A technical discussion of integration between IBM® Rational® RequisitePro® 2003 and IBM® Rational® XDE™ Developer v2003
June, 2003 Rev. 1.00

Rational. software

IBM

Keeping Software Designs In-line with Requirements

Integration between IBM Rational® RequisitePro® 2003 and IBM Rational® XDE™ Developer v2003

Catherine Connor
Requirements Specialist
Worldwide Technical Marketing
IBM Rational software

Contents

Introduction

This paper describes how the integration between IBM Rational® RequisitePro® 2003 and IBM Rational® XDE™ Developer v2003 helps software teams maintain design activities in compliance with requirements. The goal being to deliver software that actually solves customer needs.

Rational RequisitePro is the IBM Rational market leading requirements management tool. With Rational RequisitePro you manage requirements in Microsoft Word documents that are linked to a requirements database, where additional requirement information can be more effectively organized and managed.

Rational XDE Developer, a complete visual design and development environment, adds UML (Unified Modeling Language) modeling to developers Integrated Development Environments (IDE) to enhance communication and documentation of software design. The integration described in this paper applies to the following members of the Rational XDE Developer family:

- *Rational® XDE™ Developer - Java™ Platform Edition, can be implemented alone via the included Eclipse IDE or installed into the IBM® WebSphere™ Studio Application Developer and Integration Edition IDEs.*
- *Rational® XDE™ Developer - .NET Platform Edition, extends the Microsoft® Visual Studio™ .NET IDE.*
- *Rational® XDE™ Modeler Edition, which provides a subset of the Rational XDE Developer features to enable architects and designers to create platform-independent UML models of architecture, business needs, reusable assets, and management-level communication.*

This paper first reviews the challenges of keeping requirements and design synchronized, then delves into how the integration between Rational XDE Developer and Rational RequisitePro provides solutions to these challenges.

The second part of the paper describes in detail how to use the integration to achieve its benefits.

Background: The Challenge of Going From Requirements To Design

Requirements define what software developers are responsible for building. They document the customer needs the software will have to solve in order to provide value to customers. As such, requirements should drive all software activities needed to get to the end goal: delivering software that provides value to customers.

In practice, once the requirements are documented, software teams need to follow through and ensure that the design and coding efforts respect these requirements.

Historically requirements tools and design tools have lived in their respective silo, each on one side of the wall that often divides the business analysts from the developers. This often results in poor software development execution, where in the end, the application produced does not look like what was specified in the requirements. This common situation has root causes in poor communication of requirements to developers, and especially poor communication of requirement changes to developers. Communicating the first cut of the requirement specification to developers is not difficult, but as requirements have a high tendency to change, unless subsequent requirement changes are communicated as well, developers forge ahead with obsolete requirements, and end up delivering an application that does not satisfy the evolving of users. Additionally when building large systems, it is fairly easy for development teams to forget about some requirements. Assessing the coverage of requirements in design is essential to guarantying that the application will live up to the expectation set with customers early on in the project.

Without a way to answer, in a timely fashion, questions such as:

- *What is the latest state of the requirements?*
- *Are all the agreed upon requirements accounted for in design?*
- *What part of the design model is affected by the latest requirement changes?*
- *How do you know that your design will lead to an executable that fulfills users' needs?*

To answer the first question, the IBM Rational RequisitePro–IBM Rational XDE Developer integration solves the challenge of communicating requirements effectively to developers by connecting use-case diagrams stored in Rational XDE Developer models with their respective use case requirement information (flow of events, priority, difficulty, risk, etc.) stored in Rational RequisitePro.

To answer the two other key questions, the Rational RequisitePro-Rational XDE Developer integration solves the challenge of maintaining software designs in-line with requirements by establishing direct relationships between the requirements and the associated design elements that are created to implement this requirement.

Going From Requirements to Design with IBM Rational Tools

To explain the value of the integration between IBM Rational RequisitePro and IBM Rational XDE Developer, this section positions this combination of tools in the daily work of a software team that follows the software development best practices described in the IBM Rational Unified Process®. In italic text, we list the specific value that, respectively, Rational RequisitePro, Rational XDE Developer, and the integration between the two, brings to that software activity.

Ensuring Software Creates Value to Customers

At the start of a project, a vision document (typically written by an analyst, but widely circulated for team and customer approval) describes the customer problem and the proposed solution to that problem. High-level product features in the vision document describe the proposed solution.

Rational RequisitePro manages the vision document and the high-level product features, as well as detailed requirements that will be derive from these product features and additional elicitation work with stakeholders.

In Rational RequisitePro, relationships (called 'traceability links') between requirements are maintained to provide coverage reports (to ensure all higher level requirements are fulfilled by more detailed requirements) and impact analysis reports (to measure the impact of a requirement change to other requirements)

Once the vision is agreed upon, to keep the software team focus on customer value, a use case diagram describes the value of the application from a user perspective. Use cases describe a system's behavior when interacting with its users and other systems. In the past decade, use cases have been proven to be an effective way to document system functionality from a user's perspective, a perspective that provides both software teams and their customers a common understanding of the expected behavior of the system to build. By minimizing the risk of misunderstanding, use cases improve the chances a software team has to deliver a successful system.

Use case diagrams are created in Rational XDE Developer as well as actors and use case brief descriptions.

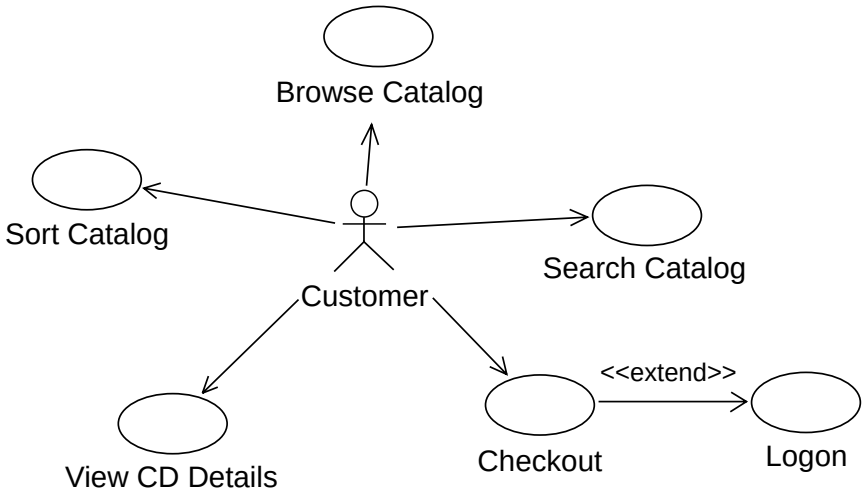

Figure 1: A typical use case diagram

Prioritizing Use Cases

Once the team and customer agree upon the value described in the use case diagram, a use case prioritization takes place to rank use cases so that the development team can focus on the most important use cases first. Most important use cases are those that automate core business processes and drive the software architecture.

To add objectivity to the prioritization process, you attach attributes to each use case. Especially helpful in iterative software development, attributes provide an easy way to scope manage each iteration of your project. Typical use case attributes include: difficulty to implement, architecture-significance, risk, and specific customer value.

From IBM Rational XDE Developer, using the Rational XDE Developer – Rational RequisitePro integration, you assign requirement properties to use cases. These requirement properties serve as use case attributes.

IBM Rational RequisitePro provides default use case requirements attributes.

In Rational RequisitePro you create use case reports that sort and filter use cases based on their attributes.

Detailing Prioritized Use Cases

To implement the high priority use cases, developers will create sequence diagrams to start building the software architecture and identify design classes. In order to create sequence diagrams, developers need to have access to more details than just the use case oval in the use case diagram. Therefore, each high priority use case is further detailed into a set of steps that describes the dialog between the actor (Customer in Figure 1) who typically initiates the use case, and the system accomplishing the functionality described by the use case (e.g. Sort Catalog in Figure 1). That set of steps is often referred to as the flow of events of the use case. Typical use cases have a single basic flow of events (describing an expected user interaction with the system) and several alternate, sometimes called exceptional, flow of events (that record how the system should handle unusual events, like missing paper in a printer). Note that at this point, there is no need to detail every single use case; only those selected as high priority use cases.

Because the flow of events in a use case is most often documented in Microsoft® Word, and IBM Rational RequisitePro allows its users to edit requirements in Microsoft Word documents, Rational RequisitePro is an ideal requirements management tool to manage use cases. Detailing use cases in Rational RequisitePro documents provides key advantages over attaching a simple Word document to a use case in Rational XDE Developer:

- *Functional requirements included in use case flow of events text are clearly identified*

Because in Rational RequisitePro requirement text is visually differentiated from additional descriptive information in the document (See Figure 9), functional requirements expressed in the use case flow of events are easy to identify. You can choose to mark functional requirements at the use case flow level, or at the individual flow step level. Marking requirements at the flow level facilitates the creation of test cases from use cases, as a test scenario typically map to a combination of basic flow steps and an alternative flow.

- *Any modification to use-case documents is automatically tracked*

An audit trail (who, what, when, and why) of every requirement change is stored in the Rational RequisitePro database. These revisions help you gain control of use case changes.

- *Functional requirements are tracked*

Functional requirements included in the use case flow of events are marked as software requirements, assigned requirement attributes (priority, difficulty, risk, etc.) and link to higher level business requirements or product features.

- *Requirements in use-case documents can be linked to other requirements they relate to*

By tracing use cases to business requirements, or product features, you can more easily measure the impact of change on related requirements and verify coverage.

Generating Design Classes from Detailed Use Cases

From the use case specification flow of events, designers build sequence diagrams to express flow of events as a succession of messages between objects.

- *Sequence diagrams are created in IBM Rational XDE Developer, and linked with Rational XDE Developer notes to the use case diagrams they represent. Note: we often get asked whether tools can automate the transformation from use case flow of events to sequence diagrams. As good as it may sound, poor designs would likely derive from such approach, as good designs are optimized to represent all critical flow of events.*

From the objects identified in sequence diagrams, class designs emerge.

- *Class diagrams and other UML diagrams are created in Rational XDE Developer, and linked to their originating sequence diagrams. The collection of UML diagrams that represent the design for a use case is often referred to as 'use case realization'. This name represents the fact that the design is realizing (aka implementing) the requirements in the use case.*

Maintaining Design Compliance to Requirement Changes

While design activities are taking place, requirements that drove the creation of the original use cases (and subsequently sequence diagrams and classes) change. Changing requirements is a fact of (the software) life and a sign of

(the software) life and a sign of healthy projects. Requirements and customer needs can never be 100% known at the start of the project, so change reflects the convergence to final requirements from the initial set of requirements with which you had to get started (if you wait to get all requirements, you never get started and end up in analysis paralysis). To ensure the delivered software does satisfy its users, designs need to be kept in sync with changing requirements.

IBM Rational RequisitePro traceability matrices link requirements to other requirements.

With the Rational RequisitePro–Rational XDE Developer integration you also link key design elements to requirements they implement. The IBM Rational XDE Developer design element is represented in Rational RequisitePro traceability matrices.

Traceability matrices between requirements and design can be filtered to access the specific information you need.

Using The IBM Rational XDE Developer – Rational RequisitePro Integration

This section details the steps to take in Rational XDE Developer and Rational RequisitePro to exercise the integration. Screenshots are provided with Rational XDE Developer - Java™ Platform Edition, but as mentioned in the introduction, this integration is available for Rational XDE Developer - .NET Edition and Rational XDE Modeler Edition as well.

In Rational XDE Developer, there are two places from which to invoke the integration: the Tools Ë Rational RequisitePro menu and the context sensitive right-click menu, when a Rational XDE Developer element is selected, either from a diagram or from the Model Explorer. The context-sensitive menu options vary depending on whether the selected Rational XDE Developer element is a package, a use case or another UML element:

Menu options for use cases:

- ***Open / New Use Case Document*** *to create a new use case document or associate the use case with an existing IBM Rational RequisitePro use-case document*
- ***View Requirement Properties*** *to view and edit attributes and traceability (dependency) links to the use case*

Menu options for design elements (classes, actors, diagrams, operations):

- ***Add / View / Remove Traceability*** *to establish, view and remove a link to requirements from the selected design element*

Menu options for packages:

- ***Associate / Disassociate to RequisitePro*** *project to specify the Rational RequisitePro project that all elements stored in this package will be integrated with.*

Setting Up The Integration

By default IBM Rational XDE Developer models are not enabled for this integration. The integration is enabled in the Rational XDE Developer model by including the 'RequisitePro' profile in the model AppliedProfiles property.

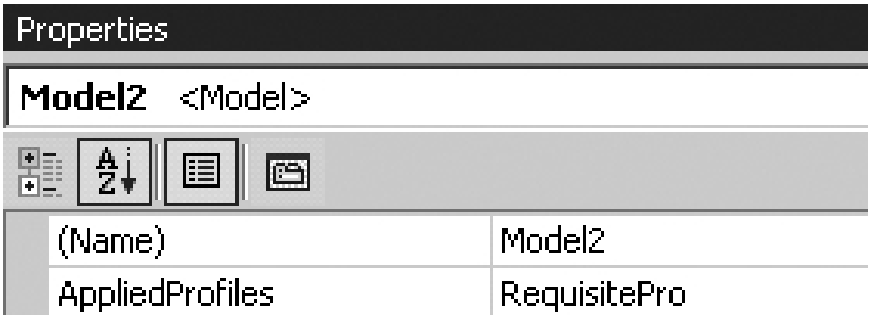

Figure 2: Enabling the integration in Rational XDE Developer

Once the 'RequisitePro' profile is included, every time you open the IBM Rational XDE Developer model, requirement management capabilities are available from the main Rational XDE Developer menu (Figure 3), as well as from the contextual right-click menu, available from the Model Explorer or from the use-case diagram.

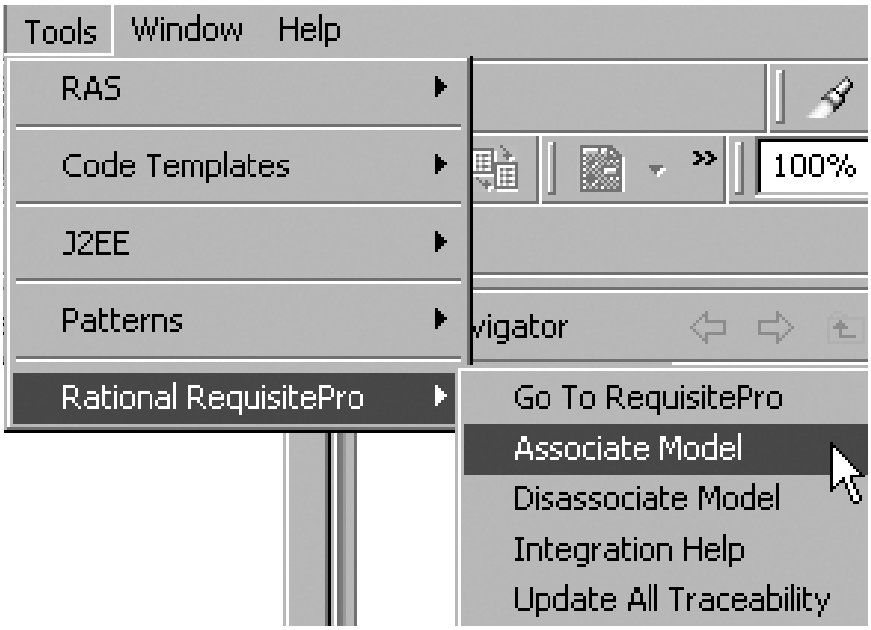

Figure 3: Requirements options on the main Rational XDE Developer menu

By default the entire Rational XDE Developer model is associated with an IBM Rational RequisitePro project, but you can override this model-level association for individual packages (Figure 4). This allows use cases (or design elements) stored in different packages to use different Rational RequisitePro project to store their requirements information. This package-level association lends itself to large software projects that might use either multiple Rational RequisitePro projects (typically one per subsystem) or different use case document templates (i.e. system-level use cases and low-level use cases).

Figure 4: Package associations with a Rational RequisitePro project

The next step is to specify the IBM Rational RequisitePro project that will integrate with the IBM Rational XDE Developer model.

From the Rational XDE Developer Tools menu, select Rational RequisitePro > Associate Model to Project and locate the Rational RequisitePro project to associate to this model (Figure 5). A Rational RequisitePro project includes document types and requirement types. Document types determine the Microsoft Word document templates you will use to document various requirements in Word, and requirements types group requirements by categories that share the same attributes (priority, risk, difficulty, status, etc.).

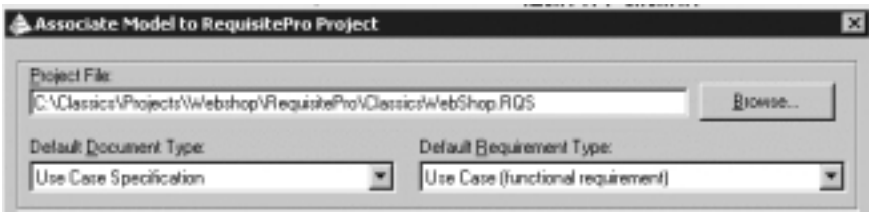

Figure 5: Rational RequisitePro project and use case information selection

Select a document type to be used as use-case document template, when detailing new use cases in Rational XDE Developer, and select a Use Case requirement type. Rational RequisitePro provides default project structures with which we recommend you start. As you become more familiar with the tool, you can create your own document types and requirements types, and project templates for reuse.

Once you specify a Rational RequisitePro project in your Rational XDE Developer model, you are ready to associate use case documents to Rational RequisitePro from Rational XDE Developer use case diagrams, and add traceability to Rational XDE Developer design elements.

Adding Requirements Properties to Use Cases

In preparation for the use case prioritization process discussed in the first part of this paper, to set requirements properties of a use case in IBM Rational XDE Developer, right-click on the use case and select RequisiteProËView Requirement Properties. In the dialog box (see Figure 6), click the Attributes tab, and set use case attribute values. Note that you can change the out-of-the-box use case attributes and their default values in the IBM Rational RequisitePro project associated with the Rational XDE Developer model. From this same dialog box, click the Traceability tab to establish traceability between the use case and other requirements.

Figure 6: Viewing and editing use case requirement properties in Rational XDE Developer

Prioritizing Use Cases

Once you've either attached a use-case document or assigned requirement properties to a use case in Rational XDE Developer, the use case becomes part of your requirement set in Rational RequisitePro. As such, you can use Rational RequisitePro requirement management capabilities to sort your use cases (by priority, by iteration, etc.), to query on specific use cases (i.e. only the use cases planned for the next iteration), and produce use case metrics.

An Attribute Matrix View in IBM Rational RequisitePro (see Figure 7) provides a view of all, or a selected subset of, use cases and their respective attributes. This helps you organize the use case information answering the first question at the beginning of this article: How do I organize my use cases? You can run queries to determine which use cases are assigned to which designer, how difficult they are to implement, or in which release they should be implemented. This answers another question presented at the beginning of the article: How can I tell in which release a particular use case is implemented?

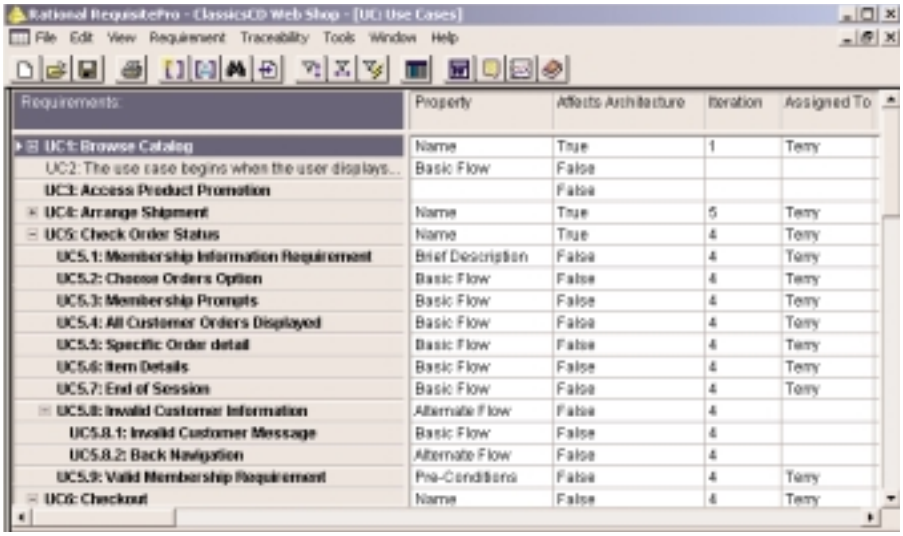

Figure 7: Use case Attribute Matrix View in Rational RequisitePro

Creating Use Case Documents

To detail a use case (by attaching a use-case document to a use case) in IBM Rational XDE Developer, right-click on the use case in Rational XDE Developer (either from the Rational XDE Developer Model Explorer or from the use case diagram), and select RequisitePro>New Use Case Document.

Microsoft Word, controlled by Rational RequisitePro, is launched and your template-based use-case document is displayed and ready for editing. If you document your use case specifications in Microsoft Word and later import them in Rational RequisitePro, you can associate an existing Rational RequisitePro document to a use case in Rational XDE Developer by using the RequisitePro Ë Associate to RequisitePro menu option.

At this point, you should have use case documents completed in Rational RequisitePro.

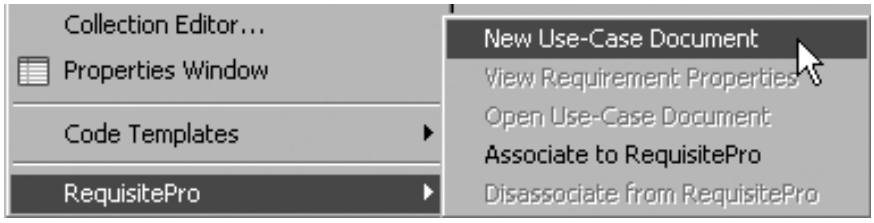

Figure 8: Creating a use case specification from Rational XDE Developer

Managing Requirements in Use Case Documents

In IBM Rational RequisitePro, once you have documented the various flows of events of your use cases, you identify functional requirements and mark them as requirements (Figure 9). This allows you to attach requirement properties to these functional requirements and to trace them to the higher-level requirements they represent.

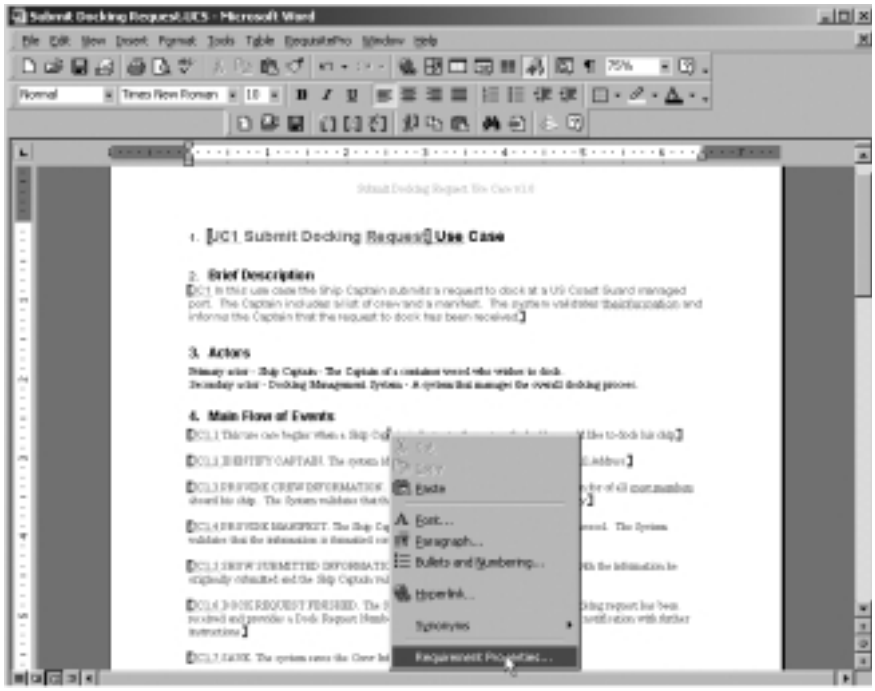

Figure 9: Functional Requirements in a Rational RequisitePro document

Maintaining Use Cases In-line with Business Needs

Once you have prioritized the use cases to be implemented, you should ensure that these use cases stay in line with the evolving business needs they are intended to fulfill. The Traceability Matrix View in Figure 10 shows the relationships established between use cases and business needs. Using traceability matrices, you can query on business needs not yet traced to use cases, answering the question: Which specific business needs does a use case address? Additionally, if business needs change you know immediately which uses cases are potentially impacted by that change, ensuring that your use cases always reflect the evolving business needs. A suspect link (red slashed arrow in Figure 10) indicates that use case UC1.2 may need to be revisited due to a change in business need BUS1.4. Querying on suspect links answers: Are my use cases staying in touch with the evolving business needs they are supposed to solve?

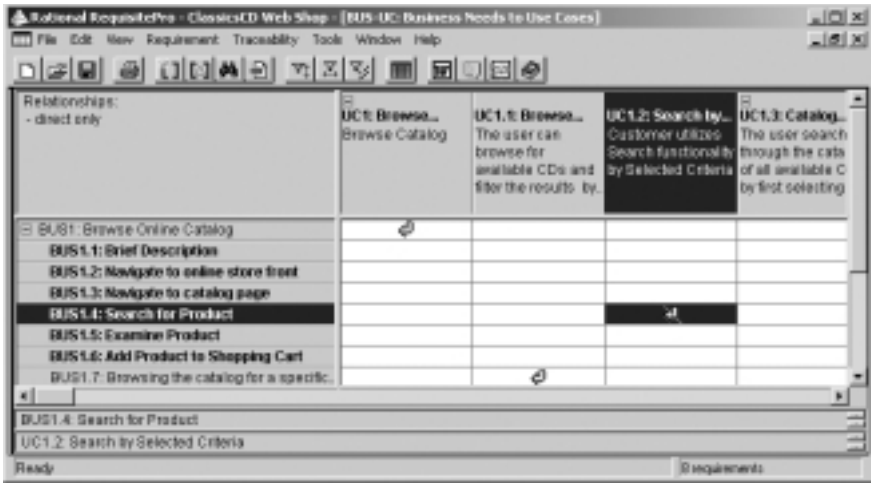

Figure 10: Traceability relationships between use cases and business needs

Ensuring Design Implements All Required Functionality

Similarly, in order to ensure your design truly implements your requirements, you link requirements in IBM Rational RequisitePro to the IBM Rational XDE Developer design elements that implement these requirements.

Once design elements (classes, diagrams, etc.) have been created in Rational XDE Developer, to link them to the requirements that drove their creation, right-click on the design element and select RequisitePro ËAdd Traceability (Figure 11). This will add a representation for this design element in Rational RequisitePro.

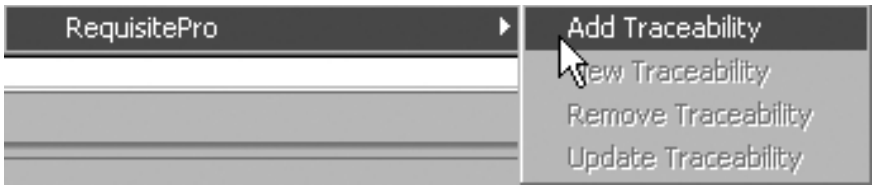

Figure 11: Propagating a design element change to Rational RequisitePro

When the IBM Rational XDE Developer design element name or brief description changes, you can invoke the RequisiteProË Update Traceability option on that same menu. If you make multiple changes to your Rational XDE Developer model, you can propagate all changes at once by selecting Tools Ë Rational RequisitePro Ë Update All Traceability (Figure 12).

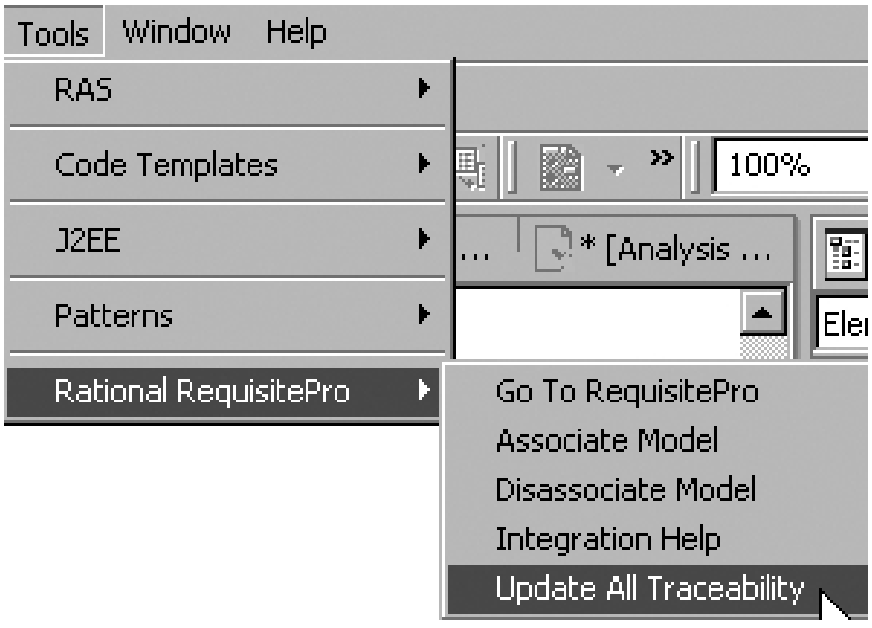

Figure 12: Propagating several design changes to Rational RequisitePro

Once traceability has been added to Rational XDE Developer design elements, in IBM Rational RequisitePro, you create requirements to design traceability matrices (Figure 13). Traceability matrices between requirements (such as use cases) and design are a way to validate design activities are covering all agreed upon requirements.

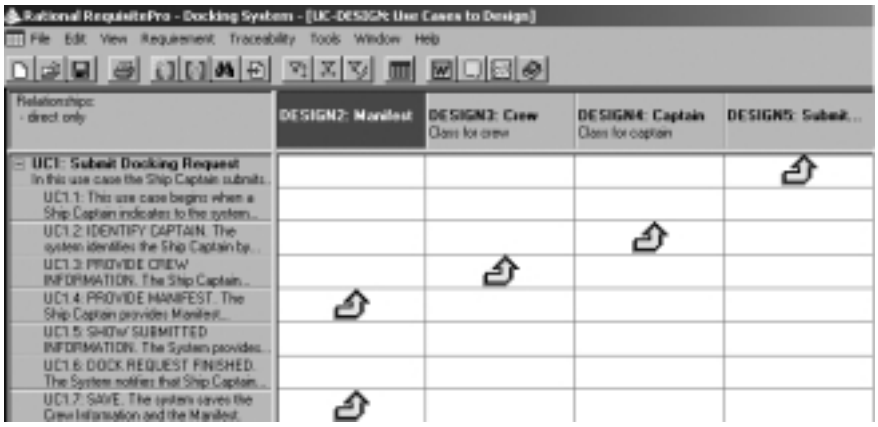

Figure 13: Traceability relationships between use cases and design

Once that linkage is establish, you can run reports in IBM Rational RequisitePro to assess the design coverage of requirements, and provide a quantitative answer to “Are all the agreed upon requirements accounted for in design?” to account for each signed off requirement in the design model that will implement them.

That linkage between requirements and design is also critical at detecting the impact a requirement change has on design elements. When a requirement that has a link to a design element changes, a suspect link (red slash) replaces the blue arrow on Figure 13. Querying on suspect links between requirements and design answers critical questions such as: What part of the design model is affected by latest requirement changes?

Figure 14: Suspect relationships between use cases and design

For more information on working with the integration, refer to the IBM Rational XDE Developer online help.

Summary

In summary, the integration between IBM Rational RequisitePro and IBM Rational XDE Developer enables you to clearly communicate requirements to developers and to keep design in line with changing requirements.

This is key to reaching the end goal: delivering applications that actually solve customer needs.

Without such integration, use cases are documented in Microsoft Word documents, which does not allow for tracking the functional requirements included in use case specifications. This deficiency leads to designs that do not comply with requirements.

Without such integration, software designs run a risk of straying away from requirements. When a requirement changes, quickly assessing which part of the design is affected by that change maximizes your chances to design solutions that solve the customer needs expressed in requirements.

You can view this integration in action at http://www.rational.com/events/webinars/index.jsp.

References

- *Books*
  - *Kurt Bittner & Ian Spence, Use Case Modeling*
  - *Alistair Cockburn, Writing Effective Use Cases*
- *Papers*
  - *Integrated Use Case Management http://www.rational.com/products/whitepapers*
  - *Requirement Management Best Practices for Developers http://www.therationaledge.com/content/jul_02/m_requirementsManagement_cc.jsp*
  - *Traceability Strategies for Requirements Management With Use Cases http://www.rational.com/products/whitepapers/022701.jsp*
- *IBM Rational webinars on use cases at http://www.rational.com/events/webinars/index.jsp*
  - *Use Cases Across the Lifecycle*
  - *Use Cases Storyboards*
  - *Writing Good Use Cases*
- *Online product demonstrations at http://www.rational.com/tryit/reqpro/seeit.jsp*
  - *Rational XDE at http://www.rational.com/tryit/xde/seeit.jsp*
  - *Rational RequisitePro at http://www.rational.com/tryit/reqpro/seeit.jsp*
  - *Rational XDE-Rational RequisitePro Integration at http://www.rational.com/tryit/reqpro/seeit.jsp*