Workspace Design Document

Kenny Gruchalla

1.0 Introduction

This document describes the design for the re-implementation of the RiverWare workspace. The RiverWare workspace is the RiverWare main window and provides the primary view of the simulation space where models are built and modified.

The impetus behind this work is the need to visualize RiverWare accounting networks. The current workspace architecture cannot support the visualization of RiverWare accounting networks. Furthermore, the RiverWare workspace lacks many modern GUI features that users have come to expect in a commercial software application. In effort to modernize the RiverWare workspace and provide support for new types of information visualization (i.e., the visualization of the account network), the RiverWare workspace is being redesigned.

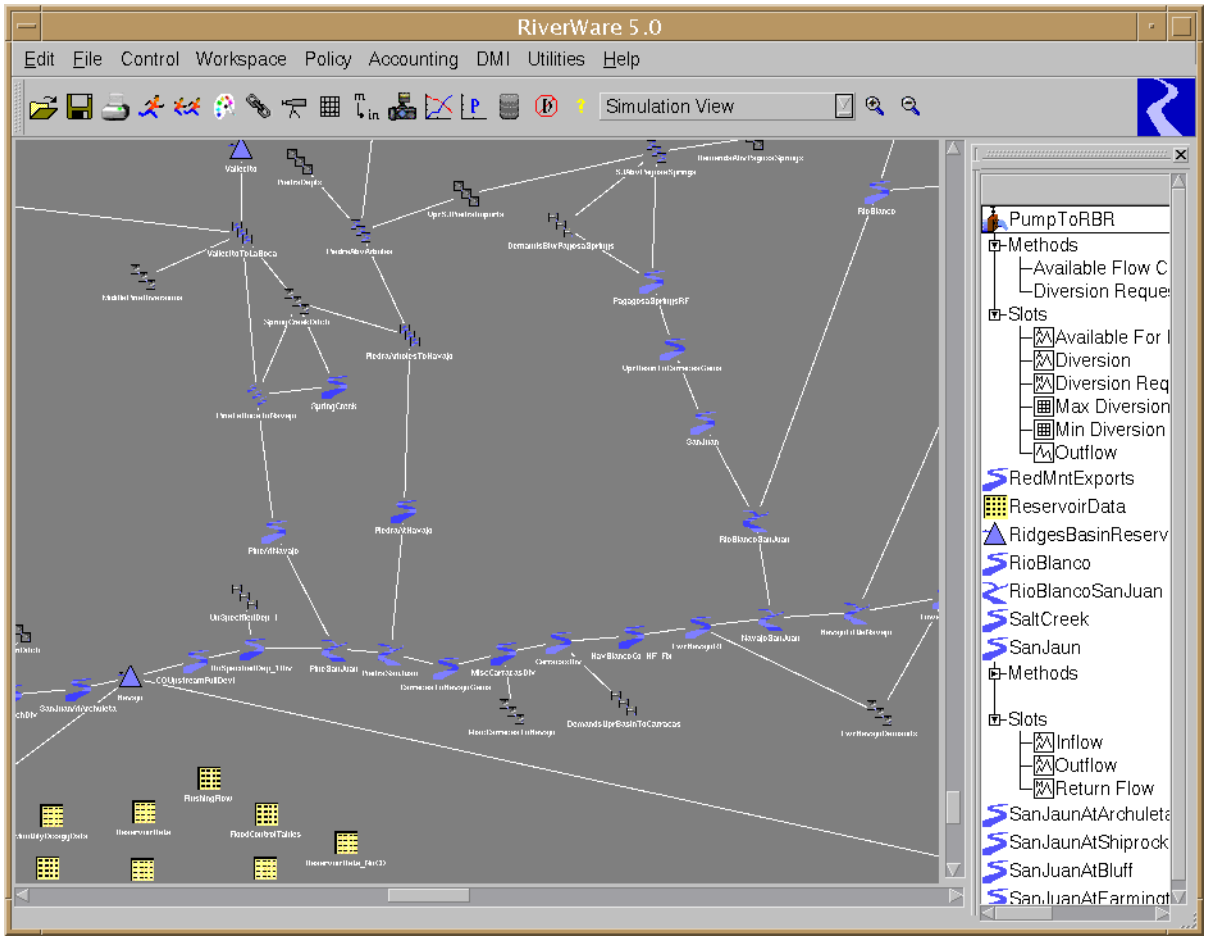


Figure 1: Artist's conception of the new RiverWare workspace.

2.0 Functional Requirements

2.1 Existing Functionality

The new workspace shall support all the functionality currently provided by the RiverWare workspace.

2.1.1 The workspace shall provide a menu bar with the same layout and a superset of the functionality of the current implementation.

2.1.2 The workspace shall provide a toolbar that provides a superset of the functionality of the current implementation.

2.1.3 The workspace shall be capable of displaying the simulation model as a network of object icons with links connecting the objects.

2.1.4 The workspace shall be capable of creating simulation objects by dragging them from an object palette.

2.1.5 The workspace shall provide a model overview functionality (locator view) with a search feature.

2.1.6 The workspace shall provide scroll bars as a mechanism to pan across the model.

2.1.7 The workspace shall provide the ability to select objects on the workspace by selecting a region, or selecting individual objects.

2.2 New Requirements

The following requirements were derived from the workspace functional analysis document*.

2.2.1 The workspace shall provide a rich zooming interface, which should include the ability to zoom-in on an area by selecting the area with a drag box, as well as, predefined zoom level buttons to zoom in and out.

2.2.2 The workspace shall provide an ability to reconfigure the size of the model canvas.

2.2.3 The workspace shall provide context sensitive popups

2.2.4 The workspace shall provide the ability to hide links, and stylize links.

2.2.5 The workspace shall provide the ability to print the graphical representation of the workspace. The print feature shall be capable of printing a selected region, the current view of the model, or the entire model canvas.

2.2.6 The workspace shall provide the ability to export an image of the graphical representation of the workspace. The image export feature should be capable of exporting a selected region, the current view of the model, or the entire model canvas. Standard (open) graphics image formats should be supported including: PNG,

*.The workspace functional analysis document can be found at: /projects/riverware/doc/guiRework/workspace/workspaceFA.fm

BMP, JPG, PMB, PGM, PPM, XMP, XBM.

3.0 Design

The ultimate goal of the workspace re-architecture is to design for change. There are many different visualization and interactive features that are applicable to the RiverWare workspace. The visualization of the accounting network is just one example. RiverWare users have expressed interest in GIS features, animation features, and annotation features. The goal is to design a framework that is flexible enough to accommodate these and other advanced visualization features. This is in contrast to the current design that provides only a single static view of the model.

The design recognizes that not all visual data representations are compatible. That is, there maybe visualizations that are fundamentally different and cannot occupy the same space simultaneously. Therefore, the design provides the idea of an information view. These views will be unique visualizations into the RiverWare data. The idea of an information view can be thought of as a coordinate system. We may want to view various elements of a RiverWare model in one coordinate system, and view a superset of those elements in a different coordinate system. Each view must be displayed independently since we cannot view two coordinate systems simultaneously without conflict. For example, in the future we may want to provide a three-dimensional perspective view in the workspace. This is a completely different view than say the traditional two-dimensional plan view currently used by RiverWare. Presumably both views would provide different benefits -- the two-dimensional view would likely provide a better modeling interface, while the three-dimensional might provide a better analysis interface. Therefore, the design will support the idea of multiple information views. The user will be able to switch between views using an option menu located on the toolbar.

In order to support a wide range of possible visual representations, the design aims to support two types of Qt drawing areas: QCanvas and QGLWidget. The QCanvas provides a high-level two-dimensional drawing area interface. The QCanvas is highly optimized and is capable of handling large collections of items. These items can have arbitrary shape, size, and content. The QCanvas should be sufficient to handle most of RiverWare's current visualization needs. The QCanvas will be used as the drawing area for all the of the work described in this document. However, QCanvas would be likely be insufficient to handle a graphics intensive or three-dimensional representation of the workspace. Therefore, the design will provide hooks to support the use of a QGLWidget in the future. The QGLWidget provides access to OpenGL, the standard API for three-dimensional graphics rendering. With the addition of an OpenGL enabled window, this design will be capable of supporting the state-of-the-art visualizations of the RiverWare model, if the need arises.

~~The design supports the idea of information layers. Layers can be thought of as transparencies that can be laid one on top of the other and interactively enabled or disabled. Any particular information view might consist of multiple layers. Layers would provide users the means to organize and selectively view elements of the data. For example, users might want to move all links to the Thermal object to a separate hidden layer to reduce the visible clutter. The initial design includes only a single type of layer whose content will be user defined; however, future capabilities (e.g., model annotation) may introduce additional types of layers whose membership maybe automatic.~~
After some review, the concept of layering was replaced by the display groups feature (see the links as first class objects design document).

The RiverWare workspace will support dock windows. A dock window is a dialog that can be docked and integrated into the main window. Every dock window has a handle that is represented by two gray lines. Using these handles, users can move dock windows within the main window. Users can also detach the dock windows, treating the window like a separate dialog. Toolbars are commonly dock windows, and the RiverWare toolbar will be a dock

window. In addition, RiverWare will support several dock windows including the locator view and a simulation object list view.

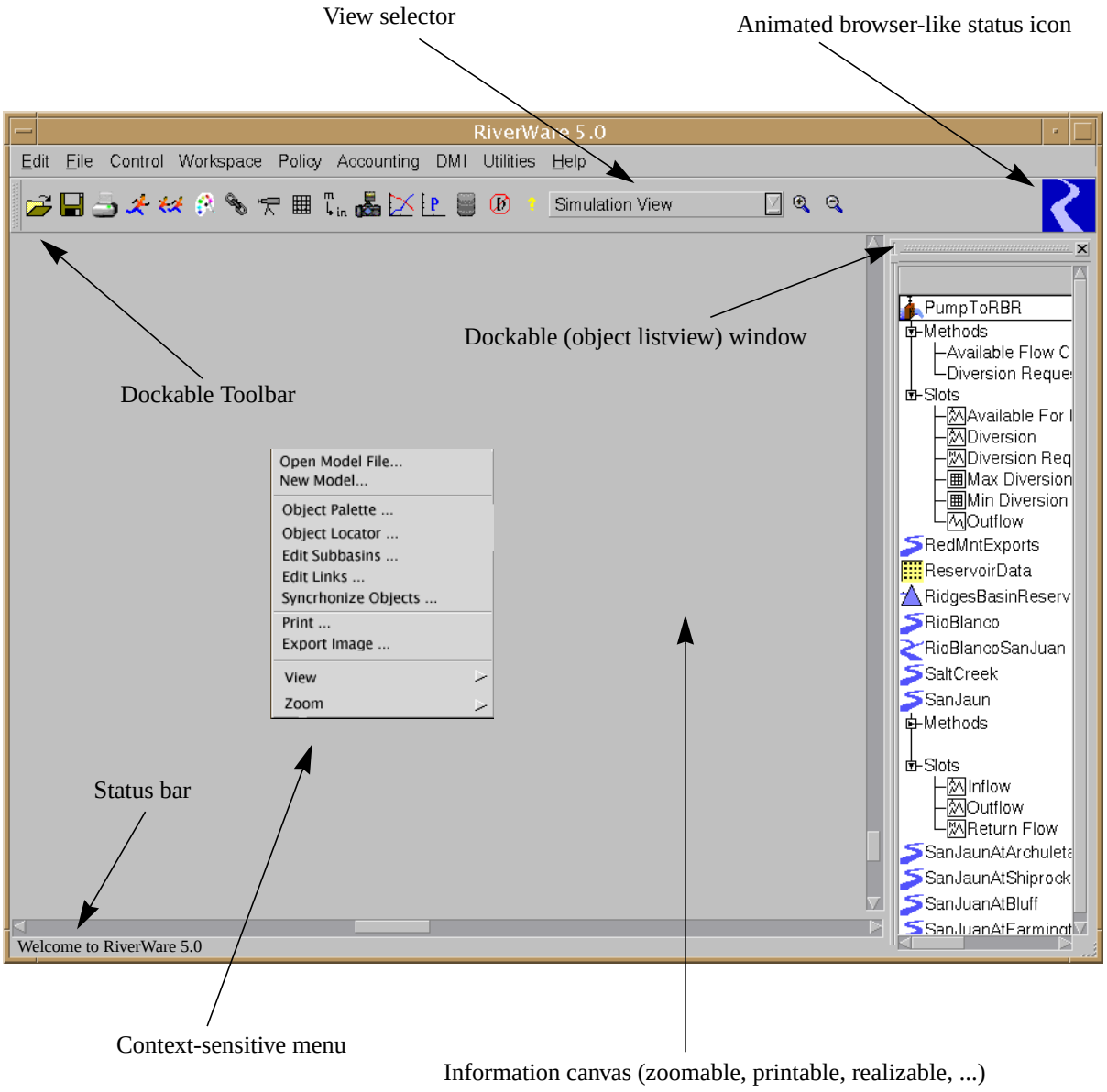


Figure 2: Annotated workspace dialog.

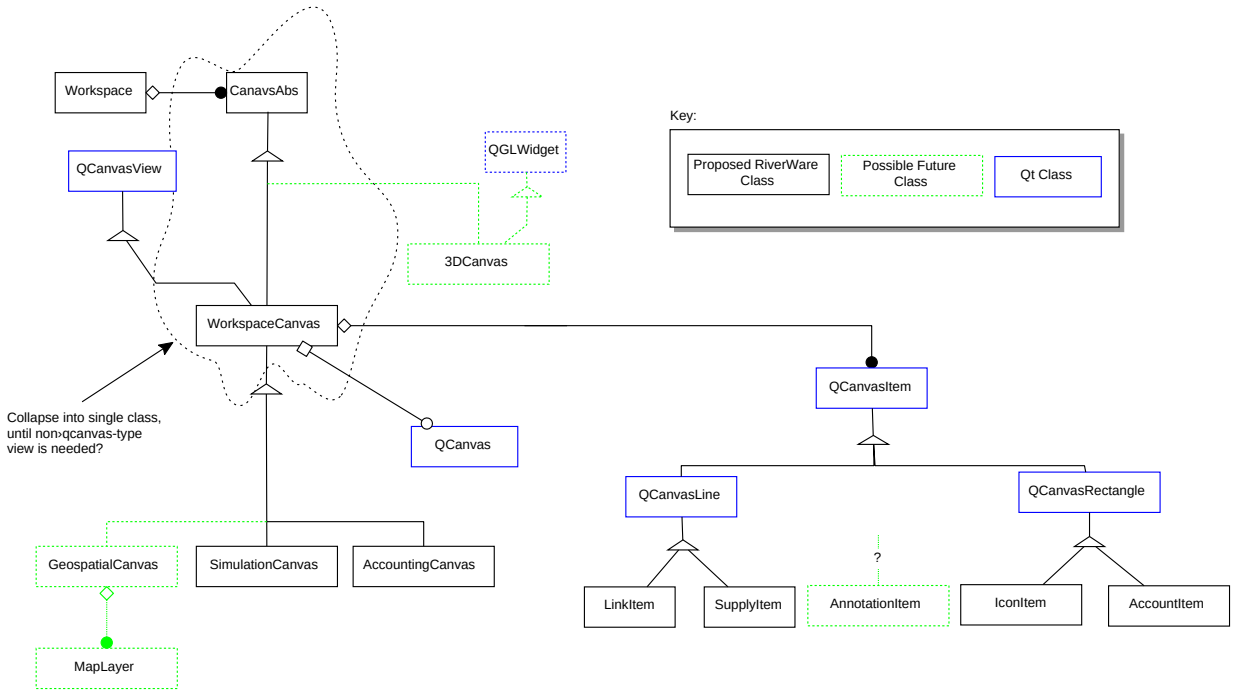


Figure 3: Class diagram for new workspace related classes.

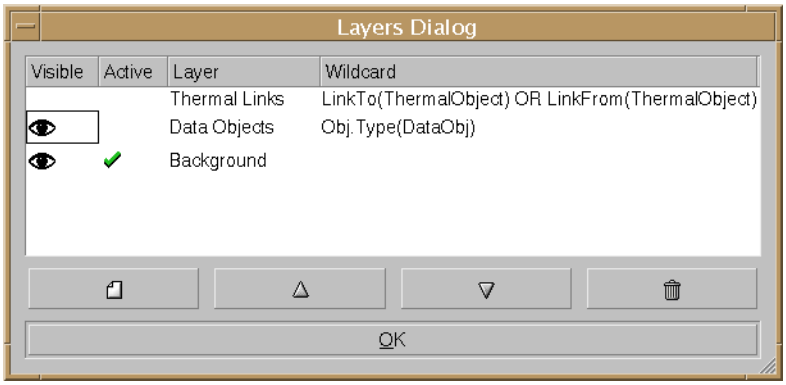
3.1 Layers

No layering scheme was implemented for the initial version of the workspace. The z-component of the QCanvasItem is used to ensure that supplies and links do not occlude underlying objects and accounts. Only the relative z ordering of icon objects can currently be controlled by the user. All display attributes of the workspace items (objects, accounts, links, supplies) are controlled by display groups (see the links as first class objects design document). Layering may be necessary for future GIS efforts, but was not considered necessary for the current functionality.

~~Layers can be thought of as transparencies that can be laid one on top of the other and interactively enabled or disabled. The concept of layers is common in most GIS programs and many image manipulation programs (e.g., Photoshop). The initial design includes only a single type of layer, a ModelLayer. A ModelLayer type will contain model elements (e.g., simulation objects, links, accounts, supplies, ...). An information view may consist of many of these layers. The composition of each layer will be user defined using one of two methods: manually or through dynamic membership selection. Selected objects can be moved to other layers through the context-sensitive "Move To" drop-down menu, which will provide a list of applicable layers. Each layer will also accept a wildcard regular expression that describes the objects, which should be included in that layer. There will be one layer that will be considered to be the "active" layer, where new objects will be placed by default, if no wildcard expression applies.~~

~~Visual elements (e.g., simulation objects, links, accounts, supplies, ...) may exist in multiple layers simultaneously. Changes to the elements' attributes will be propagated to all layers that contain that element. For example, if the position of simulation object is moved in one layer, the position of that object will be updated in all layers in which it is contained. Simulation objects and links will be treated as individual objects; therefore, hiding object will not automatically hide its associated links (and vise versa).~~

~~Layers have an ordering. Objects in higher layers will occlude objects in lower layers. The layer ordering will be controlled through the Layers Dialog (see Figure 4). Also through this dialog, the visibility of each individual layer can be controlled. That is, each layer can be visible, meaning that its constituent objects will be drawn; or invisible, meaning that each of its constituent objects will not be drawn.~~



~~Figure 4: Layer Dialog~~

3.2 Views

As mentioned, a information view is a high level entity that provides an unique view into some aspect of a RiverWare model. It may consist of multiple layers. Initially, two views will be available to the user:

1. A Modeling / Simulation view that presents the user with the model topology through simulation objects and links.
2. An Accounting view that presents an augmented model topology complete with accounting information. (Note: the requirements for the accounting view are still in the preliminary stages. Accounts might be presented as separate layers in the simulation view; however, most likely, the model topology will need to be represented with an expanded coordinate system to spatially accommodate the accounting information necessitating the need for an separate view.)

These views will be accessible through an option menu on the toolbar. The user may also switch between views via the context-sensitive menus.

3.3 Zooming & Panning

Large models require views at various levels of detail. RiverWare will provide the user with the ability to zoom in and focus on a particular area of the model, but also be able to zoom out to understand the context of the detailed areas. The zooming scheme will be the same as that currently used by the Plot dialog. User will be able to select a

rectangular region to which the viewable area of the model will snap. Two buttons will also be provided on the toolbar, allowing users to gradually zoom in and out. In addition, predefined zoom levels will be accessible through the main menu and a context sensitive menu.

3.4 Canvas Configuration

The size of the model canvas will automatically grow as objects are dragged past the extents of canvas. User may also adjust the size of the model canvas through the canvas configuration dialog. In addition, the user will have control over the background of the canvas. User's will be able to control the color of the background, and they also have the option of importing an image into the background (see Figure 5).

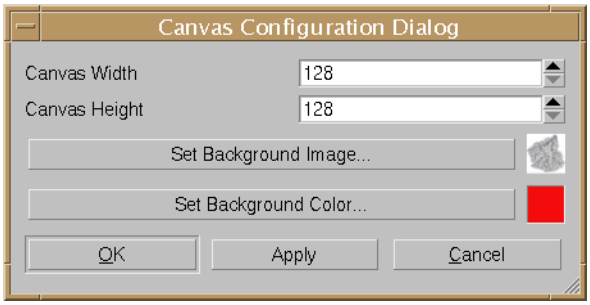


Figure 5: Canvas Configuration Dialog

3.5 Context-sensitive Popups

A context-sensitive menu is a menu that appears in response to a user action (typically, a right-mouse button click). The menu contains functions (that may also be available on in the main menubar, or toolbar) that are particularly relevant to the area of the window or object from where the click originated from. The original galaxy workspace did not support context-sensitive menus. The new workspace will support context sensitive menus ubiquitously.

3.5.1 Over Workspace or Workspace selection

Button	Enabledness	Comment
Import Objects...	Always	
Export Objects...	When objects are selected	
Synchronize Objects...	Always	
Select Object Methods...	Always	
Object Palette....	Always	Or possibly replace with a drop down icon menu
Locator View...	Always	
Edit Links...	Always	
Edit Subbasins...	Always	
Print...	Always	
Export Image...	Always	
Zoom >	Always	drop down > Zoom in, Zoom out

Button	Enabledness	Comment
View >	Always	drop down > listing of available data views
Move to >	When objects are selected	drop down > listing available data layers
Copy to >	When objects are selected	drop down > listing available data layers

3.5.2 Over Object

Button	Enabledness	Comment
Open Object...	Always	
Delete Object	Always	
Create Link...	Always	Possibly provide linkable slots via drop down?
List Subbasin Membership...	Always	
Export Object...	Always	
Object Color >	DataObj Only	Only visible for data objects
Move to >	Always	drop down > listing available data layers
Copy to >	Always	drop down > listing available data layers

3.5.3 Over Link

Button	Enabledness	Comment
Edit Link...	Always	
Delete Link...	Always	
Link Manager...	Always	
Move to >	Always	drop down > listing available data layers
Copy to >	Always	drop down > listing available data layers

3.6 Links

Links are the connections between objects that pass information between objects during a simulation. A link is represented on the workspace by straight line that connects two objects. In the new workspace design, a link will be a first class GUI object (see class diagram). As a first class object the user will be able to select and modify the links. User will be able to control the line style, line width, and color of the links (as shown in Figure 4). Support will be added to the Simulation library to classify types of links. This will enable users to select and stylize links based on the link's type (e.g., main flow, return flow, diversion, ...).Visibility of the links can be controlled by moving the links to hidden layers.

Future enhancements may also include an alpha channel on the link's color. This would allow users to control the transparency of a links.

Open Issues: Should we support curved links, in order to show multiple links between objects? Qt does support splines and bezier curves with in the QCanvas. Therefore, this should be tractable in the current design framework.

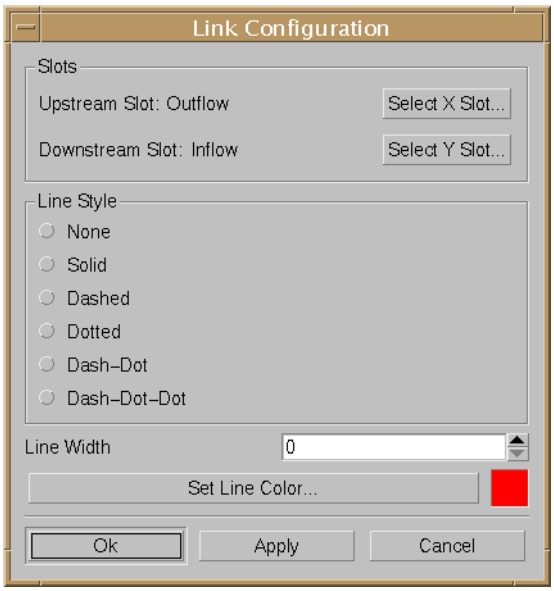


Figure 6: Link Configuration dialog.

3.7 Printing & Image Capture

Both printing and image capture will rely on the QPainter interface. Both printing and image capture will be available through the File menu and the context-sensitive workspace popup. The user will be able to print or capture the entire workspace, only the visible workspace, or a selected area of the workspace. The image export feature will support all the image formats supported by Qt (i.e., PNG, BMP, JPG, PMB, PGM, PPM, XMP, and XBM).

Open Issues: How do we want to handle the image scale when printing? How do we want to handle printing across pages?

3.8 Object / Slot Locators

The current locator view serves two purposes. It provides a global overview of the model, and provides an ordered list of objects (a rudimentary search facility). Two new dialogs will replace the old locator view. The new object locator dialog will provide a global overview (complete with current functionality) and an advanced search feature. The object locator will be dockable. A dockable object list will also be provided, providing the user with a more familiar sorted interface.

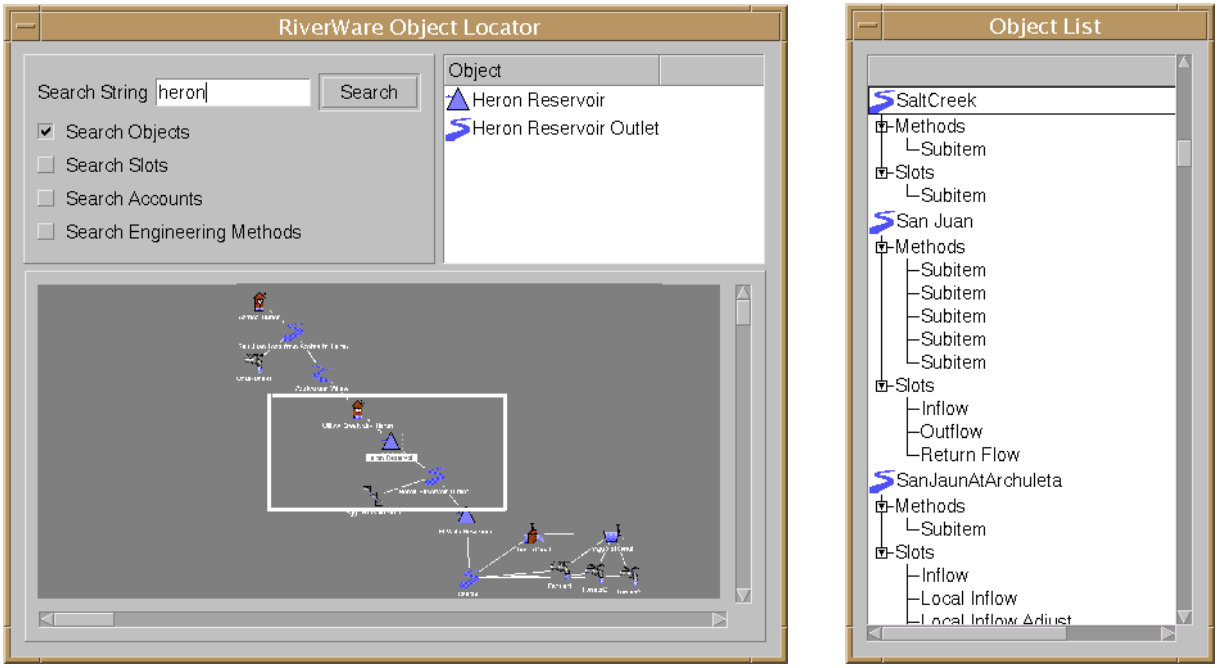


Figure 7: Snapshots of the Object Locator and the Object List dialogs. Both dialog will be dockable in the main RiverWare workspace

3.9 Copy, Cut & Paste

The requirements do not currently specify any cut and paste behavior. In the future, we may want to allow the copy and paste of objects between instances of RiverWare.

3.10 Menu Bar, Tool Bar, & Status Bar

The RiverWare menubar and toolbar will remain largely unchanged. The accelerator keys will be updated to reflect the newly adopted RiverWare GUI Standards. Tooltips will be provided for all of the toolbar buttons.

A status bar will be added to the RiverWare workspace. The statusbar provides a single line of read-only text that provides status information to the user.

The RiverWare icon will be animated and will communicate the status of the run. The icon will depict running water through a meandering channel, while model is being run. The icon is similar in spirit to the animated icons found in most internet browsers (e.g., windows explorer, netscape, ...).

3.11 Workspace Object Selection

Users will be able to select and deselect objects on the workspace by single clicking on individual objects. The users will be able to select multiple objects by dragging a rectangle around objects of interest. The selection mechanism will adhere to standard user interface practices, allowing users to select multiple objects with use of the control key.

4.0 Open Design Issues

4.1 Links

- Should we support the display of multiple links between objects?
- Links and supplies are similar concepts. Supplies may or may not be handled differently than links in the accounting visualization. What issues need to be considered at this stage of the design for supplies?

4.2 General

- How do we want to address model file bloat?
- How should we handle zooming/scaling issues when printing?
- How should we handle printing across pages?

5.0 Resource estimates

These estimates do **not** currently consider any of the above open design issues.

Task	Time Estimate (8 hour days)
Main Window GUI (Toolbar, Menubar, Statusbar)	9 days
Layer & View Object structure	15 days
Canvas Callbacks	4 days
Zooming interface	5 days
Layer Interface[a]	7 days
Canvas Configuration	2 days
Locator & List views	5 days
Link Configuration[b]	3 days
Printing & Image Capture	TBD
Total	**50 days**

a. Estimate does not include dynamic wildcard selection

b. Estimate does not include link type work in the simulation library