RiverWare Workspace port to Qt4 Graphics View

Requirements and High Level Design

Phil Weinstein, CADSWES

This document describes the requirements for, and high level design of a port of the RiverWare 5.1 Workspace GUI from its Qt3 QCanvas-based implementation to a Qt4 “Graphics View” implementation.

0.1 Document Status

01-20-2009: High Level Design completed -- ready for review.
01-09-2009: Requirements section completed -- ready for review.

0.2 Other Relevant Documents

Source directory: /projects/riverware/doc/guiRework/workspace/

- Workspace Design Document [January 2006]
- Accounting Visualization Design [July 2005]
- Accounting Visualization Functional Requirements [June 2004]
- RiverWare Workspace Functional Analysis [May 2004]

0.3 Contents

1.0 Requirements

The Qt3 QCanvas-based components used in implementing the RiverWare Workspace need to be replaced by Qt4 “Graphics View” -based components. This includes:

- **Simulation View** (SimObj Icon items and Links between those items).
- **Accounting View** (SimObj rectangles containing Account shapes, with Supply Links).
- **Locator View** (Small, fully-zoomed out view with a draggable rectangle to scroll a main view).

All functionality of the Qt3 QCanvas-based implementations must be preserved. Most of this is enumerated in the following sections.

1.1 Simulation Object Palette

- A Simulation Object can be created by dragging an icon from the Simulation Palette to the Workspace.
- The Object is initially positioned at the “dropped” point in both workspace view geometries (i.e. simulation and accounting).
- If the palette icon being dragged is dropped outside of the Workspace window, then a Simulation Object is not created (i.e. the object creation operation is aborted).
- A drag operation can also be aborted by hitting the Esc (escape) key before releasing the mouse button.

1.2 Simulation Object Selection

- The set of Simulation Objects which are “selected” is globally maintained. The change of the selected SimObj set made in any of the workspace views, or within the Simulation Object listview, is automatically reflected in all of the others (workspace views and listview).
- During any selection operation, if the Shift-Key or Control-Key is pressed, then the previously selected Simulation Objects remain selected. Otherwise, they are deselected.
- An Individual Simulation Object can be selected by clicking on it.
- Clicking on the Workspace (not on a Simulation Object) causes all objects to be **deselected,** unless the Shift-Key or Control-Key is pressed.
- Rectangle “Rubber-Band” multiple-object selection is supported (by dragging a rectangle).

1.3 Simulation Object Moving

- The position of a Simulation Object on a workspace canvas can be changed by dragging the Simulation Object item.
- If the drag operation is started on a Simulation Object item which is among several selected items, all items are moved (translated) together.
- The positions of an object in the various workspace views is independent. So dragging a Simulation Object icon will not effect the object’s position in the other views (i.e. simulation vs. accounting).
    - **An exception** to this during development of the new Qt4 QGraphicsViews: Common positions will be maintained for old and new workspace views for any particular geometry (i.e. simulation or accounting).
- If a Simulation Object item is dropped outside of the Workspace window, then the position of the Simulation Object is restored to its original position, and an error dialog with the following message is shown:

```
Move Error:  Some objects were moved off the canvas;
all objects have been restored to their original position.
```

1.4 Selection-based Operations / Workspace Menus

The following menu operations in the **RiverWare Workspace menubar** operate on the set of selected Simulation Objects:

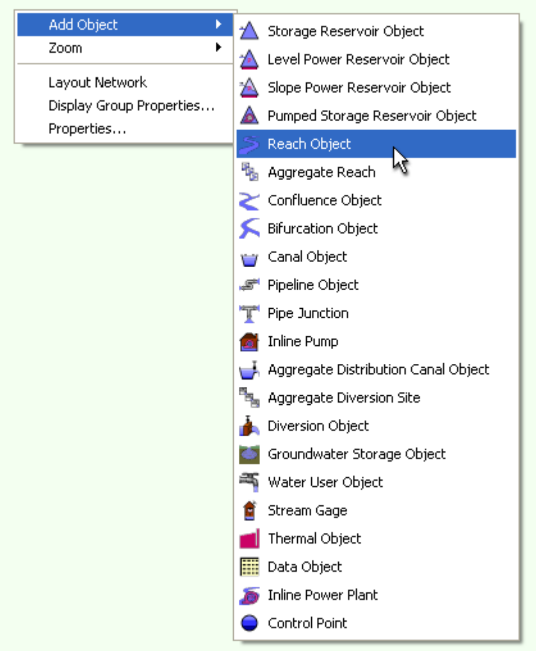
- File >> Export Objects ...
- Control >> Synchronize Objects ... (when one of the “Selected Objects” options is picked).
- Workspace >> Objects >> Open Selected Objects ...
- Workspace >> Objects >> Delete Selected Objects
- Workspace >> Objects >> Export Objects...
- Workspace >> Objects >> Synchronize Objects ... *(same as Control item, above).*

Also, the following operations in the **Subbasin Manager** operate on the set of selected Simulation Objects:

- Subbasin >> Select Member Objects on Workspace
- Subbasin >> Insert new Subbasin Before
- Subbasin >> Append New Subbasin
- Subbasin >> Append Typed Subbasin >> ...
- Subbasin >> Replace Objects from Workspace Selection
- Subbasin >> Append Objects from Workspace Selection
- Object >> Select Object on Workspace

1.5 Simulation and Account View Context Menu

By right-clicking on the workspace Simulation View or Accounting View background (not on an icon or object rectangle or link), a **context menu** with the following operations is displayed:



- **Add Object** >> ... (SimObj Type submenu, with icons and names, e.g. “Storage Reservoir Object”)
- **Zoom** >> (radio buttons): 400% .. 300% .. 200% .. 100% .. 80% .. 70% .. 50% .. 30% .. 25% .. 20%. These operations also center around the clicked point.
- **Layout Network** ... *(Accounting View only).* This shows a confirmation message: `The operation will override current object placements. Current layout will be lost. [OK] [Cancel].`
- **Display Group Properties** ... (shows the Display Group Editor).
- **Properties** ... (shows the Canvas Configuration Dialog which allows the setting of the Canvas Size, Background Color, Text Color and Canvas Font).

1.6 Simulation Object Operations

By right-clicking on a Simulation Object Icon in the Simulation View, a **context menu** with the following operations is displayed:

- **Link** >> (list of Slots on the Simulation Object)
- **Open Object ...**
- **Display Group Membership** >> ...
- **Delete Object**

The **Link** operation initiates a sequence of creating a link between Slots on Simulation Objects. *See the illustration to the right.* Slots which are already linked are shown with an icon (three linked hoops). After picking the first Slot, the following message is displayed in the status bar:

```
Right-click on object to complete linking
from <slot name>.
```

When hovering on a Simulation Object Icon, a **Tooltip** is displayed with the name of the Simulation Object.

> NOTE: This is redundant with the visible label.
> We might consider showing the Simulation Object Tooltip only if the canvas is zoomed out.

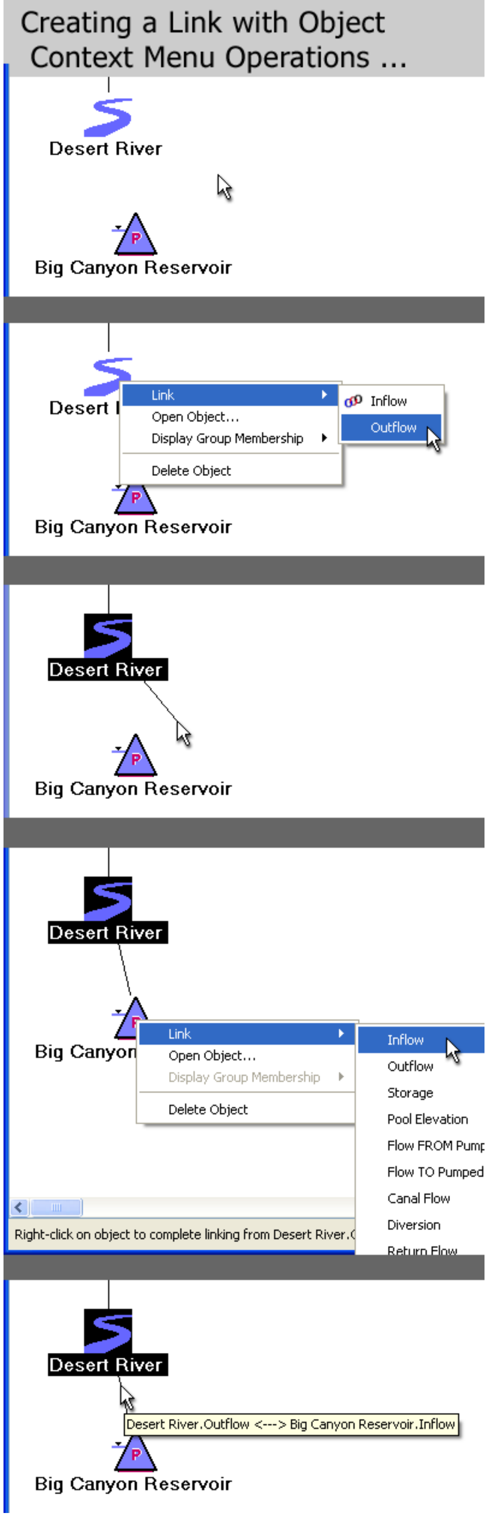
The **Open Object** operation shows the Open Object Dialog for the Simulation Object.

The **Display Group Membership** operation shows a submenu of the Object Display Groups to which the Simulation Object belongs. Clicking on a subitem shows the Object Display Group Configuration dialog. For Simulation Objects, the Object Label can be turned off. (There are other settings which apply only to SimObj Region rectangles on the Accounting canvas).

The **Delete Object** operation bring up a confirmation dialog with one of the following messages, depending on whether or not the clicked object is one among several selected objects:

- `<object name> will be removed from the workspace. [OK] [Cancel].`
- `Selected objects will be removed from the workspace. [OK] [Cancel].`

In the **Account View,** Simulation Objects are represented by rectangles, containing Account “shapes” (more info below). When right-clicking within a Simulation Object rectangle (but not on an Account), the context menu described above is show, with also the following two items (as the 2nd and 3rd items):


Creating a Link with Object Context Menu Operations ...

Right-click on object to complete linking from Desert River.(

Desert River.Outflow <---> Big Canyon Reservoir.Inflow

- **Add Account** >> (list of the available Account types for the object: Storage Account, PassThrough Account, Diversion Account, and Instream Flow Account). Clicking on one of the available Account Types tentatively creates an Account, and shows the Open Account dialog, where the Account's name can be changed or otherwise configured. Cancelling that dialog aborts the creation of the Account.
- **Account Summary....** This shows the Object Account Summary Dialog for the Simulation Object.

1.7 Physical Link Operations

When hovering over a Link between Simulation Object Slots, a **Tooltip** is displayed indicated the linked Slots. See the bottom image in the prior illustration. The drawn Link actually represents all the links between the physical Slots of two Simulation Objects, and the Tooltip shows one line for each such link.

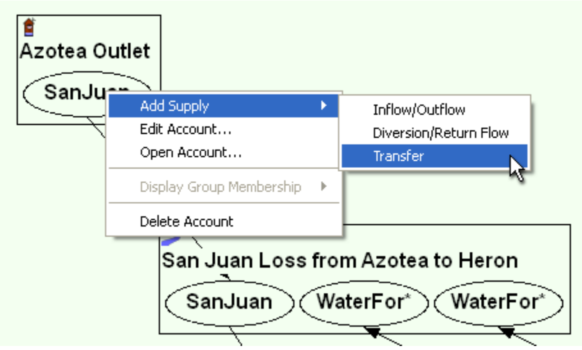
By right-clicking on a Physical Link in the Simulation View, a context menu with the following operations is displayed:

- **Delete Link** >> (list of links, of the form: Obj.Slot1 <---> Obj.Slot2)
- **Display Group Membership** >> ...

No confirmation dialog is shown when the user selects a link to **delete** -- the selected link is just immediately deleted.

The **Display Group Membership** operation shows a two-level cascaded submenu. The top level is a list of Links (of the form: Obj.Slot1 <---> Obj.Slot2). Each of those links has a second level menu enumerating the Link Display Groups to which the link belongs. Selecting a Link Display Group item shows the Display Group Editor. In that editor, the set of links in the group can be edited, and the following properties can be changed: Line Style, Line Color, and Line Width.

1.8 Account Operations



Accounts are shown only in the **Accounting View** as labelled "shapes" managed within a Simulation Object rectangle. This is described in detail in the **Accounting Visualization Design** document.

The positions of the Account shapes are managed automatically in a single row. The user can change the order by dragging.

Only single-selection is available for Accounts, though there aren't actually any operations which effect the selected Account. [**Note:** There is, perhaps, a bug with Account Selection in RiverWare 5.1. The user must drag (just a little) within an Account shape to cause it to be selected. Just clicking on the Account doesn't cause the Account to become selected].

By right-clicking on a Account shape in the Accounting View, a **context menu** with the following operations is displayed *(see also illustration above):*

- **Add Supply** >> (the three supply types: Inflow/Outflow, Diversion/ReturnFlow, Transfer). Clicking on one of the Supply types initiates a linking sequence similar to the Add Link function for Simulation Objects.
- **Edit Account.** Shows the Edit Account dialog (which shows the Series data within the Account).

- **Open Account.** Shows the Open Account dialog (with tabs for the Account Slot list, Methods, Supplies, General configuration and Units).
- **Display Group Membership** >> (list on the Account Display Groups associated with the Account). Clicking on an Account Display Group item shows the Account Group Display Configuration dialog. In that dialog, the user can change the set of Accounts in the group, and can edit these properties: Account shapes width (in pixels), Account shape aggregation enabling (on or off), Account Shape fill pattern, fill color, and border color.
- **Delete Account.** A “Delete Confirmation” dialog is shown with the following message:

```
<account name> will be removed from the workspace.
```

When hovering on an Account shape, a **Tooltip** is displayed with the name of the Account.

> NOTE: This is redundant with the visible label. We might consider showing the Account Tooltip only if the canvas is zoomed out.

1.9 Supply Operations

No interactive features are supported on Supply links. When hovering on a Supply, a **Tooltip** is displayed with the name of the Supply.

1.10 Zooming and Scrolling

- The workspace view can be scaled to preset zoom factors with a workspace **context menu Zoom operation** (400% .. 300% .. 200% .. 100% .. 80% .. 70% .. 50% .. 30% .. 25% .. 20%). The view is re-centered at the clicked point.
- There are also (+) and (-) magnifying-glass **Workspace Toolbar icon buttons** to increase or decrease the Zoom factor. (QUESTION: step values?).
- Unless the entire canvas scope fits within the view, a horizontal and/or vertical **scroll bar** appears at the bottom or right side of the window.

POTENTIAL ENHANCEMENTS:

- We could easily support **Rubber Band (drag rectangle) Zooming.** It’s supported directly by the new Qt4 classes. [Technical: The Qt4 QGraphicsView widget supports both “ScrollHandDrag” (scrolling) and “RubberBandDrag” (multi-object selection) DragModes. The initial development code uses a discrete QRubberBand instance to support the latter mode, but this should be changed to use QGraphicsView’s built-in implementation, as well as the “ScrollHandDrag” mode].
- Zooming OUT to be able to see the whole model makes sense. But **Zooming IN** to high zoom factors isn’t all that useful unless zooming in displays **“more detail”** (than could reasonably be shown at the normal zoom factor). Generally, currently, zooming way in just shows REALLY BIG icons. Might there be important information which could be displayed that way?
- **Instead of a seperate Locator View window,** there could be **Toolbar Mode Buttons** for quickly Zooming Out to the full model, with the ability to either: (a) drag an inscribed rectangle representing the scope of the current normal view, and (b) drag a new rectangle to define that scope. This could be one of three mutually-exclusive mode buttons:

(a) Selection Mode (with Rubber Band rectangle multiple-selection)
(b) Hand Drag Scroll Mode
(c) Locator Mode (with moving inscribed scope rectangle, or dragging a new one).

1.11 Workspace Image Export and Printing

From the Workspace menu, the following operations are available:

- File >> Export Workspace Image >> Full Model...
- File >> Export Workspace Image >> Visible Workspace...
- File >> Print...

The Export Image operations show the Export Image dialog which allows the user to select a target file and specify the following parameters: File Type (bmp, jpeg, png, ppm, xbm and xpm), Image resolution (low, medium, or high), image width (pixels), and image height (pixels).

The Print operation brings up the native output file selection dialog.

2.0 High Level Design

The initial Qt-based RiverWare Workspace GUI was implemented using the Qt3 "Canvas" managed-graphics classes. Graphical objects are represented by C++ classes managed within Qt with implementations defined in RiverWare subclasses. In Qt4 (starting with Qt version 4.2), the Qt3 "Canvas" classes were replaced with "Graphics View" classes having a very similar architecture.

The process of porting to Qt4 Graphics View classes is fairly straight forward. Trolltech provides the following online document:

> Porting to Graphics View
> http://doc.trolltech.com/4.3/graphicsview-porting.html
>
> “Conceptually, the Graphics View classes from Qt 4 and the Canvas classes from Qt 3 provide similar functionality using a similar design. Instead of "canvas", we use the term "scene". Otherwise, the class names and functions are almost the same as in Qt 3. The easiest classes to port will be QCanvas and QCanvasView. Experience shows that most time is spent porting the item classes, depending on the complexity of the QCanvasItem classes you have been using before.” *[Trolltech document cited above].*

	Qt 3	Qt 4.2
View :	QCanvasView	QGraphicsView
Model :	QCanvas	QGraphicsScene
Model Objects :	QCanvasItem and subclasses	QGraphicsItem and subclasses

The QGraphicsView and QGraphicsScene subclass development will use the most natural application of the new graphics framework. The QGraphicsItem subclasses will be very direct reimplementations of the old QCanvasItem classes using the new framework.

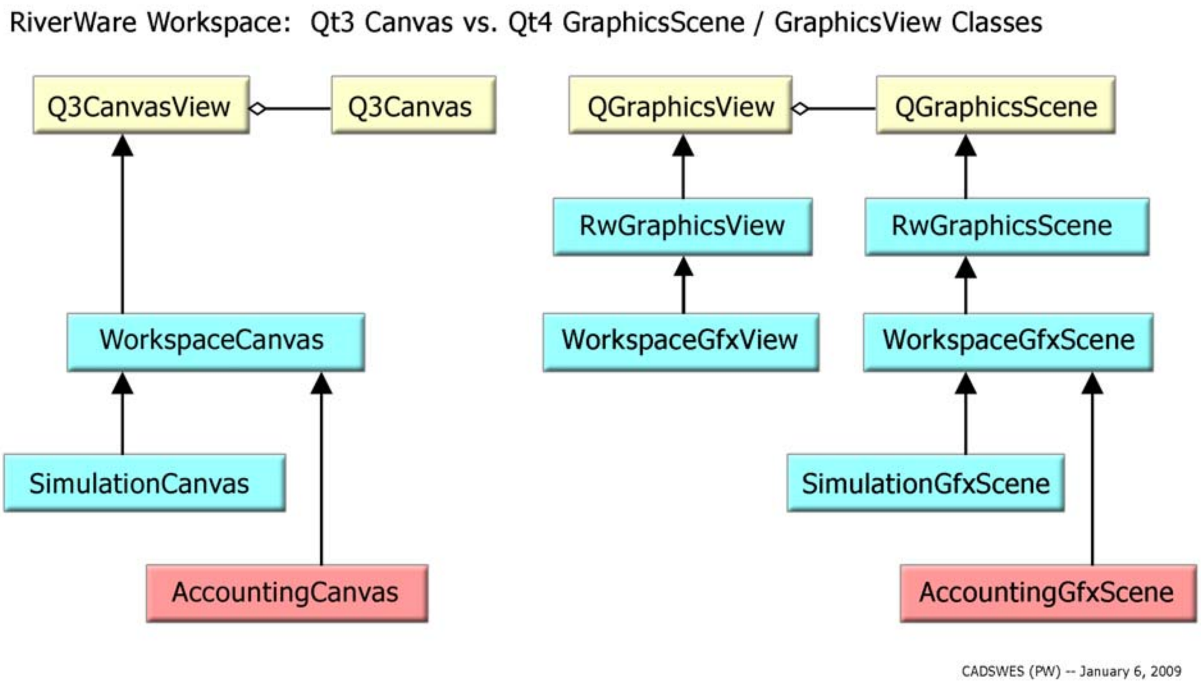
2.1 “Canvas” / Graphics View Classes

See the class diagrams on the subsequent pages.

This is the first application of the new Qt4 Graphics View classes in RiverWare. Application-level QGraphicsView and QGraphicsScene capabilities which would generally be useful in future uses of these classes will be implemented in distinct intermediate-level base classes: **RwGraphicsView** and **RwGraphicsScene.** This would support, for example, generic ItemView selection, and advanced zooming and scrolling features.

The (RiverWare GUI) **Workspace** class, which manages two supported **WorkspaceCanvases** (a **SimulationCanvas** and an **AccountingCanvas**) will instead manage **WorkspaceGfxView** and **WorkspaceGfxScene** pairs (subclasses of RwGraphicsView and RwGraphicsScene, respectively). The WorkspaceGfxScene will be abstract, specialized by **SimulationGfxScene** and **AccountingGfxScene** implementations.

- **Note:** During the porting process, the (RiverWare GUI) Workspace class will manage a mixture of both the current Qt3 “Canvas” implementations and Qt4 “Graphics View” implementations.



CADSWES (PW) -- January 6, 2009

2.1.1 class RwGraphicsView : public QGraphicsView

The Qt 4.2 QGraphicsView implementation provides only low-level support for selection, zooming and scrolling functions. (See especially the QGraphicsView "DragMode" property). This RwGraphicsView subclass will implement specific selection and scrolling capabilities, including:

- Rectangle "Rubberband" multiple-object selection.
- Interface to the "Locator View" for scrolling operations (see the **LocatorGfxView** class, below).
- **Tentative / Future:** Rectangle "Rubberband" Zooming. (This would be an enhancement).
- **Tentative / Future:** "Scroll Hand Drag". (This would be an enhancement).

2.1.2 class RwGraphicsScene : public QGraphicsScene

This class provides general "scene-related" capabilities potentially comon to all uses of QGraphicsScenes in RiverWare. This includes:

- General Display Property configuration: background color, text color and text font.
- QGraphicsItem utilities not provided by QGraphicsScene. For example, QGraphicsScene (in Qt 4.2) doesn't provide a method to delete all of the scene's QGraphicsItems.

2.1.3 class WorkspaceGfxView : public RwGraphicsView

This concrete class provides application-level “Workspace” features based on the QGraphicsView interface. This includes:

- Handling Drag and Drop operations for creating new Simulation Objects from the Simulation Object Palette.
- Program-initiated **scrolling** operations based on RiverWare objects (e.g. Simulation Objects and Accounts).

2.1.4 class LocatorGfxView : public RwGraphicsView -- (NOT SHOWN in the class diagram)

This class will be a specialized “fully-zoomed out” view on the various Workspace Scene instances. It will present a “draggable” rectangle which controls the scroll position of a client RwGraphicsView. LocatorGfxViews will be usable for any RwGraphicsView / RwGraphicsScene clients (i.e. not just for RiverWare “workspaces”).

2.1.5 class WorkspaceGfxScene : public RwGraphicsScene

Application-level public operations on the a Workspace Graphics Scene (including the scene’s QGraphicsItems) will be performed on concrete instances of the WorkspaceGfxScene abstract class, e.g. from the (RiverWare GUI) Workspace.

For operations presented in the workspace menubar and toolbar, and in various context menus, WorkspaceGfxScene will provide both:

- Qt “slots” (Qt signal handlers) implementing application level operations, many of which will apply to the currently selected set of objects.
- QActions connected to those Qt “slots”. The (RiverWare GUI) Workspace (and potentially, any client), can build menus and tools from these QActions.

This abstract class will also provide application-level “Workspace” features based on the QGraphicsScene interface. This includes:

- Support for managing workspace-*background* context menus based on the public QActions. The actual context menus will be built in the concrete subclasses (see next section). (And QGraphicItem context menus will be managed in those item subclasses).
- Program-initiated **selection** operations and queries based on RiverWare objects (e.g. Simulation Objects and Accounts).

2.1.6 class SimulationGfxScene : public WorkspaceGfxScene, and class AccountingGfxScene : public WorkspaceGfxScene

These concrete subclasses provide specialization of WorkspaceGfxScene operations, generally via virtual methods (of both Qt and RiverWare base classes). Capabilities will include:

- Building the workspace-background context menu.
- Registering for, and servicing callbacks from the RiverWare model (generally from the Sim and Accounting libraries). These will include callbacks for:

  •Object creation notification. (Note that Object deletion and property change notifications will be handled instead through the relevant QGraphicsItem subclasses).

  •Workspace clearing notification. (This is redundant with, but more efficient than handing deletions at the QGraphicsItem subclass level).

During development, there will probably be some “refactoring” of implementation between these two classes and the WorkspaceGfxScene base class.

2.2 Graphics Item Classes

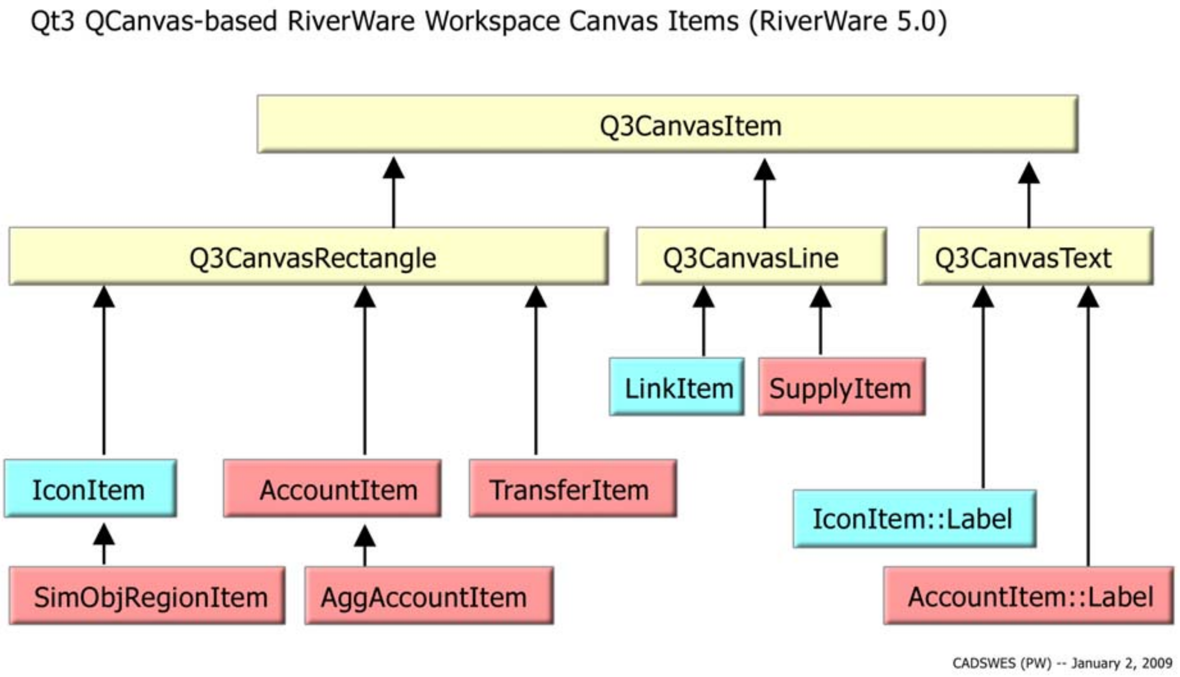
The new QGraphicsItem subclasses will be fairly direct reimplementations of the existing Qt3 QCanvasItem classes. They will be responsible for:

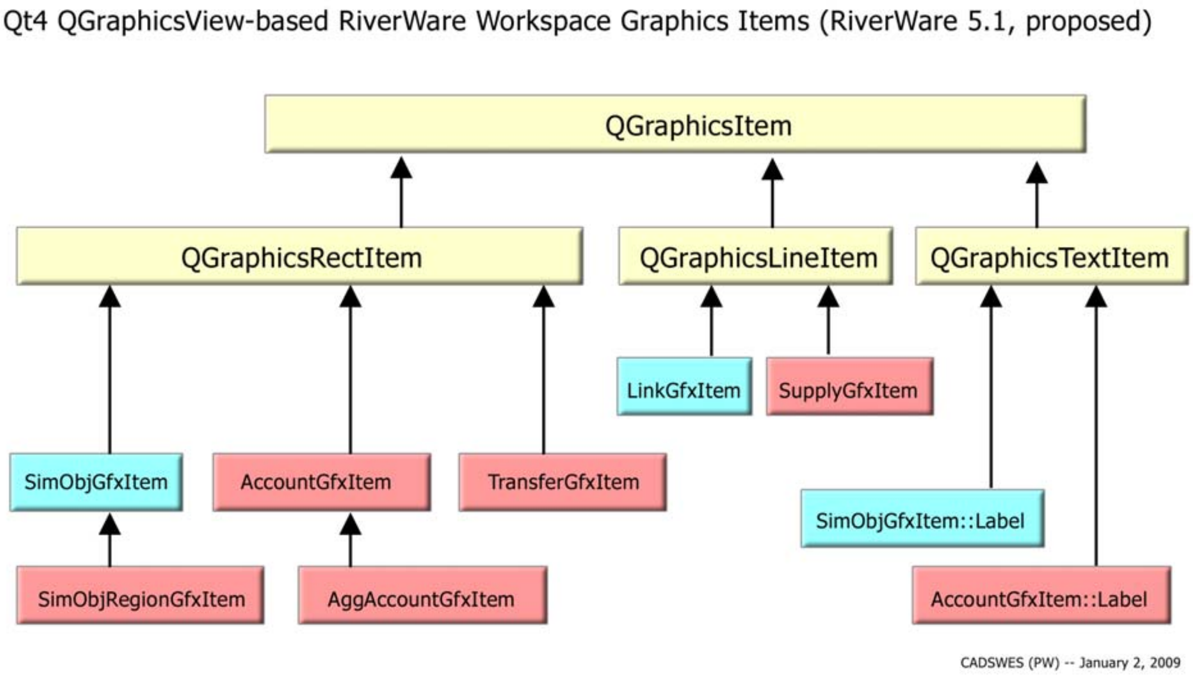
- Managing the item's internal components: labels, icons, and subitems,
- Managing item-based context menus (with QActions connected to Qt slots defined in WorkspaceGfxScene).
- Handling notifications from the RiverWare model (generally from the Sim and Accounting libraries, e.g. deletions).

Generally the graphics item subclasses will implement the following QGraphicsItem virtual methods:

- virtual QRectF boundingRect() const;
- virtual QPainterPath shape() const;
- virtual void paint (QPainter*, const QStyleOptionGraphicsItem*, QWidget* w=NULL);

The following two class diagrams show the existing Qt3 QCanvasItem and Qt4 QGraphicsItem hierarchies:





Qt4 QGraphicsView-based RiverWare Workspace Graphics Items (RiverWare 5.1, proposed)

CADSWES (PW) -- January 2, 2009

These class diagram image files and a development estimate are available from this webpage:
http://cadswes2.colorado.edu/~philw/2009/GraphicsViewWorkspace/

--- (end) ---