

IBM Innovate 2011  
Using IBM® Rational® Requirements  
Composer Sketching Tools to Visually  
Communicate and Elaborate  
Requirements

Srinivas Mandava  
Staff Software Engineer, IBM  
[smandava@in.ibm.com](mailto:smandava@in.ibm.com)  
RDM-1460

IBM Software

# Innovate2011

The Premier Event for Software and Systems Innovation



**Software. Everywhere.**

**August 9-11, Bangalore | August 11, Delhi**



## Please Note:

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal 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.

## Agenda

- Rational Requirement Composer features
- What are the “UI sketching tools” within Rational Requirements Composer?
- Benefits of visually communicating requirements
- How we used sketching tools in our work on Rational Requirements Composer
- Tips and techniques for using sketching tools more effectively

# IBM Rational Requirements Composer Features

## Requirements Management for the Development Lifecycle

### Rational Requirements Composer

#### Definition

- Rich-text documents
- Diagrams: Process, Use Case
- Storyboards, UI sketching & flow
- Project glossaries
- Templates (formal/agile)

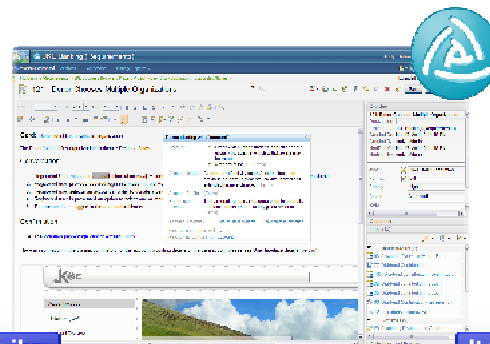
Improved!

#### Visibility

- Customizable dashboards
- Analysis views
- Collections
- Milestone tracking & status

#### Collaboration

- Review & Approval
- Discussions
- Email Notification



#### Management

- Structure, Attributes/Types
- Traceability, Filtering, Tags
- Baselines, Change History
- Reuse
- Reporting Metrics & Doc.

NEW!

Improved!

#### Lifecycle

- Central requirements, test, & development repository
- Common administration and role-based user licensing
- Warehouse reporting

NEW!

#### Planning

- Integrated planning
- Effort estimation
- Progress tracking

## What are the “UI Sketching Tools” within Rational Requirements Composer

## IBM Rational Requirements “UI sketching” tools

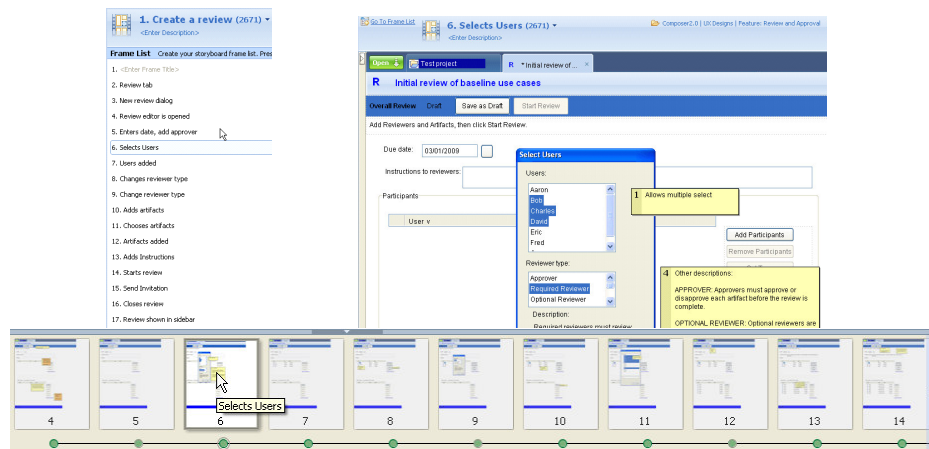
“User interface sketching tools” generically refers to the subset of Rational Requirements Composer capabilities for illustrating the user interface of a software application (storyboards, user interface sketches, and user interface parts).

Primary purpose is to quickly and easily “sketch” the user interface and scenario flow with just enough detail to facilitate conversations about the requirement in a more concrete context.

Not for creating working prototypes or high-fidelity user-interface designs.

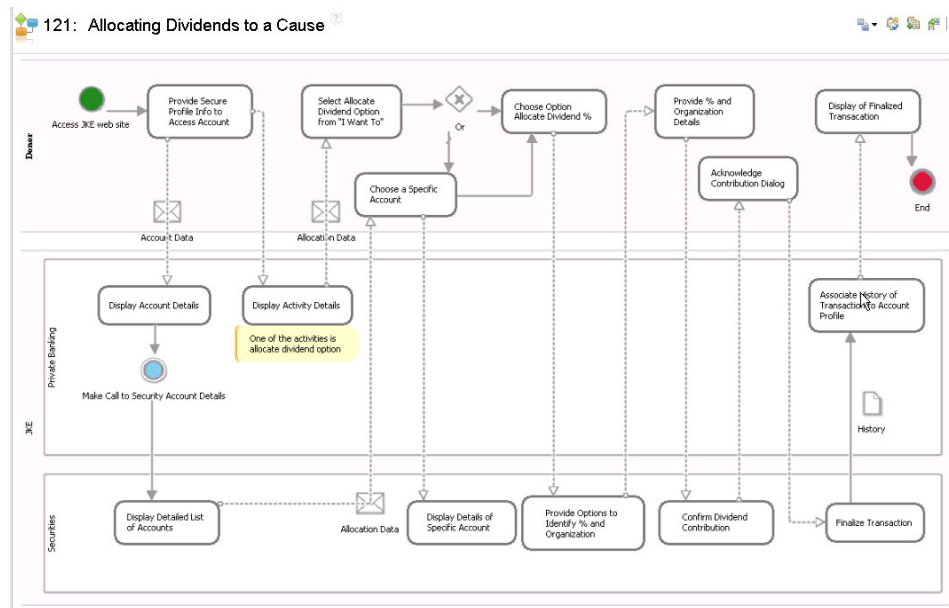
These are useful resources, but not necessary or efficient for higher-level requirement discussions at the elicitation and validation stage.

Can also be used to add more detail for direction during implementation.

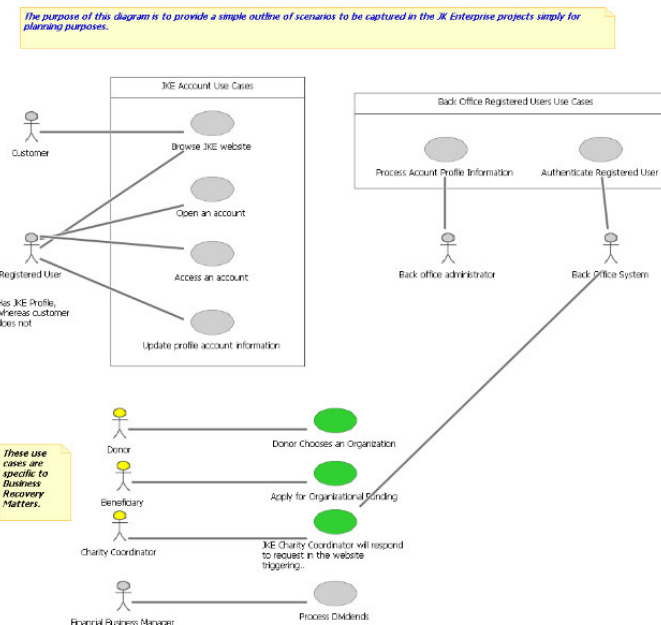


## Other visual techniques in Rational Requirements Composer

UI sketching tools are just part of the “visual” techniques for defining requirements in Rational Requirements Composer. There are also business process diagrams and use case diagrams for visually communicating and elaborating requirements.

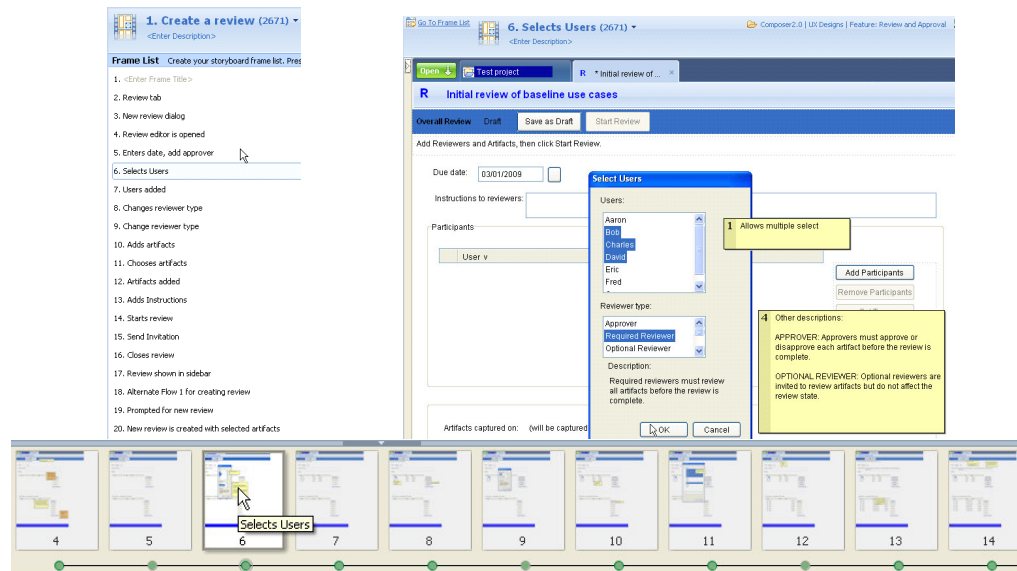


122: JKE Enterprise Diagram



# Storyboards

- Used to illustrate user scenarios and user interactions.
- Visually expresses or illustrates requirements that have a user-interface and user-interaction component.
- Uses a series of “frames” to show the progression of user-interface screens and actions performed by the user while performing primary tasks.
- Each frame can be constructed from previous frames, sketches, or parts, or by adding individual user-interface elements directly to the frame.





# User Interface Sketches

- For illustrating a single web page, screen or state.
- Can be used to visually express or illustrate requirements that have a user-interface component, but where there's not much user interaction
- Can be constructed from other sketches, parts or by adding individual user-interface elements directly to the sketch.

**R Initial review of baseline use cases**

Overall Review Draft Start Review

Due date:

Instructions to reviewers: Bob, can you make sure you focus on Artifacts 1 -3, and Charles on Artifacts 4, 5 and 9? David please review as you have time.

Participants

User v	Type	Status
Bob	Approver	In Progress (70% complete) 1 Disapproved
Charles	Reviewer	Not started
David	Optional	Reviewed

Artifacts captured on: (will be captured when review is started)

Artifact	Version	Status
Artifact1	02/01/2009 9:56 AM	Not started
Artifact2	01/31/2009 10:00 AM	Not started
Artifact3	12/15/2008 08:56 PM	Not started
Artifact4	02/01/2009 9:56 AM	Not started
	01/31/2009 10:00 AM	Not started
	12/15/2008 08:56 PM	Not started

**7** Should current state and buttons go in header, similar to reqs?

**5** Table should be sortable.

**4** User pics

**3** Specify type when adding users. Change type via context menu or we could provide a button.

**1** Status:  
 \* Not started (or perhaps just 0% complete)  
 \* % complete (based on number of required artifacts they have completed)  
 \* Complete/Approved/Rejected  
 Perhaps indicate if any artifacts rejected, even if not complete.  
 Hover could give more details about individual artifacts.

**2** Status:  
 \* Not reviewed (or perhaps just 0% reviewed)  
 \* % reviewed (based on number of approvers/reviewers -- not optional)  
 \* Complete/Approved/Rejected  
 Perhaps indicate if any artifacts rejected, even if not complete.  
 Hover could give more details about individual artifacts.

**6** If we need more detailed status, e.g., reviewer x artifact matrix, we could add another section or tab for that.

## User Interface Parts

- For creating reusable components that are used in sketches and storyboards.
- Can be constructed of other parts or by adding individual user-interface elements.
- Key to rapid iteration of storyboards.

Participants

User v	Type	Status
Bob	Approver	In Progress (70% complete) 1 Disapproved
Charles	Reviewer	Not started
David	Optional	Reviewed

[Import from existing review](#)

[ content area ]

Artifacts captured on: (will be captured when review is started)

Artifact	Version	Status
Artifact1	02/01/2009 9:56 AM	Not started
Artifact2	01/31/2009 10:00 AM	Not started
Artifact3	12/15/2008 08:56 PM	Not started
Artifact4	02/01/2009 9:56 AM	Not started
Artifact5	01/31/2009 10:00 AM	Not started
Artifact9	12/15/2008 08:56 PM	Not started

[Import from existing review](#)

[ content area ]

## Benefits of visually communicating requirements

## Why “visual communication”?

- To ensure that everybody is on the same page in understanding the requirement
- Bridges the communication gap between interested parties (stakeholders, business analysts, user experience, development, testing, etc.)
- Visual representations can often communicate requirements more efficiently and more effectively than just text. Provides a concrete context that customers/stakeholders can relate to.
- Forces thinking in terms of scenario and user interaction.
- Is helpful in identifying additional, related requirements, constraints, etc.
- Can help clarify scope of requirement

## Why visual communications with Rational Requirements

- Integration with other requirement artifacts
  - ▶ Linking/traceability to related requirement artifacts (user requirements, use cases, business process diagrams, business rules, etc.)
  - ▶ Embedding visual artifacts within textual artifacts to create composite documents
  - ▶ Manage all requirement artifacts in the same system
- Integration with Rational Lifecycle Management applications
  - ▶ Relate artifacts with development work items in Rational Team Concert and test plans and results in Rational Quality Manager.

## Why visual communications with Rational Requirements

- Ease of creating and modifying storyboards, sketches and parts
  - ▶ Employs reusable parts, making it easier to build up storyboards and ensure consistency
  - ▶ “Inheritance”, along with reusable parts, makes it easy to make changes to large number of related artifacts.
- Benefit from other Rational Requirements Composer features
  - ▶ On the web
  - ▶ Reviewing
  - ▶ Commenting
  - ▶ Artifact audit history
  - ▶ Artifact filtering and tagging
  - ▶ Traceability analysis
  - ▶ Reporting

# How RRC product team used sketching tools in our work on Rational Requirements Composer

## Our primary uses of sketching tools

- Understanding and elaborating on requirements
- Exploring the user interface and user experience design
- Proposing the user interface and user interaction
- Specifying the user interface design
- Helping frame and resolve issues during development
- Illustrate and capture UI patterns and guidelines



## Understanding and elaborating on requirements

- To initially understand, elaborate and agree on the requirement
- Very early in the process. Usually based on a short text description of the requirement.
- This step is especially important for more vague and large-scope items. Text descriptions alone can leave a lot of room for misinterpretation.
- Usually created by business or user experience team, sometimes dev team
- Not elaborate or detailed. *Just enough to communicate the idea.*
- Helps facilitate discussion and elaboration of requirement
- Input for user experience designs, initial development estimates

# Example

Custom Attributes and Links

Filter by artifact type

Feature

Filter by attribute value

Feature.Priority <> Low  
User Story.Priority <> Low

[Filter by linked artifact](#)

Available link types

Related To

Traced To

Parent

Artifact type that participate to Trace To

Feature

Use Case

User Story

Endpoint is null

Well-known Attributes

[Filter by filename](#)  
Burak

[Filter by user](#)

[Filter by date](#)

Custom Attributes

Filter by attribute value of

[Use Case](#) [Feature](#)

[Status](#) [Priority](#) [Difficulty](#)

Elaborated

Not [Apply Filter](#)

**3** Checkbox to ask for "Features that do not have any Use Cases"

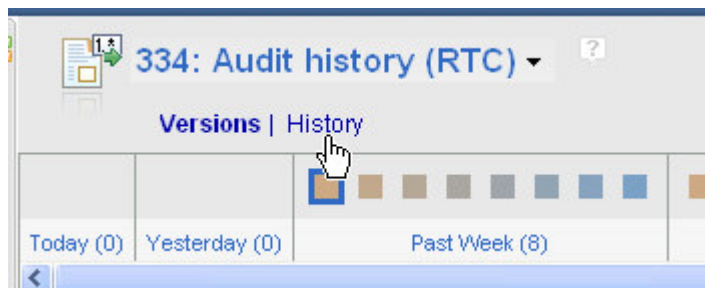
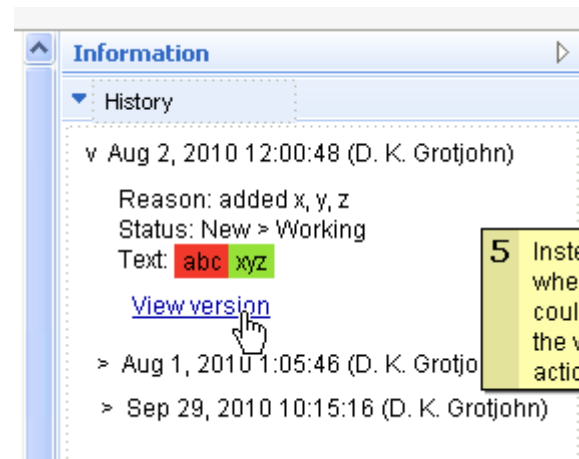
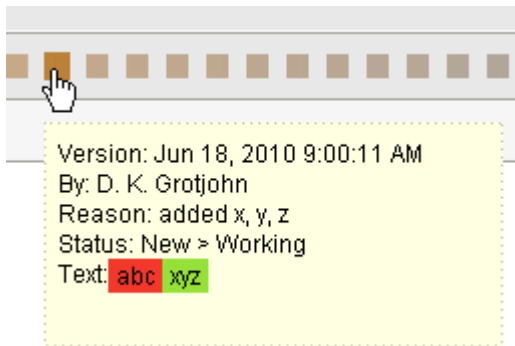
**4** No links here. Only 1-level deep is allowed.

**1** The section expands. The link types that apply to a particular type

## Exploring the user interface and user experience design

- Initial designs of the user interface and user experience
- Focuses on primary screens and primary user tasks
- More elaborate and detailed than initial sketches, but still low fidelity
- Includes many design options and iterations
- Primarily a user experience activity, but may require some input from business or development team
- Dependencies and interaction issues can start to be identified at this stage
- Taking advantage of re-use and inheritance is especially helpful here
- User experience identifies preferred design

# Examples: exploring options for audit history



# Examples: controls for attribute and attribute value filters

The screenshots illustrate the following steps in the 'Filter by Attributes' workflow:

- Initial View:** A list of attributes (Name, User, Date, Artifact type, More Attributes, Linked artifacts) is shown.
- Select Attribute:** A dialog box allows selecting an attribute. A callout '1 Can filter list' points to the selection area.
- Search:** A search input field is used to find attributes. A callout '4 click to open dialog' points to the 'More...' link.
- Search Results:** A list of search results is displayed, including '<Item>' and '<Show all items>'. A callout 'e ahead searches against the ver' is present.
- Filter by Value:** A dropdown menu for 'Priority' is shown with options: High, Medium, Low. Callouts 'x' are next to the selected value.
- Add Item:** A specific attribute value 'Planned For' is added to the filter. A callout '2 click to add item' points to the 'Add' button.
- Final Filtered View:** The final state shows the filter applied, with 'Priority' set to 'High' and 'Planned For' selected. Callouts 'x' are next to the filter values.

## Proposing the user interface and user interaction

- User experience team narrows down and “presents” proposed UI designs
  - ▶ Usually just storyboards at this point
- More detailed, but still fairly low fidelity and focused on primary user tasks
- Business team confirms that it satisfies requirement
- Development team confirms that they can implement it, updates estimates if necessary
- Will often require further modifications to UI designs
- Can use to get feedback from stakeholders
- Other teams (e.g., test, user assistance, etc.) can begin to see how the requirement will be manifested
- Note that teams can see this work in progress as well

# Example: Filtering by attributes

Folder  
Name  
Artifact type  
Last modified date  
Last modified by user  
Linked artifacts  
More attributes

**1** Click to add a condition based on other attributes

Name  
Artifact type  
Last modified date  
Last modified by user  
Linked artifacts  
More attributes

Add attribute

P

Planned for

Priority

More attributes

Add attribute

Priority

is v

Values: (any)

High  
 Medium  
 Low

Done

More attributes

Add attribute

attribute1  
attribute2  
attribute3  
<Item>  
<Item>

Priority

is v

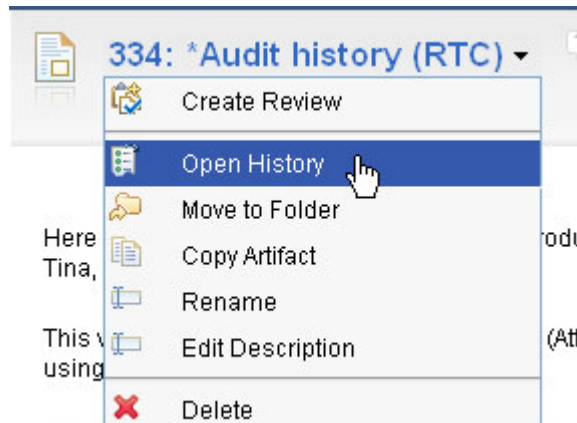
Values: High

High  
 Medium  
 Low

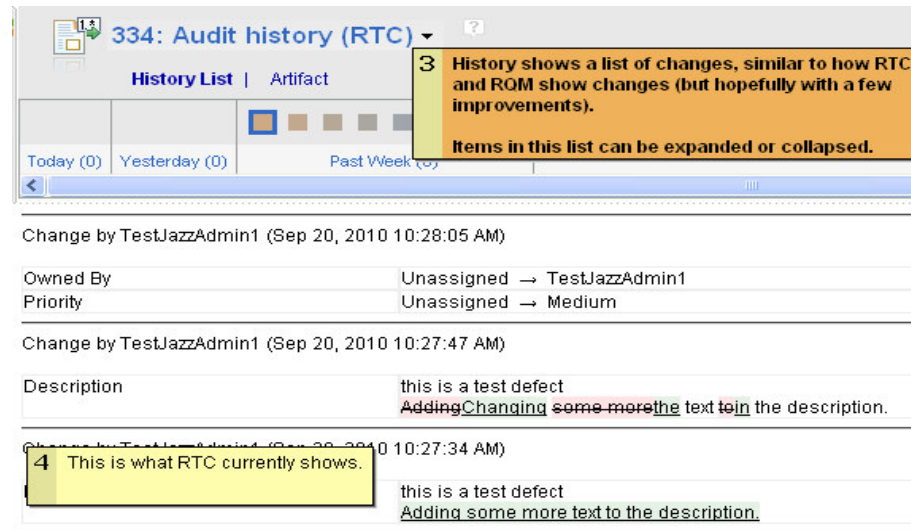
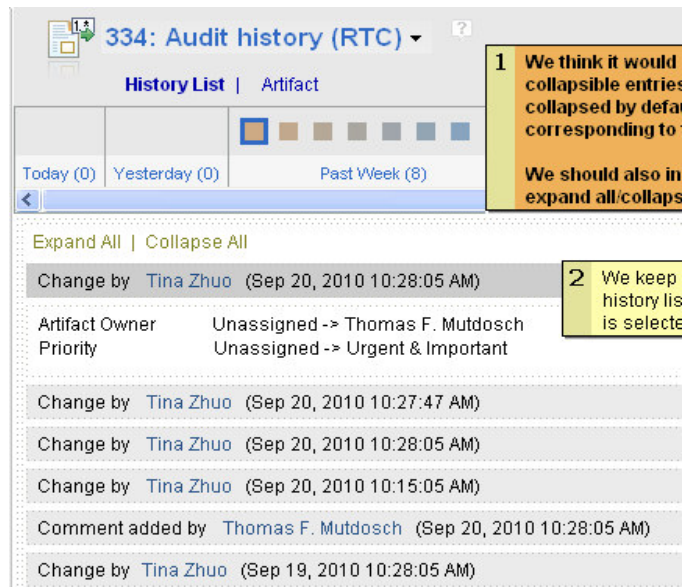
**1** clicking value selects and applies it immediately (is listed above)

Done

# Example: audit history



The audit history in DOORS RP shall record the fi





## Specifying the user interface design

- Usually need to specify the design in more detail for implementation
- Include storyboards for secondary or alternative tasks and flows
- Add detailed descriptions to user interface elements and components
  - ▶ Add notes to sketches
  - ▶ Create a composite rich text document with textual descriptions and UI sketches or storyboards embedded in the document
- Helpful in identifying gaps, inconsistencies, problems, etc.
- Higher-fidelity prototypes may be created outside of Rational Requirements Composer
  - ▶ E.g., visual design team uses graphical editing tools
  - ▶ These external resources can be added to Requirements Composer repository (and linked to sketching artifacts), or can even be pasted into storyboards.
- Primarily intended for development team
  - ▶ Also useful for any team needing more details on interactions (e.g., test, user assistance)

## Example: detailed designs for attribute filter forms

Add more detail to storyboards as needed (depending on audience and purpose)

**Filter forms for different types (9447)** >

<Enter Description>

**Frame List** Create your storyboard frame list. Press

1. Overview
2. Users (and other multi-select lists)
3. Integer type
4. Float type
5. Boolean
6. Date type
7. Time type
8. DateTime type
9. String type
10. Duration type
11. Folders
12. Tags

Owned by

is v

Value

100

is

is not

exists

does not exist

More

Last r

Done

user1

user2

user3

user4

user5

Done

Attribute using integer

is v

100

is

is not

is greater than or equal to

is less than or equal to

More

Last r

Last modified by user

Done

String

contains v

ux, ui

contains

does not contain

starts with

ends with

exists

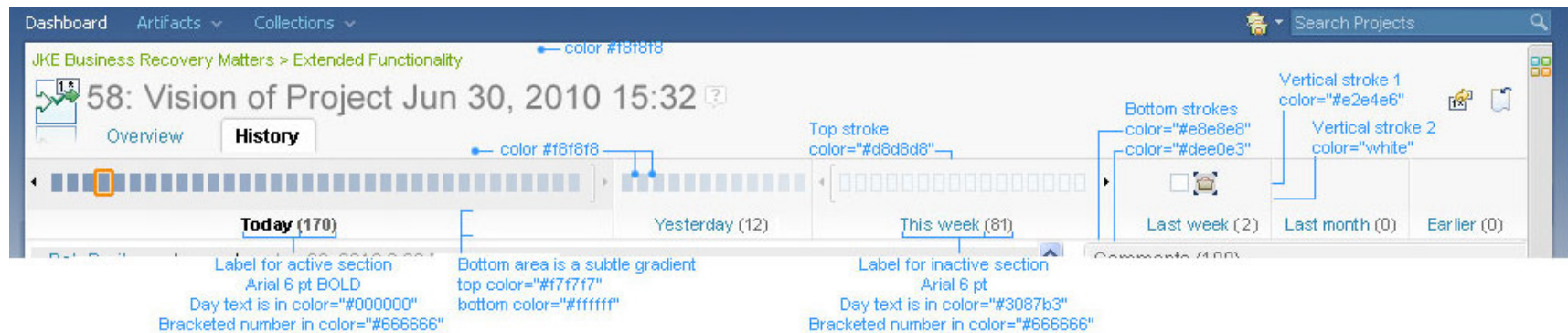
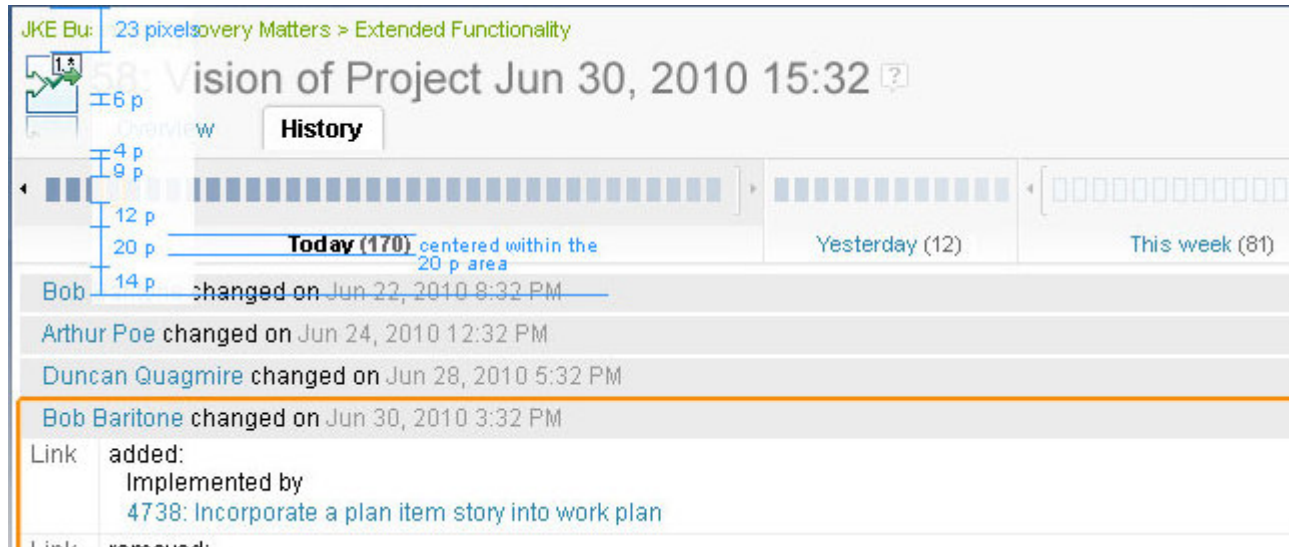
does not exist

More attribu

Last modified date

Done

## Example: detailed visual specifications



# Example: detailed UI specifications document



## 1. Review Editor - "Review State" section

This section of the review is for editing the review as well as showing the current state of the review. It is a different view of the same review as shown in the "Your Review" section (Section 2). It will primarily be used by the creator of the review.

All users can see this section, although only creators of the review or users with **admin** rights can edit it (for all others, the action buttons should be hidden). When the review is in draft state, this will be the top section (after the review is started, "Your Review" will be the top section).

The following document describes what a user can see and do based on their role and the state of the review: [Views and actions for reviewers and non-reviewers](#).

**C. Due date, Instructions**

- Due date (optional):
  - Creator can set the due date of the review.
  - Should have a graphical calendar widget for choosing date, or can manually type in date.
  - When the review is started, we should provide some visual indicator that it is past the due date.
- Instructions to reviewers (optional):
  - Creator can add text instructions to reviewers.
- If the review is in Started state, changes to either of these fields adds the **Notify Participants of Changes** button to the Status bar (section 1B).

**D. Reviewers list**

- Contains all the users that are part of the review.
- Table contains columns for:

## Helping frame and resolve issues during development

- Issues and problems will arise during development
- Being able to visualize the problem as well as potential solutions can increase clarity and reduce the time to resolve the problems.
- Often used within development teams themselves, as well as discussions between development and user experience.

## Illustrate and capture UI patterns and guidelines

- Use to illustrate “UI patterns” or “UI guidelines” for your product
  - ▶ Mock up examples, or capture and annotate existing UI
- Using common parts and sketches as building blocks for all UI mockups helps achieve consistency
- Sharing UI artifacts within team and across related product teams can also help achieve consistency, common patterns and synergy
- Can use to store screen captures of existing UI (to show patterns, identify inconsistencies, etc.)

## Tips and Techniques for Using Sketching Tools

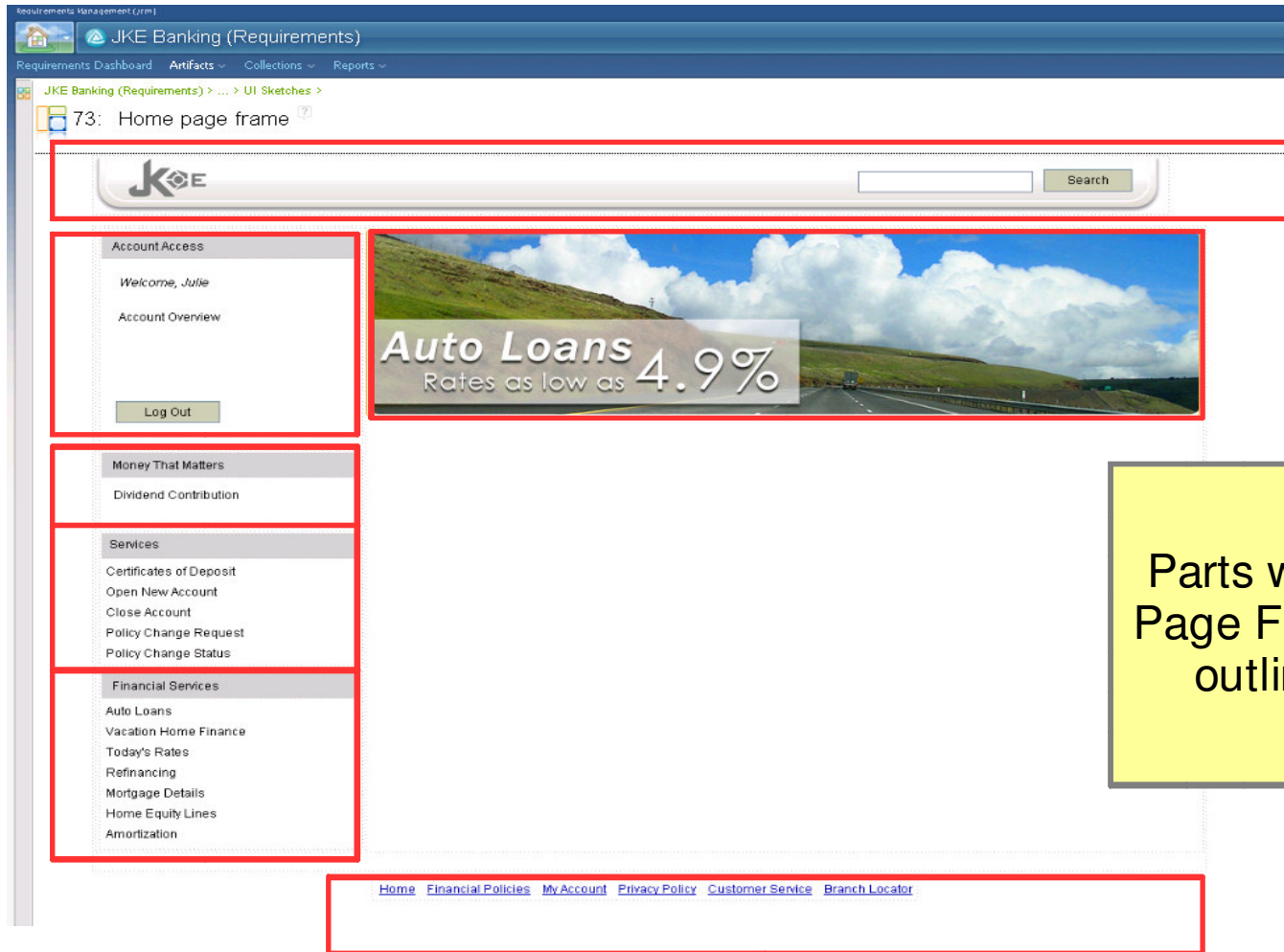
## Tips and techniques for using sketching tools

# Make use of re-use (before you get too far)

- When you find yourself creating something that already exists
- Dialogs
- Common components, widgets, UI elements, etc.
- But don't let it interfere with your design thinking – you can extract parts and sketches later.
- Not just parts – sketches can inherit sketches
  - ▶ Good for designing alternatives
  - ▶ Can use as “templates” (different from actual artifact templates)
  - ▶ Storyboards are also good for variations on similar designs
  - ▶ Think twice before copying a sketch and breaking inheritance

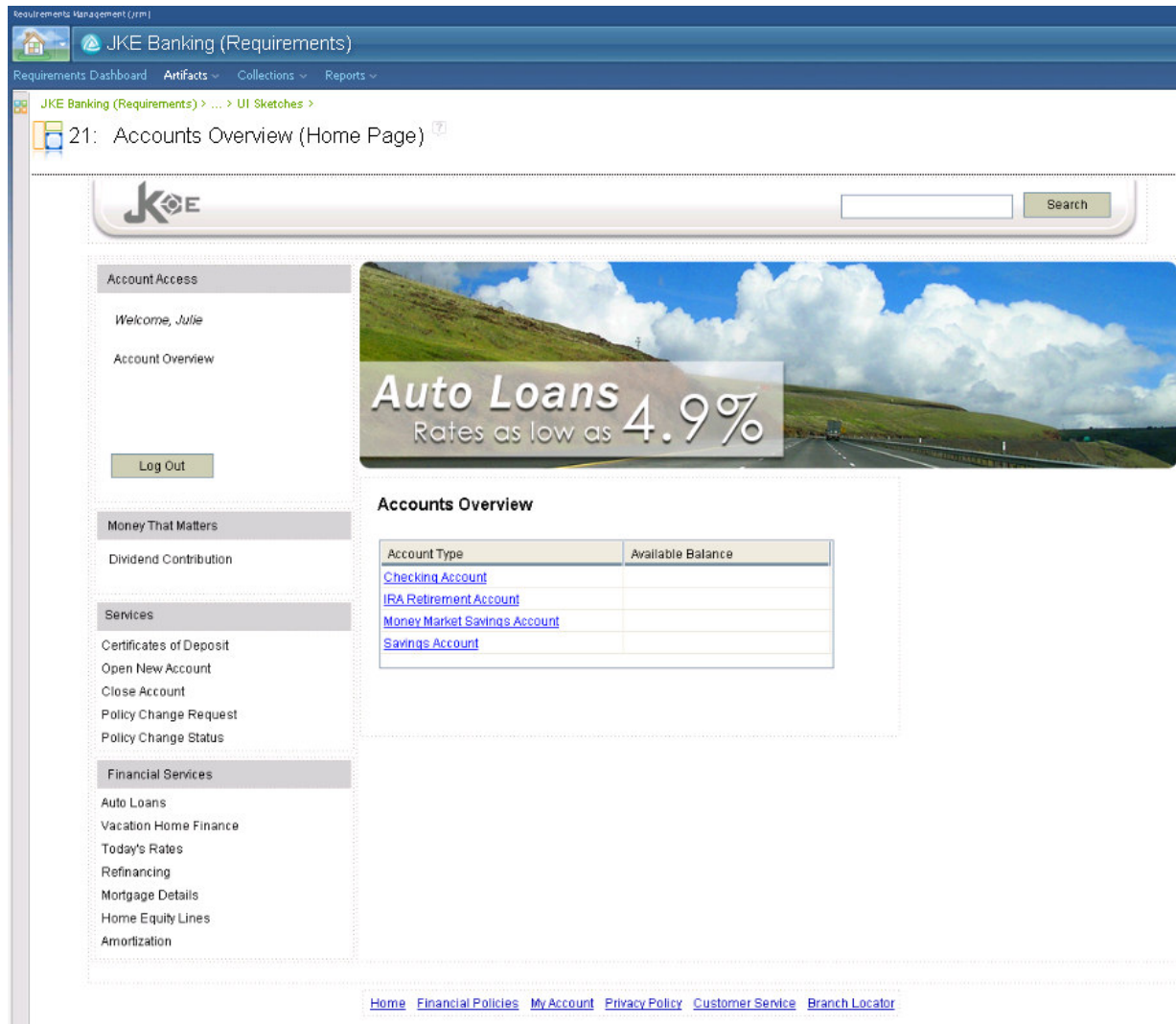


# Example: re-use in JKE Banking sample project



Parts within "Home Page Frame" sketch outlined in red

# Example: re-use in JKE Banking sample project



“Accounts Overview (Home Page) sketch uses “Home Page Frame” sketch, and adds additional parts.

# Example: re-use in JKE Banking sample project

The screenshot shows the Requirements Management (JRM) interface for the JKE Banking project. The main storyboard is titled "14: Dividend Contribution" and contains several frames. A yellow callout box highlights a frame with the text "1 Clicks Dividend Contribution". A larger yellow box on the right contains the text "Frames within storyboard then use page sketches". At the bottom, a frame list shows 14 frames, with frame 3 selected.

Account Access

Welcome, Julie

Account Overview

Log Out

Money That Matters

Dividend Contribution

Services

Certificates of Deposit

Open New Account

Auto Loans Rates as low as 4.9%

Accounts Overview

Account Type	Available Balance
<a href="#">Link Account</a>	
<a href="#">Retirement Account</a>	
<a href="#">Money Market Savings Account</a>	
<a href="#">Savings Account</a>	

1 2 3 4 5 6 7 8 9 10 11 12 13 14

## Tips and techniques for using sketching tools

# When exploring, create many design alternatives

- Ease of creating sketches should enable this
- Don't get caught up in details, especially early on; just make it “good enough to get the idea across”

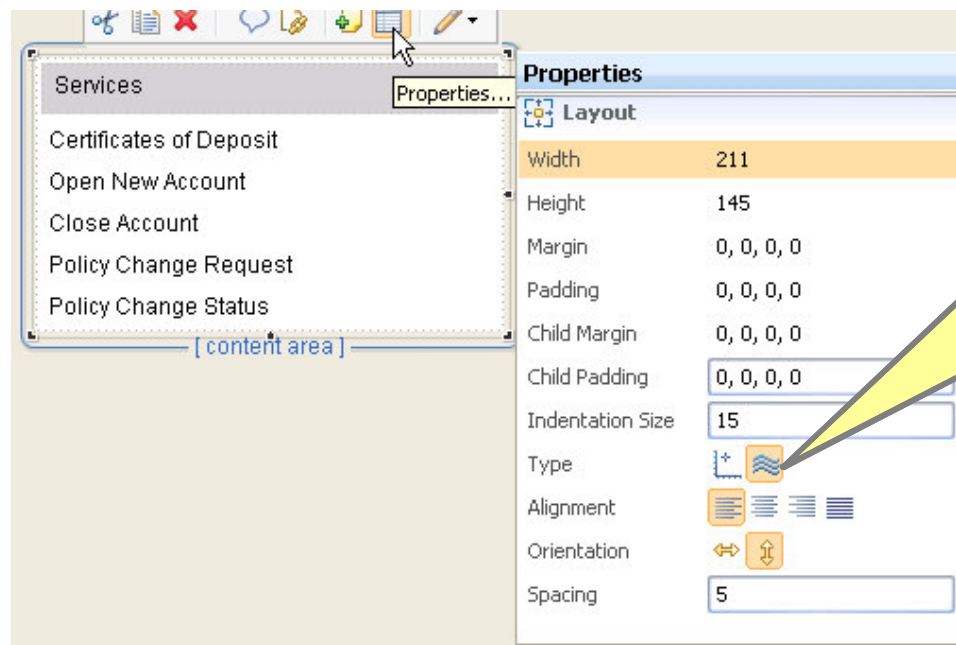
# Think twice before deleting sketches/designs

- Good to always keep design alternatives around
- Be sure to keep them separate from official designs though

## Tips and techniques for using sketching tools

### Use Flow Panels for automatic layout

- Especially when you will be adding/removing things frequently from a container in a storyboard
- Can use nested flow panels
  - ▶ Set padding and margins to avoid too much white space

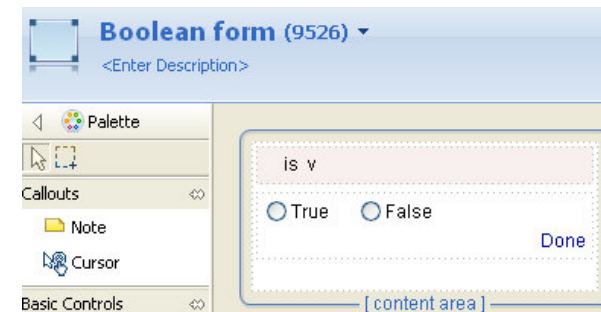
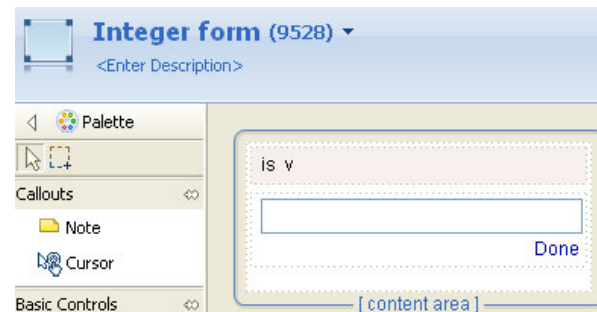
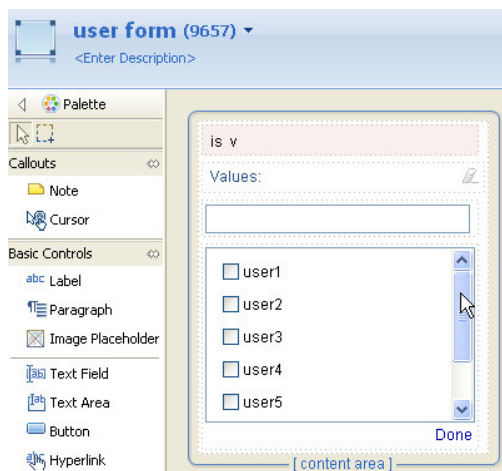


From Properties, click 2nd icon to set Type to “Flow”. Can also set alignment, orientation and spacing.

## Tips and techniques for using sketching tools

- Use parts to create your own “palette” of controls
- Improves efficiency in creating new sketches
- Improves consistency across UI designs and implementations

### Examples



## Tips and techniques for using sketching tools

### “Share” sketching designs early

- ▶ Make it clear that it's in progress, but make sketching work visible early
- ▶ Make use of commenting and reviews

### Add external visual artifacts to your repository

- ▶ If you create visual artifacts (e.g., high-fidelity UI design prototypes), add them to your Requirements Composer repository.
- ▶ If they are multiple images representing a work flow, paste the images into storyboard frames
  - ▶ Easier to navigate the flow than looking through a series of images
  - ▶ Can take advantage of commenting, reviews, linking, etc.
  - ▶ Can access via web

## Tips and techniques for using sketching tools

# Some lesser-known, but useful, techniques

Arrow keys will move focus between UI elements

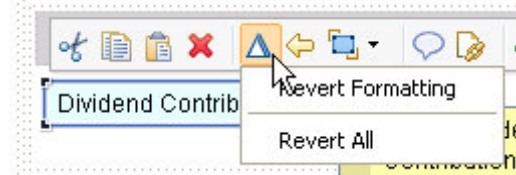
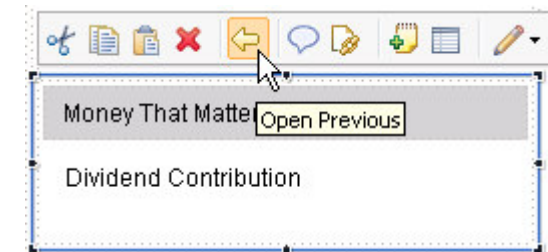
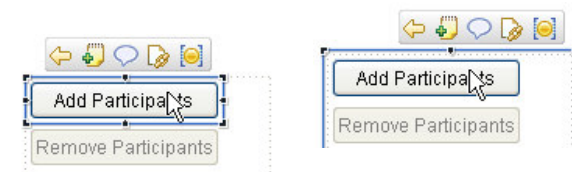
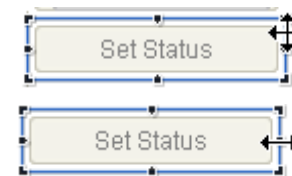
To use arrow keys to “nudge” UI elements, press “.” (period) first. Press Enter when done (or Esc to cancel move).

To use arrow keys to resize UI elements, press “.” multiple times until the cursor is on the side you want to move; then use arrow keys to resize UI element.

Alt + Up arrow to move focus to parent container, Alt + Down arrow to move focus within container

Open inherited sketches and parts using Previous button in contextual toolbar

Reset override values of inherited elements using Revert Changes button in contextual toolbar





## Tips and techniques for using sketching tools

Some general changes in RRC Next that affect sketching editors

- Everything is web based.

- Web interaction vs tabbed IDE interaction

- Read mode vs edit mode

- Editor toolbar has moved to within editor

- Customizable type system

# Tips and techniques for using sketching tools

## Some changes to sketching editors in RRC

### Next.

#### Palette

Palette on top instead of left side.

Grouping of elements in palette

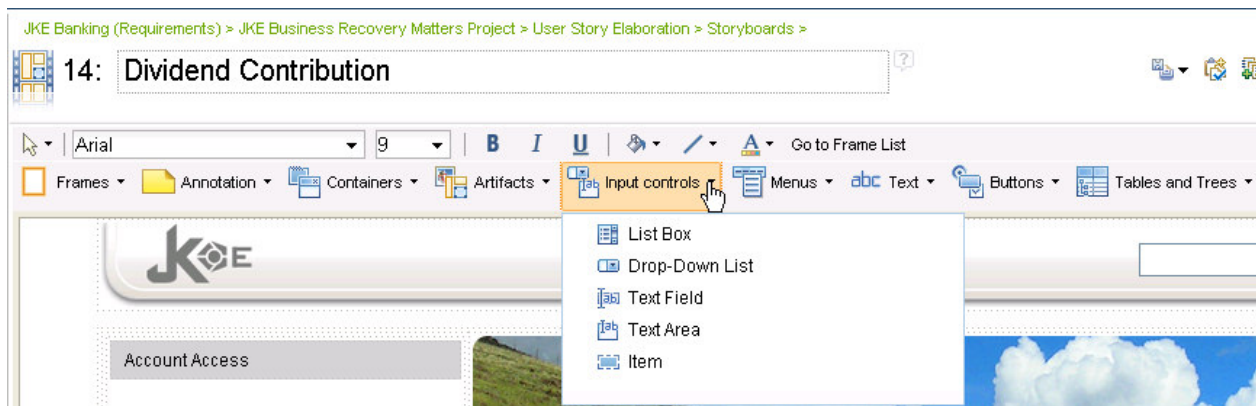
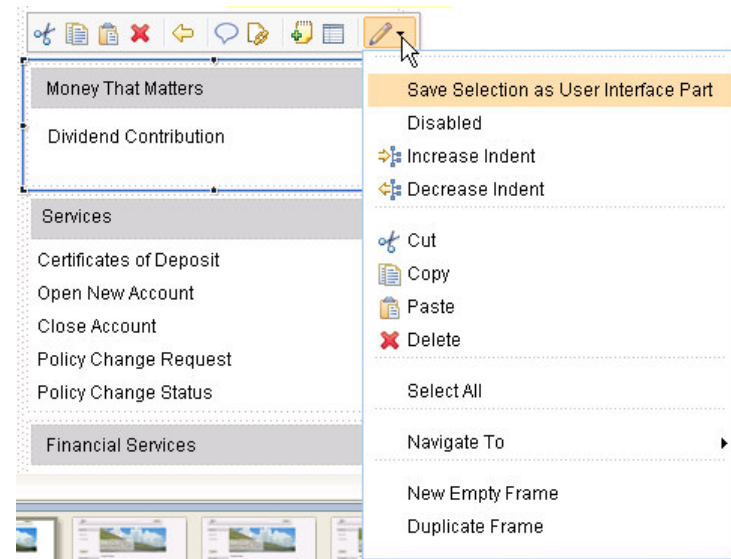
Slightly different palette interaction

#### Element interaction

No context menu

Expanded contextual toolbar

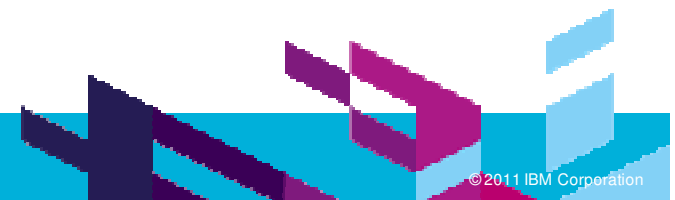
#### Frame interaction slightly different



# QUESTIONS



[www.ibm/software/rational](http://www.ibm/software/rational)





[www.ibm/software/rational](http://www.ibm/software/rational)

© Copyright IBM Corporation 2011. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.