What’s new in Rational Rhapsody

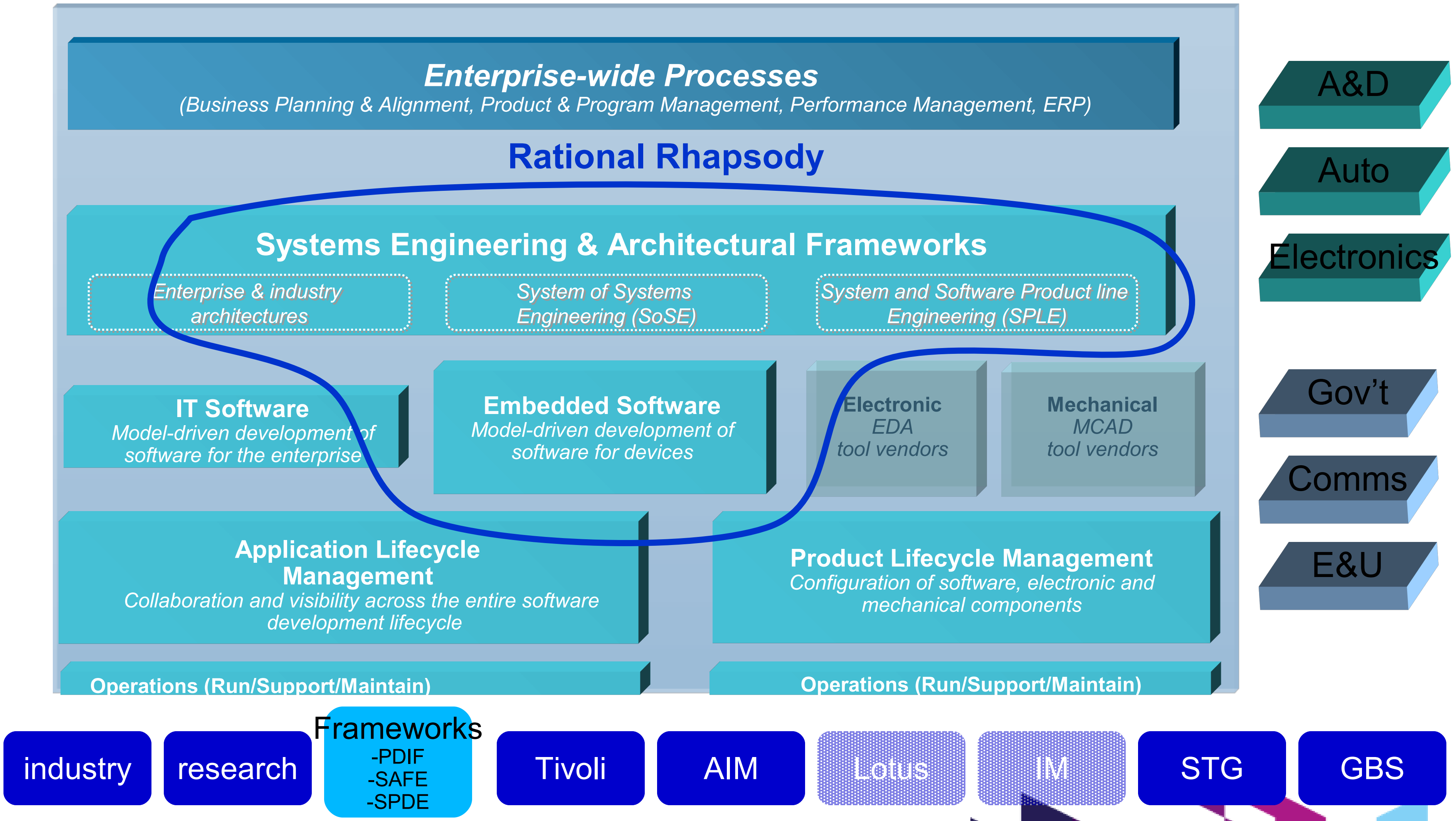
Andy Lapping
Steve Rooks
IBM Rational Technical Specialists

DISCLAIMER

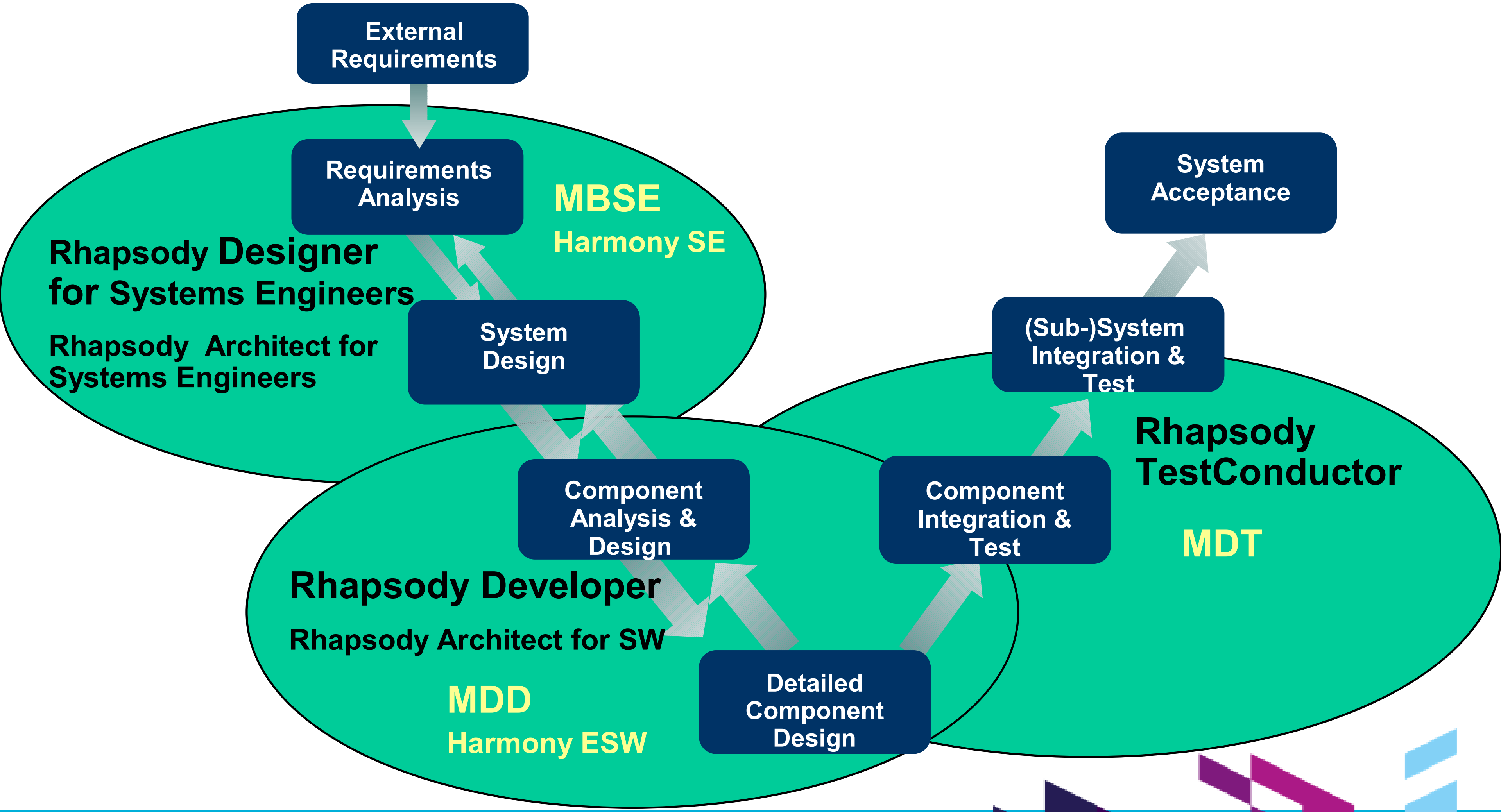
- ***IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.***

Rational Vision for Systems and Software

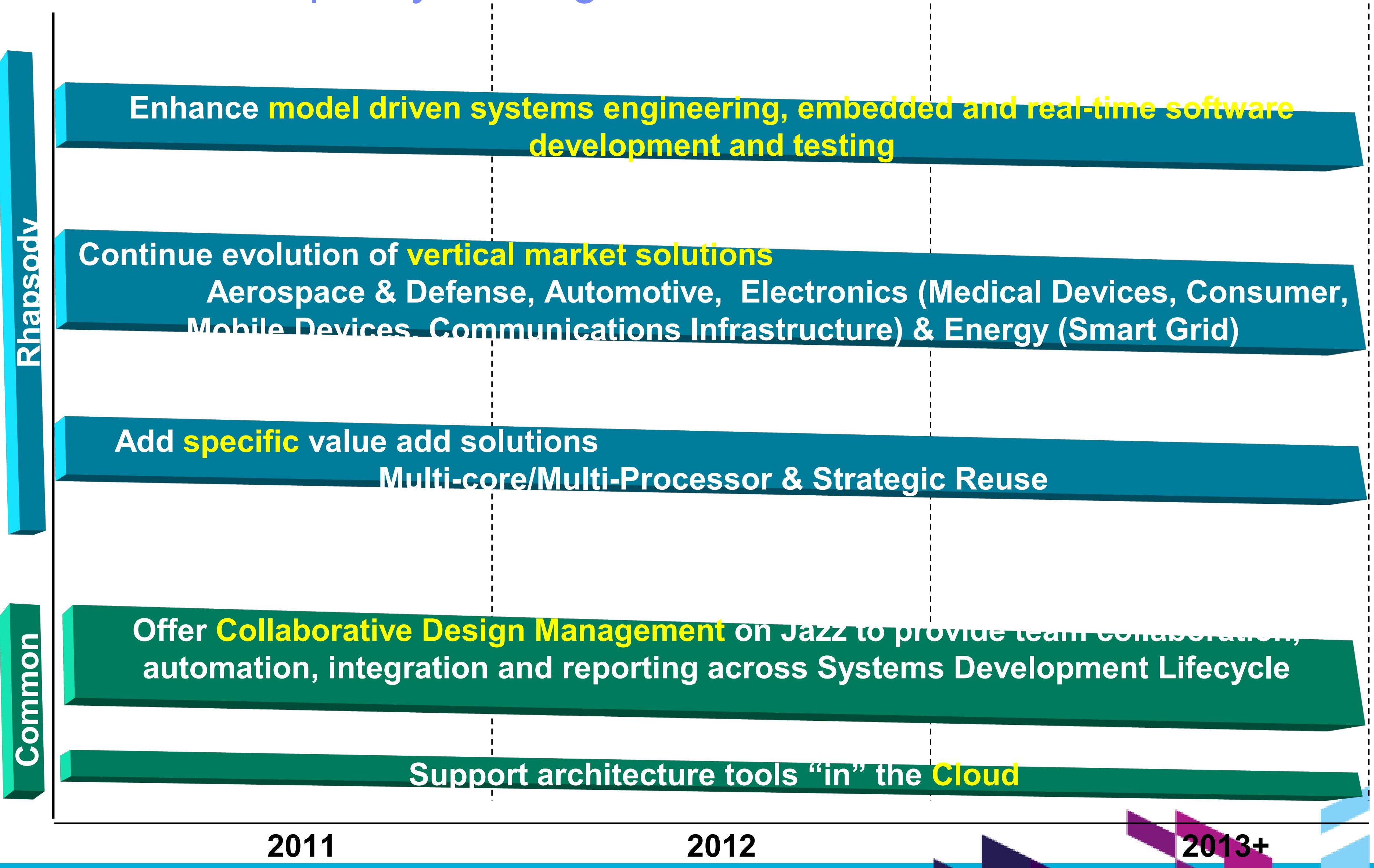
Rhapsody for making Smarter Products that enable a Smarter Planet


Enterprise-wide Processes
(Business Planning & Alignment, Product & Program Management, Performance Management, ERP)
Rational Rhapsody
Systems Engineering & Architectural Frameworks
Enterprise & industry architectures
System of Systems Engineering (SoSE)
System and Software Product line Engineering (SPLE)
IT Software
Model-driven development of software for the enterprise
Embedded Software
Model-driven development of software for devices
Electronic EDA tool vendors
Mechanical MCAD tool vendors
Application Lifecycle Management
Collaboration and visibility across the entire software development lifecycle
Product Lifecycle Management
Configuration of software, electronic and mechanical components
Operations (Run/Support/Maintain)
Operations (Run/Support/Maintain)
A&D
Auto
Electronics
Gov't
Comms
E&U
industry
research
Frameworks
-PDIF
-SAFE
-SPDE
Tivoli
AIM
Lotus
IM
STG
GBS

Rhapsody and the systems "V" lifecycle


External Requirements
Requirements Analysis
MBSE
Harmony SE
Rhapsody Designer for Systems Engineers
Rhapsody Architect for Systems Engineers
System Design
Component Analysis & Design
Rhapsody Developer
Rhapsody Architect for SW
MDD
Harmony ESW
Detailed Component Design
Component Integration & Test
(Sub-)System Integration & Test
System Acceptance
Rhapsody TestConductor
MDT

Rational Rhapsody Strategic Themes



Enhance **model driven systems engineering, embedded and real-time software development and testing**

Continue evolution of **vertical market solutions**
Aerospace & Defense, Automotive, Electronics (Medical Devices, Consumer, Mobile Devices, Communications Infrastructure) & Energy (Smart Grid)

Add **specific** value add solutions
Multi-core/Multi-Processor & Strategic Reuse

Offer **Collaborative Design Management** on Jazz to provide team collaboration, automation, integration and reporting across Systems Development Lifecycle

Support architecture tools “in” the **Cloud**

Strategic Themes for Rhapsody/MDSD (1/3)

- Lifecycle integration supports complete end-to-end traceability and support integrated solution oriented work-flows
  - Jazz enablement: integration into Jazz Foundation Server (JFS) and Jazz Integration Architecture (JIA)
  - Realize ALM scenarios for systems development enabling seamless product and data integrations between DOORS, RTC, RQM, etc. via OSLC
  - View, comment and mark-up of models, review and approval process
  - Web access to all modeling data
- Make Systems Modeling more accessible to Systems Engineers
  - Foster Systems Model as a multi-discipline architectural framework
    - Leverage Jazz to utilize System Model as the hub of engineering data
    - Refined process coverage from Systems thru Software, Electrical and Mechanical
    - Hardware/Software co-design
  - Support evolution of systems DSLs – e.g. SysML, UPDM (xDAF)
  - Utilizing the UML Action Language for model simulation

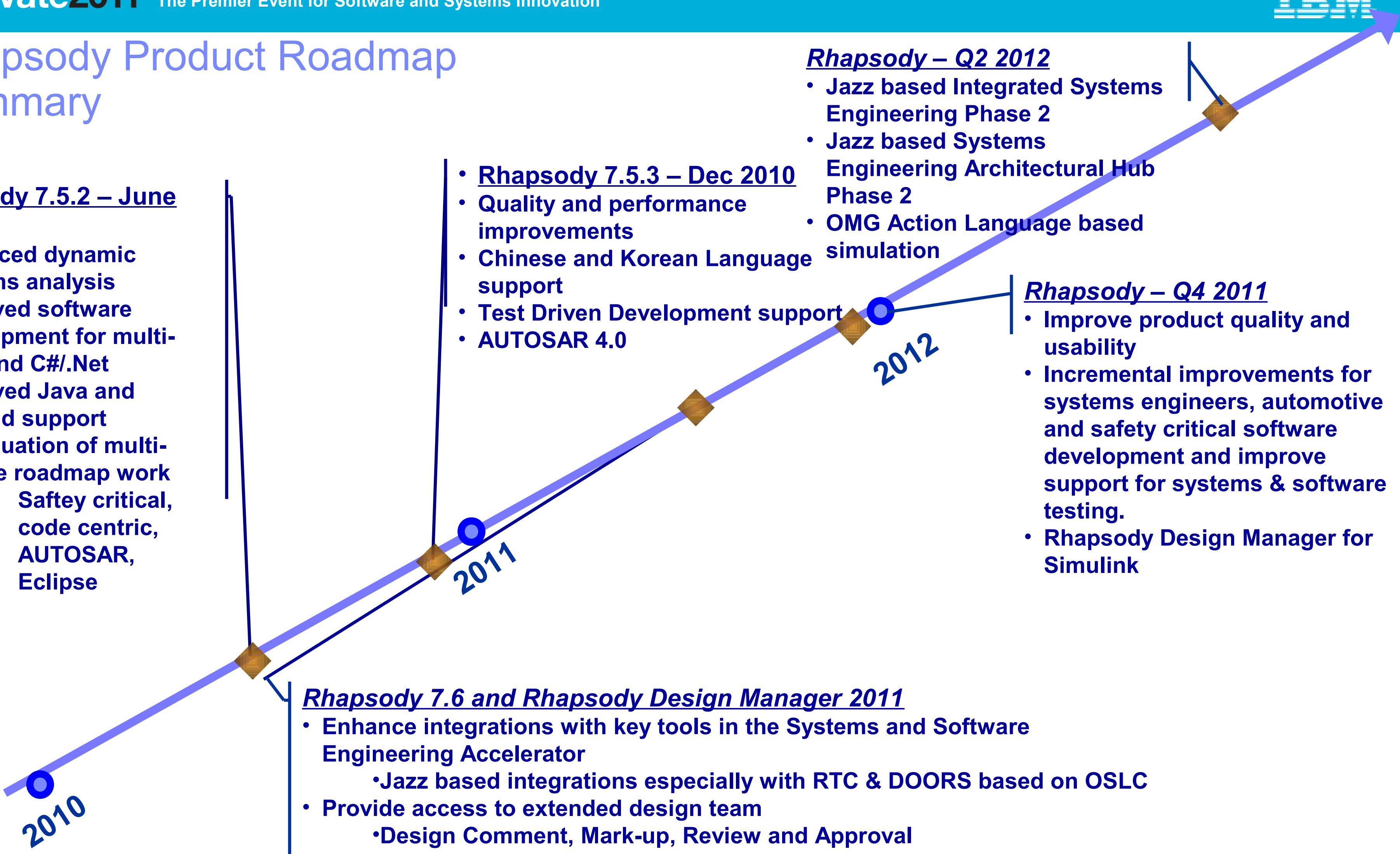
Strategic Themes for Rhapsody/MDSD (2/3)

- System design space optimization
  - Facilitate architectural trade studies
  - Enhanced integration with external simulation tools (e.g. Simulink, Modelica)
  - Multi-core and performance simulation (leveraging MARTE and discrete event resources)
- Enhanced value for software developers
  - Enhanced support for safety critical development
    - Coding standards (e.g. MISRA C/C++)
    - Enabling qualification/certification (ISO 26262, DO-178B, IEC 61508, etc)
  - Improved design for multi-core/multi-processor systems
  - More efficient support for constrained real time systems: footprint, performance
  - Advanced legacy code modernization – architectural mining
- Extend and leverage Model Driven Testing & Verification
  - Expand areas of Model Driven Test coverage enabling Test Driven Development (systems engineering, code-centric, Java, Ada, code coverage, etc)
  - Model checking: Leverage technology and practices from HW design domain

Strategic Themes for Rhapsody/MDSD (3/3)

- Driving greater productivity through strategic asset reuse
  - Modeling focus on variant management and Software Product Line Engineering (including third party integration)
  - Integration with reusable asset repository
- Continued investment in Vertical/Domain solutions
  - Aerospace & Defense: IMA, Safety Critical
  - Automotive: AUTOSAR, 26262, efficient microcontoller code generation
  - Electronics (Communications, Consumer, Industrial, Medical): Android
- Enhance productivity and Usability
  - Improved out of the box experience
  - Greater success of the first project
  - Support for model metrics
  - Ease and quality of graphical modeling
  - Innovative collaboration
  - User assistance
  - Performance

Rhapsody Product Roadmap Summary



Rhapsody 7.5.2 – June 2010
- Enhanced dynamic systems analysis
- Improved software development for multi-core and C#/.Net
- Improved Java and Android support
- Continuation of multi-release roadmap work
  - Saftey critical, code centric, AUTOSAR, Eclipse

- **Rhapsody 7.5.3 – Dec 2010**
- Quality and performance improvements
- Chinese and Korean Language support
- Test Driven Development support
- AUTOSAR 4.0

Rhapsody – Q2 2012
- Jazz based Integrated Systems Engineering Phase 2
- Jazz based Systems Engineering Architectural Hub Phase 2
- OMG Action Language based simulation

Rhapsody – Q4 2011
- Improve product quality and usability
- Incremental improvements for systems engineers, automotive and safety critical software development and improve support for systems & software testing.
- Rhapsody Design Manager for Simulink

Rhapsody 7.6 and Rhapsody Design Manager 2011
- Enhance integrations with key tools in the Systems and Software Engineering Accelerator
  - •Jazz based integrations especially with RTC & DOORS based on OSLC
- Provide access to extended design team
  - •Design Comment, Mark-up, Review and Approval
  - •Web based Design Review
- Improve product quality and usability
- Incremental improvements for systems engineers, safety critical software development and improve support for systems & software testing.

Rhapsody 7.6

Rhapsody 7.6 highlights

- Integration with Design Management
  - "modeling on Jazz" phase1
  - Publish Rhapsody models into a Jazz repository
    - Search information across projects in the repository
  - Create OSLC based links between Rhapsody elements and elements from other tools
  - Design review and markup capability
  - Web reviewing and markup of models
- Systems Engineering
  - Enhanced Activity Diagram execution
  - Simulation: Plant Model Integration (Simulink)
  - Support for systems Viewpoints
  - Enhancements for trade studies
  - RPE templates for SDD (Software Design Document)
- Software Development
  - Enhancements for safety critical
    - Simplified Framework (C and C++)
    - Reduction in MISRA C++ violations
  - Code customization enhancements
  - Ports optimization

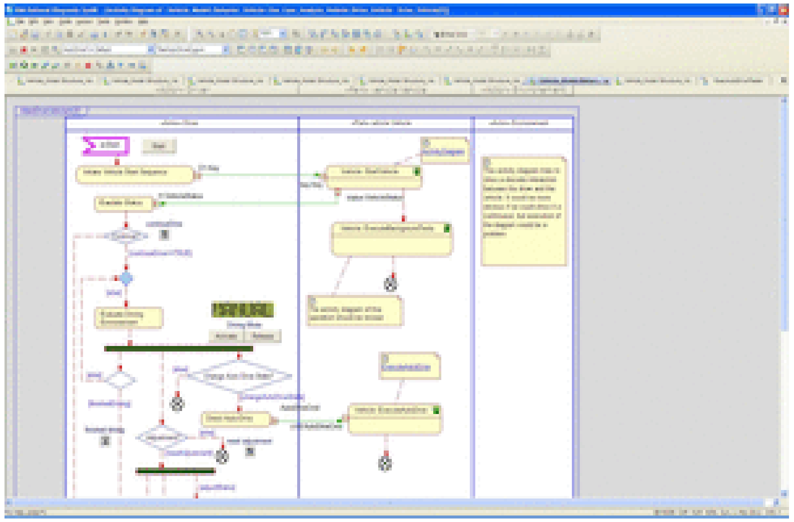
7.6 Highlights - continued

- Systems and Software Testing
  - Support for timing and Sequence Diagram operators in TestConductor
  - Provide out of the box code coverage analysis for C in TestCondutor
- General Usability
  - New diagrams look & feel
  - Action Intellisense
  - Performance: Dynamic Loading
  - Properties multi-select
  - Menu customization
- Integrations
  - Eclipse integration Enhancements
  - Gateway enhancements
- A&D
  - ARINC 653 adaptor
- Automotive
  - MicroC code generation enhancements
  - Behavioral code generation enhancements for AUTOSAR
  - AUTOSAR XML enhancements

Systems Engineering Enhancements

- Activity Diagram Simulation enhancements
- Plant Modelling integration via Simulink
- Viewpoints for visual filtering
- RPE templates for SDD
- Usability/Workflow enhancements
  - Functional Decomposition
  - Harmony
  - UI simplification
- Improved trade study capabilities using instance specification

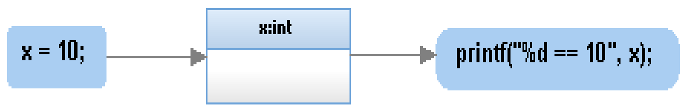
Enhanced support for Activity Diagram Execution



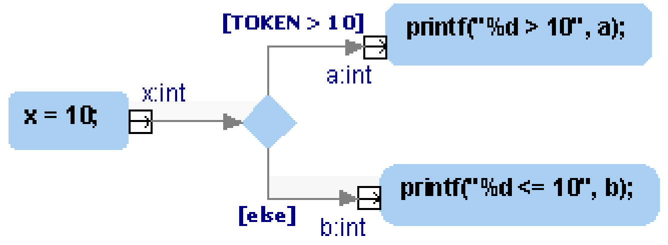
- Control and object flow execution support
  - In line with UML 2 / SysML token based semantics
  - Support control tokens and data tokens
- Can now also simulate behaviors associated with:
  - Actors, Use Cases, Operations
- Activity diagrams can call each other by
  - call operation and call behavior actions
- Activities of operations
  - Parameters correspond to arguments
  - A parameters named RETURN represents the return value
- Support for time events
- New simulation features:
  - New command: Go Action (Go until tokens are ready for an action to start)
  - Automatically open activity diagrams during execution when actions are ready (optional)
- Focus on Systems Engineering usage
  - Requires cygwin or Microsoft C++ compilers

Activity Diagram Simulation enhancements

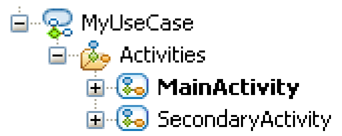
- Object Node
  - Object Nodes serve as data specification of Object Flows
  - Object Nodes can be accessed directly on the connected Actions
- Routing of data tokens
  - Data tokens can be routed by Control Nodes Decision, Merge, Join or Fork
  - The token is accessible by the Control Node using the keyword TOKEN
- Activity under Use-case
  - An Activity can be modelled under Use-case
  - Such Activities can be executed and simulated just as Activity under Class
- Activity under Operation
  - An Activity can be modelled under an Operation
  - Invoking the operation will result in executing its Activity
  - The Activity execution can be visualized



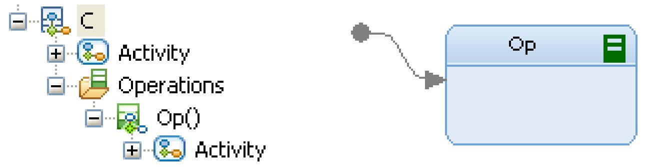
Object Node "x" is accessed directly on the Actions



DecisionNode route "int" token by testing TOKEN



"MyUseCase" own multiple Activities



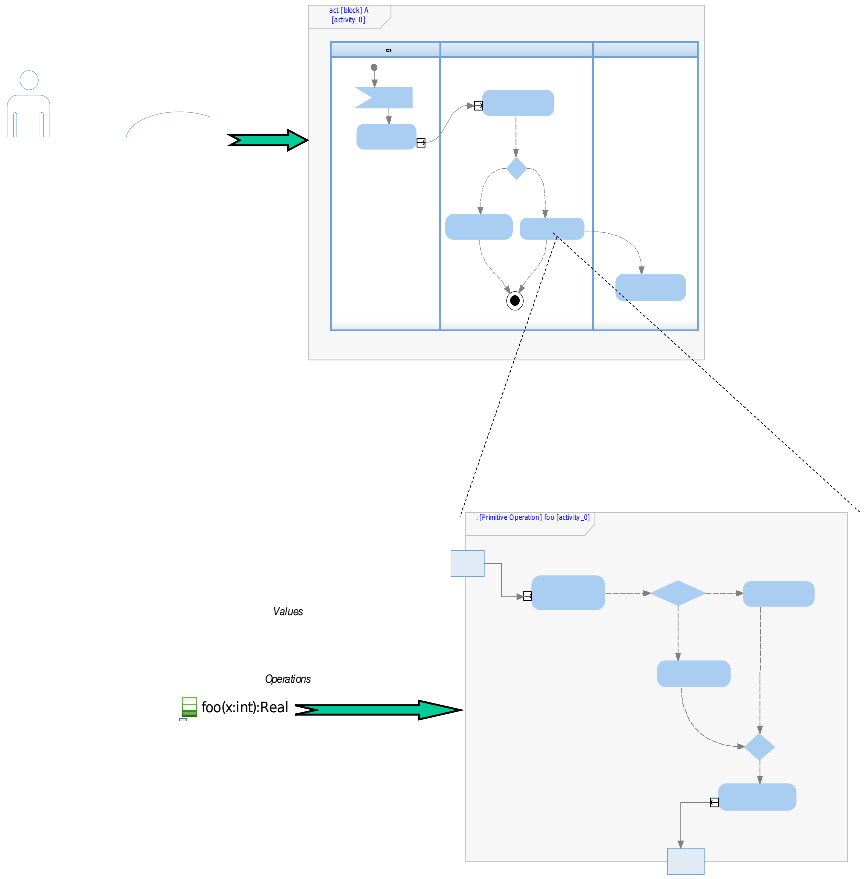
Invoking "Op" will result in executing its Activity

Activity Diagram Simulation enhancements

- Support arguments/return for “Activity under Operation”
  - Pins will be mapped to Operation arguments and return value
  - Auto-synch. mechanism for quick and safe mapping
  - Check for forcing consistency
- Support TimeEvent
  - User will be able to model and simulate TimeEvent
- Enable data passage on Events
  - Modeller will be able to specify Pins for Send/Accept Event Actions
  - Auto-synch. mechanism for quick and safe mapping
  - Check for forcing consistency
- Improve user experience and robustness
  - “Go Action” – Step between Actions
  - Ports-Activities integration issues
  - Other general modeling and execution issues

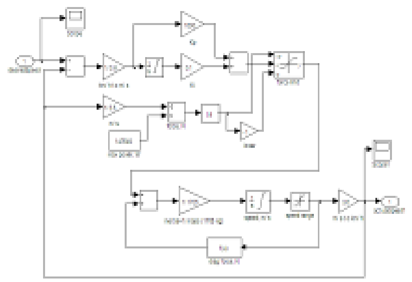
Functional Decomposition Workflow Enhancements

- Execute Use Cases with operations and activities
- Refinement of “atomic” actions to Call Operation Action and Call Behavior Action by bi-directional conversion
- Synchronizing parameters
  - Operation Arguments<->Activity Parameters/Action Pins
  - Activity Parameters<->Action Pins for call behavior actions
  - Event Parameters => Accept/Send Event Action Pins
- Navigation:
  - Call Operation Action => Activity of the operation

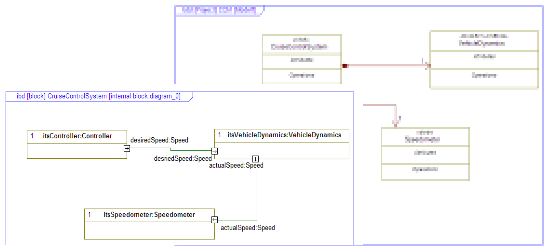

act [block] A [activity_0]
:[Primitive Operation] foo [activity_0]
Values
Operations
foo(x:int):Real

Plant Modeling integration via Simulink

- Enables the simulation of control blocks within the overall UML/SysML model

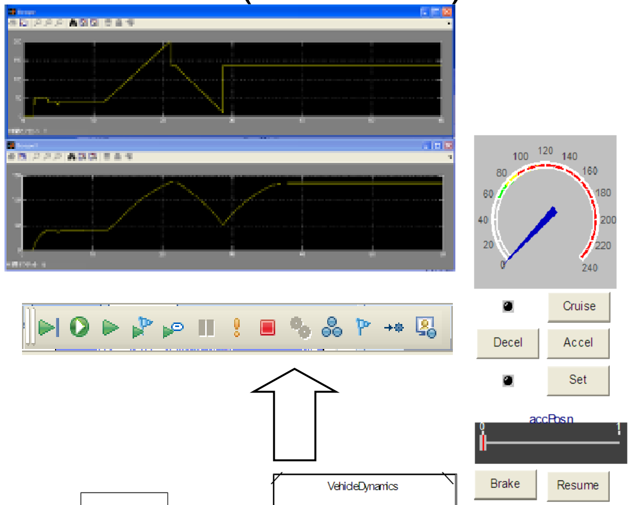


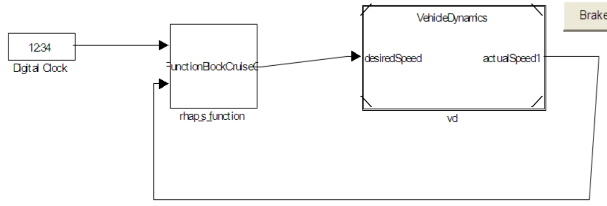
Behaviors in Simulink


ibd [block] CruiseControlSystem [internal block diagram_0]
1 itsController:Controller
desiredSpeed:Speed
1 itsVehicleDynamics:VehicleDynamics
actualSpeed:Speed
1 itsSpeedometer:Speedometer
actualSpeed:Speed

Architecture in Rhapsody (control and plant blocks)

Simulation (both tools)

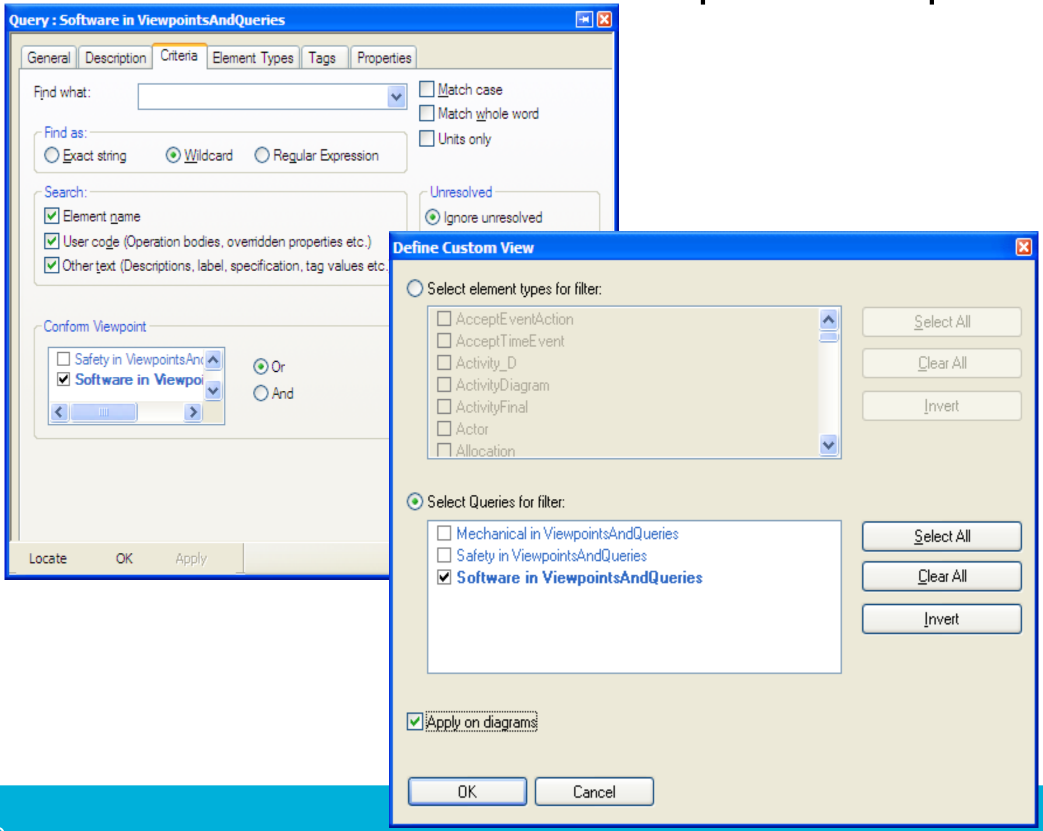




Simulink model for co-simulation

Viewpoints for visual filtering (SysML specific)

- Allows displaying different views of the model, for example: Mechanical, Safety
- Can be applied to the browser and/or to diagrams
- Conforms to the notion of Viewpoints as specified in SysML



Query : Software in ViewpointsAndQueries

Tabs: General | Description | Criteria | Element Types | Tags | Properties

Find what: (empty)

- [ ] Match case
- [ ] Match whole word
- [ ] Units only

Find as: Exact string [✓ Wildcard] Regular Expression

Search:
- [x] Element name
- [x] User code (Operation bodies, overridden properties etc.)
- [x] Other text (Descriptions, label, specification, tag values etc.

Unresolved: [✓ Ignore unresolved]

Conform Viewpoint:
- [ ] Safety in ViewpointsAnc
- [x] Software in Viewpoi

[✓ Or] And

Locate | OK | Apply

Define Custom View

Select element types for filter:
- [ ] AcceptEventAction
- [ ] AcceptTimeEvent
- [ ] Activity_D
- [ ] ActivityDiagram
- [ ] ActivityFinal
- [ ] Actor
- [ ] Allocation

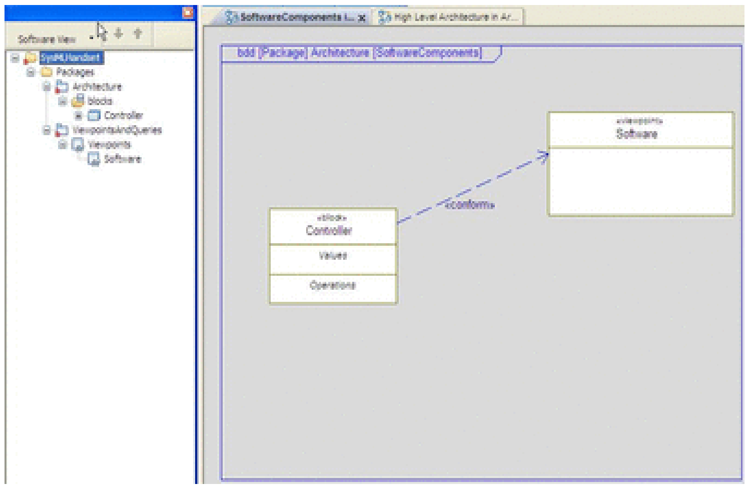
Select All | Clear All | Invert

[✓ Select Queries for filter:]
- [ ] Mechanical in ViewpointsAndQueries
- [ ] Safety in ViewpointsAndQueries
- [x] Software in ViewpointsAndQueries

Select All | Clear All | Invert

- [x] Apply on diagrams

OK | Cancel

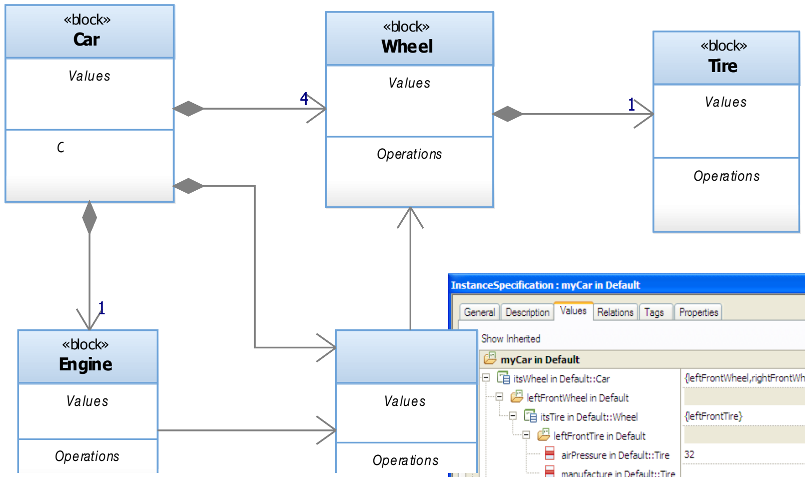


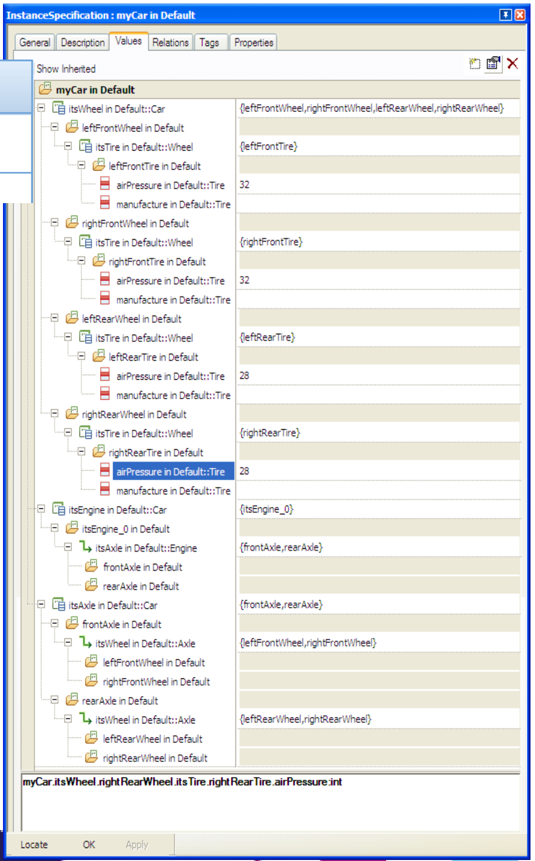
Systems Eng. Usability workflow enhancements

- **Harmony improvements**
  - Inverse actor pins to support sequence when the system is triggering the actor
  - Launching the model toolbox from the right-click context menu (auto selecting the element)
  - Support for send event, accept event and time event actions when generating sequence diagrams

Improved trade study capabilities using instance specification

- Specifying different instances of (sub-) systems
  - Values of attributes (Different weight, size, etc.)
  - Different kinds of parts (A diesel engine vs. gasoline engine)
  - Different composition schemes (Engine with 4 cylinders vs. 12 cylinders)
- Values can be automatically filled in by default based on the part hierarchy
- Intended for use by Rhapsody PCE (Parametric Constraint Evaluator) to compare design alternatives
- Compliant with instance specification in UML/SysML





InstanceSpecification : myCar in Default

myCar in Default

itsWheel in Default::Car	{leftFrontWheel,rightFrontWheel,leftRearWheel,rightRearWheel}
leftFrontWheel in Default	
itsTire in Default::Wheel	{leftFrontTire}
leftFrontTire in Default	
airPressure in Default::Tire	32
manufacture in Default::Tire	
rightFrontWheel in Default	
itsTire in Default::Wheel	{rightFrontTire}
rightFrontTire in Default	
airPressure in Default::Tire	32
manufacture in Default::Tire	
leftRearWheel in Default	
itsTire in Default::Wheel	{leftRearTire}
leftRearTire in Default	
airPressure in Default::Tire	28
manufacture in Default::Tire	
rightRearWheel in Default	
itsTire in Default::Wheel	{rightRearTire}
rightRearTire in Default	
airPressure in Default::Tire	28
manufacture in Default::Tire	
itsEngine in Default::Car	{itsEngine_0}
itsEngine_0 in Default	
itsAxle in Default::Engine	{frontAxle,rearAxle}
frontAxle in Default	
rearAxle in Default	
itsAxle in Default::Car	{frontAxle,rearAxle}
frontAxle in Default	
itsWheel in Default::Axle	{leftFrontWheel,rightFrontWheel}
leftFrontWheel in Default	
rightFrontWheel in Default	
rearAxle in Default	
itsWheel in Default::Axle	{leftRearWheel,rightRearWheel}
leftRearWheel in Default	
rightRearWheel in Default	

myCar.itsWheel.rightRearWheel.itsTire.rightRearTire.airPressure:int

RPE templates for SDD

- **The Software Design Description (SDD) describes the design of a Computer Software Configuration Item (CSCI). It describes the CSCI-wide design decisions, the CSCI architectural design, and the detailed design needed to implement the software**

- **Out of the box template is provided for generating Software Design Description (SDD)**
  - Compatible with SDD specification (DI-IPSC-81435A)

Support for Safety Critical Software Development

Enhancing support for using Rhapsody generated code with safety critical standards (e.g. DO178) for C and C++

- Making generated code to better comply
- Simplifying certification of framework code
- Test coverage with TestConductor (for C)

Safety critical features:

- Improved MISRA C++ compliancy
- SXF: Simplified Framework for C++
  - Improved MISRA/C++ compliance
  - Support for ARINC 653
- Refactoring and cleanup of MXF as safety framework for C
- Automating lowlevel code traceability to requirements (LLT)
- Provide out-of-the-box code coverage analysis in C via TestConductor

Enhanced support for MISRA/C++

- Generate const where required
  - Attribute and relation mutator argument
  - Attribute and relation accessor return type
- Attributes and Relations default visibility is Private
- Statechart without inheritance is compliant
  - Attributes are private and defined as enum
- Event’s Argument
  - Argument visibility is Private, mutator is generated

```
//## class A
class A {
public :
    //## auto_generated
    const int* getAtt(void) const;

    //## auto_generated
    void setAtt(int* const p_att);
private :

    int* att;       //## attribute att
};
```

Simplified C++ Framework (SXF) components

- Active classes (multi threading)
  - Mutexes
  - Event flags
- Static architecture
  - Static memory manager for events allocation
- MISRA C++ 2008 compliance
  - Safety critical C++ settings
  - Checks to support MISRA compliant modelling style
- Events
  - Asynchronous events
  - Synchronous events (triggered operations)
  - Timeouts

Simplified C++ Framework (SXF) components

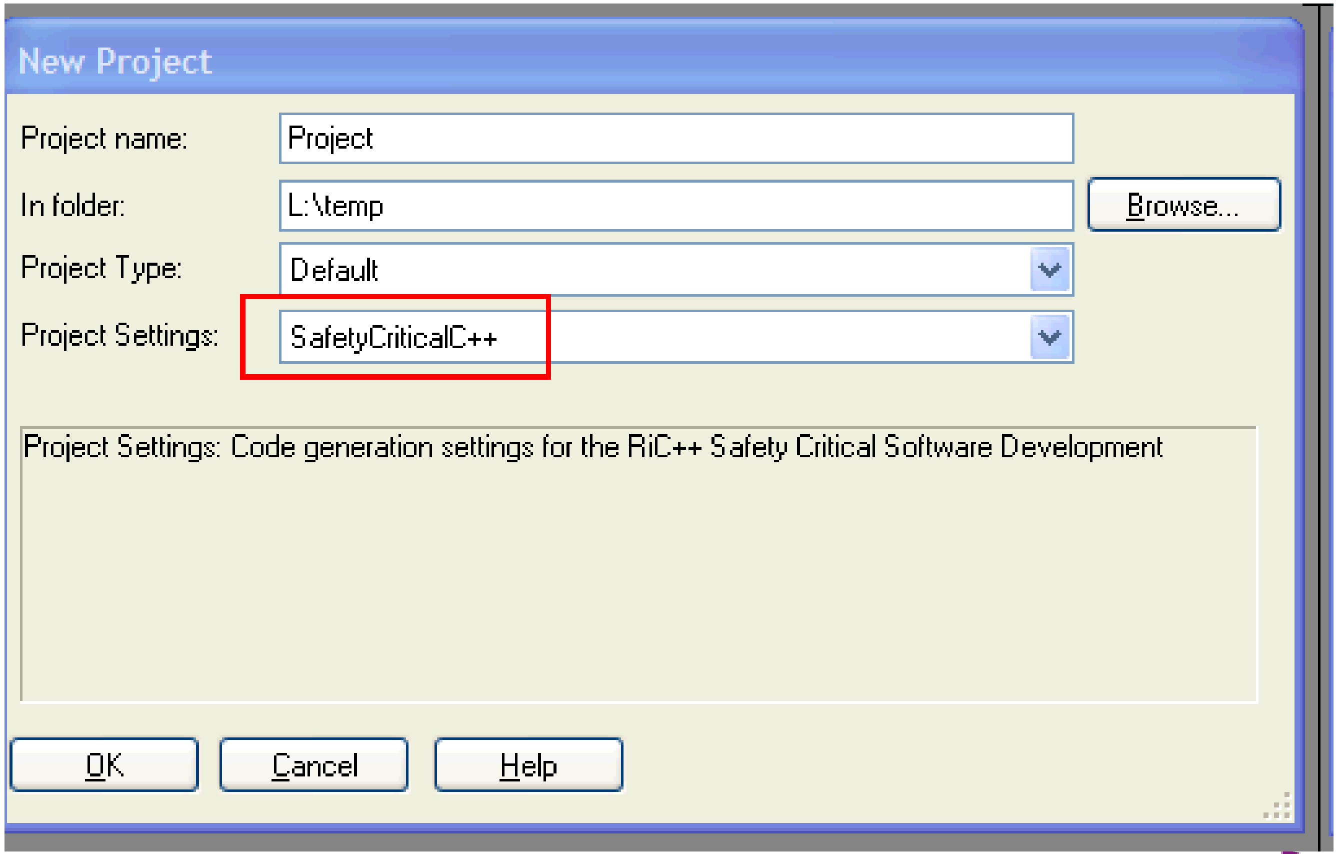
- Adapters
  - Workbench Managed 653 (APEX API based)
  - Microsoft (VS 2008/2010)
- Flat state charts
- Testing package
- No Ports
- No Animation/Tracing

SXF C++ vs OXF C++

SXF C++	OXF C++
Static architecture	Dynamic allocation
No animation/tracing	Animation/Tracing
Only Real Time	Real Time/Simulated Time
No containers (can be added)	Containers
Static memory manager (only BasedNumberOfInstances)	Static memory manager
Flat state charts	Flat/Reusable state charts
No Multi core in 7.6	Multi core
No Interfaces	Interface based
No Ports	Ports

New Safety Critical profile for C++

New model is created using Safety Critical C++ Settings


New Project

Project name: Project

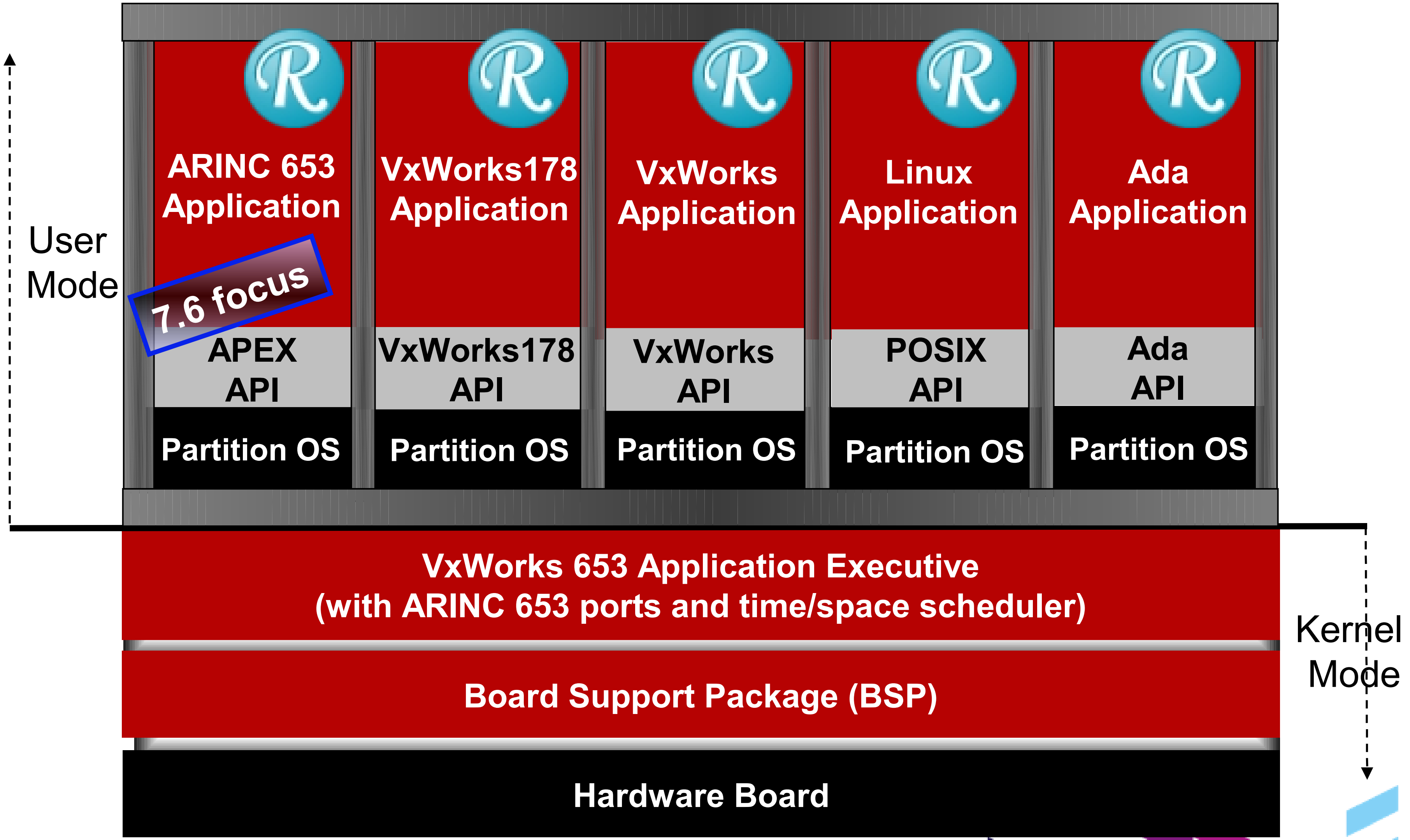
In folder: L:\temp

Project Type: Default

Project Settings: SafetyCriticalC++

Project Settings: Code generation settings for the RiC++ Safety Critical Software Development

Targeting Wind River VxWorks 653 with SXF

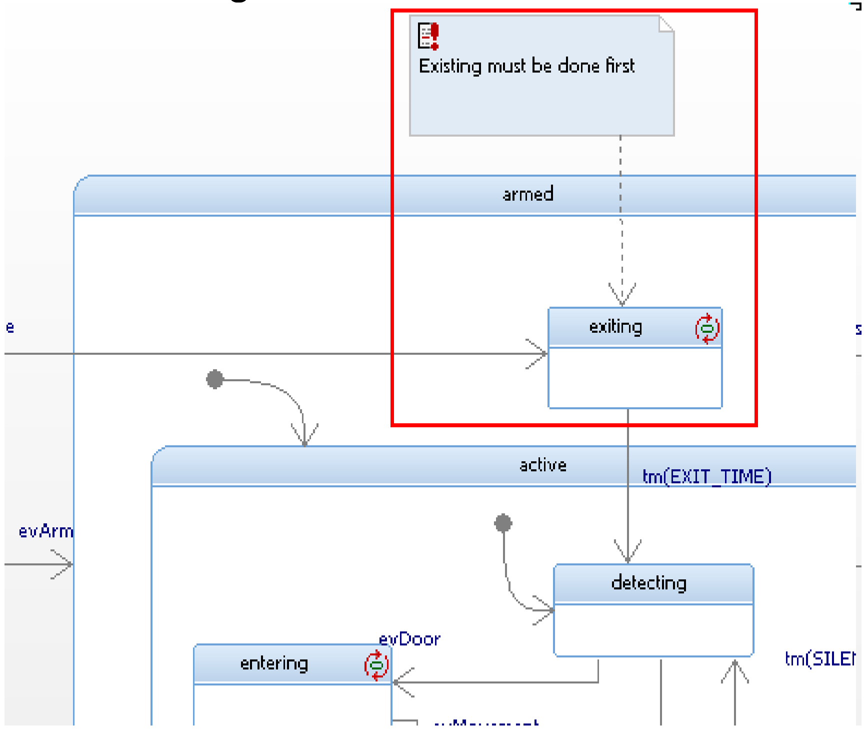


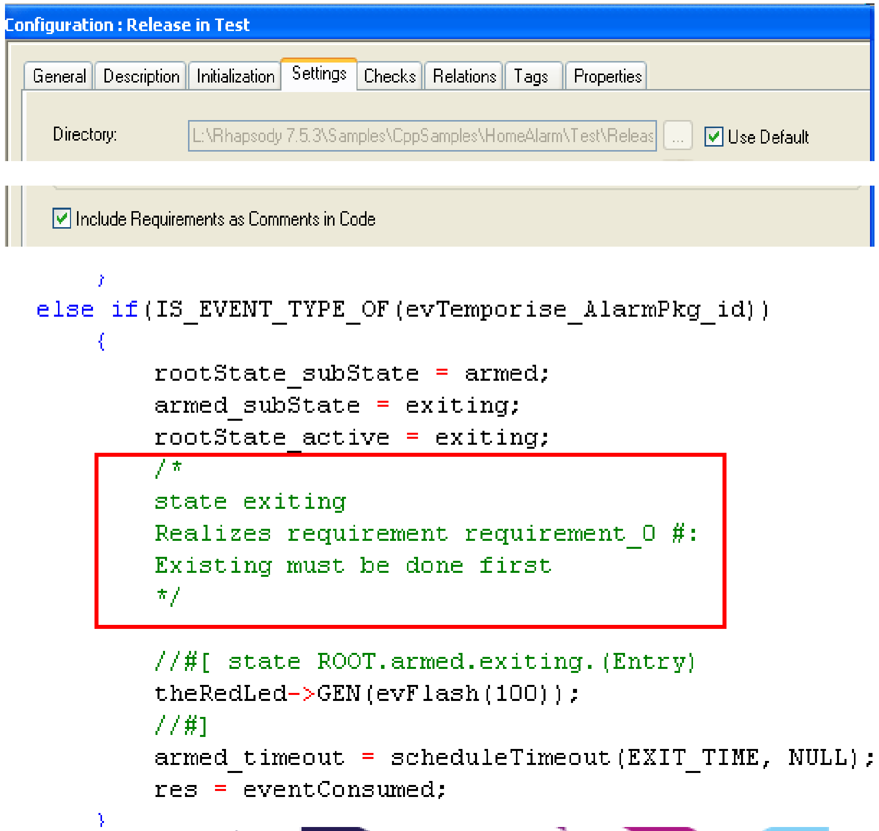
Cleanup MXF for S/C C applications

- Re-factor MXF files structure to separate core capabilities from utilities
  - Defining MXF core based on 7 components (files)
    - RiCEvent, RiCReactive, RiCTaskEM, RiCTimer, RiCtypes, RiCEventQueue, RiCOXF
- Simplify implementation and cleanup some MISRA-C violations with RiCReactive and RiCTimer
  - Remove dependancies on generic utilities such as RiCHeap
  - Use dedicated, MISRA-C compliant, code

LLT: code traceability coverage

- Mapping model level traceability into generated code
- Every generated code segment is traced to a requirement via inline code annotations
  - Operations
  - States, Transitions
  - Other generated code




Configuration : Release in Test

General | Description | Initialization | Settings | Checks | Relations | Tags | Properties

Directory: L:\Rhapsody 7.5.3\Samples\CppSamples\HomeAlarm\Test\Releas [✓] Use Default

[✓] Include Requirements as Comments in Code

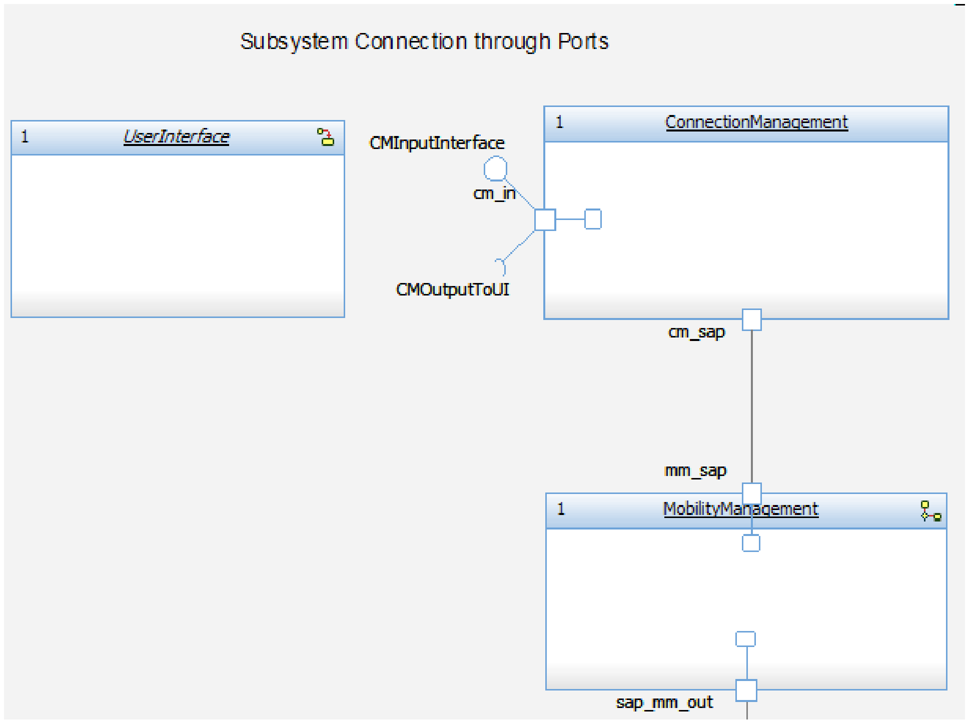
```
    }
else if(IS_EVENT_TYPE_OF(evTemporise_AlarmPkg_id))
    {
        rootState_subState = armed;
        armed_subState = exiting;
        rootState_active = exiting;
        /*
        state exiting
        Realizes requirement requirement_0 #:
        Existing must be done first
        */

        //#[ state ROOT.armed.exiting.(Entry)
        theRedLed->GEN(evFlash(100));
        //#]
        armed_timeout = scheduleTimeout(EXIT_TIME, NULL);
        res = eventConsumed;
    }
```

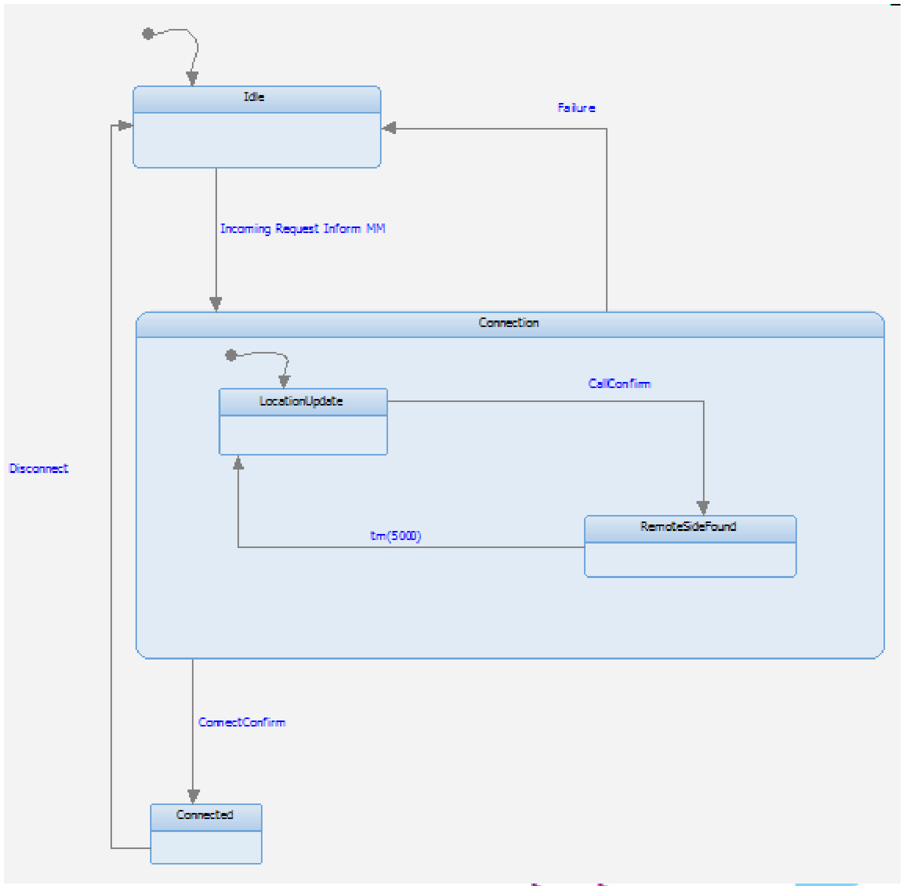
Modeling Usability

- New diagrams look & feel
- Performance improvements
- Auto completion assist by improving “intellivisor”
- Eclipse plug-in workflow enhancements

New Diagrams Look & Feel

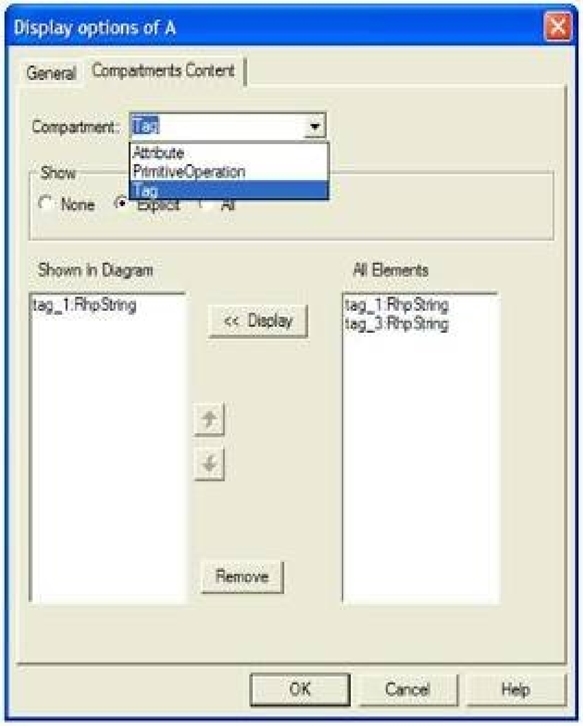
- New out-of-the-box Look & Feel





Enhanced control for compartments on diagrams

- Control compartments
  - Unified support for all compartments
  - No modification/warnings of diagram on class change
  - Avoid creation of unresolved elements by compartments


Display options of A

General | Compartments Content

Compartment: Tag

Attribute
PrimitiveOperation
Tag

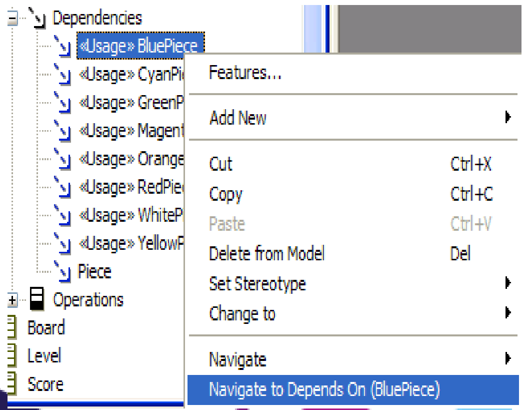
Show: None [✓ Explicit] All

Shown in Diagram:
tag_1:RhpString

All Elements:
tag_1:RhpString
tag_3:RhpString

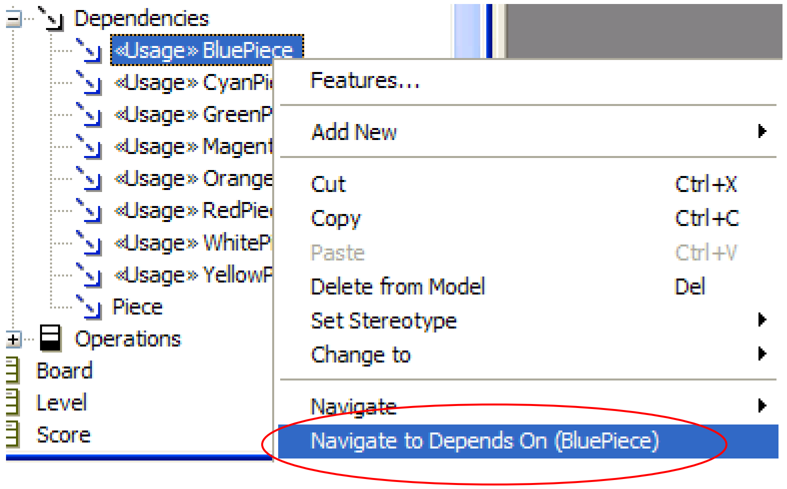
Performance - Incremental (on demand) load of units

- Only Necessary parts of model are loaded “on demand”
- Model is “loaded” faster, consume less memory, Rhapsody works faster
- To the extent possible, let the user feel that the whole model is loaded
  - Expand (iterate) browser
    - Also on open feature dialog, double click, right click)
  - Open diagram
  - Target of relation (dependencies, generalizations, etc.)
  - Load entire project/package/scope as needed
    - Code generation (scope), Reporter plus (Package), Save as (Project)
  - Sometime Ask user (dialog) entire project/package/scope….
    - Search in model / look in
- Deletion
  - Does NOT trigger loading
    - May create a UR element



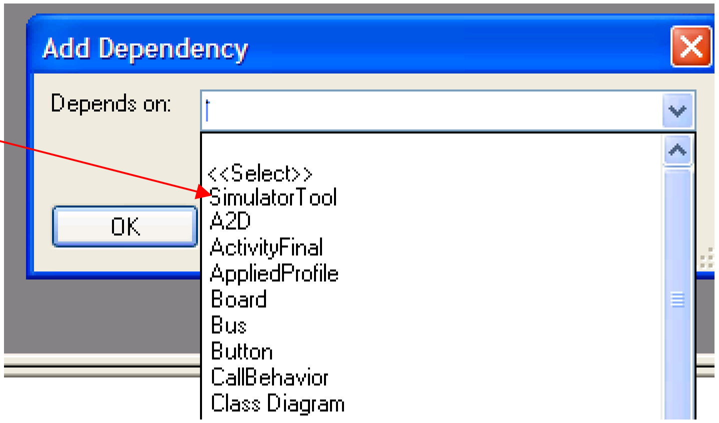
How load on demand occurs

- Implicitly loaded
  - Expand (iterate) browser
    - Also on open feature dialog, double click, right click)
  - Open diagram
  - Target of relation (dependencies, generalizations, etc.)



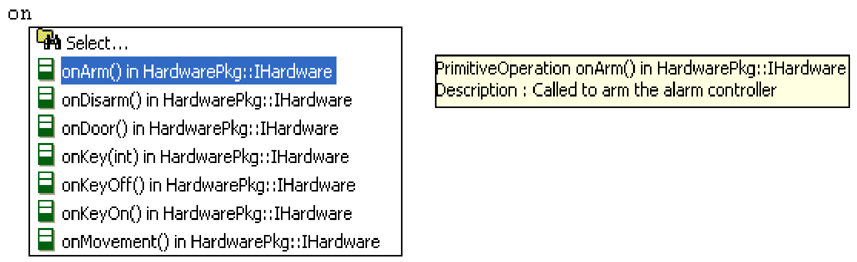
How load on demand occurs (cont)

- Tools
  - Load silently entire project/package/scope….
    - Code generation (scope), Reporter plus (Package), Save as (Project)
  - Load by Ask user (dialog) entire project/package/scope….
    - Search in model / look in
  - Not loading, using the currently loaded only
    - Select in combo

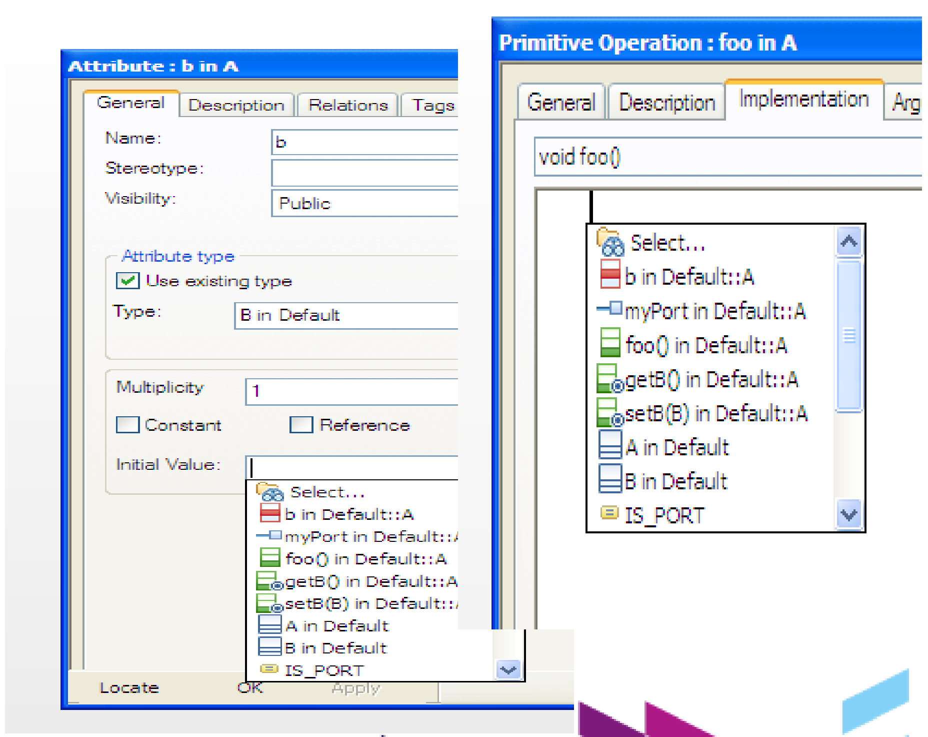


Auto completion assist by improving Intellivisor

- Context sensitive suggestions list
  - **Control-Space** (Show a current scope c possible)
  - "**.**" or "**->**" (Show the local context completior
  - "**::**" (Show the namespace completions)
- Filtering list based on current prefix
- Improved completion for e.g. GEN
- Available at:
  - Browser
  - Dialogs
  - Code Editor
  - Diagrams


on
Select...
onArm() in HardwarePkg::IHardware
onDisarm() in HardwarePkg::IHardware
onDoor() in HardwarePkg::IHardware
onKey(int) in HardwarePkg::IHardware
onKeyOff() in HardwarePkg::IHardware
onKeyOn() in HardwarePkg::IHardware
onMovement() in HardwarePkg::IHardware

PrimitiveOperation onArm() in HardwarePkg::IHardware
Description : Called to arm the alarm controller


Attribute : b in A

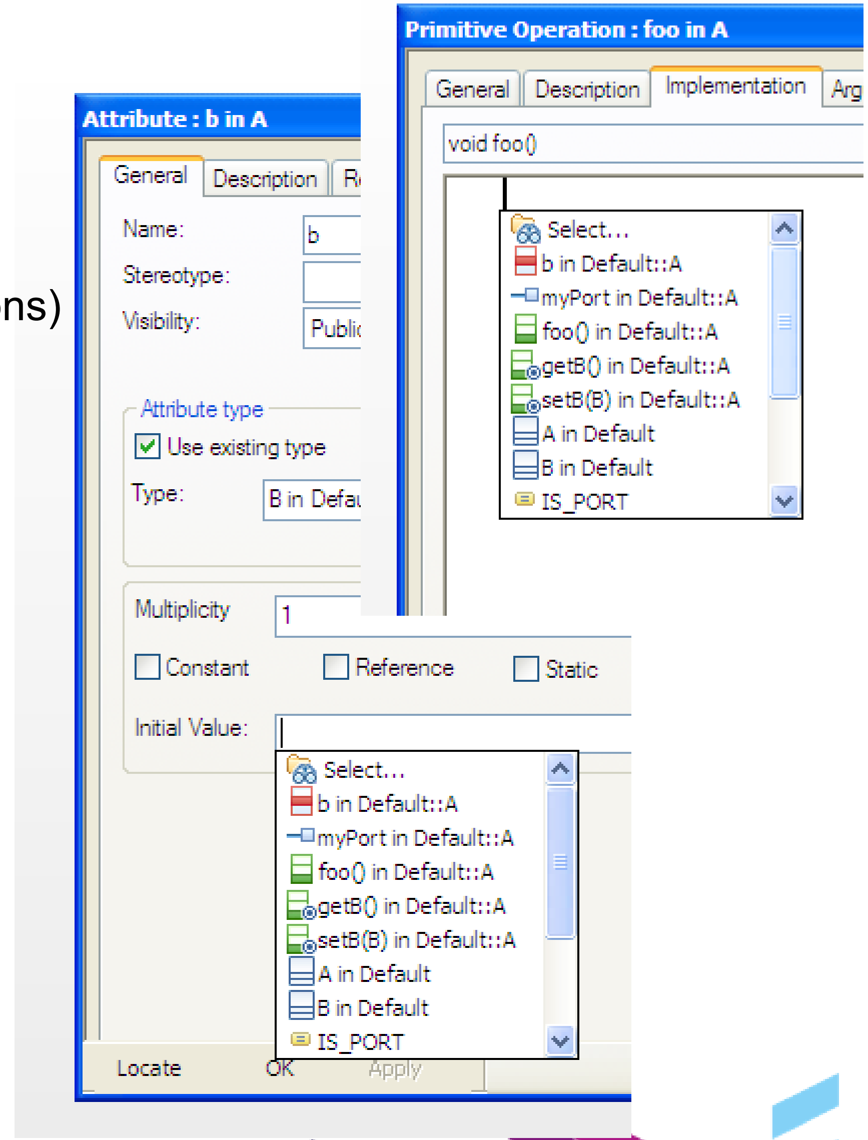
Name: b
Stereotype:
Visibility: Public
Attribute type
Use existing type ✓
Type: B in Default
Multiplicity 1
Constant ☐ Reference ☐
Initial Value:
Select...
b in Default::A
myPort in Default::/
foo() in Default::A
getB() in Default::A
setB(B) in Default::/
A in Default
B in Default
IS_PORT

Primitive Operation : foo in A

void foo()
Select...
b in Default::A
myPort in Default::A
foo() in Default::A
getB() in Default::A
setB(B) in Default::A
A in Default
B in Default
IS_PORT

Details

- Invoke IntelliVisor by:
  - **Control-Space** (Show a current scope completions or auto complete if possible)
  - "**.**" or "**->**" (Show the local context completions)
  - "**::**" (Show the namespace completions)
- Available at:
  - Browser
  - Dialogs
  - Code Editor
  - Diagrams

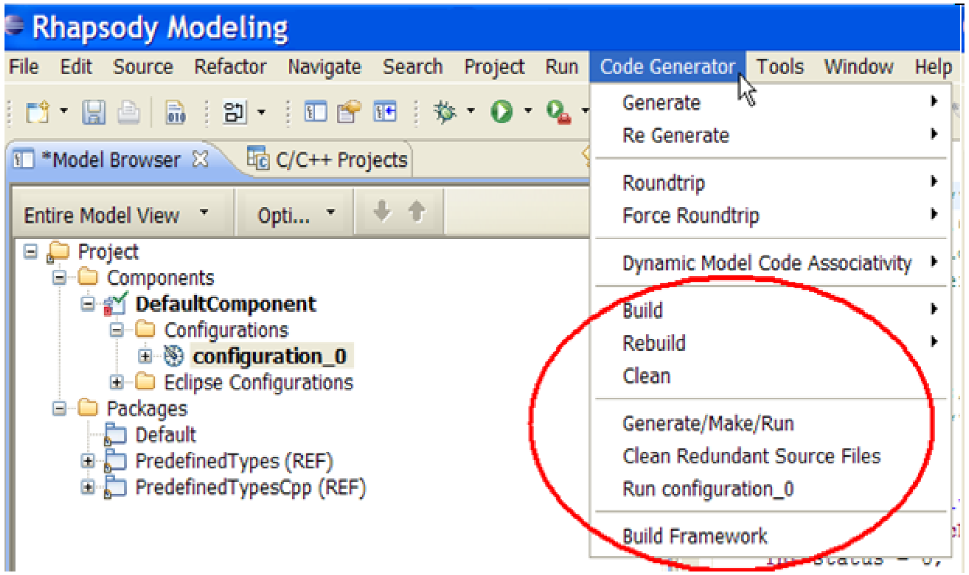


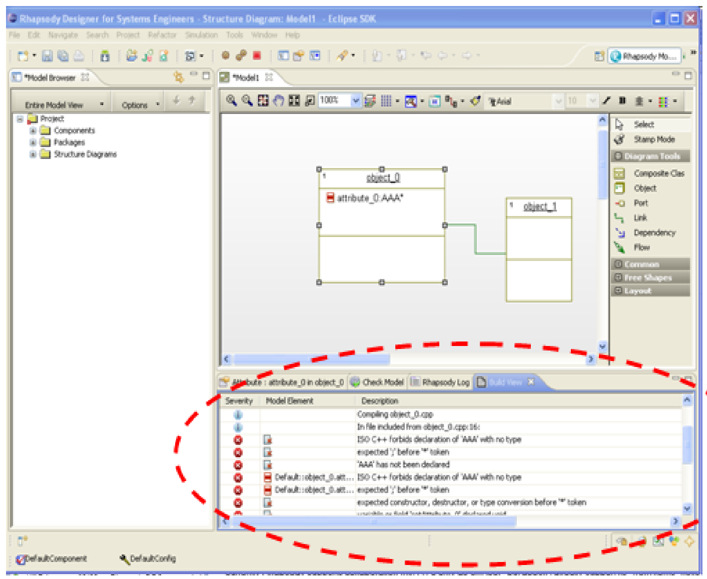
Eclipse plug-in workflow enhancements

- Ease creation of Eclipse projects using default settings
- Ability to roundtrip file(s) without matching active configuration in Rhapsody to the active project in Eclipse.
- New Rhapsody toolbar improves usability for working with multiple projects and components
- Added synchronization button, update model with code changes after disconnected development
- Improved Rhapsody & Eclipse communication
  - UDP Ports availability
- Eclipse Platform Integration (Windows) will be available in all languages that Stand-alone is available

Using Eclipse Platform Integration for non-eclipse targets

- To allow integration with RTC (as CM tool) for NOT native Eclipse environments we add more abilities for code operations for Eclipse Platform Integration
  - Build, Clean, GMR for non-Eclipse configurations
  - Build output window
  - DMCA support for Eclipse code view

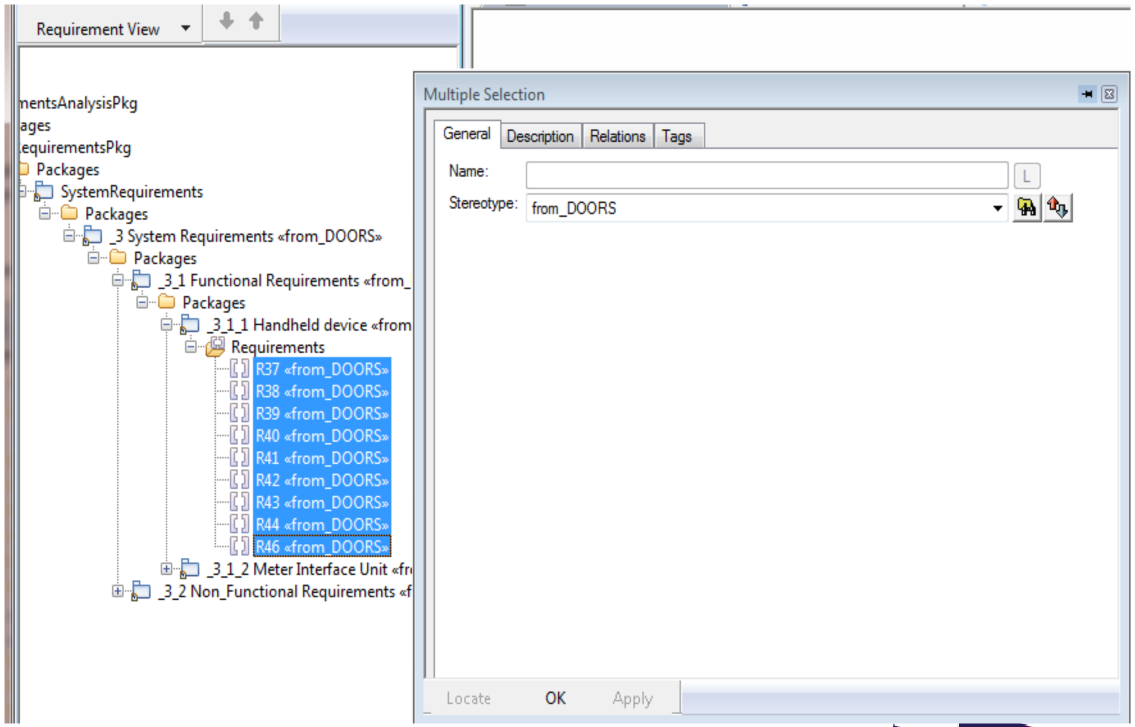




Severity	Model Element	Description
		Compiling object_0.cpp
		In file included from object_0.cpp:16:
		ISO C++ forbids declaration of 'AAA' with no type
		expected ';' before '*' token
		'AAA' has not been declared
	Default::object_0.att...	ISO C++ forbids declaration of 'AAA' with no type
	Default::object_0.att...	expected ';' before '*' token
		expected constructor, destructor, or type conversion before '*' token

Usability of User Interface

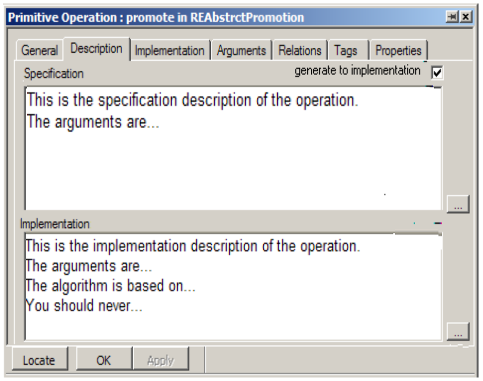
- Non blocking external editors
  - Allows to work inside Rhapsody while the external editor is open
- Support for editing multiple elements
  - Description, Tags and Properties along with other attributes are now available to edit when multiple elements are in selection



Name:
Stereotype: from_DOORS

Usability for Software Developers

- Better support for C++ <<Friend>> dependency
- Generate Operation descriptions into both .h and .c/.cpp files
- Simultaneous usage of external code editors and navigation in Rhapsody
- Improve support for CORBA
  - CORBA Unions
  - Support CORBA strings in struct with multiplicity
- Simultaneous usage of external code editors and model navigation
  - user is able to run an external editor in a separate thread while still being able to navigate the model see more information while editing the code
  - Only one external editor allowed open at a time
  - No support for merge if there are conflicting differences
- Identify header guards during Reverse Engineering

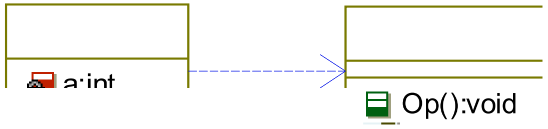


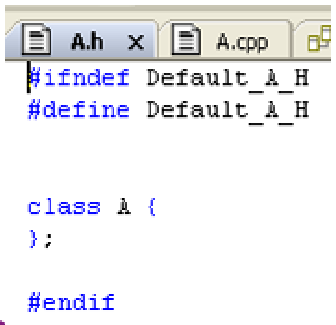
Specification generate to implementation ✓

This is the specification description of the operation.
The arguments are...

Implementation

This is the implementation description of the operation.
The arguments are...
The algorithm is based on...
You should never...



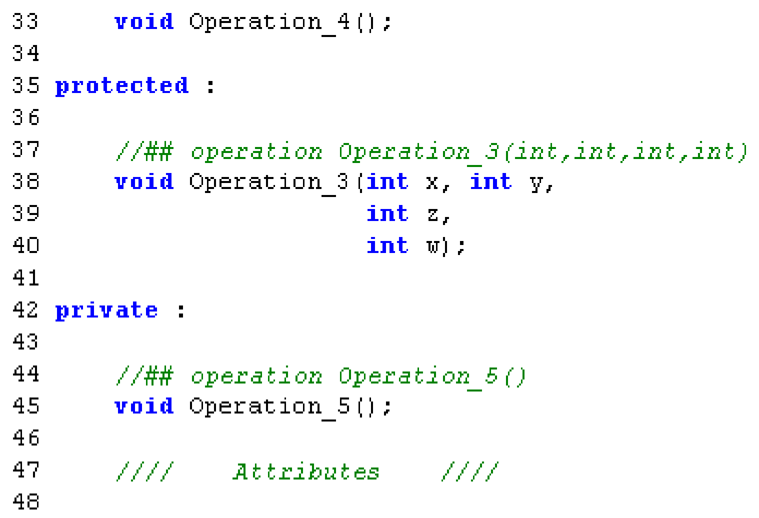


```
#ifndef Default_A_H
#define Default_A_H


class A {
};

#endif
```

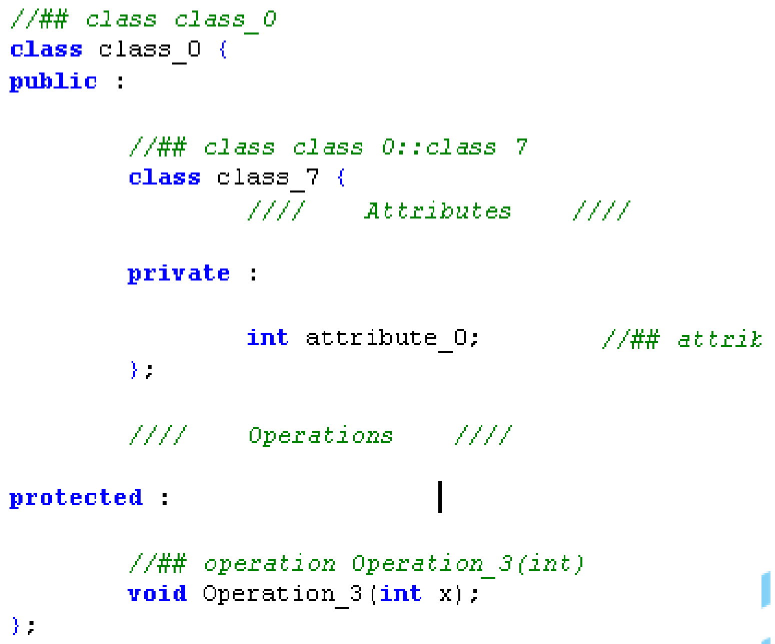
Generated Code Customization Enhancements

- Option to format generated code for functions with more than 2 parameters
- Option to specify a string to be used for code indentation
- Easier way to generate only the header file of a class

A sample that shows how to group operations into visibility groups



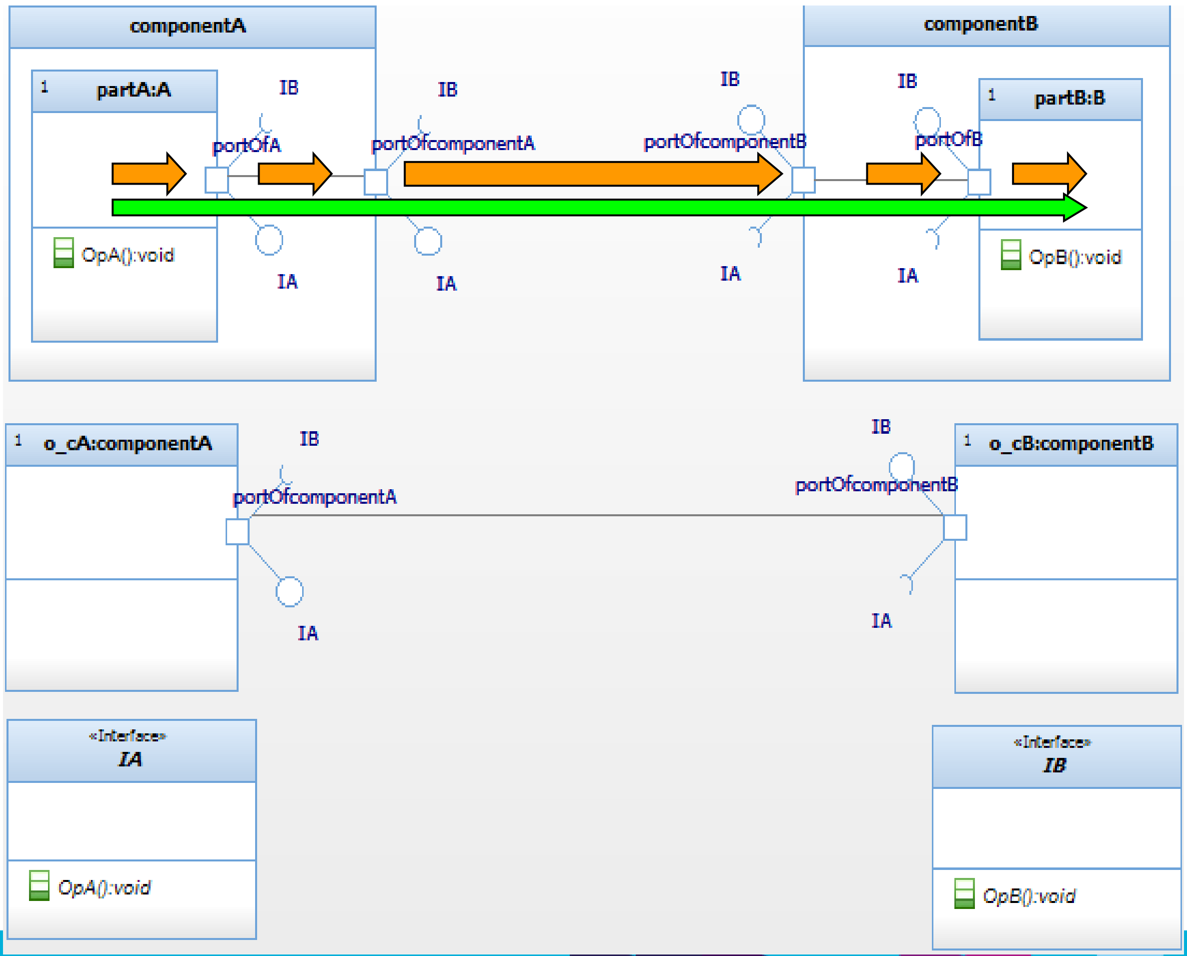
```
    void Operation_4();

protected :

    //## operation Operation_3(int,int,int,int)
    void Operation_3(int x, int y,
                     int z,
                     int w);

private :

    //## operation Operation_5()
    void Operation_5();

    ////    Attributes    ////
```



```
//## class class_0
class class_0 {
public :

        //## class class_0::class_7
        class class_7 {
                ////    Attributes    ////

        private :

                int attribute_0;        //## attrib
        };

        ////    Operations    ////

protected :

        //## operation Operation_3(int)
        void Operation_3(int x);
};
```

Run-time optimization for Ports

- Provide direct calls instead of call delegation between non-behavioral ports
  - Improve run time performance
  - Improve debugging experience
  - Scope: C++



Rhapsody in Ada Enhancements

- Roundtripping
- Removed Booch components
  - User is advised via checks when Booch containers are required
    - Unbounded relations
    - Qualified relations
    - Multicast ports
- Type reversing improvement
  - Record with discriminant
  - Subtypes
  - Arrays
- Property C++ alignments
  - Keywords substitution
  - DescriptionTemplate
  - Post processor
  - Pragma management

Extendibility Enhancements

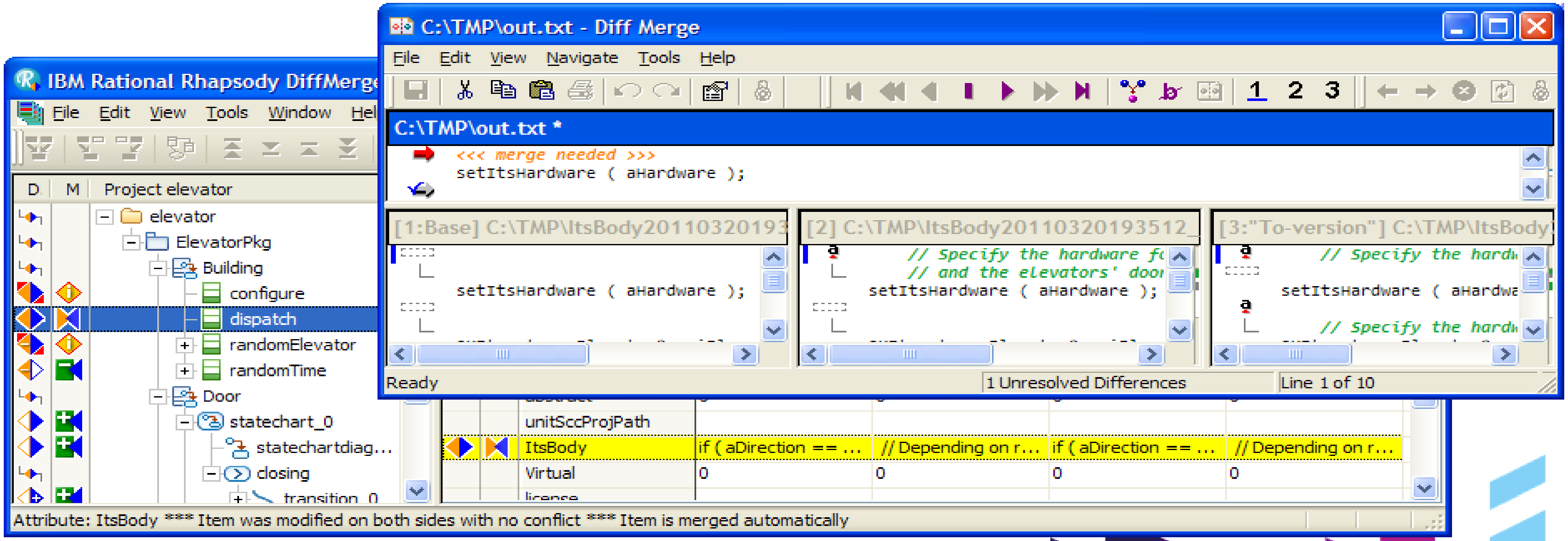
- Improvements in Diagrams API: nodes and styles
- Java plug-ins to have arguments
- Add Application Listener sample to Linux Install
- RulesPlayer in Rhapsody Architect
  - Allowing the integration of Rhapsody models with other inputs and outputs of the engineer's process
  - The engineer would define his mapping using the Rulescomposer
  - The engineer would deploy his mapping rule as tool menu option in Rhapsody or a plugin across his team
  - Any engineer using Architect or Developer version would be able to execute the transformation defined.
  - WYSIWYG templates are disabled

Diff/Merge Usability

- Align Statechart and Activity diagrams with other diagrams
  - E.g. “Take from left”, “Take from right”
- Improved 3-way text diff/merge
  - Merge operation body when there are changes on both sides of the comparison
- Allow users to merge references
  - i.e. stereotypes, component scope, etc.
  - Enable addition of references from both sides instead of either-or

New Textual Diff/Merge Utility

- Base-aware textual comparison and merge
- Automatic resolution of non-conflicting textual differences
  - For instance, the Building::dispatch operation is not a conflict anymore and DiffMerge has resolved it automatically
- As the result, some merging cases that used to require user to do it manually – through UI – can be done now automatically



C:\TMP\out.txt - Diff Merge

```
<<< merge needed >>>
setItsHardware ( aHardware );
```

[1:Base] C:\TMP\ItsBody20110320193

```
setItsHardware ( aHardware );
```

[2] C:\TMP\ItsBody20110320193512

```
// Specify the hardware fo
// and the elevators' door
setItsHardware ( aHardware );
```

[3:"To-version"] C:\TMP\ItsBody

```
// Specify the hardw
setItsHardware ( aHardwa
// Specify the hardw
```

Ready | 1 Unresolved Differences | Line 1 of 10

IBM Rational Rhapsody DiffMerge

Project elevator: elevator > ElevatorPkg > Building > configure, dispatch, randomElevator, randomTime; Door > statechart_0 > statechartdiag..., closing, transition_0

		unitSccProjPath				
		ItsBody	if (aDirection == ...	// Depending on r...	if (aDirection == ...	// Depending on r...
		Virtual	0	0	0	0

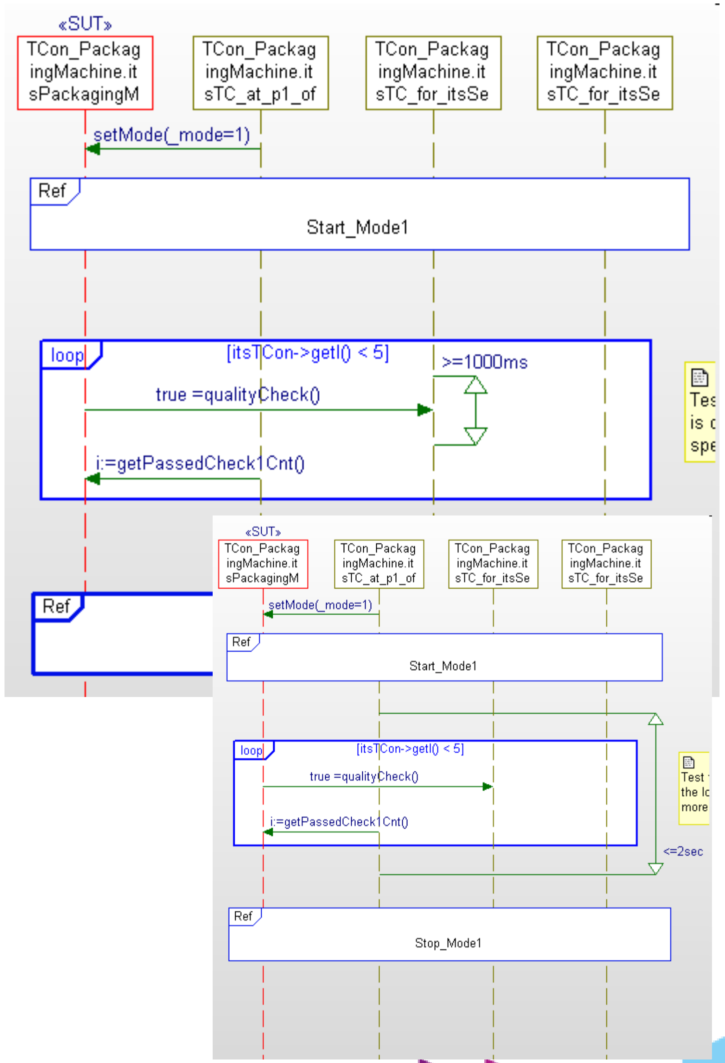
Attribute: ItsBody *** Item was modified on both sides with no conflict *** Item is merged automatically

Model Based Testing

- Enriched test case specifications with Sequence Diagrams Operators
- On Target Offline Testing (without Rhapsody in the loop)
- Safety critical development with Rhapsody
  - Code coverage computation for RhapsodyC code
- Android
  - Testing support for Android applications

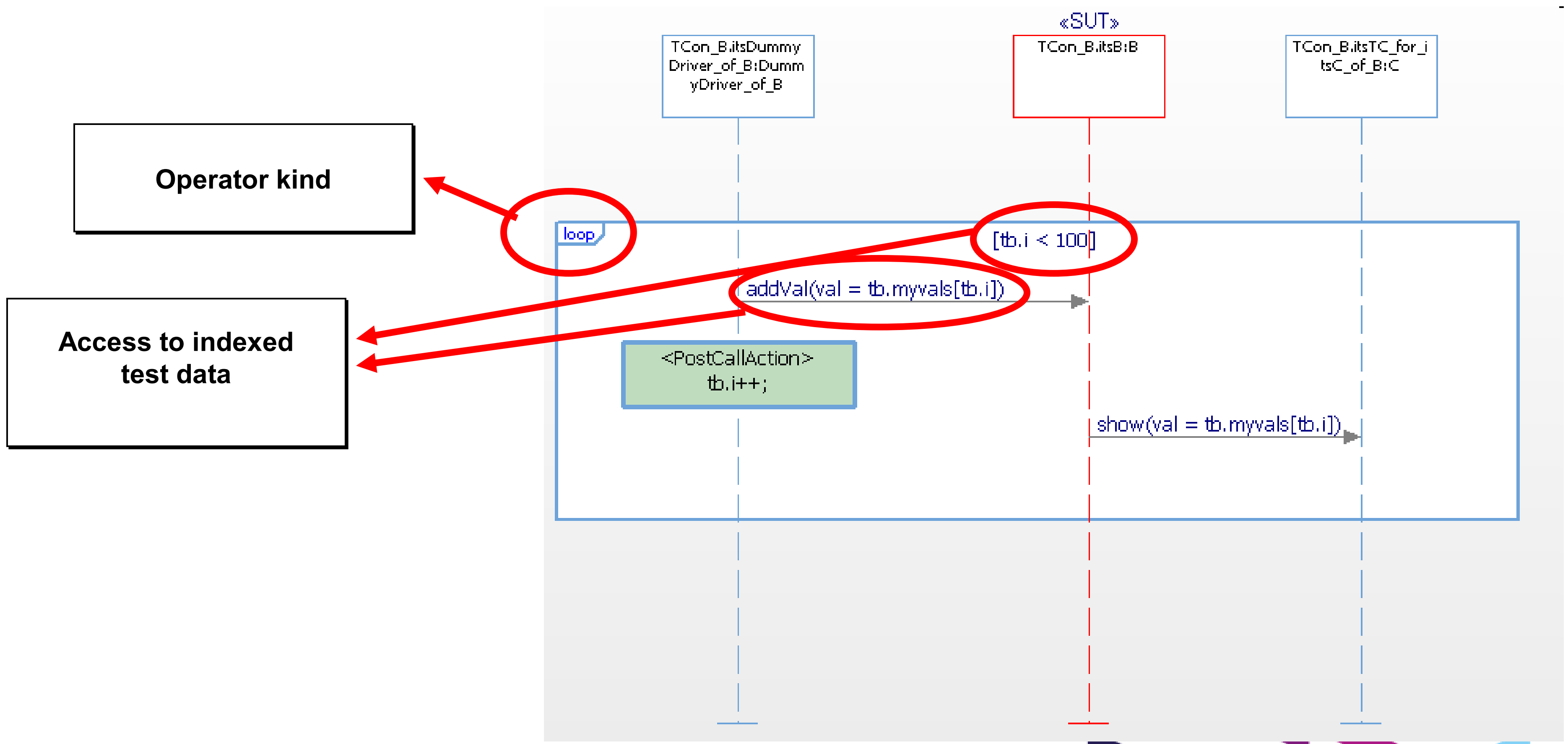
Enhanced test case specification with SD

- **Support of interaction operators**
  - opt, alt, loop, break, parallel, consider
- **Support of timing checks**
  - e.g. minimal/maximal response time
- Full support of variables and test data in SDs
  - parameterized tests
  - access to attributes of test components
- Code Editor in SD TestActions
  - Auto completion when defining actions in SDs
- Stubbing of non virtual functions (support of replacements)
  - allows overriding behavior of non-virtual functions
- Using auto generated operations in test cases
- Improved Failure SDs ("Show as SD")
  - Complete error path is shown


«SUT»
TCon_PackagingMachine.itsPackagingM
TCon_PackagingMachine.itsTC_at_p1_of
TCon_PackagingMachine.itsTC_for_itsSe
TCon_PackagingMachine.itsTC_for_itsSe
setMode(_mode=1)
Ref
Start_Mode1
loop
[itsTCon->getI() < 5]
>=1000ms
true =qualityCheck()
i:=getPassedCheck1Cnt()
Ref
«SUT»
TCon_PackagingMachine.itsPackagingM
TCon_PackagingMachine.itsTC_at_p1_of
TCon_PackagingMachine.itsTC_for_itsSe
TCon_PackagingMachine.itsTC_for_itsSe
setMode(_mode=1)
Ref
Start_Mode1
loop
[itsTCon->getI() < 5]
true =qualityCheck()
i:=getPassedCheck1Cnt()
<=2sec
Ref
Stop_Mode1

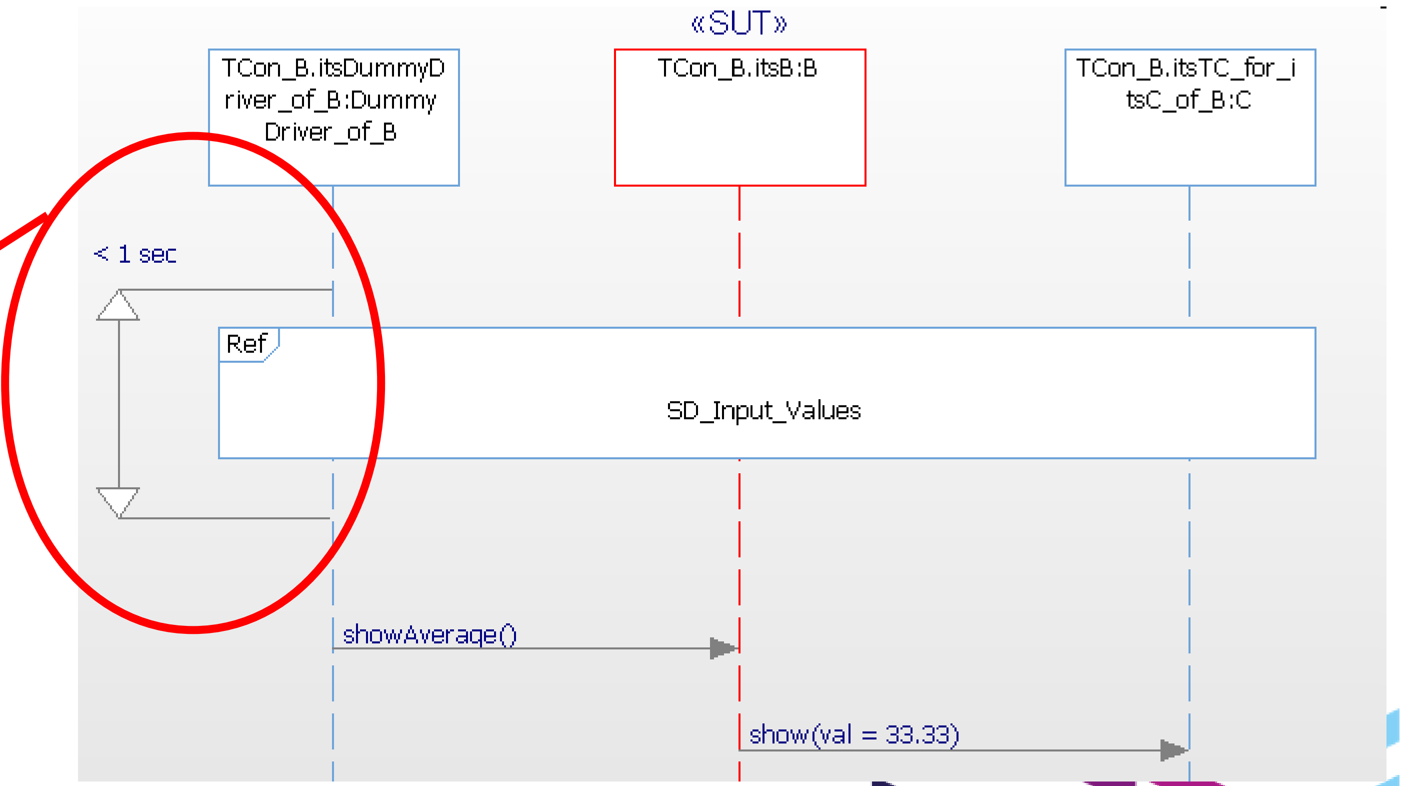
Support of interaction operators

- Support of interaction operators
  - opt, alt, loop, break, parallel, consider
  - Example: loop


Operator kind
Access to indexed test data
«SUT»
TCon_B.itsDummyDriver_of_B:DummyDriver_of_B
TCon_B.itsB:B
TCon_B.itsTC_for_itsC_of_B:C
loop
[tb.i < 100]
addVal(val = tb.myvals[tb.i])
<PostCallAction> tb.i++;
show(val = tb.myvals[tb.i])

Support of timing checks

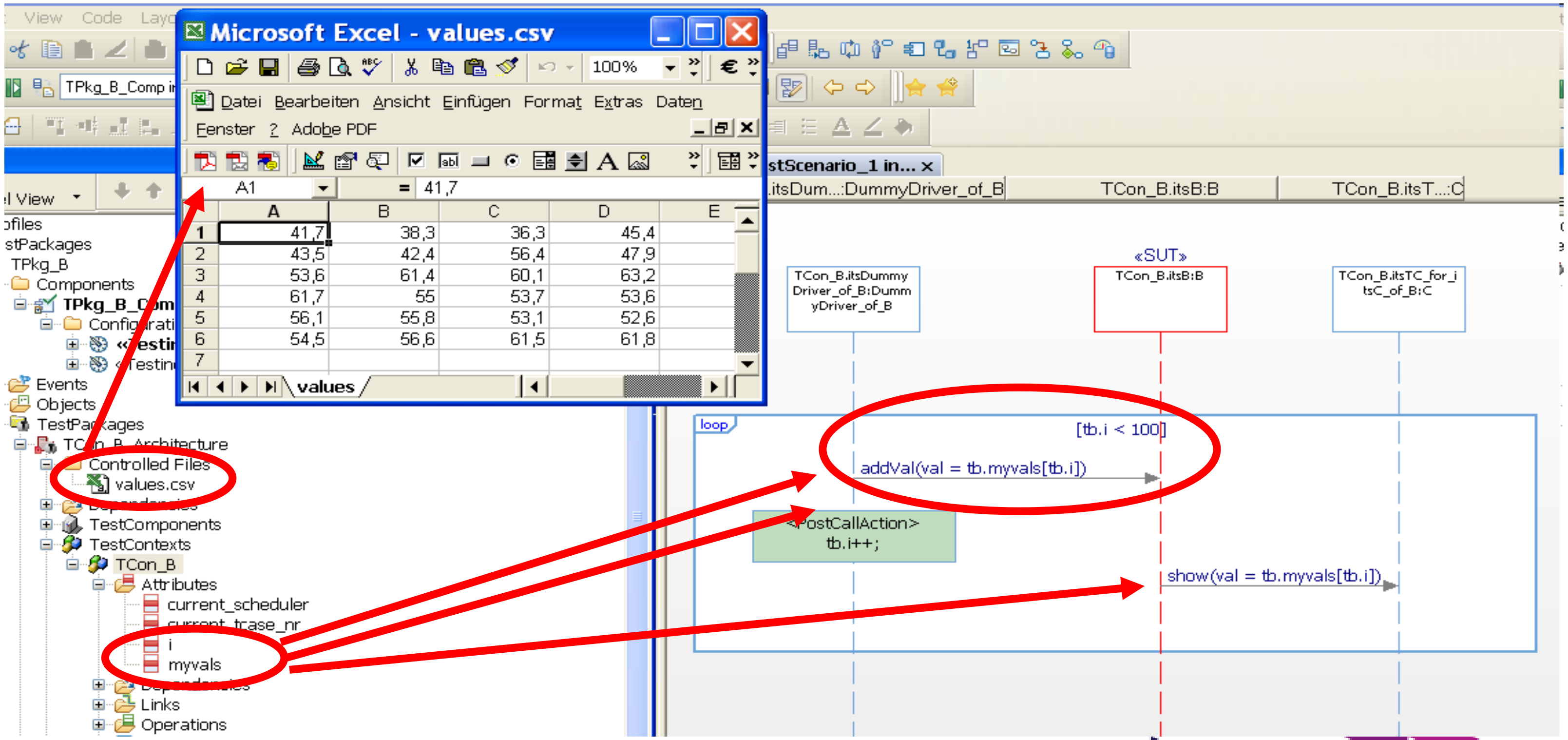
- Support of timing checks
  - e.g. minimal/maximal response time
  - Example: checking upper bound



Everything inside the scope of the time interval should complete within 1 second

Variables and test data in SDs

- Full support of variables and test data in SDs
  - separating test data from test control, improves maintainability of test cases



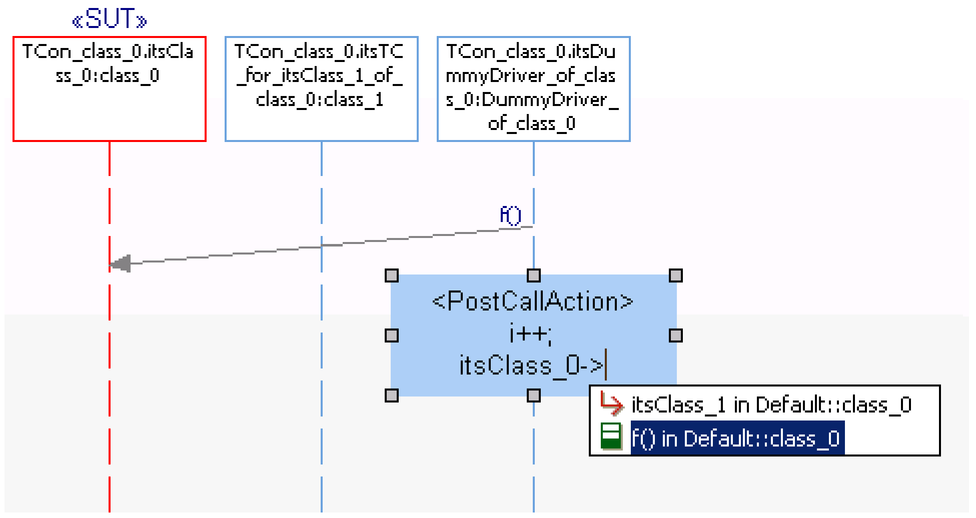
A1 = 41,7

	A	B	C	D	E
1	41,7	38,3	36,3	45,4	
2	43,5	42,4	56,4	47,9	
3	53,6	61,4	60,1	63,2	
4	61,7	55	53,7	53,6	
5	56,1	55,8	53,1	52,6	
6	54,5	56,6	61,5	61,8	
7					

```
[tb.i < 100]
addVal(val = tb.myvals[tb.i])
<PostCallAction>
tb.i++;
show(val = tb.myvals[tb.i])
```

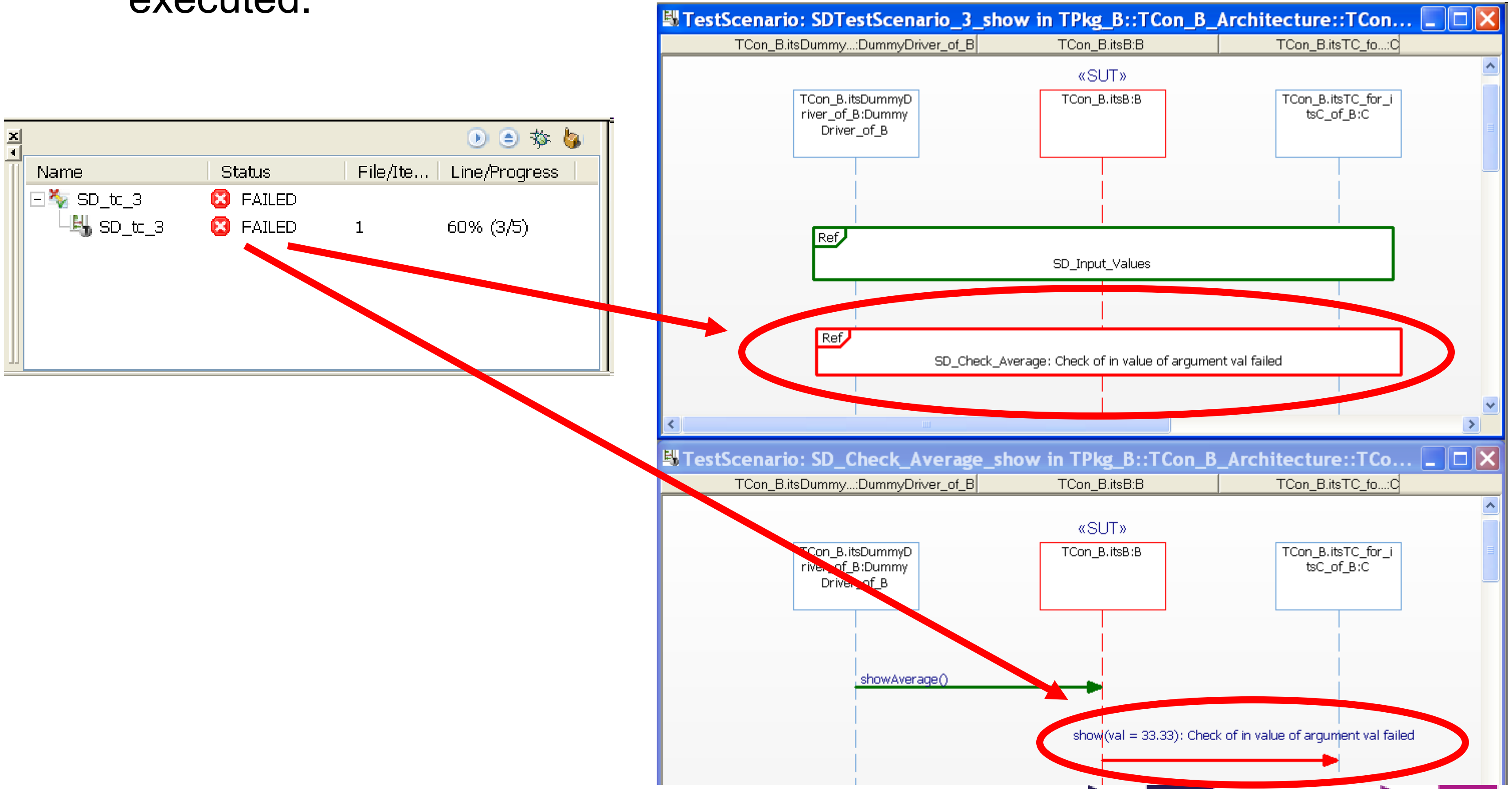
Code editor for SD test actions

- Code Editor in SD TestActions
  - Intellisense when you define test actions in SD test cases


«SUT»
TCon_class_0.itsClass_0:class_0
TCon_class_0.itsTC_for_itsClass_1_of_class_0:class_1
TCon_class_0.itsDummyDriver_of_class_0:DummyDriver_of_class_0
f()
<PostCallAction>
i++;
itsClass_0->
itsClass_1 in Default::class_0
f() in Default::class_0

Improved Failure SDs

- Complete error path is shown in red
- Green and blue colors indicate parts of the SD that has been executed/not executed.



Name	Status	File/Ite...	Line/Progress
SD_tc_3	FAILED		
SD_tc_3	FAILED	1	60% (3/5)

TestScenario: SDTestScenario_3_show in TPkg_B::TCon_B_Architecture::TCon...

TCon_B.itsDummy...:DummyDriver_of_B
TCon_B.itsB:B
TCon_B.itsTC_fo...:C
«SUT»
TCon_B.itsDummyDriver_of_B:DummyDriver_of_B
TCon_B.itsB:B
TCon_B.itsTC_for_itsC_of_B:C
Ref
SD_Input_Values
Ref

SD_Check_Average: Check of in value of argument val failed

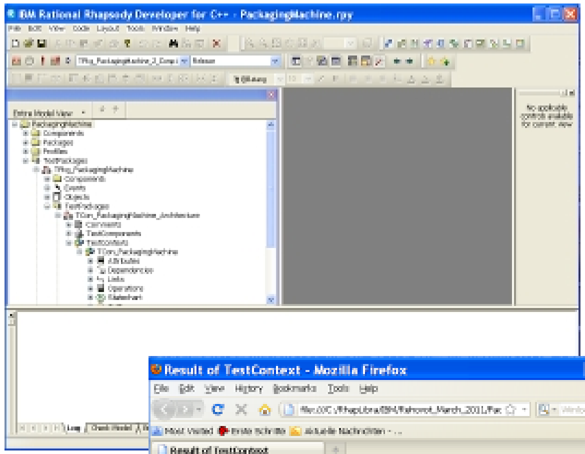
TestScenario: SD_Check_Average_show in TPkg_B::TCon_B_Architecture::TCo...

TCon_B.itsDummy...:DummyDriver_of_B
TCon_B.itsB:B
TCon_B.itsTC_fo...:C
«SUT»
TCon_B.itsDummyDriver_of_B:DummyDriver_of_B
TCon_B.itsB:B
TCon_B.itsTC_for_itsC_of_B:C
showAverage()

show(val = 33.33): Check of in value of argument val failed

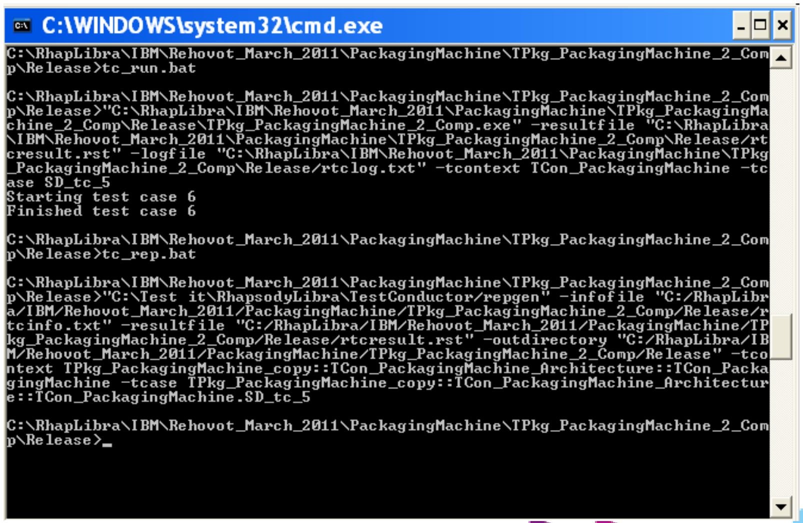
Testing without Rhapsody in the loop

- Enables testing completely without animation
- Enables testing on almost every target system

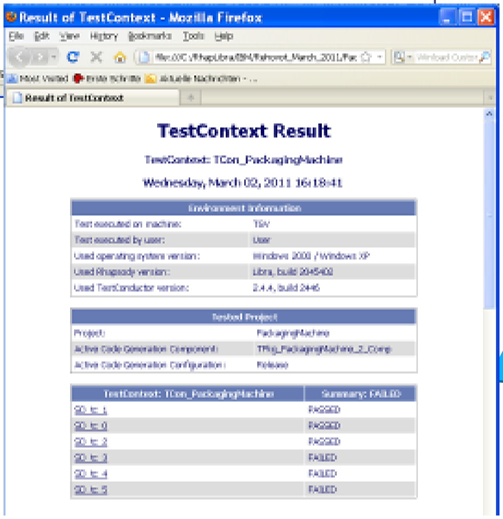


generate test code

- Executable contains all test code
- Execute test cases outside Rhapsody
- Generate reports outside Rhapsody



```
C:\RhapLibra\IBM\Rehovot_March_2011\PackagingMachine\TPkg_PackagingMachine_2_Com
p\Release>tc_run.bat

C:\RhapLibra\IBM\Rehovot_March_2011\PackagingMachine\TPkg_PackagingMachine_2_Com
p\Release>"C:\RhapLibra\IBM\Rehovot_March_2011\PackagingMachine\TPkg_PackagingMa
chine_2_Comp\Release\TPkg_PackagingMachine_2_Comp.exe" -resultfile "C:\RhapLibra
\IBM\Rehovot_March_2011\PackagingMachine\TPkg_PackagingMachine_2_Comp\Release/rt
cresult.rst" -logfile "C:\RhapLibra\IBM\Rehovot_March_2011\PackagingMachine\TPkg
_PackagingMachine_2_Comp\Release/rtclog.txt" -tcontext TCon_PackagingMachine -tc
ase SD_tc_5
Starting test case 6
Finished test case 6

C:\RhapLibra\IBM\Rehovot_March_2011\PackagingMachine\TPkg_PackagingMachine_2_Com
p\Release>tc_rep.bat

C:\RhapLibra\IBM\Rehovot_March_2011\PackagingMachine\TPkg_PackagingMachine_2_Com
p\Release>"C:\Test it\RhapsodyLibra\TestConductor/repgen" -infofile "C:/RhapLibr
a/IBM/Rehovot_March_2011/PackagingMachine/TPkg_PackagingMachine_2_Comp/Release/r
tcinfo.txt" -resultfile "C:/RhapLibra/IBM/Rehovot_March_2011/PackagingMachine/TP
kg_PackagingMachine_2_Comp/Release/rtcresult.rst" -outdirectory "C:/RhapLibra/IB
M/Rehovot_March_2011/PackagingMachine/TPkg_PackagingMachine_2_Comp/Release" -tco
ntext TPkg_PackagingMachine_copy::TCon_PackagingMachine_Architecture::TCon_Packa
gingMachine -tcase TPkg_PackagingMachine_copy::TCon_PackagingMachine_Architectur
e::TCon_PackagingMachine.SD_tc_5

C:\RhapLibra\IBM\Rehovot_March_2011\PackagingMachine\TPkg_PackagingMachine_2_Com
p\Release>_
```


Result of TestContext - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Most Visited
Erste Schritte
Aktuelle Nachrichten
Result of TestContext

TestContext Result

TestContext: TCon_PackagingMachine

Wednesday, March 02, 2011 16:18:41

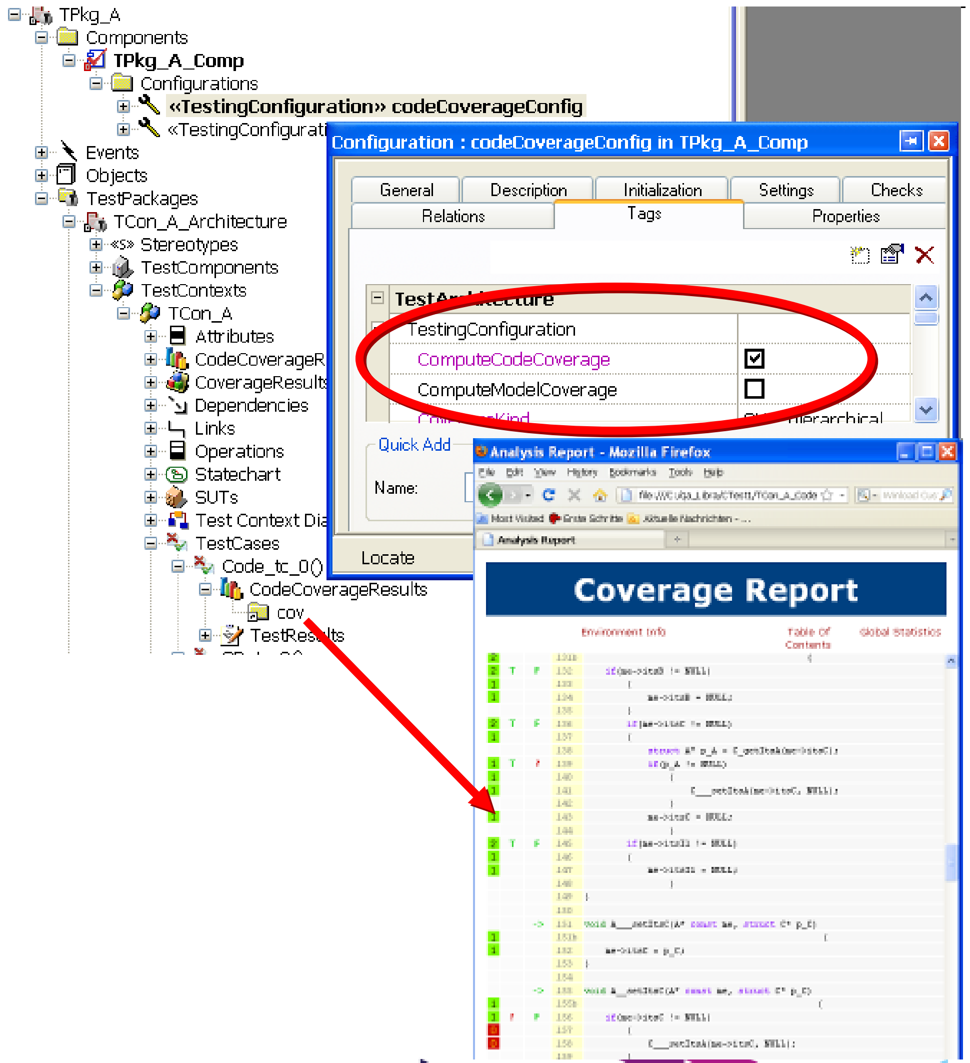
Environment Information	
Test executed on machine:	TSV
Test executed by user:	User
Used operating system version:	Windows 2000 / Windows XP
Used Rhapsody version:	Libra, build 2045402
Used TestConductor version:	2.4.4, build 2446

Tested Project	
Project:	PackagingMachine
Active Code Generation Component:	TPkg_PackagingMachine_2_Comp
Active Code Generation Configuration:	Release

TestContext: TCon_PackagingMachine	Summary: FAILED
SD_tc_1	PASSED
SD_tc_0	PASSED
SD_tc_2	PASSED
SD_tc_3	FAILED
SD_tc_4	FAILED
SD_tc_5	FAILED

Code coverage computation

- Computes code coverage for individual test cases and complete test contexts
  - Statement, Condition/Decision(CD), Modified Condition/Decision (MCDC) coverage
- In particular needed when using Rhapsody for safety critical development according to safety standards, e.g. IEC 61508, ISO/DIS 26262, DO-178B/C
  - In safety critical users have to demonstrate that all requirements are successfully tested and that the complete code has been tested



```
TPkg_A
  Components
    TPkg_A_Comp
      Configurations
        «TestingConfiguration» codeCoverageConfig
        «TestingConfigurati
  Events
  Objects
  TestPackages
    TCon_A_Architecture
      «S» Stereotypes
      TestComponents
      TestContexts
        TCon_A
          Attributes
          CodeCoverageR
          CoverageResults
          Dependencies
          Links
          Operations
          Statechart
          SUTs
          Test Context Dia
          TestCases
            Code_tc_0()
              CodeCoverageResults
                cov
              TestResults
```

Configuration : codeCoverageConfig in TPkg_A_Comp

General | Description | Initialization | Settings | Checks
Relations | Tags | Properties

TestArchitecture	
TestingConfiguration	
ComputeCodeCoverage	☑
ComputeModelCoverage	☐
	Hierarchical

Quick Add

Name:

Locate

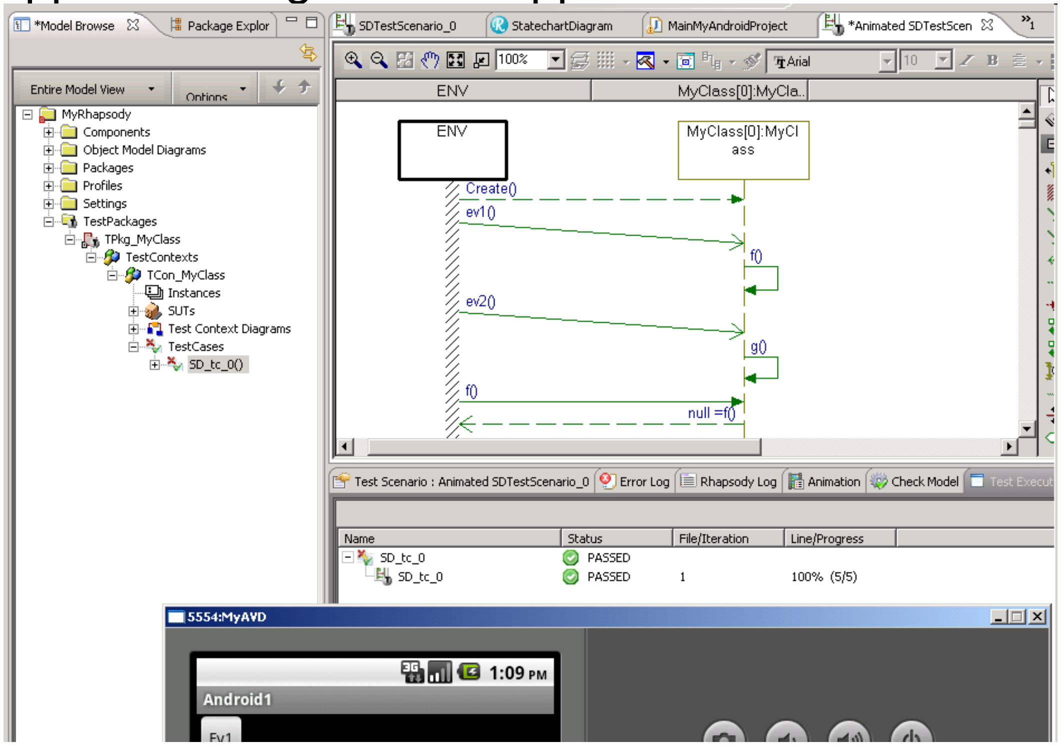
Analysis Report - Mozilla Firefox

Coverage Report

Environment Info | Table of Contents | Global Statistics

Testing support for Android applications

- Rhapsody can generate and animate android applications
- TestConductor now supports testing Android applications



Name	Status	File/Iteration	Line/Progress
SD_tc_0	PASSED		
SD_tc_0	PASSED	1	100% (5/5)

Documentation Enhancements

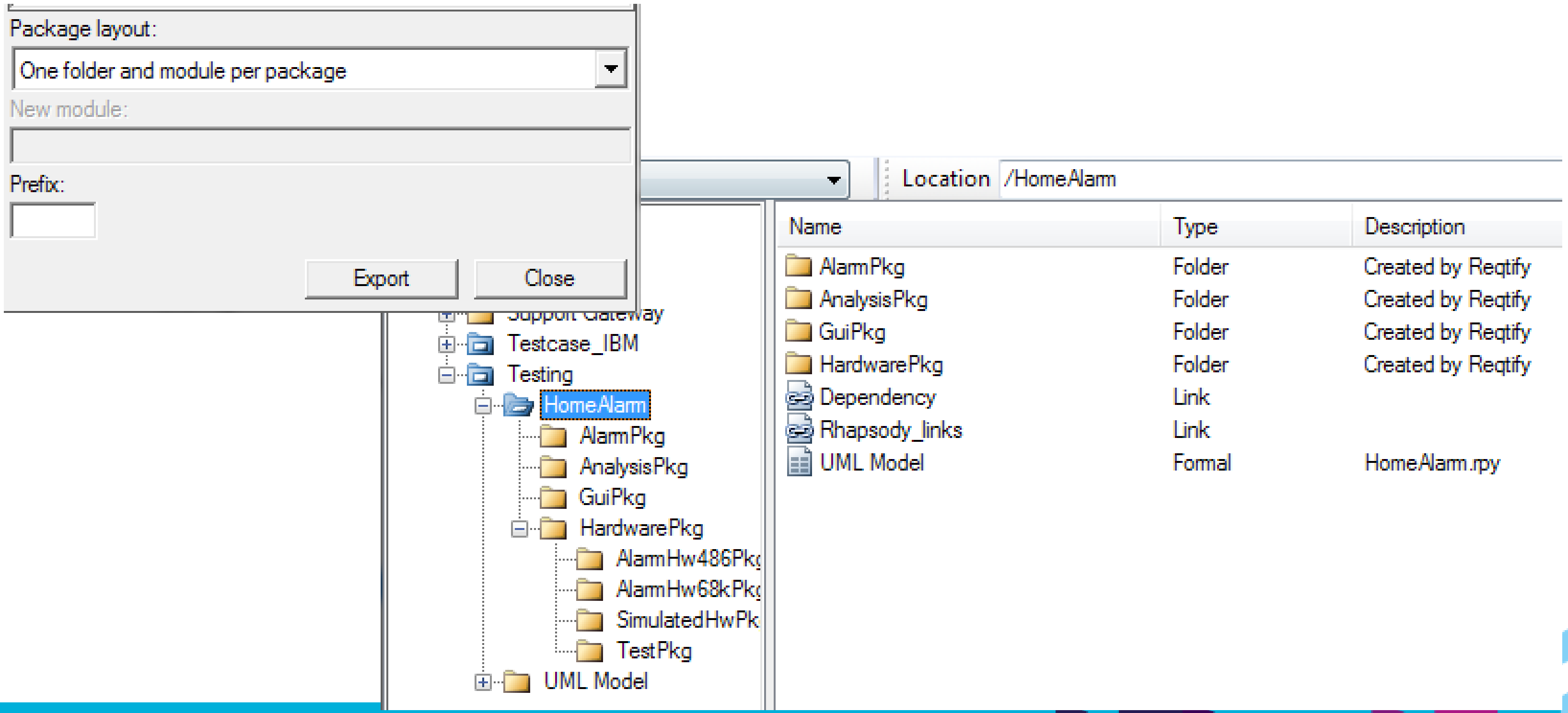
- Move product documentation to the web to reduce installation size and to allow continuous update
  - For Rhapsody user with **no limitations on Internet access** on the computer they are using Rhapsody on, who doesn't want to have the help on their computer at all, and **will access the web-based IC** each time they want the help.
  - Support users **with limitations on Internet access**
  - Support **download the IC to local server/computer**
- Properties without documentation
- Rhapsody API documentation

Rhapsody Gateway Enhancements

- Improving RTF synchronization to Rhapsody
- Export to DOORS improvements
- Additional improvements
    - Excel to XML, CSV to XML, XML Tester

Export to DOORS improvements

- Create Package structure as Folder
  - Package layout selection:
    - Single module (regular behavior)
    - One module per package
    - NEW: One folder and module per package



Package layout: One folder and module per package

New module:

Prefix:

Location /HomeAlarm

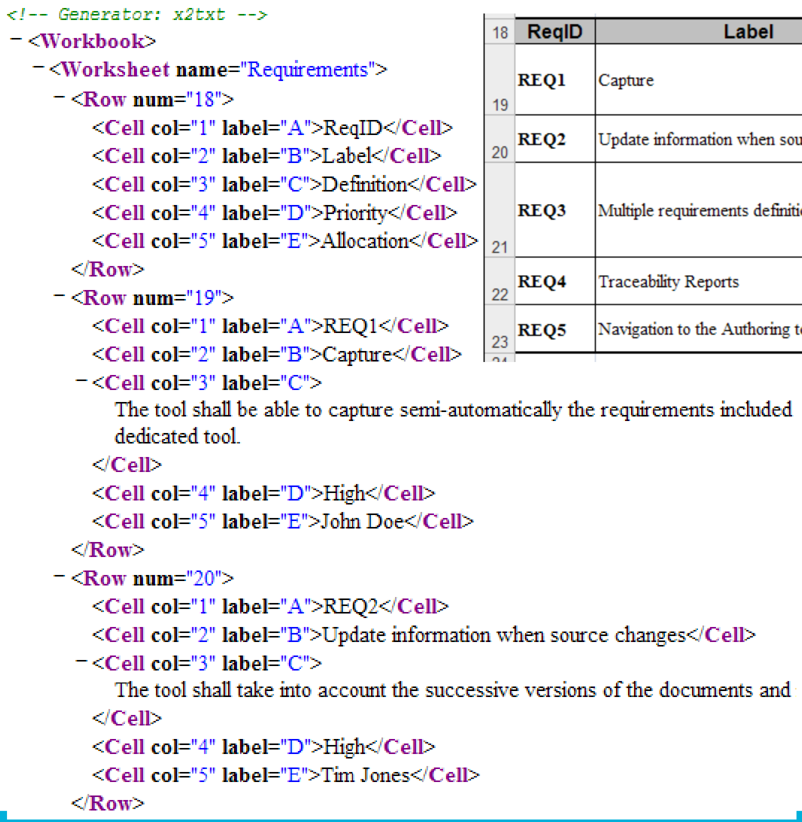
Name	Type	Description
AlarmPkg	Folder	Created by Reqtify
AnalysisPkg	Folder	Created by Reqtify
GuiPkg	Folder	Created by Reqtify
HardwarePkg	Folder	Created by Reqtify
Dependency	Link	
Rhapsody_links	Link	
UML Model	Formal	HomeAlarm.rpy

Export to DOORS improvements

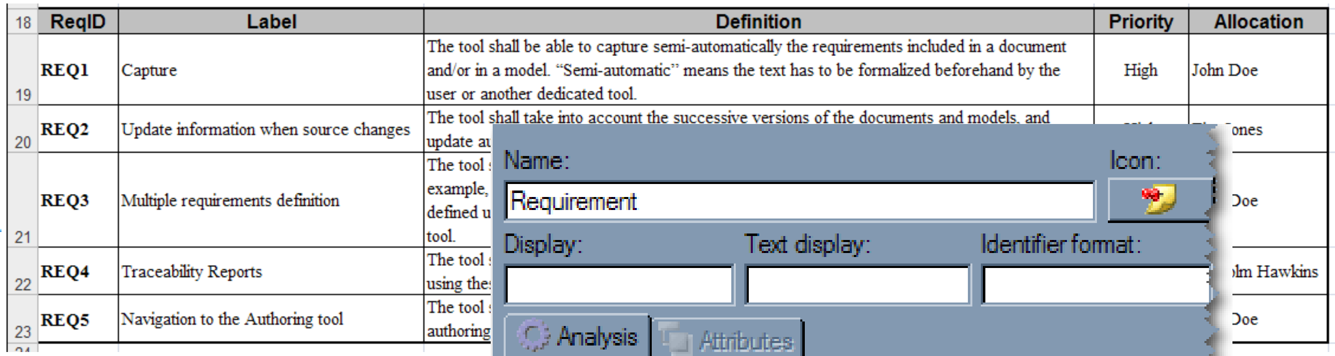
- Export reference attribute
  - Map to link attribute in DOORS (on option?)
- Customize the dependency/anchor/link module mapping
  - Solution not studied yet!
  - Need a new mapping configuration panel?

Additional improvements

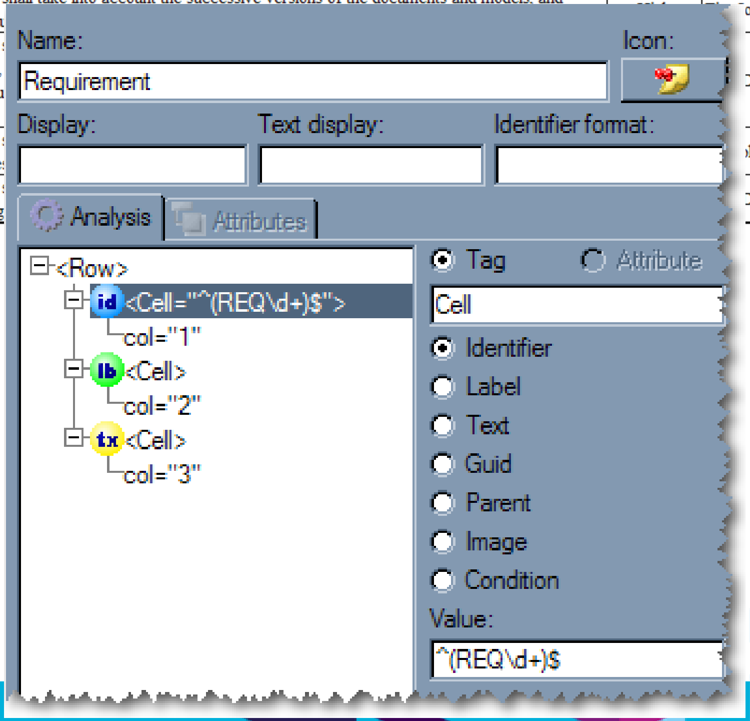
- Excel to XML capture
  - Easier pattern definition and lower the risk of erroneous extraction
  - Support Excel 2007 format (docx)



```
<!-- Generator: x2txt -->
- <Workbook>
  - <Worksheet name="Requirements">
    - <Row num="18">
        <Cell col="1" label="A">ReqID</Cell>
        <Cell col="2" label="B">Label</Cell>
        <Cell col="3" label="C">Definition</Cell>
        <Cell col="4" label="D">Priority</Cell>
        <Cell col="5" label="E">Allocation</Cell>
      </Row>
    - <Row num="19">
        <Cell col="1" label="A">REQ1</Cell>
        <Cell col="2" label="B">Capture</Cell>
      - <Cell col="3" label="C">
          The tool shall be able to capture semi-automatically the requirements included
          dedicated tool.
        </Cell>
        <Cell col="4" label="D">High</Cell>
        <Cell col="5" label="E">John Doe</Cell>
      </Row>
    - <Row num="20">
        <Cell col="1" label="A">REQ2</Cell>
        <Cell col="2" label="B">Update information when source changes</Cell>
      - <Cell col="3" label="C">
          The tool shall take into account the successive versions of the documents and
        </Cell>
        <Cell col="4" label="D">High</Cell>
        <Cell col="5" label="E">Tim Jones</Cell>
      </Row>
```



18	ReqID	Label	Definition	Priority	Allocation
19	REQ1	Capture	The tool shall be able to capture semi-automatically the requirements included in a document and/or in a model. "Semi-automatic" means the text has to be formalized beforehand by the user or another dedicated tool.	High	John Doe
20	REQ2	Update information when source changes	The tool shall take into account the successive versions of the documents and models, and update au		ones
21	REQ3	Multiple requirements definition	The tool example, defined u tool.		Doe
22	REQ4	Traceability Reports	The tool using the		lm Hawkins
23	REQ5	Navigation to the Authoring tool	The tool authoring		Doe



Name: Requirement

Icon:

Display:

Text display:

Identifier format:

```
<Row>
  id <Cell="^(REQ\d+)$">
     col="1"
  lb <Cell>
     col="2"
  tx <Cell>
     col="3"
```

[✓ Tag] Attribute

Cell

[✓ Identifier] Label Text Guid Parent Image Condition

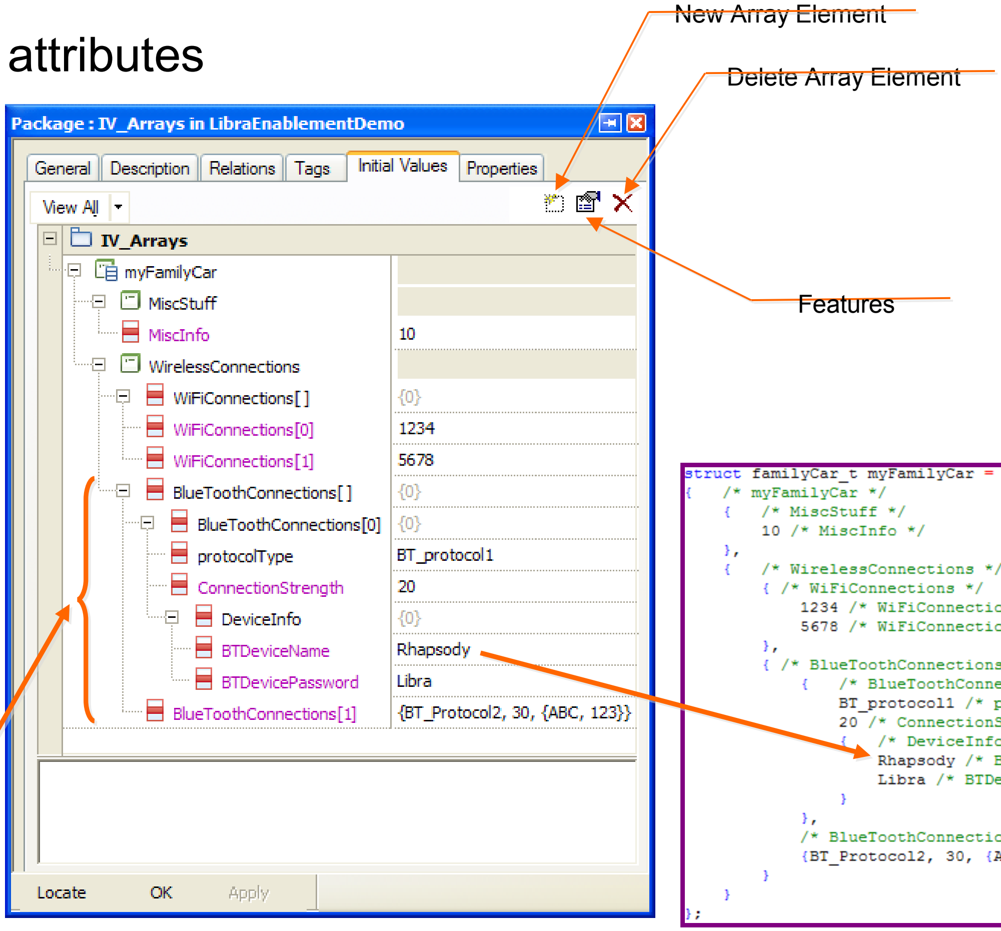
Value: ^(REQ\d+)$

Automotive/Micro C enhancements

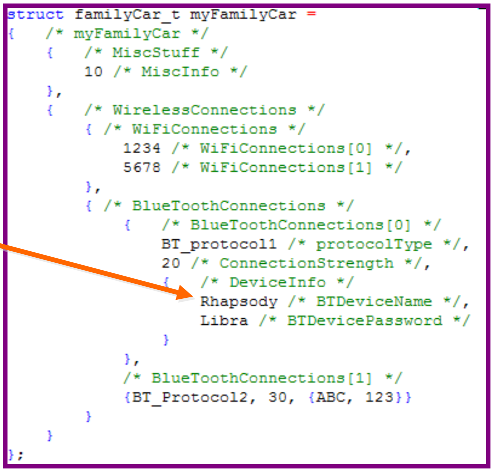
- **Enhanced Usability for MicroC and AUTOSAR**
  - Initial values capability Enhancements
  - New MicroC Checks
- **AUTOSAR code generation enhancements**
  - **Support multiple instances of AUTOSAR SW/C implementation**
  - **Automation of direct activation of triggerred operations as runnable entities**
- **AUTOSAR interchange enhancement**
  - Support ARXML roundtrip ("respect")

Enhanced Support for static initialization

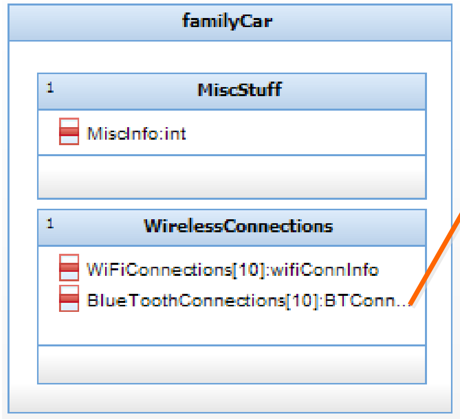
- Support statically inherited attributes
- Values for fields of record type attributes
- Array types (multiplicity > 1)
- Combinations of the above:
  - Multi Dimensioned arrays
  - Record's field of array type
  - Arrays of records



IV_Arrays	
myFamilyCar	
MiscStuff	
MiscInfo	10
WirelessConnections	
WiFiConnections[]	{0}
WiFiConnections[0]	1234
WiFiConnections[1]	5678
BlueToothConnections[]	{0}
BlueToothConnections[0]	{0}
protocolType	BT_protocol1
ConnectionStrength	20
DeviceInfo	{0}
BTDeviceName	Rhapsody
BTDevicePassword	Libra
BlueToothConnections[1]	{BT_Protocol2, 30, {ABC, 123}}

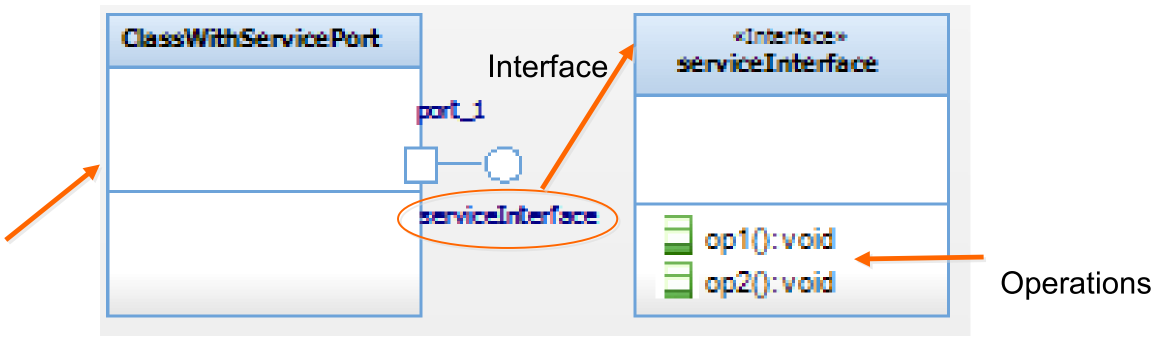


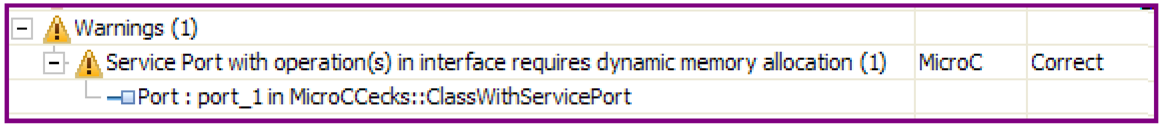
```
struct familyCar_t myFamilyCar =
{   /* myFamilyCar */
    {   /* MiscStuff */
        10 /* MiscInfo */
    },
    {   /* WirelessConnections */
        { /* WiFiConnections */
            1234 /* WiFiConnections[0] */,
            5678 /* WiFiConnections[1] */
        },
        { /* BlueToothConnections */
            {   /* BlueToothConnections[0] */
                BT_protocol1 /* protocolType */,
                20 /* ConnectionStrength */,
                {   /* DeviceInfo */
                    Rhapsody /* BTDeviceName */,
                    Libra /* BTDevicePassword */
                }
            },
            /* BlueToothConnections[1] */
            {BT_Protocol2, 30, {ABC, 123}}
        }
    }
};
```


familyCar
1
MiscStuff
MiscInfo:int
1
WirelessConnections
WiFiConnections[10]:wifiConnInfo
BlueToothConnections[10]:BTConn...

New MicroC Checks

- Object with multiplicity > 1 owning Segmented Memory Attribute(s)
- Class Part with multiplicity > 1 owning Segmented Memory Attribute(s)
- Service Port with operation(s) in interface

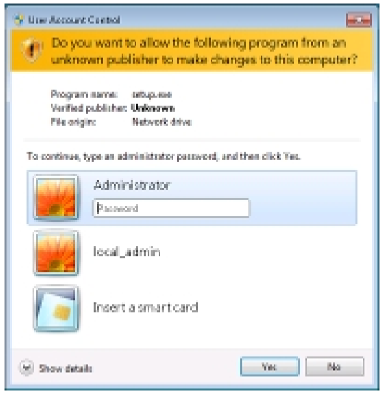




```
⊟ ⚠ Warnings (1)
  ⊟ ⚠ Service Port with operation(s) in interface requires dynamic memory allocation (1)   MicroC   Correct
      └─□ Port : port_1 in MicroCCecks::ClassWithServicePort
```

Rhapsody Install enhancements

- Ability to install Rhapsody on Windows 7 machines for "elevated" user
- Simplify install by reducing the resolution of features and components
- Improve Install/Uninstall performance
- Add JRE as a part of Rhapsody install


User Account Control

Do you want to allow the following program from an unknown publisher to make changes to this computer?

Program name: setup.exe
Verified publisher: **Unknown**
File origin: Network drive

To continue, type an administrator password, and then click Yes.

Administrator
Password

local_admin

Insert a smart card

www.ibm.com/software/rational

www.ibm.com/software/rational

Rational Rhapsody Design Manager

Collaborative Design Management

Engaging teams around the lifecycle

***Engage** all stakeholders in the design process*

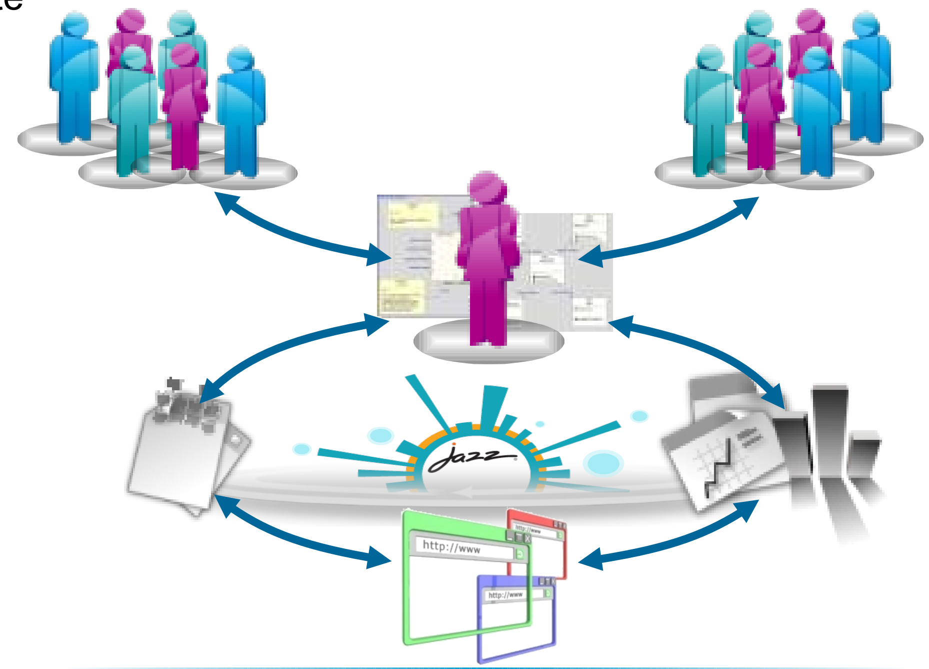
- ✓ Stakeholders better understand how their efforts relate to the overall design
- ✓ Quality is improved through automated reviews and team-wide trace analysis

***Unify** designs across domains and supply chains*

- ✓ Knowledge transfer increased through a system-wide design repository
- ✓ Real and potential impact of design changes easily analyzed and understood

***Accelerate** project delivery*

- ✓ Decision-making sped up through readily accessible information
- ✓ Reduced iteration times through direct stakeholder involvement in designs
- ✓ Regulatory demands satisfied with design process traceability and multi-discipline reporting



Collaborative Design Management

Enhance cross-team collaboration on software and systems design

NEW!

Central Design Hub

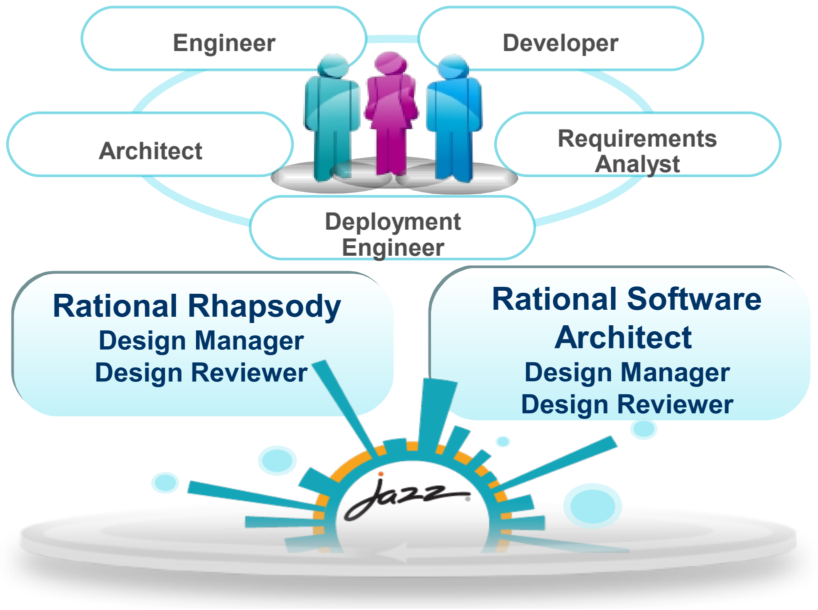
- ✓ *Enterprise-wide design storage for search, review, analysis, and reuse*
- ✓ *Links design elements to lifecycle artifacts*
- ✓ *Navigate and visualize relationships*

Stakeholder Collaboration

- ✓ *Automated design reviews at all stages of development*
- ✓ *Intuitive extended team web client for broader access to designs*

Document Generation and Reporting

- ✓ *Create documents directly from the development lifecycle*
- ✓ *Draw from information and assets linked through OSLC*



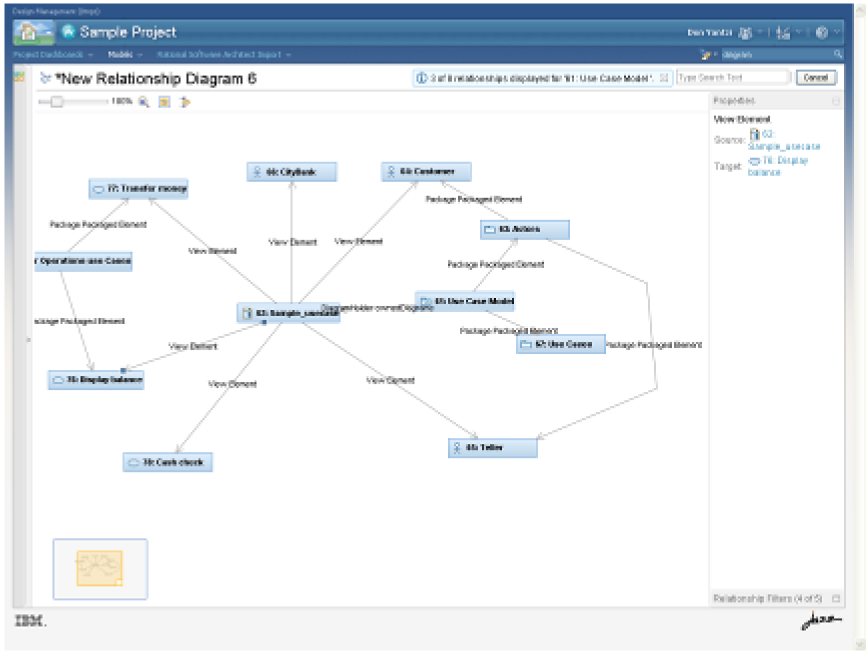
"The ability to review and comment on models from the Web client encourages feedback from a wide array of stakeholders... leading to faster consensus and improved quality of solution designs."

– Lars Tufvesson, Sellegi

Design Server

Maximize productivity and lower costs

- ✓ *Increase team knowledge through an enterprise and system-wide repository with Web-based access*
- ✓ *Leverage Jazz to quickly search across designs for review, analysis and potential reuse*
- ✓ *Analyze the impact of design changes*

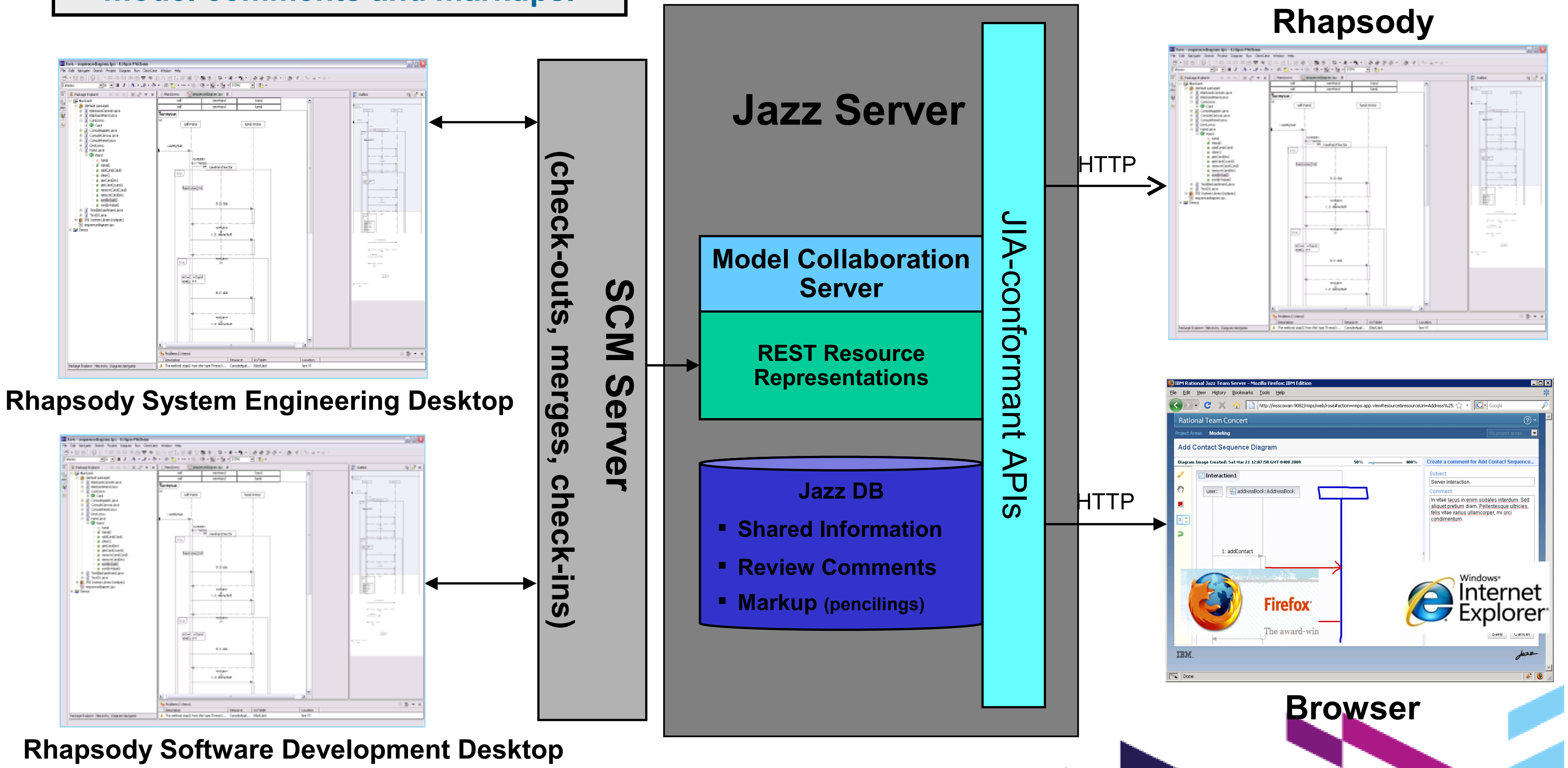


- **Teams need to quickly find existing designs to review, analyze, and identify reuse, but…**
  - × Information may be stored in multiple designs or models
  - × Desktop client installation is required to view the design information
  - × Access to SCM system also may be required to access the information
- **With Collaborative Design Management…**
  - All designs for an organization or system can be stored and accessed from a central location
  - All known designs can be searched, viewed, analyzed on the server from Rhapsody, RSA or Web client
  - The new *Relationship diagram* supports impact analysis and discovery of related design elements and resources

Web Collaboration on published models

Engineers & developers create/edit models using desktop tools with SCM system of choice publishing to model collaboration server and review model comments and markups.

Web access for stakeholders to search, query, & comment on the models, to link to requirements and to perform analysis and reporting.

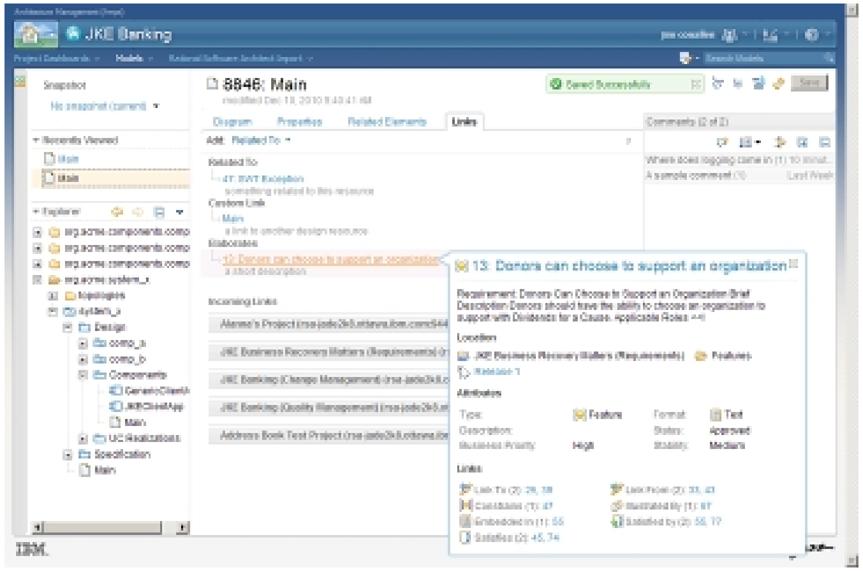

Rhapsody
Rhapsody System Engineering Desktop
Rhapsody Software Development Desktop
SCM Server
(check-outs, merges, check-ins)
Jazz Server
Model Collaboration Server
REST Resource Representations
Jazz DB
Shared Information
Review Comments
Markup (pencilings)
JIA-conformant APIs
HTTP
HTTP
Browser
Firefox
Windows Internet Explorer
Rational Team Concert
Add Contact Sequence Diagram

Stakeholder Collaboration

Easily share software architectures, deployment plans and system designs

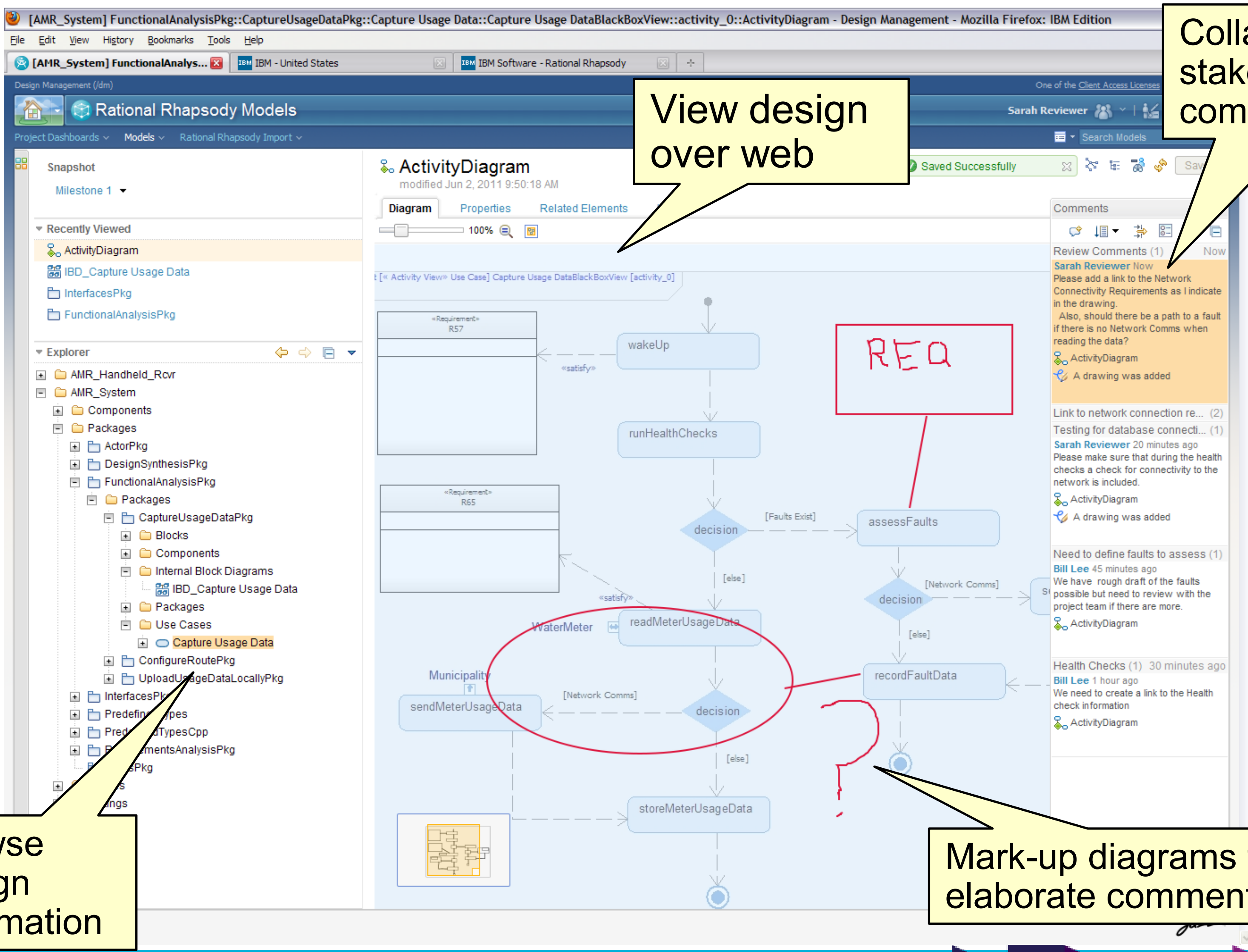
✓ *Improve quality by enabling the extended team to easily access and review designs and trace analysis*

✓ *Keep stakeholders informed on how their work relates to designs*



- **Teams need to collaborate on designs and incorporate design into the workflow, but…**
  - × Stakeholders cannot easily access the latest design information
  - × It's not clear to stakeholders which designs are related to their work
  - × Design dependencies are mismanaged, resulting in data duplication and inefficient workflows
- **With Rational Rhapsody Design Manager…**
  - Stakeholders have self-serve access to design milestones, improving collaboration and quality of designs
  - Stakeholders can determine how their task relates to designs with traceable links to work items, requirements and test cases

Rational Rhapsody Design Manager Web Client



Review Comments (1) Now

Sarah Reviewer Now
Please add a link to the Network Connectivity Requirements as I indicate in the drawing.
Also, should there be a path to a fault if there is no Network Comms when reading the data?
ActivityDiagram
A drawing was added

Link to network connection re... (2)

Testing for database connecti... (1)

Sarah Reviewer 20 minutes ago
Please make sure that during the health checks a check for connectivity to the network is included.
ActivityDiagram
A drawing was added

Need to define faults to assess (1)

Bill Lee 45 minutes ago
We have rough draft of the faults possible but need to review with the project team if there are more.
ActivityDiagram

Health Checks (1) 30 minutes ago

Bill Lee 1 hour ago
We need to create a link to the Health check information
ActivityDiagram

View design over web

Collaborate with stakeholders with commenting

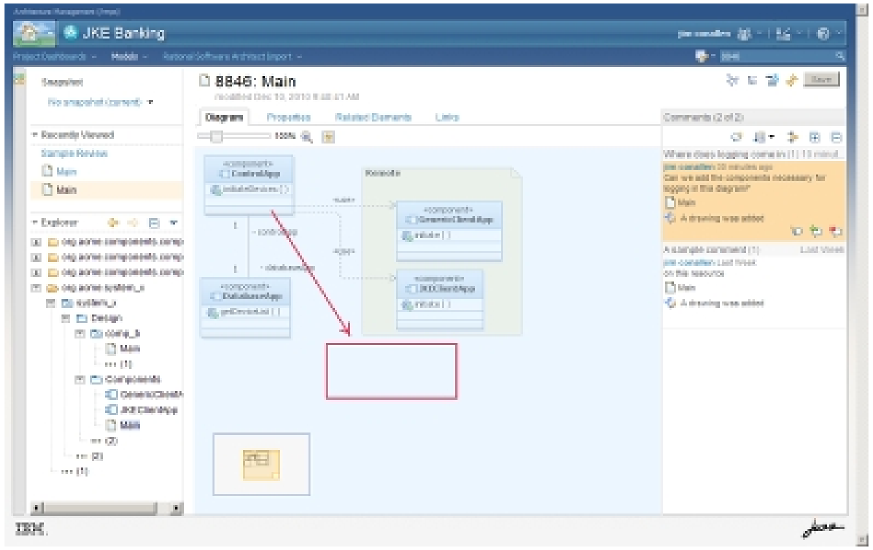
Browse design information

Mark-up diagrams to elaborate comments

Faster Design Reviews

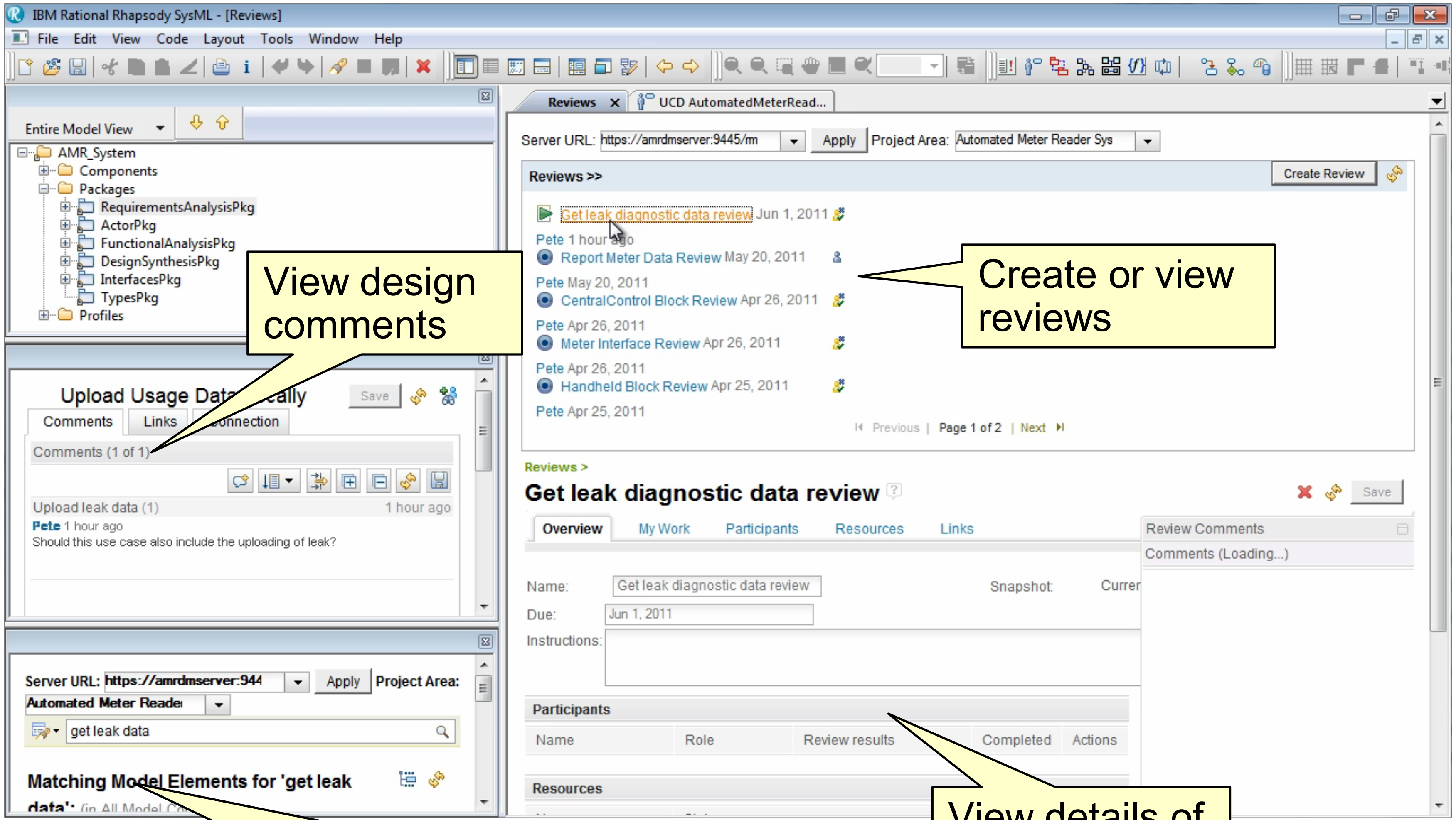
Shorten time-to-market and improve quality

- ✓ *Speed up decision-making by keeping people informed as decisions are made*
- ✓ *Improve quality by ensuring that the extended team has direct input into updates and corrections*
- ✓ *Automate the design review process*



- **Teams need to collaborate with stakeholders on software architectures, deployment plans, and system designs, but..**
  - × Design reviews are painful, tedious, and time consuming
  - × Stakeholders do not have direct access to designs so reviews need to be handled outside of the design tools; feedback is difficult to communicate and confirm
- **With Collaborative Design Management…**
  - Designers automate reviews, specifying which designs and stakeholders participate
  - Stakeholders can view the design and attach comments and mark-up via the Web or rich client
  - Design reviews can be linked to RTC work items for planning and tracking

Collaborative development in Rhapsody client


IBM Rational Rhapsody SysML - [Reviews]
File Edit View Code Layout Tools Window Help
Entire Model View
Reviews
UCD AutomatedMeterRead...
Create Review
Save
Comments
Links
Connection
Overview
My Work
Participants
Resources
Links
Previous | Page 1 of 2 | Next

AMR_System
- Components
- Packages
  - RequirementsAnalysisPkg
  - ActorPkg
  - FunctionalAnalysisPkg
  - DesignSynthesisPkg
  - InterfacesPkg
  - TypesPkg
- Profiles

Upload Usage Data ...ally

Comments (1 of 1)

Upload leak data (1) 1 hour ago
Pete 1 hour ago
Should this use case also include the uploading of leak?

Server URL: https://amrdmserver:944 Apply Project Area: Automated Meter Reade
get leak data

Matching Model Elements for 'get leak data': (in All Model C...

Server URL: https://amrdmserver:9445/rm Apply Project Area: Automated Meter Reader Sys

Reviews >>

Get leak diagnostic data review Jun 1, 2011
Pete 1 hour ago
Report Meter Data Review May 20, 2011
Pete May 20, 2011
CentralControl Block Review Apr 26, 2011
Pete Apr 26, 2011
Meter Interface Review Apr 26, 2011
Pete Apr 26, 2011
Handheld Block Review Apr 25, 2011
Pete Apr 25, 2011

Reviews >

Get leak diagnostic data review

Name: Get leak diagnostic data review
Due: Jun 1, 2011
Instructions:
Snapshot: Curren

Review Comments
Comments (Loading...)

Participants

Name	Role	Review results	Completed	Actions

Resources

View design comments

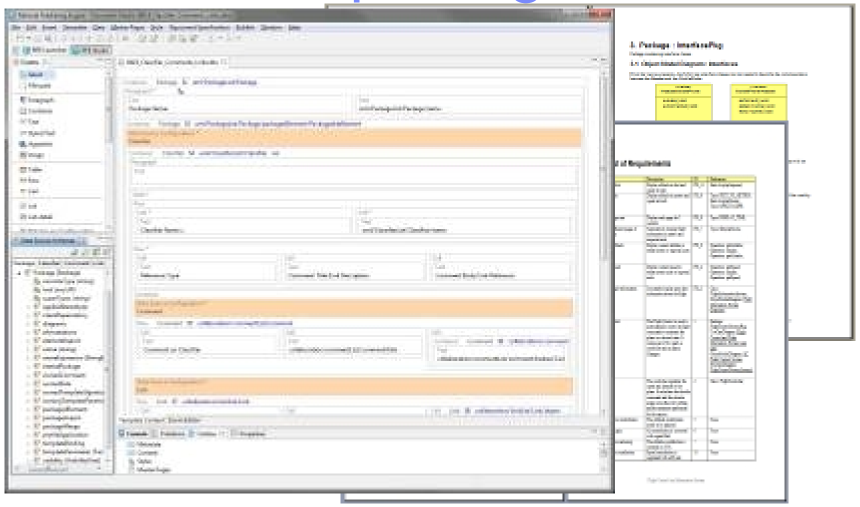
Create or view reviews

Search across design projects

View details of design review

Multi-discipline Document Generation and Reporting

Satisfy regulatory and customer demands

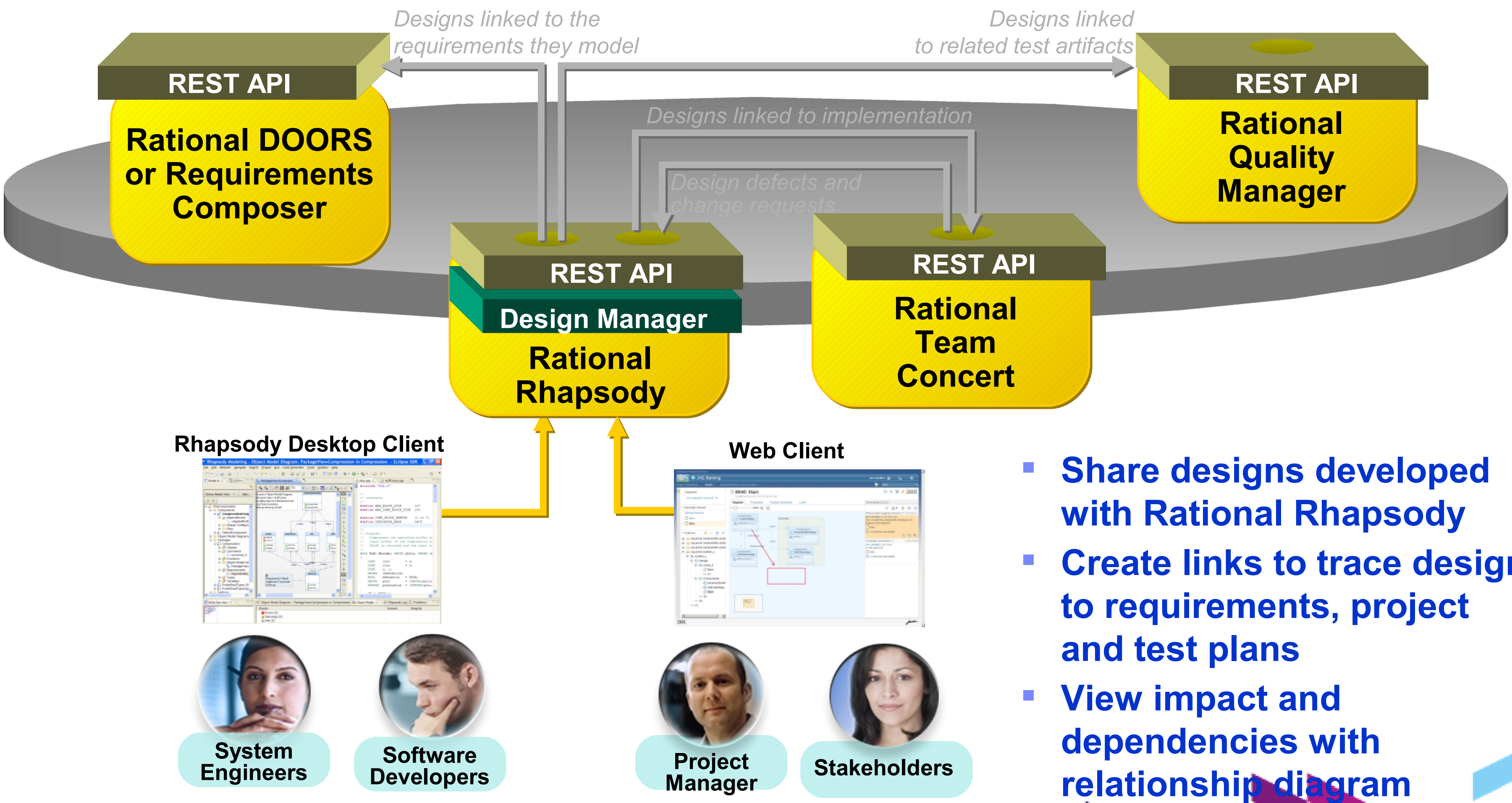


- ✓ *Easily create comprehensive documentation for specifications, communication, compliance and auditing*
- ✓ *Help prove compliance by including OSLC link information and design review information*

- **Teams must document for specifications, communication, regulatory compliance, and auditing, but…**
  - ✗ Documents need to contain information from different domains (requirements, design, change management); individual products have separate reporting solutions
  - ✗ Existing solutions are difficult to use for designs and limited in the types of information they can access
- **With Rhapsody Design Manager and Rational Publishing Engine…**
  - Users create templates and generate documents and reports pulling data from all relevant sources using open interfaces
  - Documents can show the impact of design changes on other lifecycle resources leveraging the OSLC linking data
  - Reports can include comments and details from design reviews

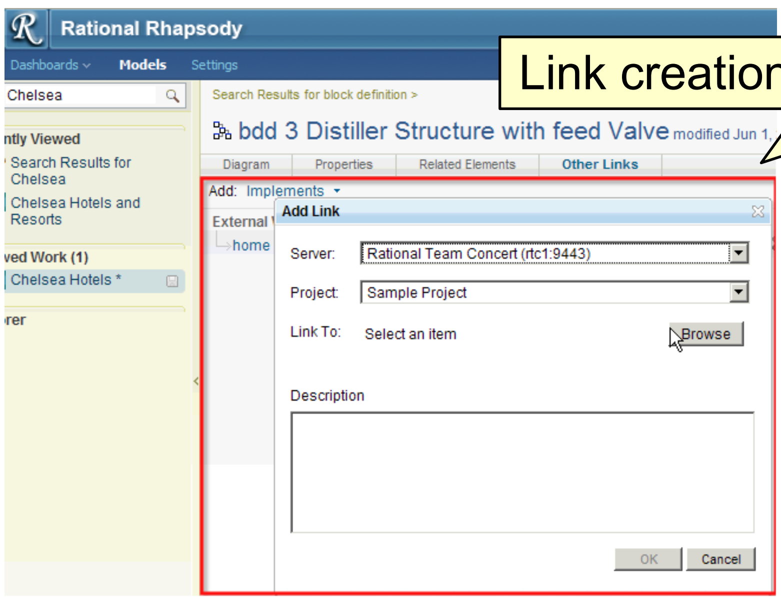
Collaborative Design Management

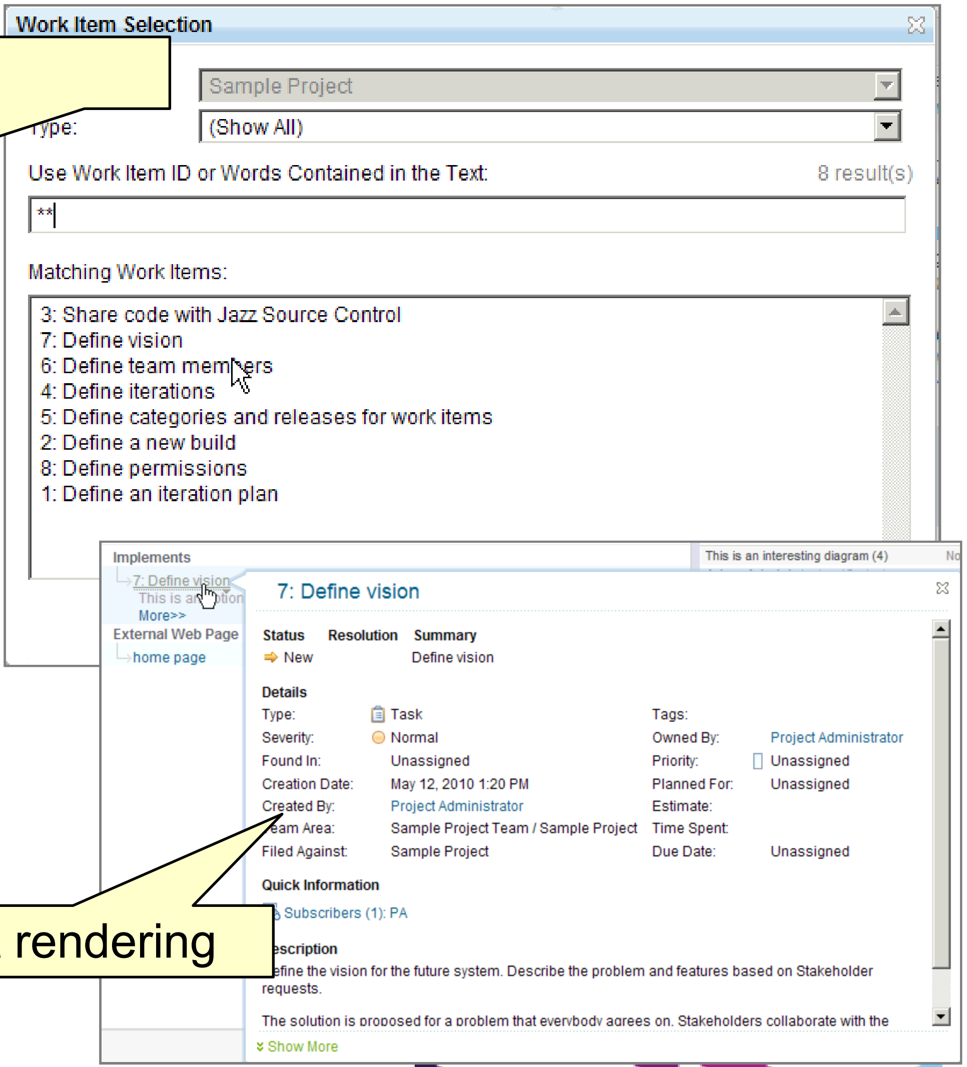
Integrating Systems Engineering and Embedded Software Design into Jazz



- **Share designs developed with Rational Rhapsody**
- **Create links to trace design to requirements, project and test plans**
- **View impact and dependencies with relationship diagram**
- **Create custom link types**

Design Manager links to Team Concert





3: Share code with Jazz Source Control
7: Define vision
6: Define team members
4: Define iterations
5: Define categories and releases for work items
2: Define a new build
8: Define permissions
1: Define an iteration plan

7: Define vision

Status	Resolution	Summary
New		Define vision

Details

Type:	Task	Tags:	
Severity:	Normal	Owned By:	Project Administrator
Found In:	Unassigned	Priority:	Unassigned
Creation Date:	May 12, 2010 1:20 PM	Planned For:	Unassigned
Created By:	Project Administrator	Estimate:	
Team Area:	Sample Project Team / Sample Project	Time Spent:	
Filed Against:	Sample Project	Due Date:	Unassigned

Quick Information
Subscribers (1): PA

Description
Define the vision for the future system. Describe the problem and features based on Stakeholder requests.

The solution is proposed for a problem that everybody agrees on. Stakeholders collaborate with the

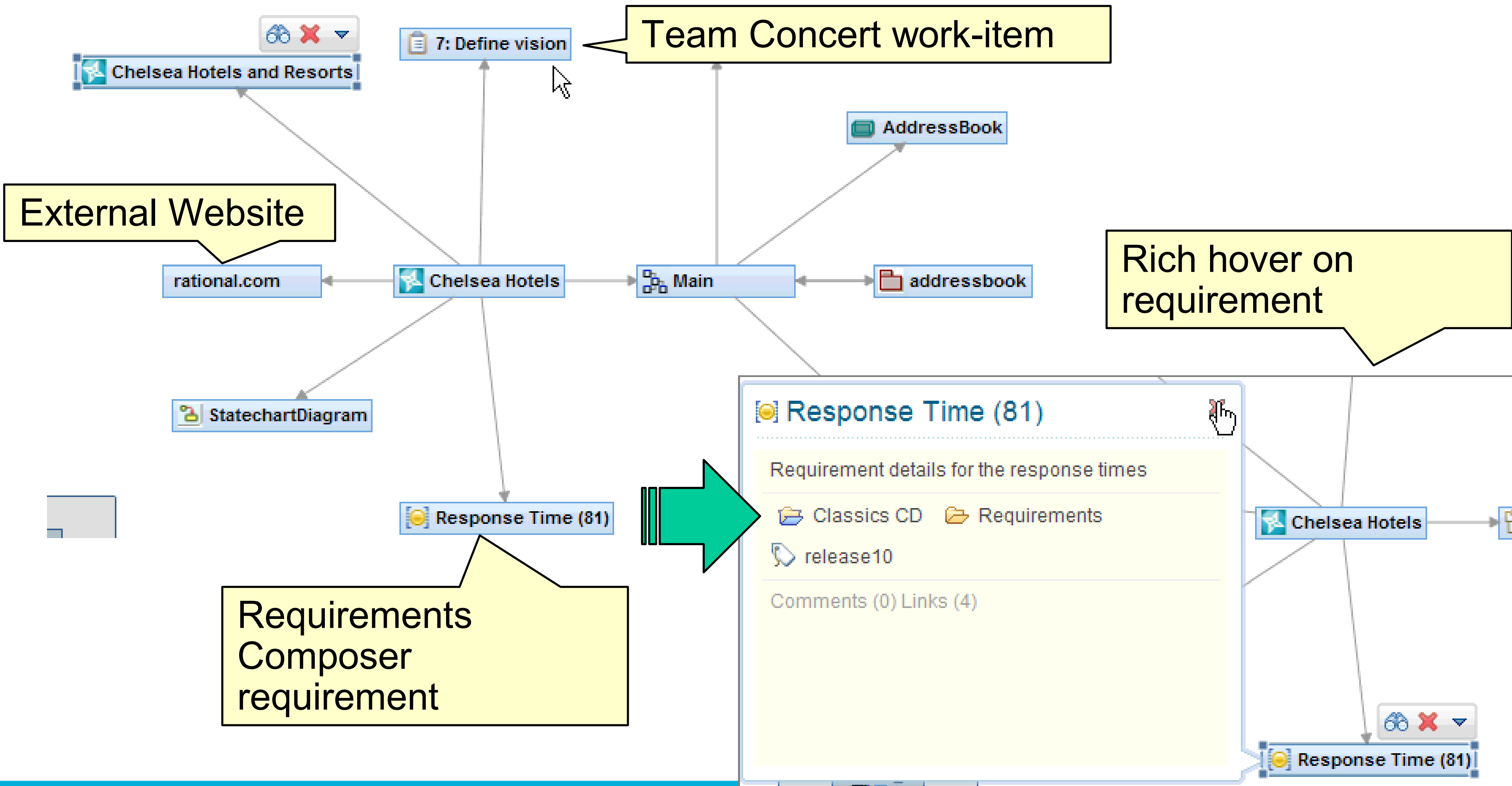
Show More

Link creation

Compact rendering

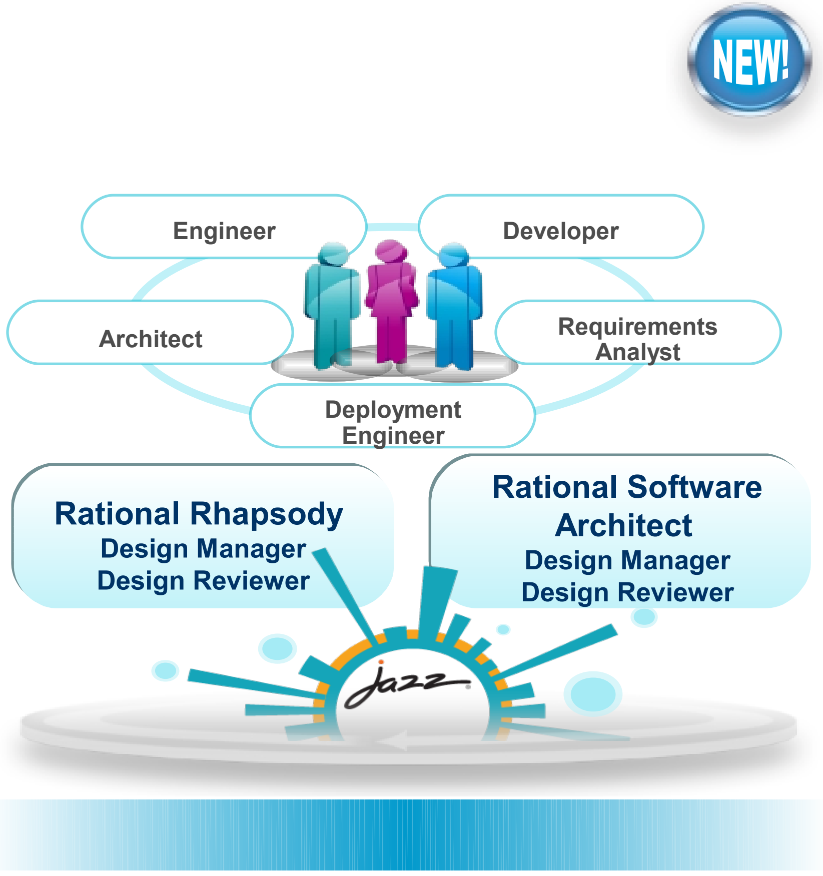
Viewing traceability links in Design Manager

- Relationship diagram shows linked elements to help impact analysis



Collaborative Design Management Offerings

- ✓ ***Rhapsody Design Manager*** *provides Collaborative Design Management for the Rational Rhapsody Family*
- ✓ *Extends Rhapsody's existing design authoring capabilities with enhanced team collaboration*
- ✓ *Connects Rhapsody into the Jazz platform, so teams can collaborate in the context of designs*
- ✓ *Team capability offered through Design Manager and Design Reviewer user roles*
- ✓ *Users access through either Web client or desktop client with Design Management client installed*



"We are excited about the capabilities in Collaborative Design Management …. We see it playing a significant role in our development process because it allows us to transition away from our home grown solutions in that area to standardized Rational products."

Hans-Peter Berger, Department Head, Application Development Infrastructure, GAD

Collaborative Design Management User Roles

Rhapsody Design Manager 3.0

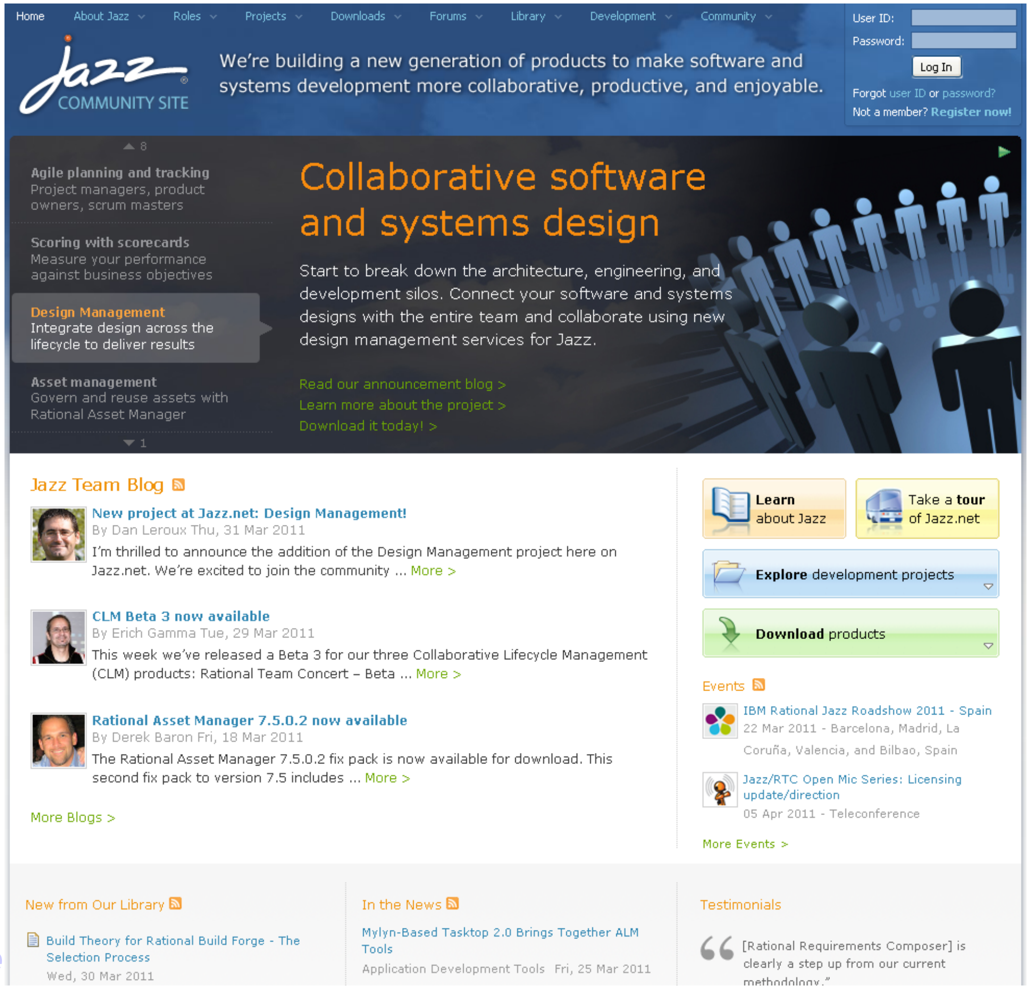
	Design Reviewer	Design Manager
	For extended team members who need to access and collaborate on models and designs	For design practitioners using Rational Rhapsody to create models and designs
View and search designs	✓	✓
Attach comments and markup	✓	✓
View designs in dependency diagram	✓	✓
View and create design links	✓	✓
Dashboards	✓	✓
Import designs directly from RSA or Rhapsody workspace		✓
Import designs from SCM		✓
Setup design reviews		✓
Create design baselines		✓

**Design Reviewer and Design Manger roles are available in both Rational Software Architect Design Manager and Rhapsody Design Manager.*

Get Involved on Jazz.net

jazz.net/projects/design-management

- Technology initiative to…
  - ✓ *Bring design management capabilities to Jazz*
  - ✓ *Provide a collection of design management services that can be used by any design tool*
  - ✓ *Involve the community in defining the services needed for design management*
- You can participate
  - ✓ *Learn more*
  - ✓ *Register on jazz.net*
  - ✓ *Download and try it out*
  - ✓ *Ask questions and give feedback*
  - ✓ *View plans and dashboards*
  - ✓ *Report defects and request enhanceme*



We're building a new generation of products to make software and systems development more collaborative, productive, and enjoyable.

Forgot user ID or password?
Not a member? Register now!

Agile planning and tracking
Project managers, product owners, scrum masters

Scoring with scorecards
Measure your performance against business objectives

Design Management
Integrate design across the lifecycle to deliver results

Asset management
Govern and reuse assets with Rational Asset Manager

Collaborative software and systems design

Start to break down the architecture, engineering, and development silos. Connect your software and systems designs with the entire team and collaborate using new design management services for Jazz.

Read our announcement blog >
Learn more about the project >
Download it today! >

Jazz Team Blog

New project at Jazz.net: Design Management!
By Dan Leroux Thu, 31 Mar 2011
I'm thrilled to announce the addition of the Design Management project here on Jazz.net. We're excited to join the community ... More >

CLM Beta 3 now available
By Erich Gamma Tue, 29 Mar 2011
This week we've released a Beta 3 for our three Collaborative Lifecycle Management (CLM) products: Rational Team Concert – Beta ... More >

Rational Asset Manager 7.5.0.2 now available
By Derek Baron Fri, 18 Mar 2011
The Rational Asset Manager 7.5.0.2 fix pack is now available for download. This second fix pack to version 7.5 includes ... More >

More Blogs >

Learn about Jazz
Take a tour of Jazz.net
Explore development projects
Download products

Events
IBM Rational Jazz Roadshow 2011 - Spain
22 Mar 2011 - Barcelona, Madrid, La Coruña, Valencia, and Bilbao, Spain
Jazz/RTC Open Mic Series: Licensing update/direction
05 Apr 2011 - Teleconference
More Events >

New from Our Library
Build Theory for Rational Build Forge - The Selection Process
Wed, 30 Mar 2011

In the News
Mylyn-Based Tasktop 2.0 Brings Together ALM Tools
Application Development Tools Fri, 25 Mar 2011

Testimonials
[Rational Requirements Composer] is clearly a step up from our current methodology."