and information about activities in the DOORS Next Generation development project are available from the jazz.net Rational DOORS Next Generation page. (jazz.net registration required to access some content.)

6.2 Use Case coverage and application

The use case needs are being assessed and described in D6.11.5. The intention is to summarise here as they become agreed with the use-case owners at subsequent milestones.

Use Case specific needs which are addressed by this Brick. Please indicate the use case specific needs with regard to the brick. Use a dedicated paragraph for each use case for which the brick is allocated.

Useful notes:

5. See table “Technology Brick”, column “Use Case Specific Needs”

(https://projects.avl.com/11/0154/Lists/Technology_Bricks/AllItems.aspx)

6.3 General Improvement

The Technical Items are being identified from then use-case analysis and will be described in D6.11.5.

Description of the developed solution based on Technical Items allocated for this Brick.

Useful notes:

4. General improvements are concerning enhancements to a state of the art solution, but the integration and interoperability aspects.

TI NAME:					
TI_ID	CRYSTAL_TI_x	**Kind of TI**	T, M, MM, G	**Contact email**	TBD
Description: TBD					
Link to internal working documents:					

6.4 Integration and Interoperability

IBM Rational DOORS Next Generation supports integrations to IBM and third party products based on the Open Services for Lifecycle Integration (OSLC) specification.

.

6.4.1 IOS: OSLC integrations of IBM products.

6.4.2

IBM Rational DOORS Next Generation 4.0.5		
Interface	**Consumer/Provider**	**Specification**
Change Management	Consumer	2.0
Quality Management	Consumer	2.0
Requirements Management	Consumer/Provider	2.0
Architecture Management	Consumer	2.0
Tracked Resource Set	Provider	1.0

6.4.3 Other interfaces

6.4.3.1 REQIF interface

Rational DOORS NG has a ReqIF (Requirements Interchange Format) interface that allows import and export of Requirements

pic.dhe.ibm.com/infocenter/clmhelp/v4r0m5/index.jsp?topic=%2Fcom.ibm.rational.rrm.help.doc%2Ftopics%2Fr_reqif_format.html

7 Crystal Brick B2_19: IBM Rational Team Concert

7.1 Description

Name:	Rational Team Concert v4.0.5
Contact:	Gray Bachelor, IBM UK, Arjen van de Wetering, IBM NL
Dependencies	Consult the installation planning information through the link provided.
License	Commercial license managed via IBM® Rational® License Key Server
Additional information	Generally Available

7.1.1 Introduction

To be addressed in a later release of this document.

7.1.2 Installation

Rational Team Concert can be installed, updated and modified using the IBM® Installation Manager.
See the detailed IBM Rational information Center.

7.1.3 Manual

Detailed information on the IBM Rational Team Concert product, key technical topics, learning and community, and support and troubleshooting is located in the IBM Rational Team Concert Information Center.

7.2 Use Case coverage and application

The use case needs are being assessed and described in D6.11.5. The intention is to summarise here as they become agreed with the use-case owners at subsequent milestones.

Use-case	Additional notes
2.2	Adopted by scenarios 1 and 3 to support functional design RELM solution is also envisaged to support traceability according to the same approach adopted by the Aerospace public use case.
2.3	Adopted as part of the RELM solution to support traceability, impact analysis, indexing and retrieval, reporting, and process automation.
2.8	
4.1	Adopted to support source control, change management and continuous integration aspects. Reporting capabilities may also be studied.

Use Case specific needs which are addressed by this Brick. Please indicate the use case specific needs with regard to the brick. Use a dedicated paragraph for each use case for which the brick is allocated.

Useful notes:

6. See table “Technology Brick”, column “Use Case Specific Needs”

(https://projects.avl.com/11/0154/Lists/Technology_Bricks/AllItems.aspx)

7.3 General Improvement

The Technical Items are being identified from then use-case analysis and will be described in D6.11.5.

Description of the developed solution based on Technical Items allocated for this Brick.

Useful notes:

5. General improvements are concerning enhancements to a state of the art solution, but the integration and interoperability aspects.

TI NAME:					
TI_ID	CRYSTAL_TI_x	**Kind of TI**	T, M, MM, G	**Contact email**	TBD
Description: TBD					
Link to internal working documents:					

7.4 Integration and Interoperability

8 CRYSTAL BRICK BX_TBD: IBM Rational Design Manager

8.1 Description

Name:	IBM Rational Rhapsody Design Manager 4.0.5
Contact:	Gray Bachelor, IBM UK, Arjen van de Wetering, IBM NL
Dependencies	Consult the installation planning information through the link provided.
License	Commercial license managed via IBM® Rational® License Key Server
Additional information	Generally Available

8.1.1 Introduction

IBM® Rational® Rhapsody® Design Manager is a collaborative web-based tool that enables a broad set of stakeholders to contribute to and influence the design of products, software, and systems. By using the Design Management products, you can integrate design into the overall application and systems engineering lifecycles and collaborate on models and designs.

The design management capabilities in Rational Rhapsody Design Manager support integrations with the Jazz™ applications for change and configuration management, requirements management, and quality management.

The key capabilities of Rational Rhapsody Design Manager include:

In-context collaboration

You can share models created in Rational Software Architect or Rational Rhapsody, edit the models by using the web client, and conduct online design reviews with peers and stakeholders.

Traceability and impact analysis

When you integrate Design Management with SSE, you can set up links between design elements, other designs, and lifecycle artefacts (architecture management, requirements management, change management, and quality management.) The bidirectional traceability between requirements and designs enables you to easily see and understand the impact of changes to the requirements or designs. You can also create and view impact analysis diagrams to examine the links to and from specific target elements in a design

Agile sketching

In Design Management, you can use the collaborative, agile sketching functionality to create sketches of architecture and design ideas. You can then share the sketches with colleagues and stakeholders, conduct reviews of the ideas, and revise the sketches before proceeding with formal modelling of the design.

Reporting and document generation

You can use the embedded document generation features and document templates to generate and share design documents, specifications, and reports. By using the Design Management web client to generate reports, project members and stakeholders can easily access information in the models and projects.

Change management for designs

You can store and manage different types of architectures, designs, and models in the SSE applications.

Traditional design tools store designs in files that are managed by source control management (SCM) systems. In Design Management, designs and models are treated as first-class artefacts, which means that you can import models to the server and manage them directly at the model level; no mapping of model elements to files is required. This approach simplifies the workflow for successfully managing architectures and designs in a team context.

More information can be found on the IBM product information web site:

Overview of Rational Rhapsody Design Manager

8.1.2 Installation

Rational Rhapsody Design Manager can be installed, updated and modified using the IBM® Installation Manager.
See the detailed IBM Rational information Center article on Installing and Configuring Design Management.

8.1.3 Manual

Detailed information on the IBM Rational Rhapsody Design Manager product, key technical topics, learning and community, and support and troubleshooting is located in the IBM Rational Design Management Information Center.

8.2 Use Case coverage and application

The use case needs are being assessed and described in D6.11.5. The intention is to summarise here as they become agreed with the use-case owners at subsequent milestones.

Use-case	Additional notes
2.2	
2.3	Adopted to support model-based systems engineering similar to HarmonySE methodology. Design Manager is envisaged to support IOS and allow to create links with other engineering artefacts such as requirements.
2.8	
4.3	The detailed use of IBM Rhapsody within UC4.3 will be defined after M9.
4.45	

Use Case specific needs which are addressed by this Brick. Please indicate the use case specific needs with regard to the brick. Use a dedicated paragraph for each use case for which the brick is allocated.
Useful notes:

7. See table “Technology Brick”, column “Use Case Specific Needs”

(https://projects.avl.com/11/0154/Lists/Technology_Bricks/AllItems.aspx)

8.3 General Improvement

The Technical Items are being identified from then use-case analysis and will be described in D6.11.5.

Description of the developed solution based on Technical Items allocated for this Brick.
Useful notes:

6. General improvements are concerning enhancements to a state of the art solution, but the integration and interoperability aspects.

TI NAME:					
TI_ID	CRYSTAL_TI_x	**Kind of TI**	T, M, MM, G	**Contact email**	TBD
Description: TBD					
Link to internal working documents:					

8.4 Integration and Interoperability

IBM Rational Rhapsody Design Manager supports integrations to IBM and third party products based on the Open Services for Lifecycle Integration (OSLC) specification.

There are several advantages to integrating Design Management capabilities with other Rational products:

- Traceability between designs, requirements, test artefacts, and work items in lifecycle management projects.
- Ability to plan and deliver solutions with Open Services for Lifecycle Collaboration - Change Management (OSLC-CM) providers such as the change and the configuration management and the requirements management applications in the Rational solution for CLM.
- Ability to easily view and understand how potential changes to designs or requirements can impact projects.

8.4.1 IOS: OSLC integrations of IBM products:

IBM Rational Rhapsody Design Manager 4.0.5		
Interface	Consumer/Provider	Specification
Change Management	Consumer	2.0
Quality Management	Consumer	2.0
Requirements Management	Consumer/Provider	2.0
Architecture Management	Consumer/Provider	2.0
Tracked Resource Set	Provider	1.0

8.4.2 Other interfaces

8.4.2.1 Simulink

Rational Design Manager provides the ability to import from or integrate with MathWorks Simulink

http://pic.dhe.ibm.com/infocenter/rdmhelp/v3/index.jsp?topic=%2Fcom.ibm.dm.rhp.simulink.doc%2Ftopics%2Fc_rhp_dm_simulink_overview.html

9 CRYSTAL BRICK BX_TBD: IBM Rational Engineering Lifecycle Manager

9.1 Description

Name:	IBM Rational Engineering Lifecycle Manager v4.0.5
Contact:	Gray Bachelor, IBM UK, Arjen van de Wetering, IBM NL
Dependencies	Consult the installation planning information through the link provided.
License	Commercial license managed via IBM® Rational® License Key Server
Additional information	Generally Available

9.1.1 Introduction

Rational Engineering Lifecycle Manager is a capability of the Rational solution for systems and software engineering. The tool indexes product engineering data created and managed in Rational DOORS® or DOORS Next Generation, Rational Rhapsody® Design Manager, Rational Quality Manager, and Rational Team Concert™. When MathWorks Simulink artefacts are used with the Rational Rhapsody Design Manager, Simulink artefacts can participate in the index as well.

Engineering data created by home-grown tools and other vendors have the potential to be indexed. Light-weight connectors are needed for this purpose. In the time frame of this initial release of Rational Engineering Lifecycle Manager, it is recommended that customers engage professional services from IBM® or select IBM Business Partners to create these connectors.

A web-based system provides a portal into the indexed data, typically by using customized, predefined views to display the latest information from the index.

Rational Engineering Lifecycle Manager enhances the Rational solution for systems and software engineering by providing the capability to visualize, analyse, and organize systems product engineering data coming from many tools. Engineering teams get a better understanding of the relationships in that data, so they can make more effective and timely engineering decisions.

Rational Engineering Lifecycle Manager uses Rational Jazz™ integration services to increase the visibility and value of engineering data in the following ways:

- Get faster, more complete answers to key engineering questions across the lifecycle. Search and report on data across engineering disciplines, regardless of data source and location.
- Visualize and analyse engineering data and relationships in the context of user roles, product structure, development process, and product release.
- Use views to navigate to specific engineering artefacts needed for a particular task, and open the artefacts in the tool where the artefact was created.
- Analyse the impact of changes with visual and report-based representations of engineering artefacts and their relationships.
- Improve reuse of engineering artefacts and the components they represent.

More information can be found on the IBM product information web site:

Overview of Rational Engineering Lifecycle Manager

9.1.2 Installation

Rational Engineering Lifecycle Manager can be installed, updated and modified using the IBM® Installation Manager.

Detailed installation notes including deployment planning, installation, configuration and license management are covered in Installing Rational Engineering Lifecycle Manager in the IBM Rational Information Center.

9.1.3 Manual

Information on using Rational Engineering Lifecycle Manager can be found in the IBM Rational Information Center

The key topics covered include:

- Getting started
- Key terminology and concepts
- Working with products and managing product hierarchies
- Designing queries and views
- Analysing the impact of change
- Building reports
- What's new in the 4.0.5 release

9.2 Use Case coverage and application

The use case needs are being assessed and described in D6.11.5. The intention is to summarise here as they become agreed with the use-case owners at subsequent milestones.

Use-case	Additional notes
2.2	
2.3	
2.8	
4.3	
4.45	

Use Case specific needs which are addressed by this Brick. Please indicate the use case specific needs with regard to the brick. Use a dedicated paragraph for each use case for which the brick is allocated.

Useful notes:

8. See table "Technology Brick", column "Use Case Specific Needs"

(https://projects.avl.com/11/0154/Lists/Technology_Bricks/AllItems.aspx)

9.3 General Improvement

The use case needs are being assessed and described in D6.11.5. The intention is to summarise here as they become agreed with the use-case owners at subsequent milestones.

Description of the developed solution based on Technical Items allocated for this Brick.

Useful notes:

7. General improvements are concerning enhancements to a state of the art solution, but the integration and interoperability aspects.

TI NAME:					
TI_ID	CRYSTAL_TI_x	**Kind of TI**	T, M, MM, G	**Contact email**	TBD
Description: TBD					
Link to internal working documents:					

9.4 Integration and Interoperability

9.4.1 IOS: OSLC integrations of IBM products

Rational Engineering Lifecycle Manager implements the OSLC Tracked Resources Set 1.0 specification which enables it to index data from any linked data source such as the IBM Rational products that make up the IBM Rational Solution for Systems and Software Engineering, including Rational DOORS, Rational DOORS Next Generation, Rational Rhapsody, Rational Quality Manager and Rational Team Concert.

The concepts of linked data and OSLC integrations are described in more detail in the IBM Rational Information Center document Open Services for Lifecycle Collaboration integrations.

OSLC specifications and other resources are available at open-services.net

9.4.2 Other interfaces

9.4.2.1 Product tree import

Rational Engineering Lifecycle Manager provides a means to import product structure trees defined in an xml file or via a RESTful service.

http://www-01.ibm.com/support/docview.wss?uid=swg21637443

10 Terms, Abbreviations and Definitions

Please add additional terms, abbreviations and definitions for your deliverable.

CRYSTAL	**CR**itical S**YST**em Engineering **A**cce**L**eration
R	Report
P	Prototype
D	Demonstrator
O	Other
PU	Public
PP	Restricted to other program participants (including the JU).
RE	Restricted to a group specified by the consortium (including the JU).
CO	Confidential, only for members of the consortium (including the JU).
WP	Work Package
SP	Subproject
T	
M	
MM	
G	

Table 10-1: Terms, Abbreviations and Definitions

11 References

Please add citations in this section.

[Author, Year]	Authors; *Title*; Publication data (document reference)

12 Annex

12.1 Expectations from UC4.3 to Brick 4.9a (Lifecycle management of simulation models)

Use Case 4.3 has a need to include model-based testing of software in the development process. To test the developed software before implementation on a real system, the hardware behaviour needs to be simulated by a dynamic model. These simulations are performed on different levels, e.g.: sub-module testing or high-level system testing. Furthermore, a software developer working on a specific issue might need to simulate a specific part of the hardware, while system validation will be done with a much more higher-level model containing less detail. It is therefore expected that there is no model which fits all these purposes. This means a large amount of different hardware models will be needed. Furthermore the real hardware consists of different modules which all can have different versions, so the models will also have a modular structure. To reduce the amount of modelling effort needed, the hardware models should consist of reusable building blocks. This large amount of models, sub-models and building blocks should remain manageable and traceable. This summarizes the need for lifecycle management tools from Use Case 4.3.

Model evolution

Typically, the models are adapted and modified within one project and then reused in the next project. In another project they can be reconfigured for certain test cases and for hardware variants. This requires an extensive version management of simulation models and their building blocks. Furthermore, the validation quality of the models need to be characterized and traced. Test results on implemented systems may result in improved models (for example using data assimilation techniques). This results in a wealth of model versions and requires strict model lifecycle management.

The sub-systems of a simulation model can be modelled at different levels of detail. The simplest models handle one key performance parameter such as processing time, whereas more advanced models open more and more black boxes up to e.g. full force finite element simulations. Typically, the model level variants are developed in the course of several development cycles and keep being used in parallel. While a software developer uses a part of the hardware model in very high detail to solve a specific issue, the verification team is running automatic tests using a complete model of the hardware with less detail. Models may be simplified using model order reduction techniques to keep simulation time manageable. Depending on simulation goal, some system functions may be simulated at advanced level whereas others are simulated at aggregate level. Software lifecycle management in this case handles the model level variants, their versions and the traceability of the models applied for certain simulation results.

Model configuration

Another important aspect from UC4.3 is that the simulation models are not only used by the people who developed these models. The *software tester* should also be able to use the hardware models without extensive knowledge of these models. Since there will be a large amount of different models and sub models, a software tester needs a tool to help him to select the right models to use. The selection of a proper model can depend on:

- Simulation purpose, e.g.: real-time simulation, high-level functional behaviour

- Machine configuration
- Automatic or manual testing
- Fault injection: replay of specific situations, or manually inject faults

By choosing the machine configuration, only a sub-set of the available model blocks can be used. Depending on the simulation purpose, sub-systems require a certain amount of detail. With this information, a tool will need to combine the proper model parts into a working simulation model (this can involve automatic code compilation).

Once a selection for a certain model is made, a tool should help the software tester to select the proper tests to be done. Furthermore a selection must be made of which results the user want to check. This can either be a manual check or an automatically derived test report. Finally. the tests and results should be linked to the requirements (link to Brick 4.06).