

Creating a Business Rule Repository Using Requisite Pro



WEST POLE

A Whitepaper by West Pole, Inc.
© 1996-2005, All rights reserved.

Why bother to build a business rule repository? If this just sounds like busywork, some bureaucratic way to write requirements, think again.

A repository allows you to use a two-tiered approach to managing your requirements, to give you an edge for maintaining your core business rules across multiple projects.

This whitepaper walks you through the process of building a business rule repository using Requisite Pro, and includes business rule guidelines, types, and attributes.

What Is a Business Rule?

Simply put, a business rule defines or constraints some aspect of the organization.¹ Some common sources for business rules are:

- Strategic decisions from executives
- Laws and regulations
- Contractual obligations

Why Create a Business Rule Repository?

The best practices of software engineering advocate using modular, reusable code. When you separate business rules from specific projects, you're adapting these key principles for effective requirements management.

GOAL: Leveraging Requirements Work

Requirements, as in software, can be designed in multiple tiers. In a standard Requisite Pro project, you identify the requirements through use cases. Some of the requirements are in fact business rules that may apply to other projects as well.

Instead of duplicating such requirements, you can leverage your requirements work from one project to the next by creating a repository.

Business rules are well-suited to this repository concept; as they often span multiple projects in an organization.

¹ Hay, David; Healy, Keri Anderson, "GUIDE Business Rules Project, Final Report," October 1997.

GOAL: Easing Maintenance Effort

Without a business rules repository, your typical work product after completing requirements is a single Requisite Pro repository for each project (one-to-one ratio), with overlapping or duplicated rules in each project.

By isolating business rules, you're adapting software engineering best practices to effective requirements management.

If you implement a business rules repository across multiple projects, you now have a single *business-level* repository with multiple projects' use cases referring back to the repository (one-to-many ratio). You can then use Rational SoDA to combine the use case text and the rules from the repository into one report.

Managing one repository that eliminates business rule duplication across multiple projects eases the maintenance effort of all projects considerably. When a business rule changes, you no longer need to update each project individually. You update the repository once, and trace the effects of the change to each project.

Example: Managing Multiple Systems

Many companies support a client-server system for internal data processing, as well as a web-based (thin-client) system for customer interaction.

The customer has the choice of either calling or mailing in their order, which is processed by an internal employee – or, they can choose to order the item from the company web site. Clearly, many rules apply whether an employee enters the data or the customer does. A simple example would be a validation rule such as, “the quantity ordered must be greater than or equal to zero.”

If you have multiple systems that must be maintained with the same business rules, creating a repository is a great approach over documenting the business rules in each Requisite Pro project.

The Difference between Business and Workflow Rules

Business rules should be identified separately from workflow rules. Business rules govern an organization, regardless of the process or procedure chosen to execute them. As such, they apply to any project the organization undertakes.

Workflow rules, by contrast, define *how* work is done. The goal of many projects is to create or improve workflow, i.e. to change the workflow rules.

Remember, business rules *constrain* some aspect of the organization. Workflow rules define the process by which business is done; they are bound by the business rules.

Business and Workflow Rules Example

For example, a bank's policy may be that when a customer withdraws over \$5,000 from an account, the teller must get approval. The workflow rules supporting that business rule describe how the teller gets the actual authorization.

Business Rule: A supervisor must authorize any cash withdrawal of \$5,000 or more.

Workflow Rule: When processing a withdrawal over \$5,000, the teller must either:

- Fill out an approval slip and have the supervisor sign it.
- E-mail the supervisor who presses a thumbprint on the screen for approval.
- Shout to the supervisor, who rings a cowbell signifying approval.

How to Identify Business Rules

It's beneficial to properly identify your organization's business rules, and make sure that you're not mixing workflow rules into your list of business rules. Business rules will tend to span projects, while workflow rules may not.

A good rule of thumb for identifying a business rule: imagine that you had to create the company from scratch, possibly changing all workflows. What are the rules that you'd still need, to rebuild the company? Typically, these rules are the business rules.

Of course, this is a general guideline, not an absolute, because executive strategy is always based on existing resources.

Business Rule Basics

If you're new to the process of defining business rules, review these basics to get up and running quickly.

Business Rule Types

One way to categorize business rules is by applying these types:

- **Definition**—defines a business concept, or relates business concepts. In ER modeling, a rule of this type may correspond to an entity, an attribute, or a relationship (although it may not correspond to any of these).

- **Constraint**—enables or prohibits action. These may be reflected in database integrity constraints, data validation rules, triggers, manual business procedures, etc.
- **Derivation**—enable discovery of new information from information already known. This may either be a mathematical formula, or simply be a derived fact of the form “if A is true, then we know B is true”.

Guidelines for Business Rules

When stating a business rule, double-check this list of guidelines to ensure that you’ve clearly, consistently, and accurately stated the rule:

- **Declarative**—Business rules are not stated procedurally. The rule is declared; how the rule is enforced isn’t part of the rule.
- **Precise**—A business rule must be open to only one interpretation. If the rule can be understood to mean more than one thing, you have to restate it.
- **Atomic**—A business rule contains a single complete thought, but not more than one. The business rule must be indivisible; if you try to break up a true business rule into parts, you’ll lose information.
- **Consistent**—A set of business rules must not contain conflicting rules.
- **Non-redundant**—A set of business rules must not contain rules stating the same information.
- **Business oriented**—The rules are stated in terms businesspeople can understand. The rules must use terms that are meaningful and confirmed across the business scope.
- **Owned by the business**—Business rules are stated by the businesspeople, and only they can own the rule, modify it, or state that the rule is no longer valid (delete the rules).
- **It may also use other rules**—Business rules build on each other. Constraint rules often state when terms and facts can or can’t exist, and inference rules state how a term or fact is derived. A common misconception is that “atomic” means “doesn’t use other rules”. Atomic means no such thing, and it’s almost impossible to write a business rule that doesn’t use other business rules.

Sample Business Rules Attributes

You’ll need to capture these business rule attributes, or other similar attributes that fit your business environment:

- Rule name
- Status (proposed, validated, approved, or archived)

- Effective date
- Expiration date
- Description
- Expression (a concise statement of the rule)
- Triggering business event
- Department or workflow category
- Source(s) of rule, such as:
 - System source code
 - Existing process documentation
 - IT and business staff
- Brief “meta-comments” to document the repository itself

Creating a Repository

With Requisite Pro, you have the ability to trace your requirements to requirements in other projects. You can use this feature to create a centralized repository for your company’s business rules, and trace to them from each of your application-specific Requisite Pro project files.

How to Do It

To create a business rule repository:

1. Open Requisite Pro.
2. In the Create Project window, click the New tab.
3. Choose the blank project template and click OK. You’ll modify this to create a new business repository template.
4. Give your new project a name (“Biz Rule Repository”, or something similar) and click OK.
5. Requisite Pro creates and opens the new project.

Creating a New Requirement Type

To easily identify your traced rules, create a new requirement type in your rule repository, called “Business Rule” (abbreviated as “BIZ”). It’s important to flag

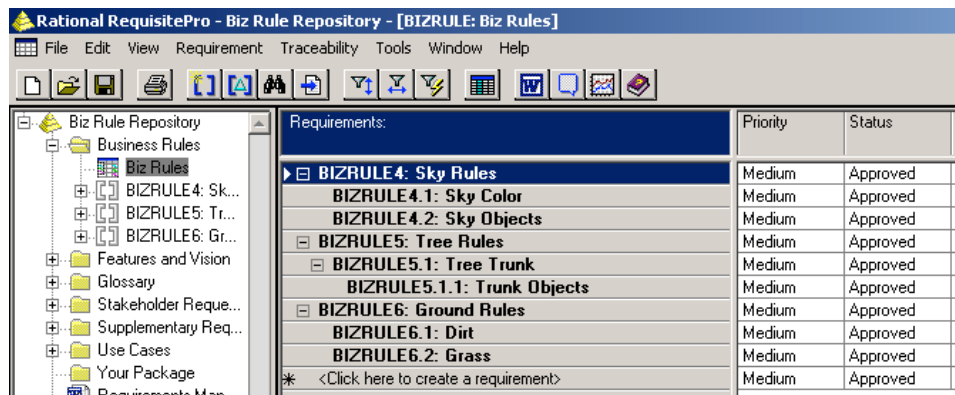
this requirement type as eligible for external traceability, so you'll be able to see your business rules from the other projects.

How to Do It

To create a new requirement type:

1. Select File/Project Administration/Properties. The Project Properties window displays.
2. In the General tab, enter "BIZRULE" in the Prefix field.
3. Click the Requirement Types tab.
4. Click the Add button. A new window displays.
5. Name your new requirement type "Business Rule".
6. Select the Allow External Traceability checkbox.
7. In the Requirement Tag Prefix field, enter "BIZ".
8. Click OK to close the new requirement type window.
9. Click OK to close the project properties window.
10. Select File/New/Package and title the new package "Biz Rules".

Sample Business Rules in the Requisite Pro Repository



Requirements:	Priority	Status
▶ BIZRULE4: Sky Rules	Medium	Approved
BIZRULE4.1: Sky Color	Medium	Approved
BIZRULE4.2: Sky Objects	Medium	Approved
▶ BIZRULE5: Tree Rules	Medium	Approved
▶ BIZRULE5.1: Tree Trunk	Medium	Approved
BIZRULE5.1.1: Trunk Objects	Medium	Approved
▶ BIZRULE6: Ground Rules	Medium	Approved
BIZRULE6.1: Dirt	Medium	Approved
BIZRULE6.2: Grass	Medium	Approved
* <Click here to create a requirement>	Medium	Approved

Now that you have set up your business rule repository, you can save it as a new project template for future use.

1. Close the Biz Rule Repository project.
2. Select File/New/Project.

3. Select Make New Template and click OK. The project template wizard displays.
4. Give your template a name, such as “Business Rule Template”.
5. In the ReqPro Project field, use the browse button to find the business rule repository project (Biz Rule Repository.rqs). **Note:** Do NOT select the Include Project Data checkbox.
6. Click Finish. Requisite Pro creates a new template based on the Business Rule Repository project file.

Tracing to Business Rules from Other Projects

Now that you’ve created your business rule repository, you need to trace your application-specific rules to your business rules.

How to Do It




Before you can trace rules to external requirements, you must first allow tracing to your use case requirements:

1. Open your Requisite Pro project where you are documenting your non-business rules.
2. Select File/Project Administration/Properties. The Project Properties window displays.
3. Enter a project Prefix under the General tab.
4. Click the Requirement Types tab.
5. Select “UC: Use Case Requirement Type” and click Edit.
6. Select the Allow External Traceability checkbox.
7. Repeat Steps 4 and 5 for each requirement type you wish to trace to a business rule.
8. Click OK when you are finished.
9. Select File/Project Administration/External Projects. The External Projects window displays.
10. Click Add, and locate the Business Rule Repository project file (Biz Rule Repository.rqs).
11. Click OK.

To trace rules to external requirements:

1. Select the Use Cases package.
2. Select File/New/View. The View Properties window displays.
3. Enter “UC to BIZ Rule” for the Name.
4. Select Traceability Matrix for the View Type.
5. Select “UC: Use Case Requirement Type” for the Row Requirement Type.
6. Select “BIZRULE.BIZ: Business Rule” for the Column Requirement Type.
7. Click OK.
8. The new traceability view displays. If you have created any business rules, they appear in the view. To trace a use case to a business rule, right-click the cell that intersects the use case requirement and business rule and select Trace To. An arrow appears in the cell.

Matrix Traces a Use Case to a Related Business Rule

Relationships: - direct only	<input checked="" type="checkbox"/> BIZ.RULE4: Sky. These are the rules associated with the sky.	BIZ.RULE4.1:...	BIZ.RULE4.2:...	<input checked="" type="checkbox"/> BIZ.RULE5:...	<input checked="" type="checkbox"/> BIZ.RULE5.1:...	BIZ.RULE5.1.1:...	<input checked="" type="checkbox"/> BIZ.RULE6:...
<input checked="" type="checkbox"/> UC2: Schedule Human... This use case describes the process by which a toy designer schedules the human trials for one of their...							
<input checked="" type="checkbox"/> UC4: Paint Landscape This use case describes the steps involved in painting a landscape picture.							
UC4.1: Draw the horizon line across the lower half of the canvass or paper.							
UC4.2: Create the sky by adding color and objects in the upper part of the picture.							
<input checked="" type="checkbox"/> UC4.3: Create the ground below the horizon line.							
<input checked="" type="checkbox"/> UC4.3.1: Add objects to the ground.							
UC4.3.1.1: The user may wish to include one or more trees.							

Repeat Step 8 for all desired traceability links.

Merging the Rules into Documents

After tracing functional requirements to your business rules, you'll need an effective way to communicate them to your shareholders. Rational Software's documentation tool (SoDA), works nicely.

SoDA allows you to create document templates that pull data from the Rational toolset.

By using the traceability link you created between your application requirement and the business rule, SoDA is able to merge the data from the two sources (project files), into one use case specification.

The End Product: A SoDA-generated Use Case Document

Use Case Specification: Paint Landscape

Description
 This use case describes the steps involved in painting a landscape picture. (UC4)

Pre-Condition
 The user begins with a blank canvas or sheet of paper. (UC4.6)

Basic Flow

- 1) Draw the horizon line across the lower half of the canvas or paper. (UC4.1)
- 2) Create the sky by adding color and objects in the upper part of the picture. (UC4.2)
 - Sky Rules**
 These are the rules associated with the sky. (BIZRULE4)
 - The color of the sky must be blue, if it is a daytime picture, or black if the picture is of the night sky. (BIZRULE4.1)
 - The sky may contain white, fluffy clouds, birds, balloons, a yellow sun, or a moon and stars. (BIZRULE4.2)
- 3) Create the ground below the horizon line. (UC4.3)
 - Ground Rules**
 These rules govern the creation of the ground. (BIZRULE6)
 - Dirt may be brown or black. (BIZRULE6.1)
 - Grass must be green and be on top of the ground. (BIZRULE6.2)
 - B.1) Add objects to the ground. (UC4.3.1)
 - B.2) The user may wish to include one or more trees. (UC4.3.1.1)
 - Tree Rules**
 These are rules to govern the creation of trees. (BIZRULE5)
 - A tree trunk must be brown and grow up from the ground. (BIZRULE5.1)

Alternate Flow

- 1A) The user may wish to add mountains in the horizon, instead of a straight, horizontal line. (UC4.4)
- At any time in the use case, the user may lose inspiration and stop. (UC4.5)

Post Condition
 A masterpiece has been created. (UC4.7)

Shortcut: Download a SoDA Template

You can download a SoDA template to get a jumpstart on this step at: www.westpole.com

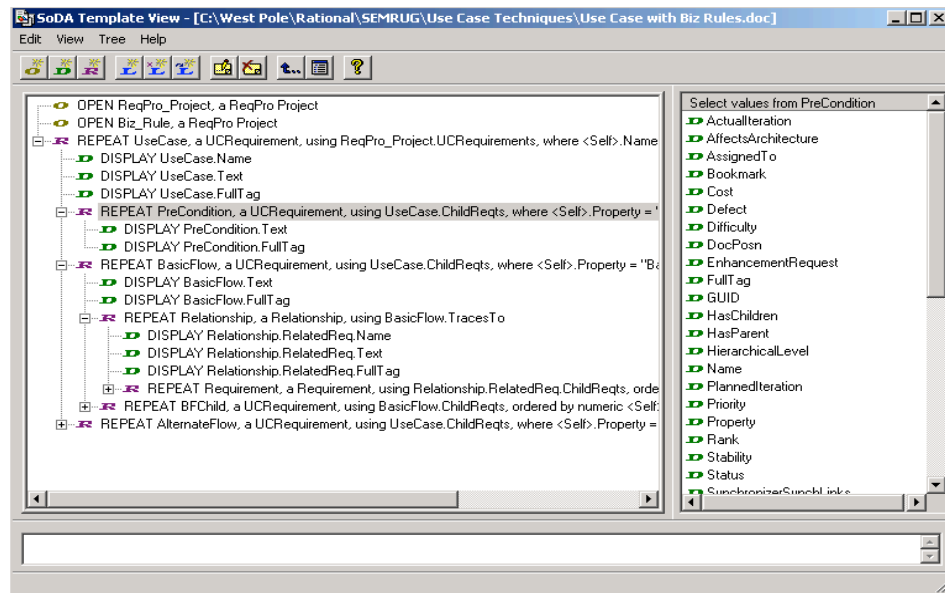
How to Do It

The steps below give you general instructions for creating a traceability link for one use case specification. Refer to the SoDA documentation for detailed instructions to complete each step. An alternative is to modify SoDA's existing traceability hierarchy template.

To merge the rules into documents:

1. Open a blank SoDA template.
2. Open two Requisite Pro projects in the template, one for the project containing the use case requirements, and the other for the business rule repository. To keep from confusing the projects in a later step, name the business rule repository project "Repository".
3. Add a Repeat command for the use case requirements (UCRequirements) in the use case project. Under the Advanced options, have SoDA prompt you for the name of the parent use case requirement.
4. Add a Repeat command for the child use case requirements to the parent use case requirement.
5. Under the child use case requirements, add a repeating Traces To command. And then add a display command for the traced-to requirement's full text, and any child requirements it may have.
6. Repeat Steps 4 and 5 for grandchild use case requirements to the parent use case requirement.

The Template View of SoDA Defines What SoDA Displays



Conclusion

When you separate business rules into a repository for requirements management across multiple projects, you apply the modularity and reuse principles that are the best practices of the software engineering industry.

More importantly, the burden of maintenance for multiple projects is eased, due to the increased accuracy of tracking requirements when there are changes to business rules.

A business rules repository can be an incredibly effective method for managing requirements over time – and Rational’s Requisite Pro and SoDA products can be applied to support your repository development process.

References

Hay, David; Healy, Keri Anderson, “GUIDE Business Rules Project, Final Report,” October 1997.

Von Halle, Barbara, “Life Beyond Data,” *Database Programming & Design*, September 1995, pages 13 to 15.