Accounting Visualization Design

Kenny Gruchalla

1.0 Introduction

This paper presents the design for providing a visualization capability of the accounting network in RiverWare. RiverWare supports two parallel representations of water: physical water and paper water. The physical water is modeled in the simulation objects, and the network of the physical water is displayed on the workspace through simulation object icons and links. The paper water represents water types and ownership and is modeled in the accounting network. RiverWare does not provide any visualization of this accounting network. Currently, the only mechanism to explore the accounting network is through large tables provided on various accounting dialogs. This paper proposes a design to incorporate a graphical view of the accounting networking into the RiverWare workspace.

2.0 Functional Requirements

The functional requirements are presented in a separate document:
/projects/riverware/doc/guiRework/accountingVis.fm

3.0 Interface Design

The accounting network will be presented in a separate *view* from the standard schematic depiction of the physical network, or *schematic simulation view.* These two views can be thought of as different coordinate systems. The accounting view will provide a representation of the physical objects, as well as, the accounts on those objects. In most RiverWare models, representations of the accounts could not be added directly to the schematic view without occluding much of the information that we are trying to present. Therefore, the accounts will be presented in the workspace as a separate view where the network can be generated without occluding or modifying the original physical network. The workspace view will be toggled through a option menu on toolbar, or though the RiverWare menu system (see Figure 1).

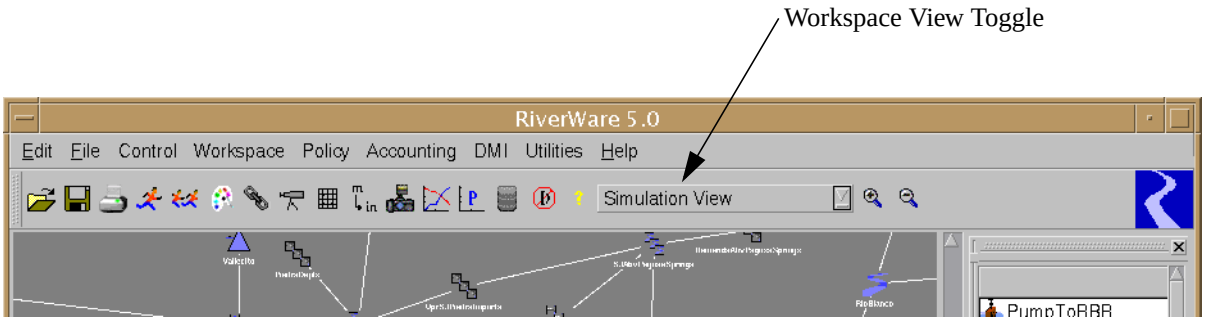

Figure 1. RiverWare toolbar. The accounting view will be accessed through an option menu on the toolbar. If accounting is disabled, the view option menu will be hidden.

3.1 Account Representations

Accounts in the accounting view will be displayed in the *object territory* of its parent simulation object, providing a spatial association of the account with its simulation object. An object territory will be an rectangular box with the object icon and name in the upper left corner. Icons representing the accounts on that simulation will then be displayed within the territory in a single row (see Figures 2 and 3). The position of the accounts in the territory will be user configurable through a drag and drop operation; however, RiverWare will provide an initial default ordering.

Initially, the accounting visualization will be represent three account types: storage accounts, pass through accounts, and diversion accounts. Storage accounts will be represented by a rectangular icons, pass through will be represented by elliptical icons, and diversion accounts will be represented by an isosceles trapezoid. The account name will be presented within its icon (see Figure 3).

The accounting visualization will represent supplies by a line. The directional supply-demand relationship between accounts will be represented by placing an arrow head on the demand-side of the supply (see Figures 2 and 3). Transfers between accounts on the same object will be represented by a curved line (see Figure 2) for clarity -- straight lines would conflict with accounts in the territory.

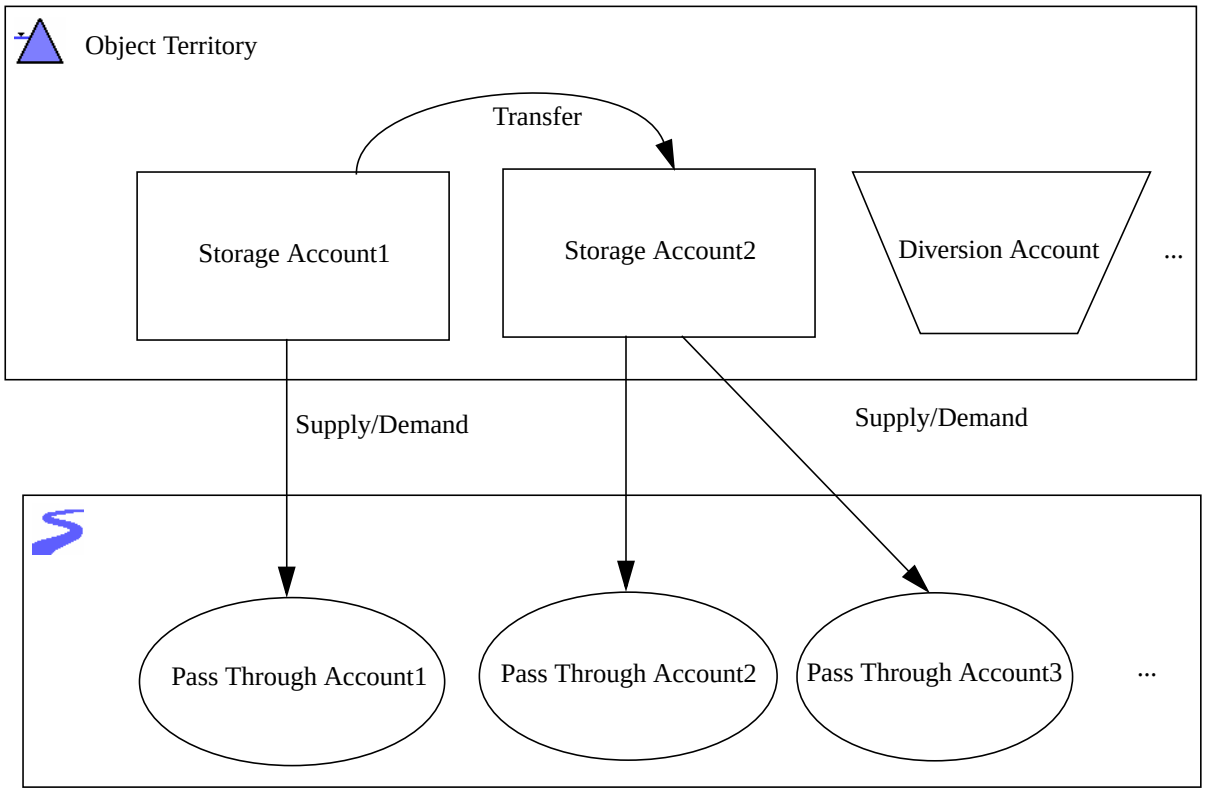

Figure 2. Account representations. Accounts will be represented by geometric icons. Rectangular icons for storage accounts and elliptical icons for pass through accounts. All the accounts belonging to an object will be encapsulated by a large rectangle “territory” representing the object. Supplies will be represented by lines, and direction indicated by an arrowhead.

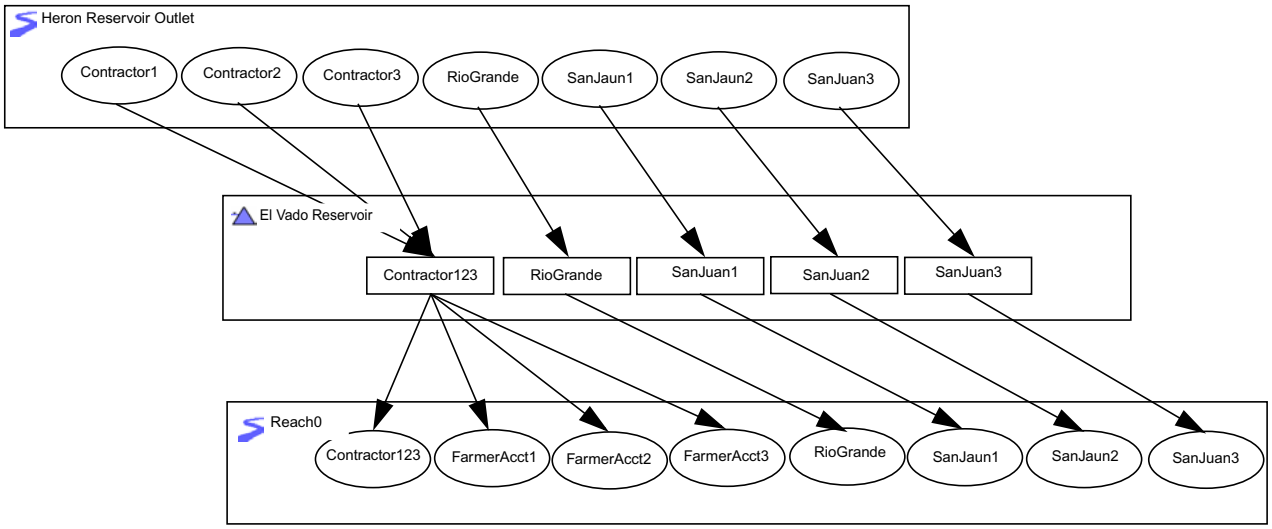

Figure 3. Example Account Depiction. A mock-up of the accounting representation using three objects taken from the accounting regression test: accountingHeronInflow.

The position of the object territories will be user configurable through a drag drop operation just as icons can be moved in the schematic view. In addition, RiverWare will provide a capability to automatically layout the accounting network. This capability will be used the first time the accounting view is instantiated for a model, and the user will be able to initiate this capability from the menu system. The layout will be based on position of the objects in the schematic view. The algorithm can be thought of a simple scaling the x (primarily) and the y axes to fit the extended representation of the simulation object (see Figure 4). That is, the automatic layout algorithm will place the object territories in the accounting view relative to the positions of the simulation objects in the schematic view. The topologies of the simulation and accounting networks may have significant differences; therefore, the positions of simulation object representations are not coupled, allowing users to move the object territories independently of the simulation objects in the schematic view after the layout has been performed.

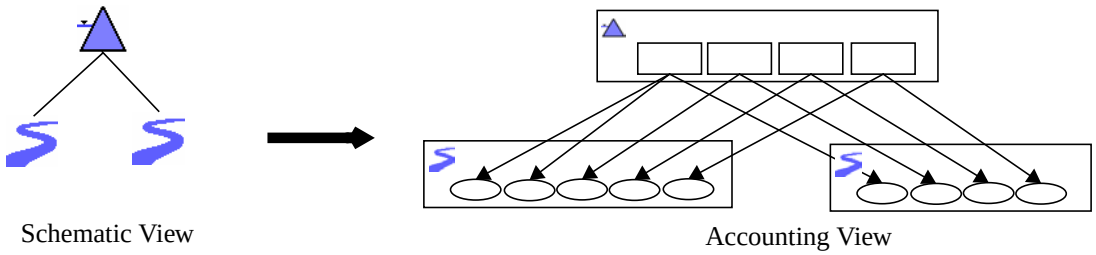

Figure 4. Depiction of automatic layout. RiverWare will provide an automatic layout capability that will place the simulation object representations relative to the positions of the simulation objects in the schematic view by scaling the x and y axes. The algorithm will be used on the first instantiation of the accounting view, and can be initiated by the user at any time. Users will also be able to manually adjust the positions of the object territories independently of the icons in the schematic view.

3.2 Display Properties

The display properties of accounts and supplies will be user configurable. Configurable account display properties will include:

- Border color - the color used to draw the outline of the account's geometric representation.
- Background color - the color used to fill the area within the account's geometric representation.
- Background fill pattern - the pattern used to fill the area with the account's geometric representation (e.g., solid vs. a crosshatch pattern).
- Maximum width - the maximum allowable width of the account's geometric representation. The geometric representation will stretch to accommodate the account's name up to this limit, at which point, the account name will be truncated. Tooltip'ing over the account will always provide the full name.
- Visibility - controls whether or not the account will be shown in the network.
- Depth / Canvas Layer - The depth at which the representation is drawn. This provides the user fine grain control over the drawing order of items on the canvas and will allow stacking (or layering) of items.
- Aggregation - a built in ability to aggregate the visual representation of one or more accounts.

Configurable supply display properties will include:

- Line Color - the color used to draw the supply.
- Line Style - the style used to draw the supply (e.g., solid line vs. dashed line).
- Line Width - the width of the line.
- Depth / Canvas Layer - The depth at which the representation is drawn. This provides the user fine grain control over the drawing order of items on the canvas and will allow stacking (or layering) of items.
- Visibility- controls whether or not the supply will be shown in the network

The display properties of the accounts and supplies will be configured the same way display properties of links and objects are configured in the simulation schematic view, through the display group editor. The display group editor allows users to define groups of objects and the display properties on those groups. The groups are prioritized, allowing an object to span multiple groups. The properties of the highest priority group containing an object will be used to "paint" the object. Default account groups will be provided foreach water type and water owner.

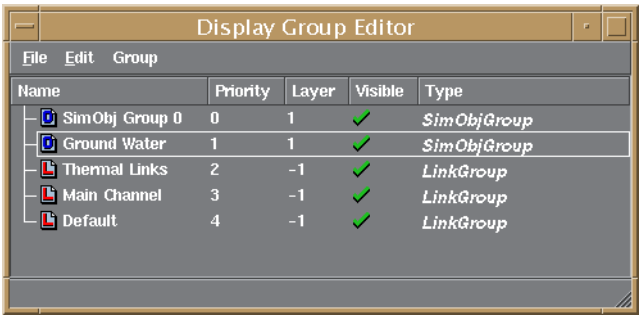

Figure 5. Display Group Editor showing link and object display groups for the schematic view. This will be extended to define account and supply groups.

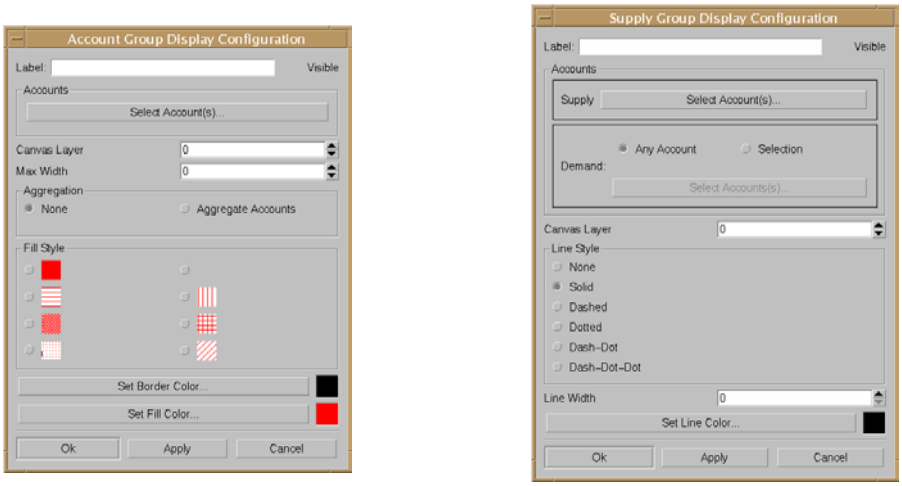

Figure 6. Account and supply group display configuration dialogs.

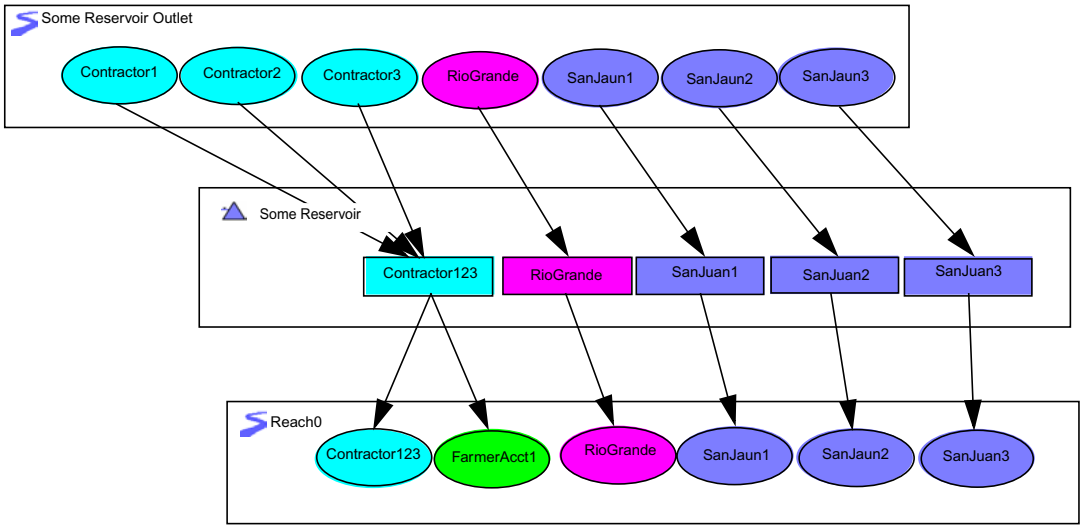

Figure 7. Contrived visualization example. Accounts colored by Water Owner.

3.2.1 Account Aggregation and Visibility

To provide level-of-detail manipulation, users will be able to aggregate accounts and hide accounts. When accounts are aggregated, all the accounts in the aggregation will be represented by a single icon. The aggregation will be represented by an icon that looks like a “stack” of two accounts (see Figures 8 and 9). This representation will be labeled with the name of the display group that aggregated the accounts. A tooltip will provide the names of all the accounts in the aggregation. Enabling or disabling the visibility of an account (or group of accounts) will hide the graphical representation of the account(s) entirely (see Figure 9). Supplies (if their visibility is enabled) will continue to depict routes even if intermediate accounts have been hidden (see Figure 9).

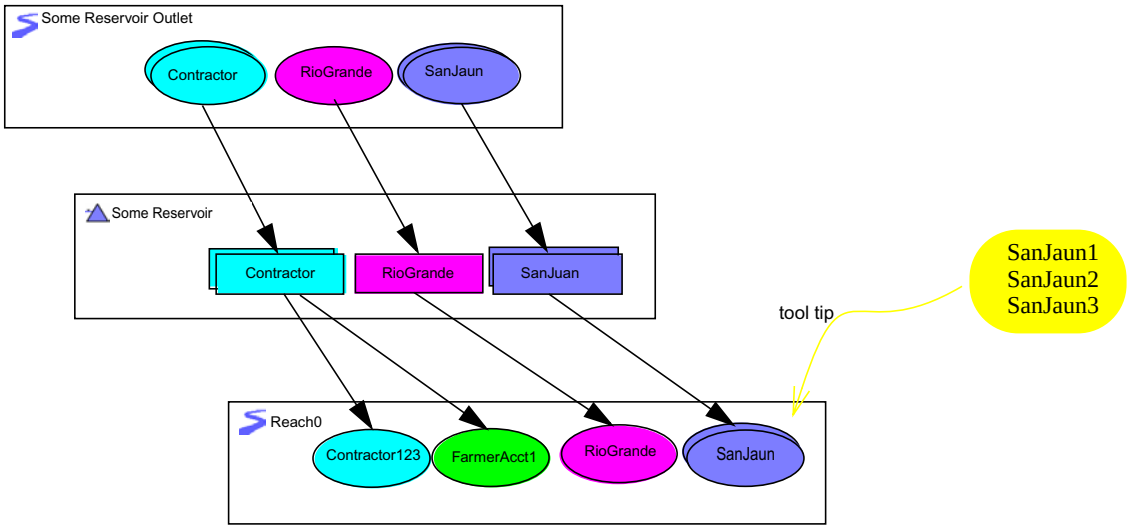

Figure 8. Aggregation example. An example of what Figure 7 would look like after aggregating on Water Owner. Tooltiping over Reach0's SanJaun aggregation will provide the name of all the accounts in the aggregation, as shown in the yellow tooltip.

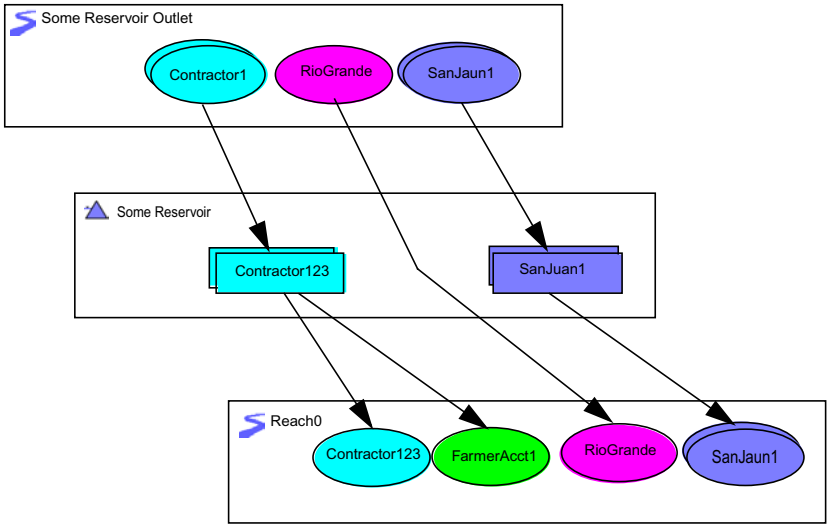

Figure 9. Account visibility example. The RioGrande storage account has been hidden (not visible). Notice the supplies to that account have not been hidden, and the ultimate route of the flow is still shown.

3.3 Interaction

The visualization of the accounting network on the workspace will provide the opportunity to allow direct manipulation of the accounts, much like current workspace interaction with simulation objects. As mentioned, the position of object territories will be editable through drag and drop operations, as will the positions of the accounts within the object territories. Double-clicking on an account icon will instantiate the account editor. Context-sensitive menus within the object-territories will provide the ability to create and delete an accounts on that object. Context-sensitive menus on the account icons will provide the ability to create supplies between accounts -- in a similar manner to the creation of links between objects in the simulation view.

3.4 Misc. Features

In addition to the graphical representations of the accounts on the workspace, accounts will be depicted in the workspace listview. When accounting is enabled, the workspace listview will provide a treeview with accounts shown as children of the simulation objects. The accounts shown in the listview are independent of any visibility or aggregration settings (i.e., all accounts are always accessible from the listview).

3.5 Time-dependent Visualization

This section is still under design. In addition to the static view of the network topology described above, the accounting view will provide a time-dependent visualization. This visualization will only show the supplies (and accounts?) with positive flows for the current (visualization) timestep. Timestep controls will need to be added to the workspace to control the current (visualization) timestep when in this mode.

4.0 High-Level Design

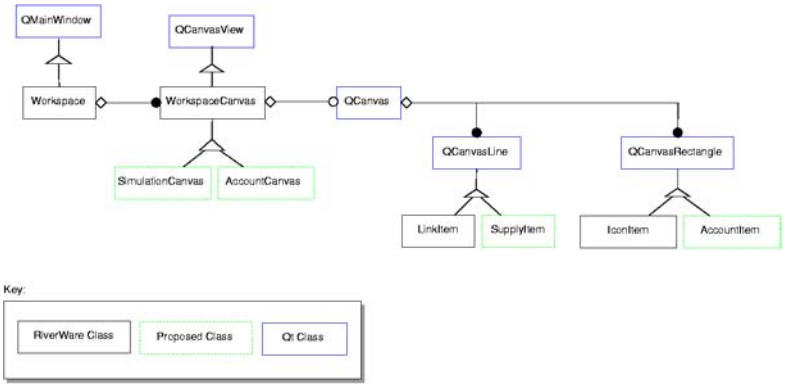

Figure 10. Class diagram that shows changes to the Workspace class that need to be made to support the accounting visualization.

The high-level design of the accounting visualization capability was largely captured in the original workspace design document. However, some details have evolved. The accounting visualization will be implemented using the Qt's QCanvas. The QCanvas provides a high-level two-dimensional drawing area interface, and is capable of handling large collections of items. The schematic simulation view is currently implement using a QCanvas. The main widget of the main window is a derived QCanvasView, a WorkspaceCanvas, that owns this QCanvas. To support accounting visualization the current WorkspaceCanvas class will be abstracted and the simuluation simulation-specific functionality will be moved into a derived SimulationCanvas class and the new accounting-specific functional-

ity will be put in a derived AccountingCanvas class (see Figure 10). When the user switches views the corresponding WorkspaceCanvas will be installed as the main widget of the workspace. The accounts and supplies will be QCanvasItems, derviving from QCanvasRectangle and QCanvasLine respectively. The graphical representation of the AccountItems and SupplyItems will be drawn using native Qt drawing commands, opposed to representing these as pixmaps like the simulation objects in the current simulation schematic view. This will allow greater flexibilty in the representations of the accounting system.

Note: Neither the current workspace nor this accounting visualization provide anti-aliasing. This was also true of the original Galaxy workspace. The lack of anti-aliasing is why diagonal lines on the workspace can appear jagged. Since Qt 3.x does not support anti-aliasing, we would have to develop our own anti-aliasing algorithms using the Qt drawing primatives. Anti-aliasing is supported natively in Qt 4.0*; therefore, the cost and effort to develop home-grown functionality does not seem warranted.

5.0 Time Estimates

task	~Hours
Development of the Account and Display display group configuration dialogs	20
Development of Group Display editor capabilities	10
Model File I/O	30
Development of CanvasItems	20
Development of layout algorithms	40
Development interaction features	30
Workspace integration	10
Time-dependent visualization	?
Documentation	20
Testing & Debugging	40
Total Estimate	220

*.Qt 4.0 was released June 28, 2005. Qt 4.0 represents a major upgrade in the API and will require a reasonable effort upgrade RiverWare. We will likely wait for the first few patch releases before considering an upgrade (i.e., let the Qt community find and deal with any major bugs/issues first).