Integrating GIS technology with RiverWare

Functional Requirements

Author: Patrick Lynn

Graphical information systems (GISs) support storage, analysis, and display of spatial data. While most types of RiverWare objects represent physical objects that are located in space, RiverWare does not represent or make use of information about the spatial extent and location of the modelled objects. This document motivates and describes requirements for the use of GIS technology within RiverWare.

1 High Level Requirements

In this section we provide a brief description of the immediate goals for incorporating GIS technology into RiverWare; in subsequent sections we motivate and describe more detailed requirements. The new functionality will support a map based view of the modelled basin. By providing an intuitive and information-rich view of a basin, the map-based view will complement the existing schematic views.

1.0.1 Support georeferenced objects

RiverWare will support the association of *spatial coordinates* with each object. The spatial coordinates for an object will be represented as a pair of floating point numbers, and will be interpreted as either the object's geographic coordinates (latitude and longitude) or cartesian coordinates in a projected coordinate system (easting and northing, or x and y). The user will be able to provide objects with spatial coordinates in the following ways:

- By interactively entering the two numbers.
- Through graphical interaction with the workspace. For example, the user could drag an object from its original location on a map display and drop it at the map location corresponding to its spatial coordinates. Alternatively, the user could select a location on a map display and then indicate which object has that locations as its spatial coordinates.
- From a file, for example in the ESRI shapefile format.

In addition, RiverWare will provide a mechanism to facilitate the sharing of object coordinates between models.

1.0.2 Support a geospatial workspace view

In the geospatial workspace view object icons, links, and a background image are displayed in a spatially consistent way. That is, the background image is taken to be a map projection and each object icon is displayed at (or near) the location on this map which corresponds to the object's spatial coordinates.

RiverWare will be able to read and display background maps in standard image and GIS formats. Note that some formats support information about the image's location in a coordinate system, and some do not. Appropriate mechanisms will be provided for user input (and persistence) of this information for images that do not already contain it.

1.0.3 Design for additional integration

Current development should facilitate further integration of GIS technology with RiverWare in the future.

2 Description of the current workspace functionality

In this section we discuss the purposes currently served by the RiverWare workspace, so that we can evaluate how GIS technology might improve or complement this functionality. Note that the main RiverWare dialog provides menu-based access to most RiverWare functionality in addition to providing the workspace display discussed here.

- **Schematic visualization of the modelled river basin**. This display indicates how the basin has been abstracted and discretized by the user into objects, and for each object shows the name, type, and relationships with other objects (i.e., links indicating information sharing between objects).

  In this schematic view, each type of object has a distinct, predefined icon (image) associated with it, each object in the model is represented by an icon of the appropriate type, and each icon is labelled below with the object's name. The user can manually arrange the object icons and the resulting object locations are saved within the model file.

  Links are drawn as lines between icons, and the color and style of these lines can be controlled by the user. Links can be organized into groups.

  Users control the focus of the workspace by scrolling and the level of detail by zooming in and out.

  Two types of schematic views are supported, Simulation and Accounting. The two views are conceptually similar, but the accounting view displays additional object information pertaining to water ownership (e.g., which accounts are associated with an object and what are their types).

  The object icons in each view are different, their locations in each view can be different, and RiverWare makes no assumption about the relationship between an object's location in either view and the object's actual, physical location (if any).
- **Access to more detailed information organized by object**. This information is accessed by double-clicking on the object's icon, and includes object data and methods.
- **Support for model creation**. The user interacts with the workspace to create and locate objects as well as to create links between the objects.
- **Selection of multiple objects**. For actions that operate on a collection of objects, such as creating a subbasin or saving a subset of the objects to a file, the workspace provides a graphical mechanism for defining the collection of objects on which the action will operate.

Each workspace view is drawn on a conceptual "canvas" which is characterized by the following user-settable attributes:

- Size (default width = 6450, height = 6450)
- Background color
- Text color
- Canvas font size

RiverWare 5.1 added support for a background image in the simulation view. Users can specify an image in a standard format and that image is displayed as the background of the simulation view of the workspace. The image file must be in the same directory tree as the model file and its relative path to the model file is retained in the model file. If the background image is a map of the modelled basin, then the user can manually rearrange the object icons to correspond to the actual locations of the associated objects on the map. In other words, the user can use this functionality to create a simulation workspace view in which the schematic view of the model basin is overlaid on top of, and in registry with, a map of the basin. See the document “Georeferencing Simulation Objects in RiverWare: Proposal” for details and limitations of this approach as well as possible extensions.

3 GIS technology and RiverWare

In this section, we briefly discuss GIS technology and suggest ways in which RiverWare might usefully access this functionality.

3.1 Coordinate systems and projections

This section introduces some basic GIS terminology and concepts.

Locations on the earth are most often represented as a pair of coordinates representing latitude and longitude. For some applications, a third coordinate representing something like elevation relative to mean sea level is also used. Apart from this spherical coordinate system, GIS systems typically represent locations as cartesian coordinates in a plane onto which the surface of the globe has been projected.

Several standards have been developed to provide agreed-upon coordinate systems for a given area. These standard coordinate systems define a set of map projections with together cover the target area. One common coordinate system which covers the globe is the Universal Transverse Mercator (UTM) coordinate system. This system defines a projection for the northern and southern areas of each of 60 longitudinal zones spanning the globe. For example, in the projection UTM Zone 11 North, the coordinates eastings = 397,800 m, northings = 4,922,900 m, corresponds to a location in central Oregon.

Another standard coordinate system is the State Plane coordinate system which defines a set of projections for the United States. In this system, each state is divided into one or more zones, so a location is specified by a state, zone designation, and x, y coordinates.

Note that a standard coordinate system defines a mathematical coordinate system corresponding to each projection; when these two uses of the term “coordinate system” would be confusing, we use the term to refer to the standard, and use the term “projection” or “map projection” for the component coordinate systems.

Coordinates, whether they be geographic (latitude and longitude) or correspond to a map projection, are specific to a datum. A datum is a reference surface and surveyed coordinates for a set of actual points and lines. Examples of common datums include the North American Datum of 1983 (NAD83) and the World Geodetic System of 1984 (WGS84).

From “GIS Fundamentals” (Bolstad, 2008):

> Exact or approximate mathematical formulas have been developed to convert to and from geographic (latitude and longitude) to all commonly used coordinate projections. These formulas are incorporated into “coordinate calculator” software packages, and are also integrated into most GIS software. For example, given a coordinate pair in the State Plane system, you may calculate the corresponding geographic coordinates. You may then apply a formula that converts geographic coordinates to UTM coordinates for a specific zone using another set of equations. Since the backward and forward projections from geographic to projected coordinate systems are known, we may convert among most coordinate systems by passing through a geographic system.

That is, it is relatively easy to convert back and forth between geographic coordinates and coordinates from a known projection; whereas a method for direct conversion between two different projections is not generally available.

To display a part of the curved surface of the earth on the screen necessarily requires a projection, so it will always be the case that any geospatial view will have some map projection associated with it, and everyting displayed in that view (e.g., object icons, background image) will need to be displayed in that same projection.

On the other hand, to support our initial goals, we do not need to know anything whatsoever about the projection associated with the geospatial view, rather what we require is that the geospatial view, the object coordinates, and the map all share the same projection. Note that this allows us to support the maps for which the projection is not known, which I believe will be common.

Another relevant point is that there are many ways of describing map projections, and some systems exist in multiple variations. For example, UTM zones are sometimes designated by hemispheres and sometimes by latitude band, leading to different notations for the same zone (such as, “UTM Zone 11T” and “UTM NAD83 Zone 11 North”).

In the future if we need to convert between coordinate systems, RiverWare will of course need to know the details of each projection involved (including the equation describing the projection). For example, if objects locations are provided as geographic coordinates, or if the user would like to change the coordinate system of the spatial view to replace the map with a different one from a different projection.

At the river basin scale, I think the most common conversion will be between geographic coordinates and a single projection. For example, a weather service might provide rainfall forecasts in geographic coordinates, and the geospatial view might use the appropriate UTM zone projection.

At any rate, when we choose to support conversion between coordinate systems, we can start requiring that the user specify the projection for the geospatial view, map, and objects. Presumably we would identify a set of supported common projections for RiverWare to support (we will need to be able to do the conversion math, so supporting arbitrary projections is not feasible).

My bias is to avoid representing (or requiring user input of) information that RiverWare does not use for some purpose, so I suggest that initially we not collect any information about the projection.

Note that without knowledge of the projection will not know the units of the view, although the user is free to include a legend in their image.

3.2 Display

While GIS applications usually support tabular displays of data, the spatial aspect of the data allow it to be organized visually by geospatial location. Thus a primary feature of most GIS applications is the visual rendering of spatial data, i.e., the drawing of maps.

Displays can be very flexible, allowing user control over both which data are displayed at any given moment as well how those data are displayed. The data and display are typically organized into layers, and the user might have control over the following aspects of the display:

- Whether or not each layer is visible and with what level of transparency.

- How the data associated with a layer is visualized, e.g., for raster elevation data, is a contour plot drawn or are the elevations indicated on a per cell basis using a grey scale.
- Whether or not icons are associated with data, which icons and how they are presented (e.g., size, color, and orientation to the datum's coordinates).
- Whether or not labels are present, the label font, orientation, etc.
- Whether or not a legend is drawn and its appearance.

Stand-alone GIS applications can be very sophisticated in their displays, especially in how they deal with zooming in and out. For example, at different levels of resolution, an application might automatically change font size and move or omit labels.

Note that the GIS data and maps can be displayed on a screen or saved to a file. Output formats vary greatly in the degree to which the structure of the original information is retained as well as with respect to which other applications are likely to be able to understand and display files in that format. For example, the ESRI shape file format is supported by most GIS applications and allows retention of the original structure of the data. By contrast, the JPEG image format, while supported by many applications that display images, does not retain any spatial information or even the structure to allow, for example, toggling of the layers used to generate the image.

3.2.1 The clustering problem

Phil described one solution to the clustering problem, supporting callouts.

Another option is to support more flexibility in how we display icons, specifically to support smaller, less obtrusive icons, with perhaps more control over the icon labels. This suggestion is motivated by several observations:

- I think our existing, cartoonish icons look bad when displayed on a nice map. The icons seem appropriate for our other views, but I would not want to seem them on top of a map.
- Most maps address this problem with the combination of small (typed) markers and a legend, why not RiverWare? For example, Mike Basin brochures show maps with modelled objects indicated by small dots or triangles.
- Many if not most map images will have been generated for some other purpose and so might already have labels and indicators for various map features. Allowing smaller RiverWare icons and object name location would minimize conflict between the background image(s) and our schematic.

There is no reason why we shouldn't support both solutions to this issue, though I find the callout solution complicated: for some but not all objects there are two icons, lines can be links or callout indicators, the user needs to place both the object itself and its icon.

Note that there is a third solution to the callout problem, one which the users will probably frequently use, which is simply to give clustered objects spatial locations that allow for a clear display, deviating somewhat from their actual location at times.

3.3 Analysis

Analysis of spatial data has many application to water resources modelling. For example, contour information in the form of digital elevation maps (DEMs) can be used to compute direction of stream flow, determine upstream/downstream relationships between reservoirs, compute rainflow runoff, or delineate catchments. The ability to perform, or import the results of, these and similar analyses could be useful within RiverWare.

3.4 Data access

Spatial databases employ both vector and raster representations of spatial data, and support the association of attributes (non-spatial data) with each datum. Most spatial data are stored and shared in one of a small number of formats (e.g., USGS DLG format, ESRI shape file format).

To the extent that the attributes and object abstractions represented within a spatial database match those required by a RiverWare model, that database could be used to help provide information to a RiverWare model. For example, a GIS database might contain representations of reservoirs in a basin with attributes which include the reservoir name. RiverWare could use such a database to provide objects with their spatial coordinates for the geospatial workspace view, as well as to initialize other modelled object attributes. RiverWare's namemap capability could be used to facilitate access to this information by relaxing the requirement that names in the database match those in the RiverWare model.

Current RiverWare DMI functionality allows RiverWare to access series data in spatial databases, though we could consider creating new direct database connections for specific database formats.

4 Phased development of GIS display technologies within RiverWare

In this section I present one proposal for phased integration of GIS within RiverWare. I focus on the use of GIS technology in the workspace display because I think other uses of GIS within RiverWare require relatively independent lines of development and so don't affect design decisions we are making now.

For example, to support the immediate goals and make decisions about future workspace issues, we don't need to think too much about what sort of spatial analyses we might want to support in the future. The main commonality or shared development is the georeferencing of objects, something that is at any rate definitely a part of the immediate work.

4.1 Phase 0: Simulation View Background Image (done)

In this phase, RiverWare will support the display of a single background image without knowledge of its projection. This functionality is currently supported for the simulation view, and permits the display of a schematic view overlaid in registry with a map.

4.2 Phase 1: Geospatial View (immediately)

In this phase, RiverWare meets the immediate goals, which are: 1) georeferenced objects, 2) add a view in which objects are shown at their locations on a map, and 3) design for future GIS integration. This will be accomplished by the following new functionality:

- Introduce the geospatial view which displays a schematic diagram over a map.
    - Allow the user to specify the coordinate system for the spatial view by providing the coordinates of the lower left and upper right corners (or coordinates of lower left and the size).
    - Support a background image for the spatial view

      In addition to specifying the file containing the background image, the user also locates the image in the workspace by providing coordinates for two points on the image.

      A reference to the image file persists in model file

- Provide an operation to size the workspace view canvas to the minimum size that encompasses the objects and the background image.
- Add to SimObj a pair of floating point spatial view coordinates.

  Provide an interface for the user to type in the coordinates.

  Provide a mechanism for specifying an object's spatial view coordinates by dragging and dropping its icon in that view.
- Support some degree of control over the display of icons in the geospatial view

  At a minimum, allow the users to choose between the normal icon set and a small icon set, and to choose the orientation of the label with respect to the icon. Note that eventually we might want to allow this flexibility on a per-object basis, but initially we could orient the labels in the same way for all objects (as we do now).
- Current mouse coordinates could be displayed in the task bar.

Note that an object's spatial coordinates are best thought of as the location of the object's icon, not necessarily the location of the physical entitity (if any) represented by that object. Physical objects are often larger than appropriate for such a point representation and we want to avoid confusion of this spatial location with the full spatial information contained in a database. By choosing a point representation to georeference objects, we are not limiting ourselves with respect to GIS database representations of objects, be they raster, point, line, or polygon.

We might want to consider supporting image files that contain both the image itself as well as coordinate system information in some form, allowing us to place it automatically within the geospatial view.

Open issue: do we use the Simulation View's system for embedding image references (path relative to model file)?

With the functionality developed in this phase, RiverWare will be able to display a schematic view overlaid in registry with a map. The background image (map) can be changed to another with the same projection without requiring any change to the spatial locations (geospatial view coordinates) of the objects. In this scenario, RiverWare assumes that the geospatial view, the object spatial locations, and the background image(s) are from the same projection, but RiverWare does not need to know anything about that projection.

4.3 Phase 2: Layers (future)

In this phase, RiverWare will support the organization of the geospatial view into an ordered set of layers.

- Users can control (at least) the visibility and transparency of each layer, and they can easily rearrange the layers.
- Allow an arbitrary number of background images, each constituting a different layer. The user will need to provide information on how each image is located in the spatial view. Since it is likely that multiple images will share not only the same projection (as required), but also location within that projection (what Phil refers to as scale and scope), we should provide good support for this case. This could be done, for example, by providing projection information from a previous layer (image) as the initial default when the user adds a new layer (image).
- One possibility: a layer for each object type, each link group.

  Enhance user control over the schematic view, i.e., for each layer, what icon should be used its objects, label orientation, etc.
- We might want to introduce a key (legend) to the geospatial display.

This work could perhaps leverage off of the current display groups, and could potentially be extended to all views. Note that all only some image formats would be appropriate for layering, those that provide representational support for transparency sucha PNG (but not JPEG or GIF).

4.4 Phase 3: Map projection sophistication

In this phase, RiverWare would add support for representation of the coordinate systems associated with the geospatial view, object spatial coordinates, images, and spatial data.

- Projection knowledge can be used to support conversion between coordinate systems. For example, the user could request that the geospatial view display object coordinates in geographic coordinates even though the map projection is from UTM Zone 7N.
- Knowledge about a projection might include wording (e.g., “easting” versus “x”) units, and orientation.

This phase would require knowledge of GIS formats and the ability to do projection math, functionality that we will probably want to achieve by linking in 3rd party software or creating a RiverWare plugin interface and plugins which use a 3rd party library.

In theory, with knowledge of projections RiverWare could allow each of the three located entities (RiverWare objects, the geospatial view, and the map image) to use different coordinate systems. However, converting one map image to another requires sophisticated computation, and it does not seem likely that users would gain much from this sort of generality. My suggestion would be to require that the geospatial view and its maps share the same projection and that the object spatial locations be coordinates in that projection or geographic coordinates.

4.5 Phase 4: Additional GIS data access

In this phase, we will add support to RiverWare for accessing spatial data and probably for representing spatial data internally.

- Support reading of common spatial database formats such as ESRI shapefiles.
- Support display of spatial data in tabular form.
- Support interactive access to spatial data via the geospatial view.

  For example, if the user were to select a location on the geospatial view, RiverWare could then translate that screen location into projection coordinates and use those coordinates to index into the spatial data (i.e., show the elevation for that location, which data would be available from a DEM).

  As another variation, if RiverWare has read vector representations of objects in a spatial database, and the name attributes of those data correspond to modeled objects, then RiverWare can translate locations in the geospatial view to RiverWare objects. That is, if the user moused over any portion of the view on which a reservoir was drawn, RiverWare could display the name of that reservoir. This could allow the users to disable the schematic view entirely if they desired without losing any of the functionality of the view (access to open object dialogs, multiple object selection).
- Support rendering spatial data sets as a layers in the geospatial view.

  This would provide the full power and flexibility of GIS visualization within RiverWare and is likely to require some sort of tight coupling with a GIS application or library. Many issues would need to be addressed, such as would RiverWare or a GIS library interact with the user to specify display options, and how would visualization specifications persists.

  One option would be for the user to make visualization choices in a separate GIS application, then save those maps in a GIS format. RiverWare could then read the spatial data along with visualization choices and pass those along to a rendering engine, which would create the image to display as a layer. In this scenario, display options would need to be made and saved in a GIS application, and then those data would be displayed as layers within RiverWare.

There are a few aspects of traditional map displays that RiverWare will not be able to support in a fully general way until this phase; namely the legend (key), scale bar, and north arrow. In flexible displays such as the geospatial view, the legend would ideally be displayed in its own window. This would allow the user to scroll around the geospatial view, zooming in and out, while keeping the legend in view (or hidden, as they wish). Similarly the scale bar and north arrow would appear in their own windows or at a fixed location in the view, independent of the scroll location and current zoom level. To fully support this functionality requires a great deal of knowledge about what is being displayed and how, e.g., knowledge of the projection as well as which icons are associated with objects in the each layers.

5 Detailed design for Phase 1

In this section we provide detatiled requirements for the initial phase of GIS integration (scheduled for Winter 2010).

In this context, a *map projection* or simply *projection* is characterized by the following fields:

- Description: Arbitrary text input by the user, not used by RiverWare. This could include a description of the projection such as “UTM NAD83 Zone 11N”, but RiverWare will not make any assumptions about the description contents.
- Unit type: unit type of the projection coordinate system. This will be one of: Length, Angle, or None.
- User units: the preferred units for displaying coordinates.
- Scale: the preferred scale for displaying coordinates.
- Precision: number of digits to the right of the decimal point with which to display coordinates.
- *(Optional)* Horizontal dimension: label associated with the horizontal dimension of the projection’s coordinate system (usually “x” or “Easting”).
- *(Optional)* Vertical dimension: label associated with the vertical dimension of the projection’s coordinate system (usually “y” or “Northing”).

In the future we might add additional fields such as the name and datum of the projection.

A *projection rectangle* is a specific rectangular area within a map projection. For the initial implementation, the Geospatial View will be need to specify the projection and projection rectangle. If the user provides a background image for the Geospatial View, they will need to locate it within that view’s projection by specifying the image’s projection rectangle.

The user is responsible for ensuring that the projections are identical in the mathematical sense of the term.

5.1 The Geospatial View

The Geospatial View of the workspace displays a schematic view of the workspace in registry with a background map image. The Geospatial View of the workspace is described by the following attributes:

- Projection: describes the projection associated with this view.
- Projection rectangle: describes the location of the view (canvas) within its projection.
- Icon scale factor: A floating point number representing how the size of the object icons should be scaled in this view relative to the the Simulation View. the scale factor is in projection units per pixel, e.g., m/pixel. Factors greater than one lead to large, poor quality icons.

- Opacity: opacity of the icons and links.
- *(Optional)* Display schematic toggle: whether or not the schematic view (icons and links) should be displayed or not.
- Icon label location: where the icon label should be displayed relative to the icon itself. The possibilities are: Below (default), Right, Above, Left, Not displayed.
- Background image: the image to be displayed between the background and the schematic (discussed in detail below)
- The usual properties associated with all workspace canvases (Background Color, Text Color, and Canvas (Text) Font).

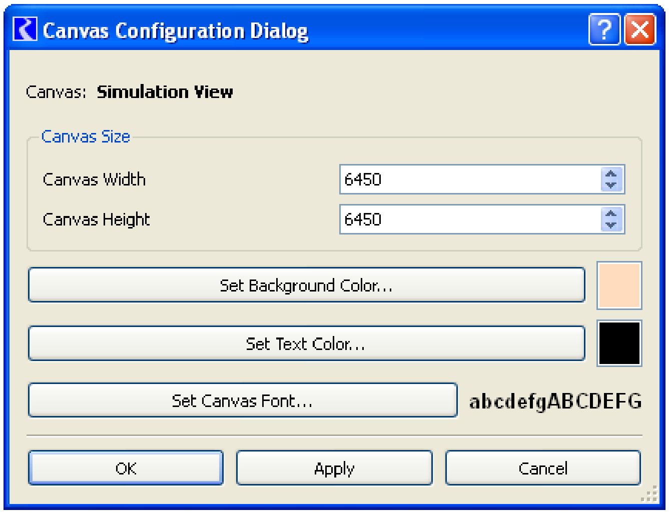
All of this information must persist in the model file. The format should extensible to support backward compatibility as we add additional information (e.g., to the specification of the map projection).

The default values for the Geospatial View's projection and projection rectangle will be based on the default canvas size for the Simulation View: the lower left corner will be (0.0, 0.0) (i.e., the origin of the projection) and the upper right corner will be (6450, 6450), with a unit type of None.

Current mouse coordinates will be displayed in the task bar.

5.2 Configuring the Geospatial View

Figure 1. The existing Canvas Configuration Dialog.



Canvas Configuration Dialog

Canvas: **Simulation View**

Canvas Size

Canvas Width 6450

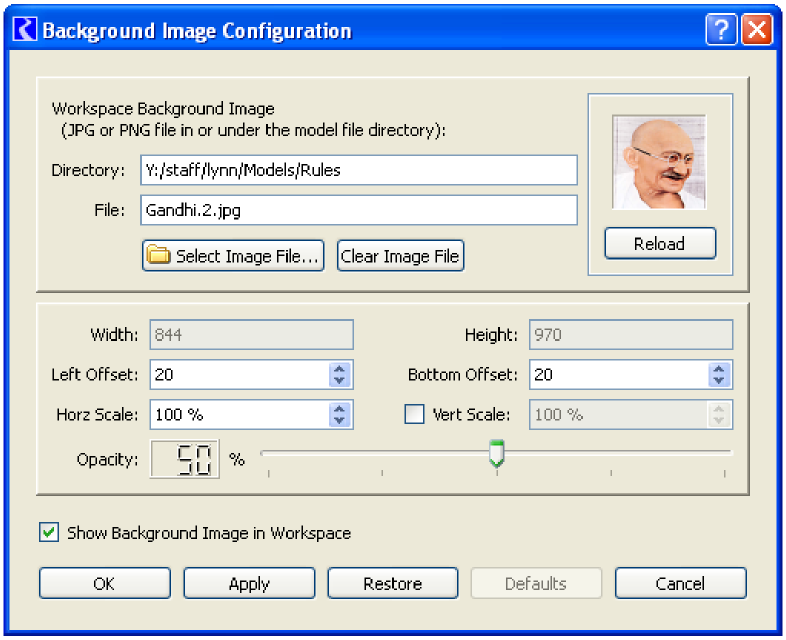
Canvas Height 6450

The existing Canvas Configuration Dialog will be extended to support the Geospatial View. This involves adding several features:

- Display and editing of the projection, projection rectangle, and projection display scale (the relationship between pixels at standard zoom and projection)
- Display and editing of the background image specification (which itself has an associated projection rectangle).
- Icon settings.

Note that the Canvas Size portion of this dialog will be hidden for the Geospatial view, since the canvas width and height are a components of the projection rectangle, which is displayed in its own panel.

Figure 2. The existing Background Image Configuration Dialog for the Simulation View



Workspace Background Image
(JPG or PNG file in or under the model file directory):

Directory: Y:/staff/lynn/Models/Rules

File: Gandhi.2.jpg

Width: 844 Height: 970

Left Offset: 20 Bottom Offset: 20

Horz Scale: 100 % □ Vert Scale: 100 %

Opacity: 50 %

☑ Show Background Image in Workspace

Specifying the background image is the most complicated piece of Geospatial View configuration. As a starting point for this discussion, consider the current interface for configuring the Simulation View's background image. In this dialog, the user:

- Specifies which image to display (by name or path relative to the model file).
- Provides guidance for how to display the image (opacity, location in Simulation View).
- Can toggle on or off the display of the background image (without otherwise affecting the image configuration).

This dialog has several limitations:

- Neither absolute paths nor variable subsitution are supported.
- The user can not access this dialog until the model has been saved. Presumably this is due to the requirement that the image be located in the same directory as the model file.
- This dialog is only accessible from the context menu of the Simulation View, whereas it seems most natural to consider background image configuration as part of canvas configuration.

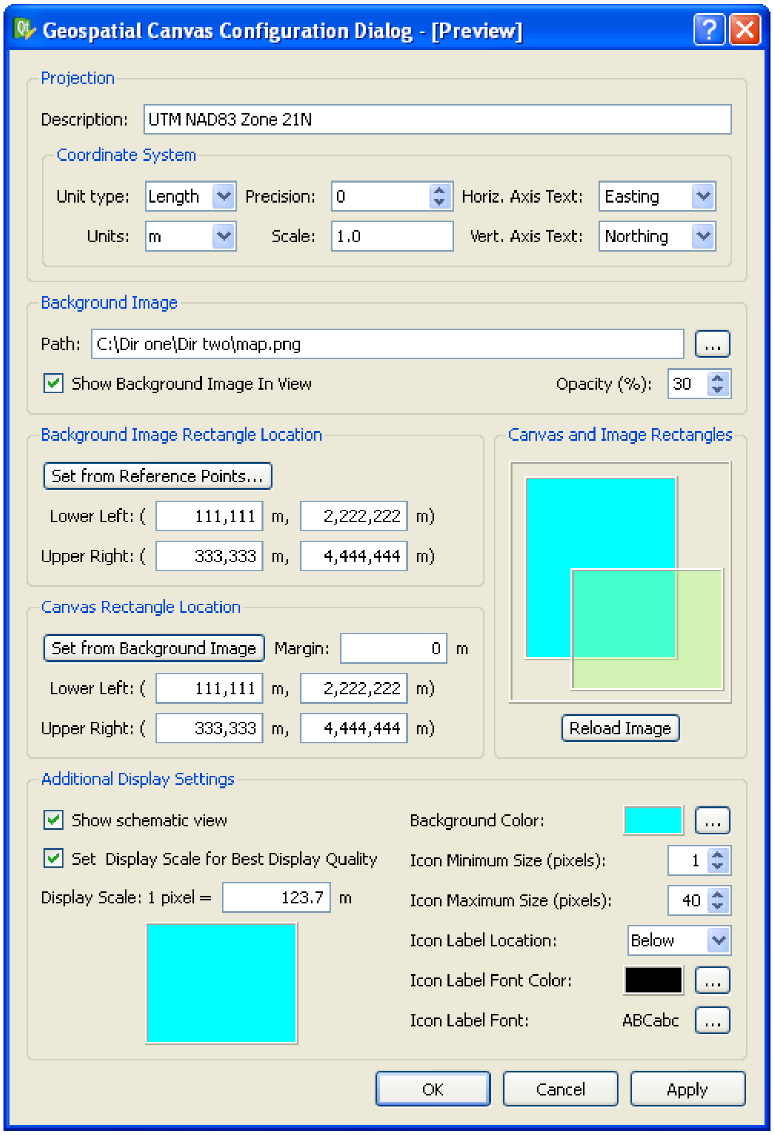
The interface for configuring the background image of the Geospatial View will address these limitations as well as provide a way to locate the image in the view's projection by allowing the user to interactively provide coordinate information for the map.

The Geospatial View and its background image are assumed to share the same projection, but the projection rectangles need not be identical. However, in one common use case the user will want the background map image to be the same as the Geospatial View's canvas, so RiverWare will provide a toggle for automatically adjusting the canvas to match the image. When this toggle is checked, at the time the background image is applied to the workspace, the projection rectangle of the workspace will be adjusted to match that of the background image. By default this toggle will be checked. If any objects are thereby no longer displayed on the Geospatial View's Canvas, RiverWare will warn the user that this is the case and refer them to the dialog in which they can view and edit object spatial coordinates.

Initially we will assume that the both axes of the background imge have the same units. Images will be displayed with thesame vertical and horizontal scaling and in the original orientation.

While we need to retain the projection rectangle associated with an image, we do not necessarily need to retain other information including the image cells with which it was specified originally. However, retaining the information used to specify the image's projection rectangle would be useful if they wanted to alter that information, so we should retain it if it is not too much work.

Figure 3. Mockup of the Geospatial View's Canvas Configuration Dialog.



Geospatial Canvas Configuration Dialog - [Preview]

Projection

Description: UTM NAD83 Zone 21N

Coordinate System

Unit type: Length Precision: 0 Horiz. Axis Text: Easting

Units: m Scale: 1.0 Vert. Axis Text: Northing

Background Image

Path: C:\Dir one\Dir two\map.png ...

☑ Show Background Image In View Opacity (%): 30

Background Image Rectangle Location

Set from Reference Points...

Lower Left: (111,111 m, 2,222,222 m)

Upper Right: (333,333 m, 4,444,444 m)

Canvas Rectangle Location

Set from Background Image Margin: 0 m

Lower Left: (111,111 m, 2,222,222 m)

Upper Right: (333,333 m, 4,444,444 m)

Canvas and Image Rectangles

Reload Image

Additional Display Settings

☑ Show schematic view

☑ Set Display Scale for Best Display Quality

Display Scale: 1 pixel = 123.7 m

Background Color: ...

Icon Minimum Size (pixels): 1

Icon Maximum Size (pixels): 40

Icon Label Location: Below

Icon Label Font Color: ...

Icon Label Font: ABCabc ...

OK Cancel Apply

Figure 4. Mockup of the Geospatial View’s Image Location Dialog

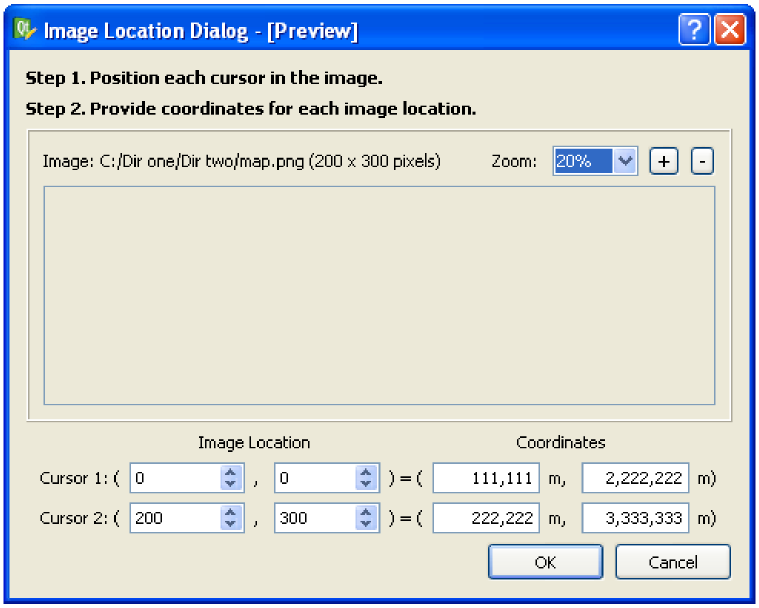


Image Location Dialog - [Preview]

Step 1. Position each cursor in the image.

Step 2. Provide coordinates for each image location.

Image: C:/Dir one/Dir two/map.png (200 x 300 pixels)

Zoom: 20%

	Image Location			Coordinates	
Cursor 1: (	0	, 0	) = (	111,111 m,	2,222,222 m)
Cursor 2: (	200	, 300	) = (	222,222 m,	3,333,333 m)

5.3 Georeferencing objects

To support the Geospatial View of the workspace, RiverWare must associated coordinates in the view’s projection with each object on the workspace. To provide more flexibility for the user, we distinguish between *actual* and *display* spatial coordinates.

These coordinates are associated with objects and are numeric, so we store them in a 2 x 2 TableSlot, called the “Spatial Coordinates” slot. The first column provides the x coordinate, the second column the y coordinate. The first row contains the actual coordinates; the second row contains the display coordinates. If in the future we add an additional geospatial view which has a different projection, then we can add additional rows labeled appropriately.

The unit type for the Spatial Coordinates TableSlot will not be user configurable, rather it will be kept in agreement with the Geospatial View’s projection. When the user changes the unit type of that view’s projections, the unit type for all Spatial Coordinates slots will be changed but the value will not be changed. Although we could consider enforcing agreement of the unit, scale, and precision in a similar way, initially we will let these be under user control in the usual way.

RiverWare will provide the following mechanisms to edit spatial coordinates:

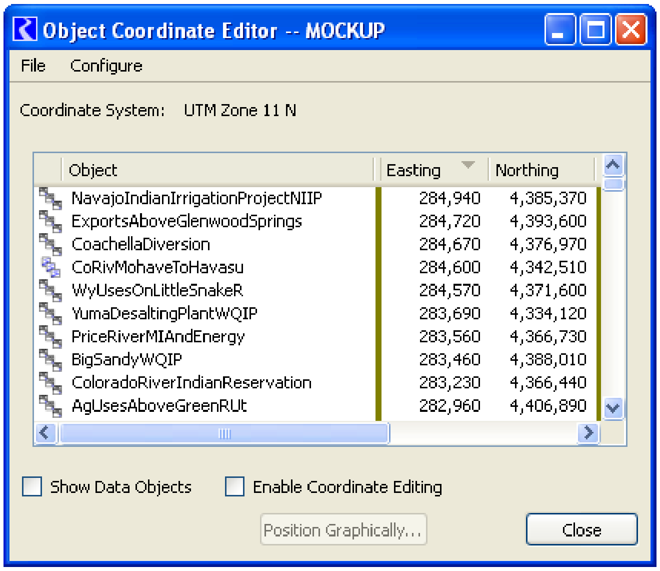
- The Spatial Coordinates TableSlot will be visible on all slots, and the users can interactively edit a single object’s spatial coordinates in the usual way. In addition DMI’s can be used to import object coordinates.
- A new dialog, the Object Spatial Coordinates Manager dialog, will display spatial coordinates for all objects in a single dialog. This dialog will support editing of individual fields as well as cut/paste/copy (see section below for more details).
- Moving an object’s icon in the Geospatial View (by using the drag and drop technique) will change the object’s *display* spatial coordinates accordingly. RiverWare will provide some sort of mechanism to facilitate reconciliation of an object’s actual and display spatial coordinates.

Internally, when an object is created it is given a location in the Simulation and Accounting Views, and the Geospatial View spatial coordinates will be initialized in a similar way.

5.4 The Object Spatial Coordinates Manager

The Object Spatial Coordinates Manager dialog displays spatial coordinates for all visible workspace objects in a single dialog. This dialog will support editing of individual fields as well as cut/paste/copy. For each object, both the display and actual spatial coordinates will be displayed.

Figure 5. A prototype Object Spatial Coordinates Manager dialog (lacking actual/display distinction).



Coordinate System: UTM Zone 11 N

Object	Easting	Northing
NavajoIndianIrrigationProjectNIIP	284,940	4,385,370
ExportsAboveGlenwoodSprings	284,720	4,393,600
CoachellaDiversion	284,670	4,376,970
CoRivMohaveToHavasu	284,600	4,342,510
WyUsesOnLittleSnakeR	284,570	4,371,600
YumaDesaltingPlantWQIP	283,690	4,334,120
PriceRiverMIAndEnergy	283,560	4,366,730
BigSandyWQIP	283,460	4,388,010
ColoradoRiverIndianReservation	283,230	4,366,440
AgUsesAboveGreenRUt	282,960	4,406,890

☐ Show Data Objects ☐ Enable Coordinate Editing

RiverWare will support the import and export of spatial coordinates. Intially RiverWare will only support the ESRI shapefile format for this operation. In this context, the shapefile format is a collection of three files with different suffixes:

- .shp - objects represented as points.
- .shx - indices into .shp file
- .dbf - object attributes (initially object name and object type)

RiverWare will support export of the spatial coordinates of all or selected objects. When coordinates are imported, the user will be warned that the existing coordinates will be overwritten and notified when coordinates for unknown objects are encountered. In the future, RiverWare might support the association of a namemap with the Geospatial View which would allow more flexibility with regard to naming.

If in the future we support other file formats for exchanging object information, then we might want to consider designing a plugin interface, allowing us to move the GIS format knowledge information out of RiverWare proper.

The Object Spatial Coordinate Manager is non-modal and accessed via the main window's Workspace->Objects menu as well as via the workspace's context sensitive menu (regardless of view).

5.5 Miscellaneous

RiverWare will add support for a new unit type, Angle, which will initially have only a single unit, Decimal Degree (DD).

6 High level software architecture

To integrate GIS functionality with RiverWare, we might want to apply one or more of the following software approaches.

- In house - we write code to do what needs to be done. This is what Riverside Technology did for their StateMod application.
- Link to a GIS library - link RiverWare to a 3rd party library with a C++ API that provides the desired functionality.
- Define a GIS plugin interface and implement one or more GIS plugins - this would support extensibility and dynamic linking, allowing only those users which use the plugins to pay their associated overhead.
- Add RiverWare as a plugin or customization of an existing GIS stand-alone tool - This is what DHI did for their Mike Basin application (using the ArcGIS application).
- Communicate in real time with a GIS application. Spatial databases might be accessed via DMIs, or in the future perhaps as a web service.

7 Open source GIS software

The following are open source GIS packages that are available on at least the Windows and Linux platforms. Many other GIS packages are more restrictive, for example, Idrisi and MapWindow are supported on Windows only, and GeoVISTA Studio consists of Java bean components.

- GRASS: Open source C stand-alone GIS application and library, supported on Window/Linux. GRASS/Qt is a library for accessing GRASS functionality from Qt developed and made available as open source by Navicon.
- Quantum GIS (QGIS): Open source stand-alone GIS application and library, supported on Window/Linux/others, uses Qt4. Documentation contains an example of accessing QGIS from other applications. From the documentation: “Required dependencies of Quantum GIS include GEOS and SQLite. GDAL, GRASS GIS, PostGIS, and PostgresSQL are also recommended.”
- GDAL/OGR: Open source C++ library for reading/writing various formats (the GDAL library supports raster formats, OGR supports vector formats).

Figure 6. Screen snapshot of Quantum GIS

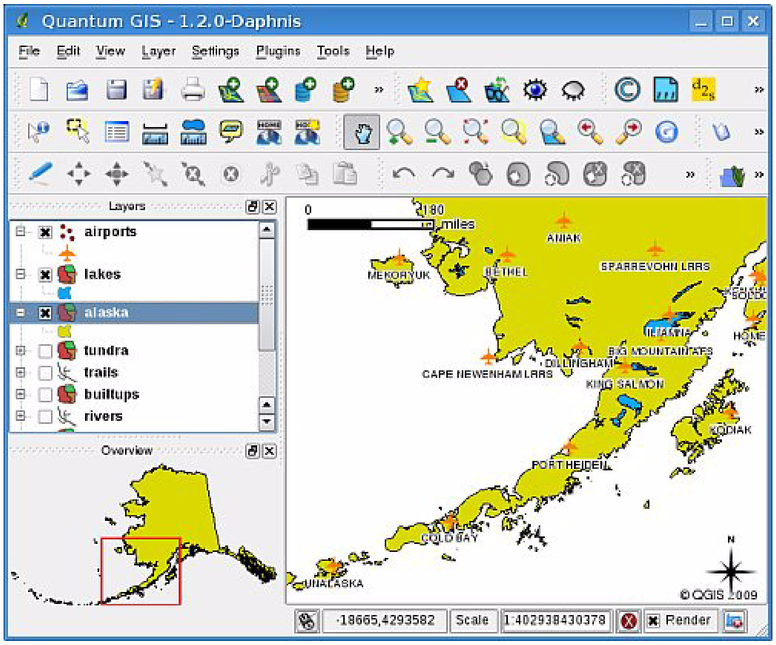
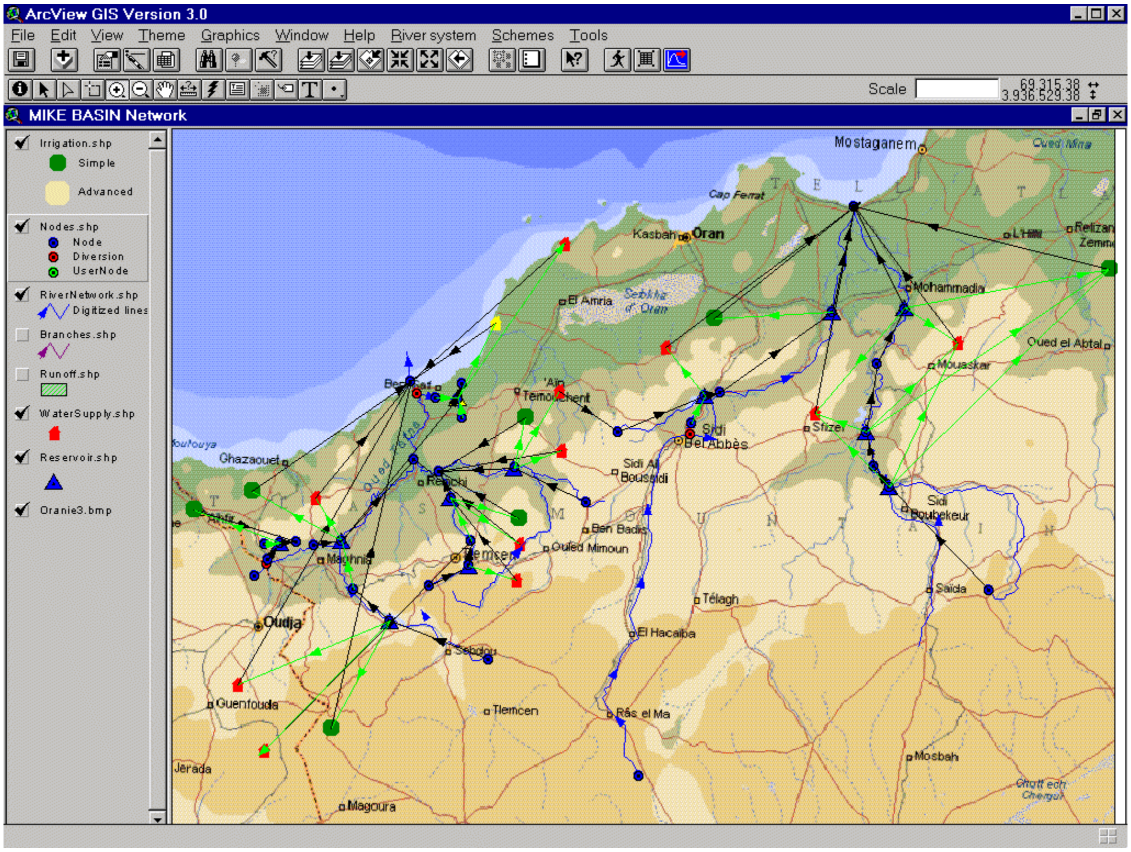


Figure 7. Screen snapshot of DHI’s Mike Basin tool & ArcGIS



8 Notes

How should we deal with objects that do not have spatial coordinates, including all objects of type DataObj and ThermalObj? Note that DataObj’s are in practice often associated with actual physical objects (e.g., each reservoir object in a model might also have an associated DataObj.

While RiverWare should be flexible with respect to scale, there is a range of scales which are more likely to be of interest at the basin level (e.g., maps with a 1:240,000,000 scale would likely not be very useful). Does this have any implications for the current design?

Consider relevance of the Spatial Data Transfer Standard, a standard developed by the USGS used to describe earth-referenced spatial data, designed to support transfer and use spatial data on different computer platforms.

Do we need to provide a mechanism to document the datum associated with spatial data?

What support should be provided for file references embedded in model files? Currently, we support pathnames relative to the model file, and there is a software hook to allow extension to other approaches. At a minimum, we should probably allow for full pathnames and environment variable expansion.

Are there any GIS formats for saving an image together with projection information? If so, we should consider using and/or supporting that format at a relatively early stage of GIS integration. What support is provided by ArcGIS for saving metadata along with images?

Phil suggests that it would be a bad idea to use world coordinates as the scene coordinates of the Qt canvas. As far as I understand, his reasoning is something like: while Qt does provide flexibility with respect to scene coordinates, RiverWare will be displaying icons on the canvas, these icons are based on 40x40 raster images, and transforming them to display in some arbitrary scene coordinate system would lead to a problem. This problem is that displaying a raster image, such as our icon images or the background images, is best done when one cell of the raster image is displayed in one pixel on the screen. When this is not true it leads to a degradation of the quality of the image as well as performance issues on X11-based platforms.

How do the users indicate that a drag and drop should affect the actual coordinates? Perhaps a toggle in the canvas configuration dialog that indicates that drag/drop should effect both, off by default, the setting saved in the model file?

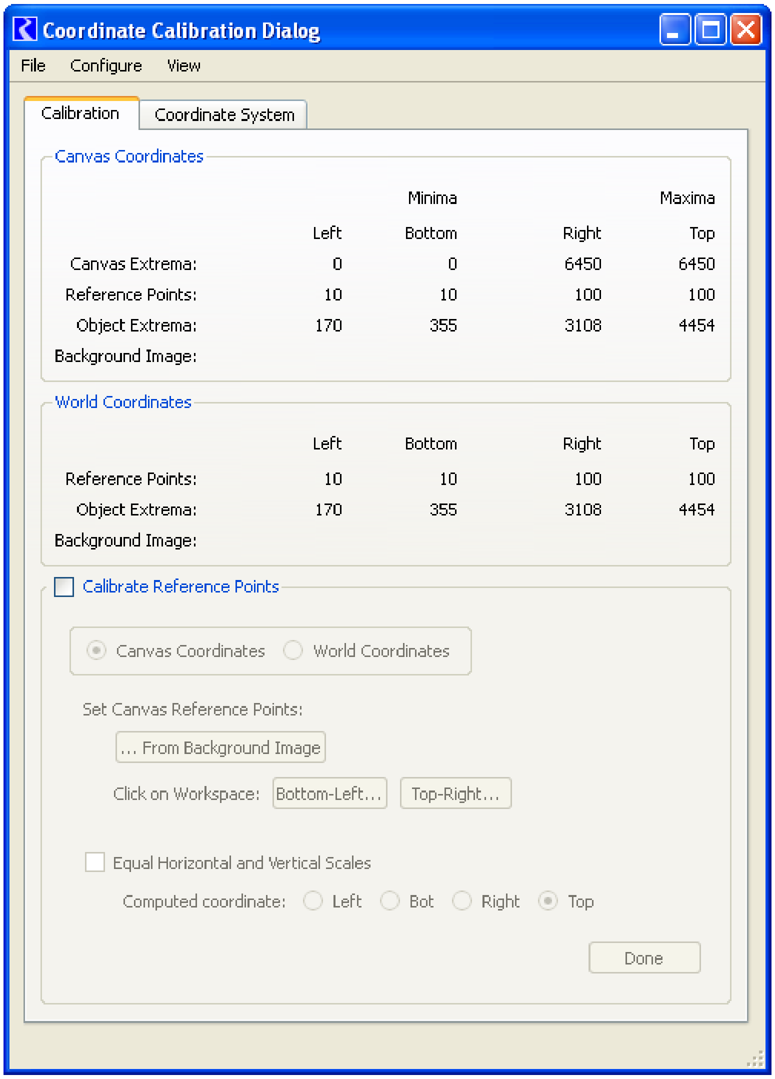
9 Schedule

Time	Who	Task
Geospatial View		
0.5		Add spatial coordinates TableSlot to SimObj
1.5		Create Geospatial View and its canvas configuration dialog (GeospatialCanvasConfigDlg, minimally functional)
1.0		User configuration of Coordinate System units
1.5		Drag/drop affects object display spatial coordinates
2.0		Basic support for background image (including environment variable expansion)
2.0		Control over maximum icon size
2.0		Control over icon label location
1.5		Configuration dialog canvas/image thumbnail (with 4 indicators)
1.5		Configuration dialog icon thumbnail
1.5		Persistence for configuration dialog settings (background image path, etc.)
0.5		Set canvas rectangle from background image rectangle (with margin)
1.5		Set display scale to maximize image quality
0.5		Background image opacity control, toggle display of image
0.5		Control over coordinate system axes text (“x”, “Easting”, ...)
0.5		Optional automatic location of objects in geospatial view
Object Spatial Coordinates Manager Dialog		
		New dialog with basic operation
		Set actual spatial coordinates from display spatial coordinates
		Import/export of coordinates
		Copy/paste

Time	Who	Task
Background Image Location Dialog		
1.		New class: ImageLocationDlg (minimal functionality)
1.5		Create cursors, support drag and drop
0.5		Connection between second cursor’s x and y coordinates
1.0		Zooming
Optional features		
0.5		Relative path specification for background image
0.5		Toggle of schematic view
0.5		Tool tip/status bar display of SimObj in geospatial view
0.5		Persistence of tics which were used to locate an image
0.5		Toggle for display of Spatial Coordinates slots
1.0		Control over minimum icon size
1.0		Add Angle unit type and units of DD (decimal degrees)
Wrap-up		
3.0		Testing/debugging
3.0		Documentation

10 Appendix: Screen snapshots of Phil’s Coordinate Calibration Dialog mockup

Figure 8. An interface for collecting projection information (part 1).



Canvas Coordinates

	Minima		Maxima	
	Left	Bottom	Right	Top
Canvas Extrema:	0	0	6450	6450
Reference Points:	10	10	100	100
Object Extrema:	170	355	3108	4454
Background Image:				

World Coordinates

	Left	Bottom	Right	Top
Reference Points:	10	10	100	100
Object Extrema:	170	355	3108	4454
Background Image:				

□ Calibrate Reference Points

● Canvas Coordinates ○ World Coordinates

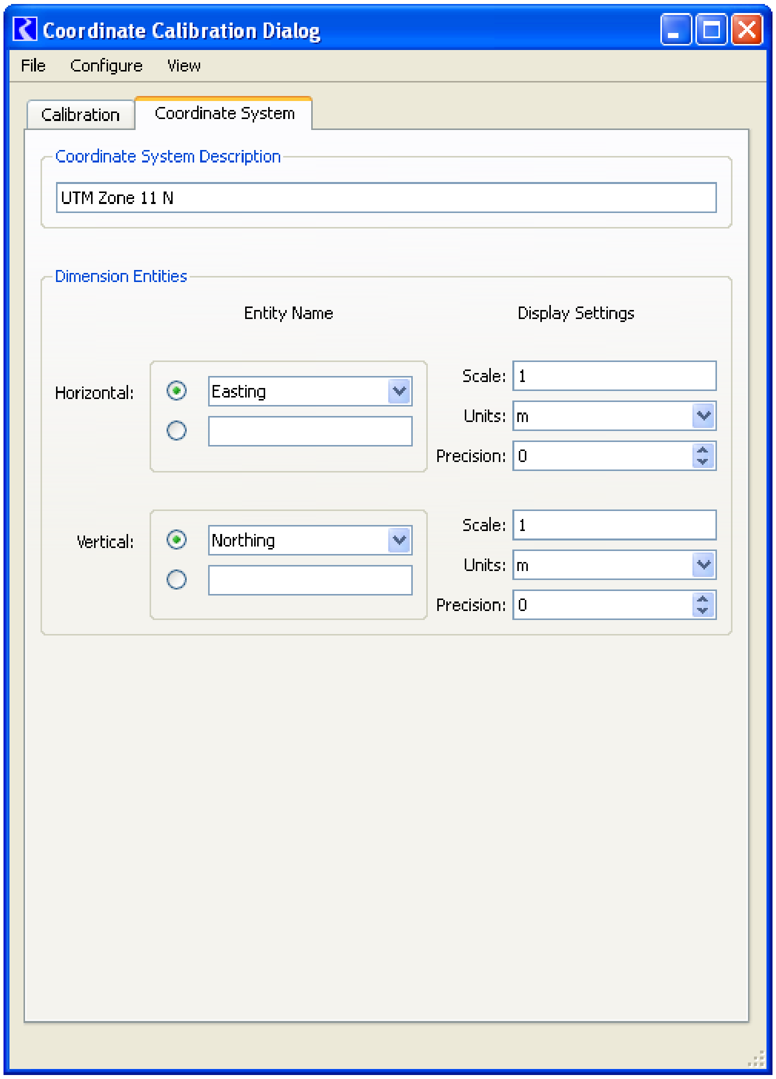
Set Canvas Reference Points:

Click on Workspace:

□ Equal Horizontal and Vertical Scales

Computed coordinate: ○ Left ○ Bot ○ Right ● Top

Figure 9. An interface for collecting projection information (Part 2)



Coordinate System Description

UTM Zone 11 N

Dimension Entities

	Entity Name	Display Settings
Horizontal:	● Easting ○	Scale: 1 Units: m Precision: 0
Vertical:	● Northing ○	Scale: 1 Units: m Precision: 0