**The Logic Laboratory Users Guide**

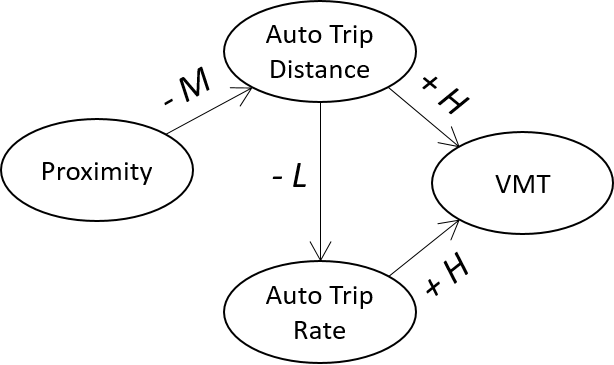
**Brian Gregor P.E.**

**Oregon Systems Analytics, LLC**

**11/15/16**

# Background on the Logic Laboratory and Fuzzy Systems Dynamics Models (FSDM)

The Logic Laboratory is used to create and run fuzzy systems dynamics models. These are variants of fuzzy cognitive map (FCM) models. The model is a directed graph whose nodes are concepts that are being modeled and whose edges specify relationships between concepts. Figure 1 shows an example. The direction of each edge (i.e. the direction of the arrow) specifies the relationship between causal and affected concepts. In the Figure 1 example, the *Proximity* concept is shown as affecting the *Auto Trip Distance* concept. Edge weights specify the strength and directionality of causal effects. A positive sign for an edge weight means that an increase in the causing concept causes an increase in the affected concept. In the Figure 1 example, an increase in *Auto Trip Distance* concept causes an increase in the *VMT* concept and a decrease in the *Auto Trip Rate* concept. In the FSDM, as in the FCM, the strength of causal effects is expressed in ‘fuzzy’ terms. For example in Figure 1, the weights are L (low), M (moderate), and H (high).



**Figure 1. A Simple FSDM & FCM Example**

The main difference between the FSDM and FCM is in the meaning of the edge weights. In the standard FCM, the value of the causal concept affects the value of the affected concept. In mathematical terms, the value of the affected concept is a function of the value of the causal concept multiplied by the value of the weight of the edge that connects them. In the FSDM, a change in the value of the causal concept affects a change in the value of the affected concept: mathematically, the proportional change in the affected concept is a function of the proportional change in the causal concept multiplied by the value of the edge weight. In other words, in the FSDM, edge weights are similar to elasticities. In the case of both models, concept values are measured in relative terms on a scale from 0 to 1 that denotes the proportion of some maximum assumed value. The FSDM approach was developed because the FCM approach was found to produce models that are unstable (i.e. don't converge to a solution) and that produce counter intuitive results. In addition, the standard FCM, which is a type of neural network, is harder to relate to public policy questions than the FSDM.

An advantage of FSDM models over other models in some circumstances is that they can be used to model systems of interactions where there is limited data. These are logical models that enable users to evaluate the consequences of a system of logical cause and effect relationships that are tied together in a system that includes feedbacks. This modeling approach has been tested with a real world problem in a joint research effort by the Oregon DOT (ODOT), Oregon State University (OSU), and Oregon Systems Analytics (OSA). A FSDM has been created to model the effects of new transportation technologies and services on metropolitan area travel behavior and transportation outcomes. Appendix B contains more detailed technical information about the FSDM and its research application to date.

Oregon Systems Analytics (OSA) developed the methods, algorithms, and code for implementing FSDM models for the Oregon Department of Transportation to support research into the potential effects of emerging transportation technologies and services on travel behavior and transportation system performance. The code is written in the R language. Models and scenarios are specified in JSON-formatted text files.

The **Logic Laboratory** is the graphical user interface (GUI) developed by OSA to simplify the process of using the FSDM and viewing the results of model runs. Since the FSDM supports the implementation of logical models, it is important that FSDM models be developed with strong logical foundations that are well articulated. Because of this, the Logic Laboratory facilitates good documentation in conjunction with model development.

# Setting up and Running the Logic Laboratory

The Logic Laboratory is an R Shiny application that has been built to run locally on a user’s computer. Shiny is a web application framework developed by RStudio to enable people to develop web applications for data analysis in the R programming language. The web-based user interface is coupled with an R server that executes R commands. In order to use the current version of the Logic Laboratory it is necessary to have your computer set up to run R and RStudio. In addition, several R packages need to be installed.

Install R on your computer. R is available for the Windows, Apple, and Linux operating systems. Installers for these systems and installation instructions are available at <https://www.r-project.org/>. If you already have R installed, you should update it to the latest version.

Install R Studio on your computer. R Studio is available for the Windows, Apple, Linux operating systems. Installers for these systems and installation instructions are available at <https://www.rstudio.com/products/rstudio/download/>. If you already have R Studio installed, you should update it to the latest version.

Install these R packages from the Comprehensive R Archive Network (CRAN): shiny, shinyBS, DT, ggplot2, and DiagrammeR. It is important that the latest versions be installed. This is particularly the case for the “shiny” package. This can be done in R Studio. It can also be done with the following command in the R console:

install.packages(c(“shiny”, “shinyBS”, “DT”, “ggplot2”, “DiagrammerR”))

After your computer has been set up with R, R Studio, and required R packages, you are ready to download and run the Logic Laboratory. The program and example models can be downloaded from the following GitHub repository: <https://github.com/gregorbj/FSDM_GUI>. If you are familiar with using “Git”, you can clone the repository. Otherwise you can download a zip archive. Just click on the button which looks like the following on the webpage to begin the clone or download process:

## The Logic Laboratory File Structure

After the Logic Laboratory has been downloaded from GitHub and extracted (if downloaded as a zip file), the files will be organized as shown in Figure 2 below.

FSDM\_GUI

| fsdm.rproj

|

|---fsdm-app

| |--- ui.R

| |--- server.R

| |--- helper.R

| |--- www

| (image files used in the interface)

|

|---models

|

|---Model1

| |--- concepts.json

| |--- relations.json

| |--- status.json

| |

| |--- scenarios

| | |--- Scenario1

| | | |--- scenario.json

| | | |--- status.json

| | | |--- Outputs\_ls.RData (created after model is run)

| | (more scenarios)

| |

| |--- analysis

| |--- Analysis1

| | |--- data.csv

| | |--- plot.png

(more models) (more analyses)

**Figure 2. Logic Laboratory File Structure**

The **fsdm.rproj** file is an *RStudio* project file. This file is the entry point for the application. Double-clicking on this file opens R studio which points to the R scripts that are located in the **fsdm-app** folder. All models built using the Logic Laboratory are stored in the **models** folder. Each model and all the scenarios and model results for the model are stored in a folder that is named with the model name. Each model folder contains three files that include all the information for describing the model. The **concepts.json** file contains information which defines and documents the model concepts and their value ranges. The **relations.json** file contains information about all the causal relationships between concepts including the directions and strengths of the relationships. The **status.json** file keeps track of important model status information such as when it was created, when it was last edited, and when it was validated. These files are created and modified by the Logic Laboratory when you work on a model. Although these three files are json-formatted text files, it is not a good idea to edit any of them in a text editor because doing could easily make them unusable in the Logic Laboratory.

The model folder contains two other folders, a **scenarios** folder and an **analysis** folder. The scenarios folder can contain specifications and results for a scenario. The folder is named with the name of the scenario. Each scenario contains two files which describe the scenario: **scenario.json** and **status.json**. These files are created and edited by using the Logic Laboratory. When the scenario is modeled, a third file is created in the folder: **Outputs\_ls.RData**. This file is an R binary file that can be loaded in R and analyzed. Its structure is described in Appendix A.

The **analysis** folder contains one or more folders containing the results of analyzing scenario results using the Logic Laboratory. These folders are created any time an analysis is performed and saved. Each folder contains a file named **data.csv** that is a “csv-formatted” text file containing the data used in the analysis, and a corresponding **plot.png** file which is a graph of those data. Appendix A describes the structure of the **data.csv** file in detail.

## Starting the Logic Laboratory

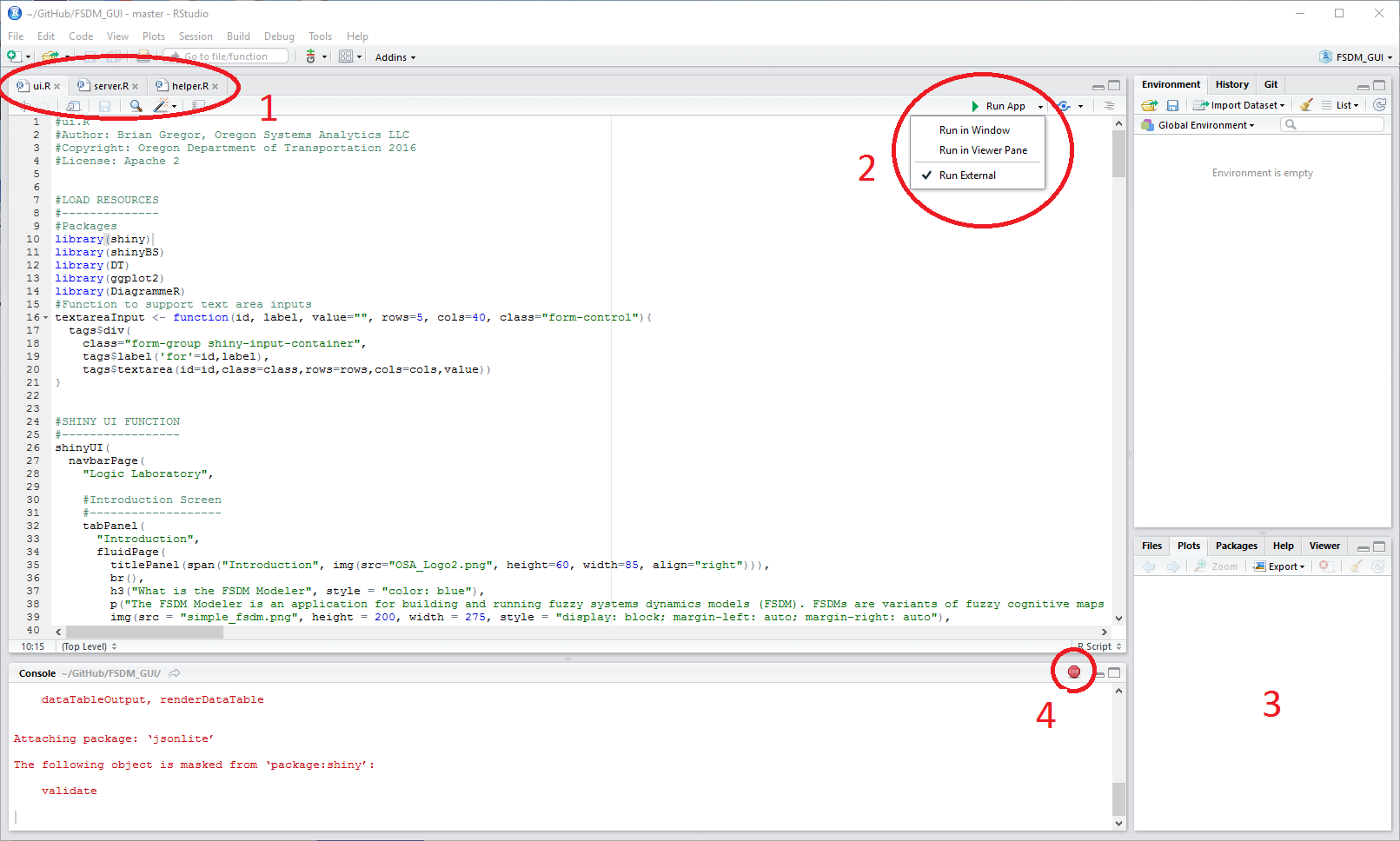
Figure 3 shows what the R Studio interface looks like when the fsdm.rproj file has been opened. On the top-left side of the window you will see three tabs with the names of the R scripts that contain the program code. (Highlighted area 1 in Figure 3) In order to run the program it is necessary for either the **ui.R** or the **server.R** tab to be selected. If either of these are selected, a **Run App** button will appear on the top-right side of the window. (Highlighted area 2 in Figure 3)

Clicking on the **RunApp** this button will start the Logic Laboratory and display the user interface. Before doing so, however, it is recommended that you select how the Logic Laboratory will be displayed. There are three display options that you can choose from a menu which appears when you click on the “down arrow” to the right of the **Run App** button. Figure 3 shows what this menu looks like. If you choose **Run in Window**, the interface will displayed in a window which is created for that purpose. If the **Run in Viewer Pane** option is chosen, the interface will be displayed in the small pane in the bottom right of the R Studio window. Finally, if **Run External** is chosen, the interface will be displayed in the default web browser on your computer.

The best option for most users will be the **Run in Window** option because it will create a dedicated window that can be easily resized to show the whole interface. Moreover, the program can be stopped by simply closing the window.

The **Run in Viewer Pane** option is not recommended because the interface will be displayed in the viewer pane at the bottom right of the R Studio window. (Highlighted area 3 in Figure 3) This viewer pane is much too small. Although it can be resized, doing so is not as easy as resizing a window and reduces the other panes in the R Studio window.

The next best option is the **Run External** option. This opens the interface in your default browser. It will open a browser window if one is not open. Otherwise it will open in a new tab in an open browser window. This option is not as convenient as the first option for a couple of reasons. First, closing the browser window or browser tab doesn’t stop the program. To stop the program, you also have to click on the stop sign in the R Studio window. (Highlighted area 4 in Figure 3.) Moreover, clicking on the stop sign will not close the browser tab. It will just make the tab inactive. Starting the program again will not reactivate the tab, it will create a new tab. Thus, if you stop and start the program multiple times with this option, you can end up with multiple inactive Logic Laboratory tabs. Second, it is not as easy to find where the interface is if multiple browser windows and/or tabs are open, especially if the application has been started and stopped multiple times and inactive windows have not been closed. One advantage of this option, however is that unlike the first option, the window will not need to be resized every time the application is started.

**Figure 3. RStudio Window with the fsdm.proj File Loaded**

## The Logic Laboratory Introductory Screen

Figure 4 shows what the interface looks like when the application is started and a menu item has been selected, showing the submenu items.

The introductory screen provides some brief background information on fuzzy systems dynamics models and the Logic Laboratory, as well as copyright and licensing information. The menu bar at the top of the screen is used to select different interface functions. The names of the menu items are number in the order of the steps involved in creating and running a model, and analyzing the model results. Some of the menu items have submenus that appear when the user selects a menu item. The four numbered menu items are as follows:

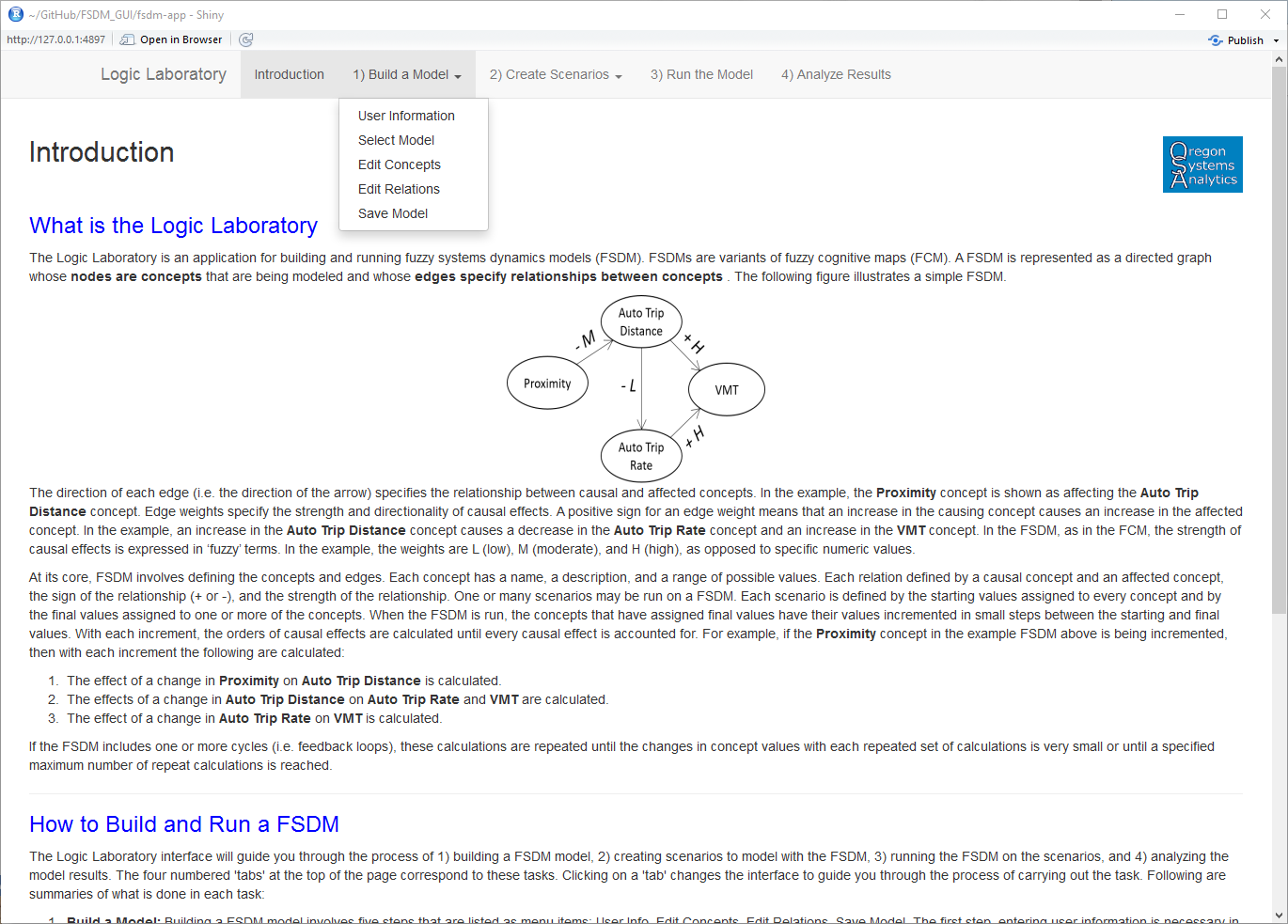
1) Build a Model: Building a FSDM model involves five steps that are listed as menu items: User Information, Select Model, Edit Concepts, Edit Relations, and Save Model. The first step, entering user information is necessary in order to attribute models and model edits. The second step, selecting a model, shows a view which enables you to start a model from scratch, copy an existing model to serve as the starting point for your model, edit an existing model, or run an existing model without editing. The third step, editing concepts, is where you define new concepts or edit existing concepts and the range of values that they may have. The fourth step, editing relations, is where you define the relationships between concepts (magnitude and direction). The final step, saving the model, permanently saves all of the model edits along with notes to document the model or edits to the model.

2) Create Scenarios: One or more model scenarios can be created for a FSDM. This 'tab' of the application assists you with creating and editing scenarios. A new scenario can be started from scratch or created from an existing scenario. Scenarios are validated against the defined model to assure that scenario values are consistent with defined concept value ranges.

3) Run the Model: The FSDM may be run for one or more of the scenarios. This 'tab' enables you to specify which of the scenarios to run. The interface shows a checkbox list of all of the scenarios that have been created and validated for a model. Entering a check in a checkbox marks the corresponding scenario to be run. A button starts the model runs. Progress of the model run is shown as the model run proceeds.

4) Analyze Results: This 'tab' of the application enables users to view the results for one or two scenarios. The user can choose which model scenarios and which concept values to display. The results are displayed in graphs. Users can choose to save the displayed graphs and corresponding data.

Following sections of the user’s guide describe each of these steps in more detail.

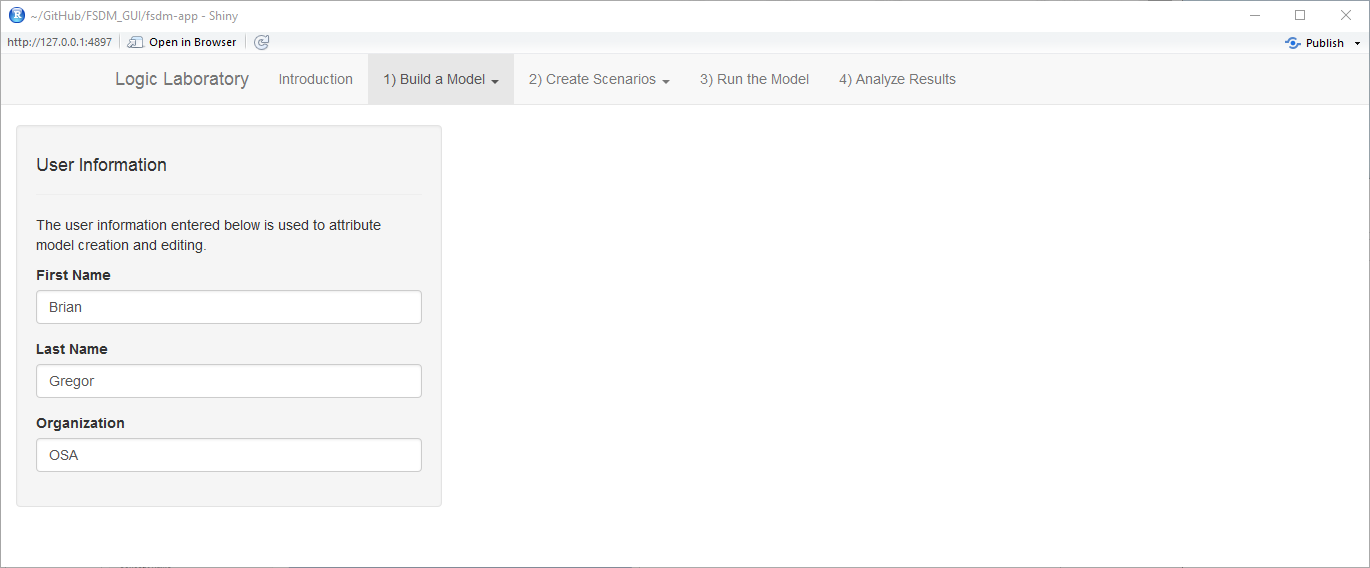
 **Figure 4. Introductory Screen for FSDM GUI Showing Overall Appearance and Drop-down Menu**

# Build a Model

There are 5 steps to building a model: entering user information, selecting the model, editing concepts, editing relations, and saving the model. Each of these steps is described in the following subsections.

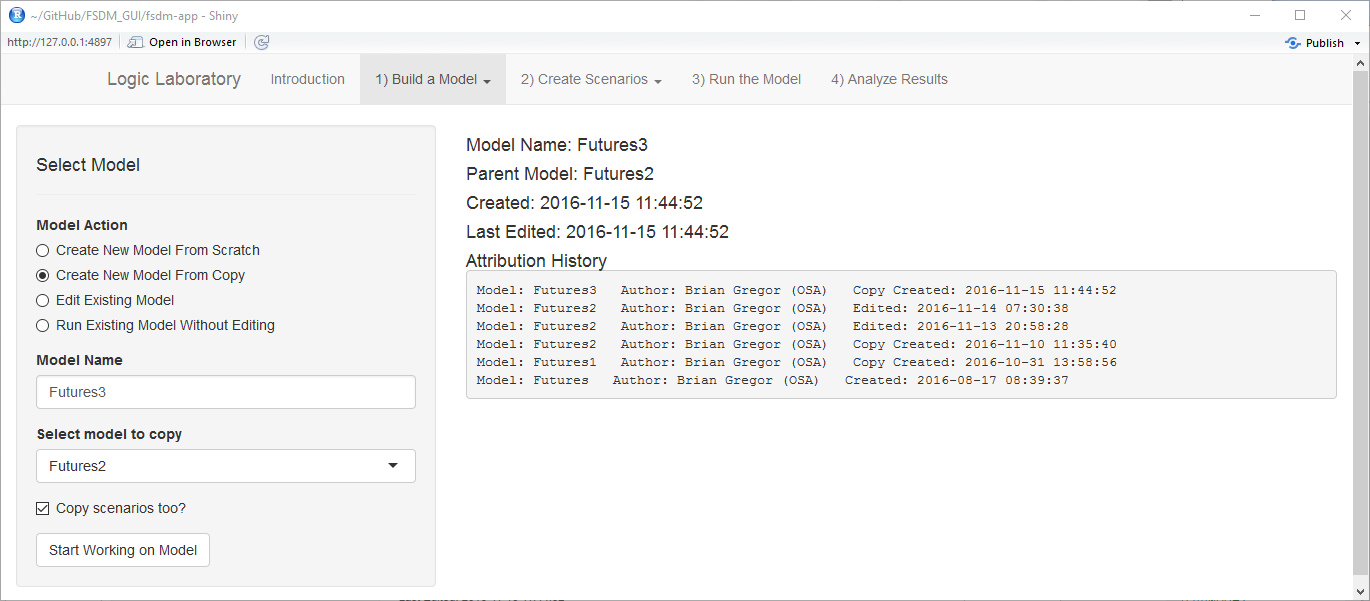
## User Information

Figure 5 shows the **User Information** screen. The user information is used by the program to attribute model creation and edits. **First Name** and **Last Name** entries are required if a model is to be created or edited. These entries are not required if you only want to run a model without making any edits. The **Organization** entry is optional. You can select an entry item by clicking on it. You can switch between items by pressing the tab key (to go forward) or the shift and tab keys (to go backward).

**Figure 5. User Information Screen**

## Select Model

Figure 6 shows the **Select Model** screen. There are four choices for selecting a model as shown in the **Model Action** list on the left hand side of the screen. The figure shows the example of when the **Create New Model From Copy** choice is selected. In this case, the user is presented with a text entry in which to type the name for the model to be created and a selector to select the model to copy. Clicking on the downward-pointing arrow head at the right side of the selector displays a listing of all of the models from which the desired model can be selected. The **Model Name** text box is also displayed when the **Create New Model From Scratch** model action is selected. The model selector box is also shown when the **Edit Existing Model** or **Run Existing Model Without Editing** model actions are selected. The labels will be different, however, depending on the chosen action. If the **Create New Model From Copy** model action is selected, a checkbox will appear to ask whether you want to copy all the scenarios for the model to be copies as well. When the **Start Working on Model** button is clicked, the model will be loaded and model status information will be displayed on the right-hand side of the screen.

**Figure 6. Select Model Screen**

## Edit Concepts

Figure 7 shows the **Edit Concepts** screen. On the left side of the screen are text input boxes and buttons that are used to edit the concepts. On the right side of the screen is a table that shows a summary of the concepts and their values and is used to select the concept to edit. In the example shown, the “Auto Miles Per Gallon” concept is selected. To select a concept, you click on that concept entry in the table. The table row for the selected concept will be highlighted. All of the information about the concept is shown in the text boxes on the left-hand side of the screen.

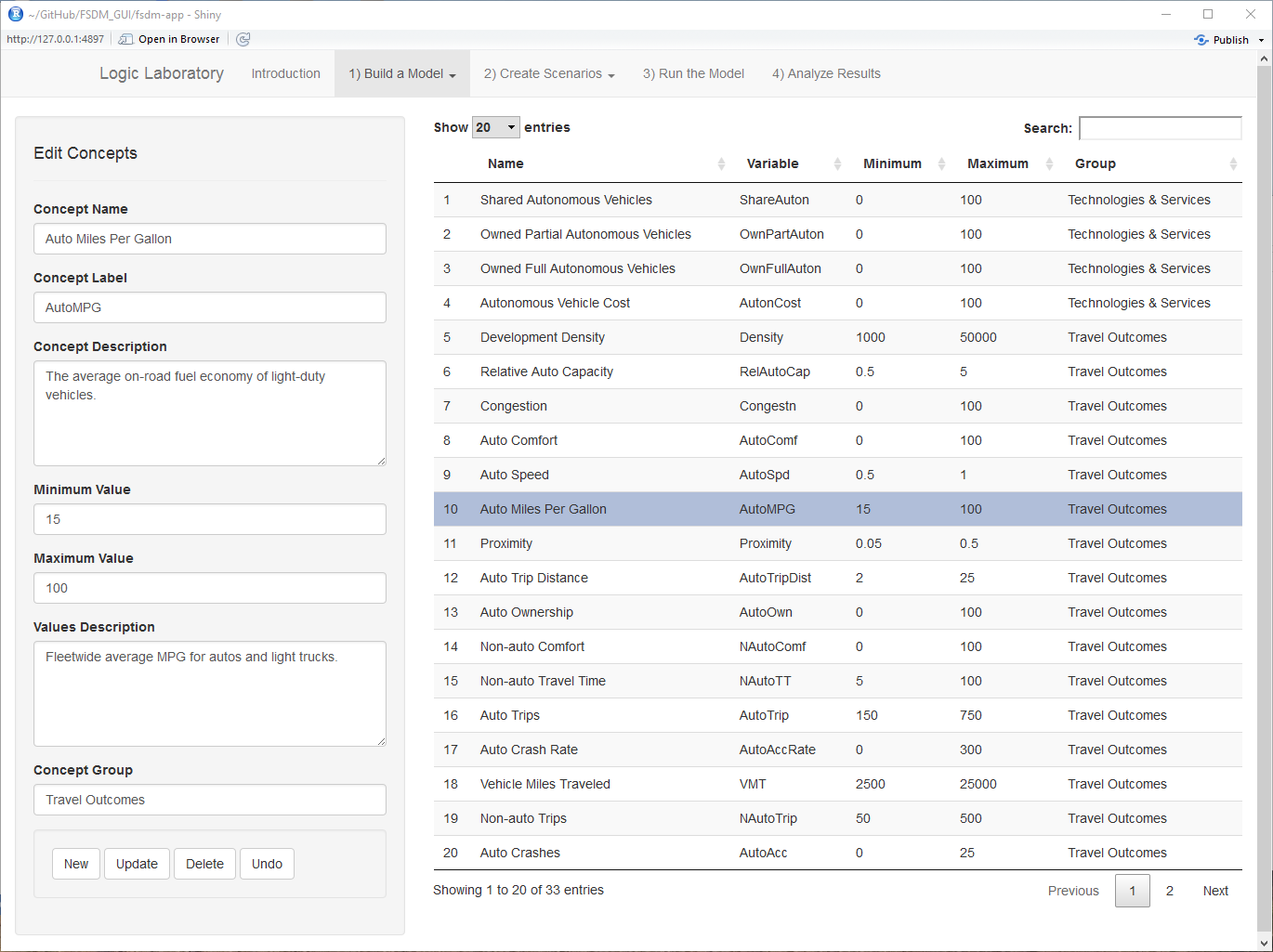
You type in the text boxes to edit the concept values. The meanings of the text boxes are as follows:

* **Concept Name**: This is a descriptive name or phrase for the concept.
* **Concept Label**: This is a one-word label for the concept. Concept labels are used for labeling diagrams and as identifiers in the program logic. It is strongly recommended that the labels be short (one word) and descriptive. It is also recommended that “Pascal” case be used to separate portions of the label to make them more descriptive. For example “AutoComf” for “Auto Comfort”.
* **Concept Description**: This is an extended description of the meaning of the concept.
* **Minimum Value**: The minimum value that the concept may have in the units of measurement described in the “Values Description”.
* **Maximum Value**: The maximum value that the concept may have in the specified measurement units.
* **Values Description**: This is an extended description of the meaning of the concept values. The description should include identification of the measurement units.
* **Concept Group**: Concepts may be assigned to groups. These group assignments are useful for partitioning the display of the FSDM, especially for complex models like the “Futures2” model shown in Figure 7. Graphical examples are shown in the next section.

At the present time there is no program logic to check whether the entries are correct and/or complete. It is the responsibility of you, the user, to make sure that you enter correct information. Needless to say, only put numeric values in the **Minimum Value** and **Maximum Value** text boxes, and make sure that the **Minimum Value** is less than the **Maximum Value**. Once the entries have been edited to your satisfaction, click on the **Update** button to change the model. The model changes will only take place if the **Update** button has been clicked.

If you want to add a new concept, click on the **New** button. This will put a new blank row at the top of the table and will highlight the new row. Then the values for the concept can be entered in the text boxes. Click on the **Update** button when done.

A concept may be deleted by highlighting it and then clicking on the **Delete** button. If you’ve deleted a concept by mistake, you can correct your mistake by clicking on the **Undo** button. Similarly you can correct an editing mistake after you’ve clicked the **Update** button by clicking on **Undo**. You should note that **Undo** only undoes the last action (Update or Delete), it does not undo previous actions.

**Figure 7. Edit Concepts Screen**

## Edit Relations

Figure 8 shows the edit relations screen. This screen is somewhat more complex than the previous screens with tabbed displays on both the left side and the right side of the screen. On the left side the **Edit Relations** tab displays text boxes showing values for selected relations. The **Relations Graph Format** tab displays text boxes which are used to modify the format of the **Relations Graph** shown on the right-hand side. The other tab on the right-hand side, **Relations Map**, displays the relationships in another format. Figure 9 shows what the display looks like when the **Relations Graph Format** and **Relations Map** tabs are selected.

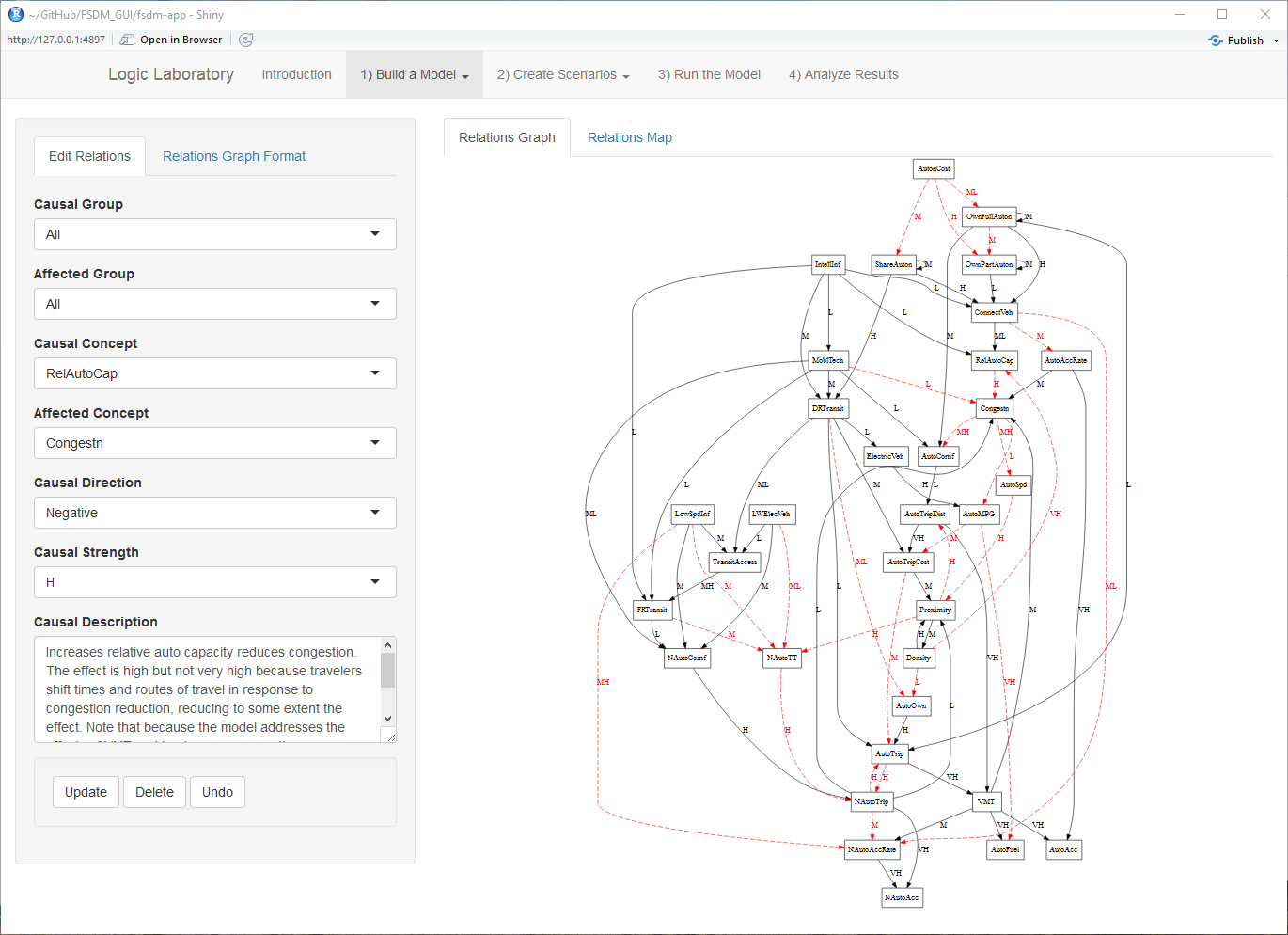
The **Relations Graph** and **Relations Map** are two different ways of looking at the relationships in the model. The **Relations Graph** is a directed graph[[1]](#footnote-1) that show concepts as nodes and relations as arrow pointing from causal concepts to the concepts they affect. Figure 8 shows what a complex relations graph looks like. Solid black arrows show positive relationships, where a change in the causal concept causes a change in the affected concept in the same direction (e.g. an increase in the causal concept causes an increase in the affected concept). Dashed red arrows show negative relationships, where a change in the causal concept causes a change in the affected concept in the opposite direction (e.g. an increase in the causal concept causes a decrease in the affected concept). The strength of the relationship is denoted by the letter codes as follows: VL = very low, L = low, ML = moderately low, M = moderate, MH = moderately high, H = high, VH = very high. The program converts these codes into numerical values behind the scenes.

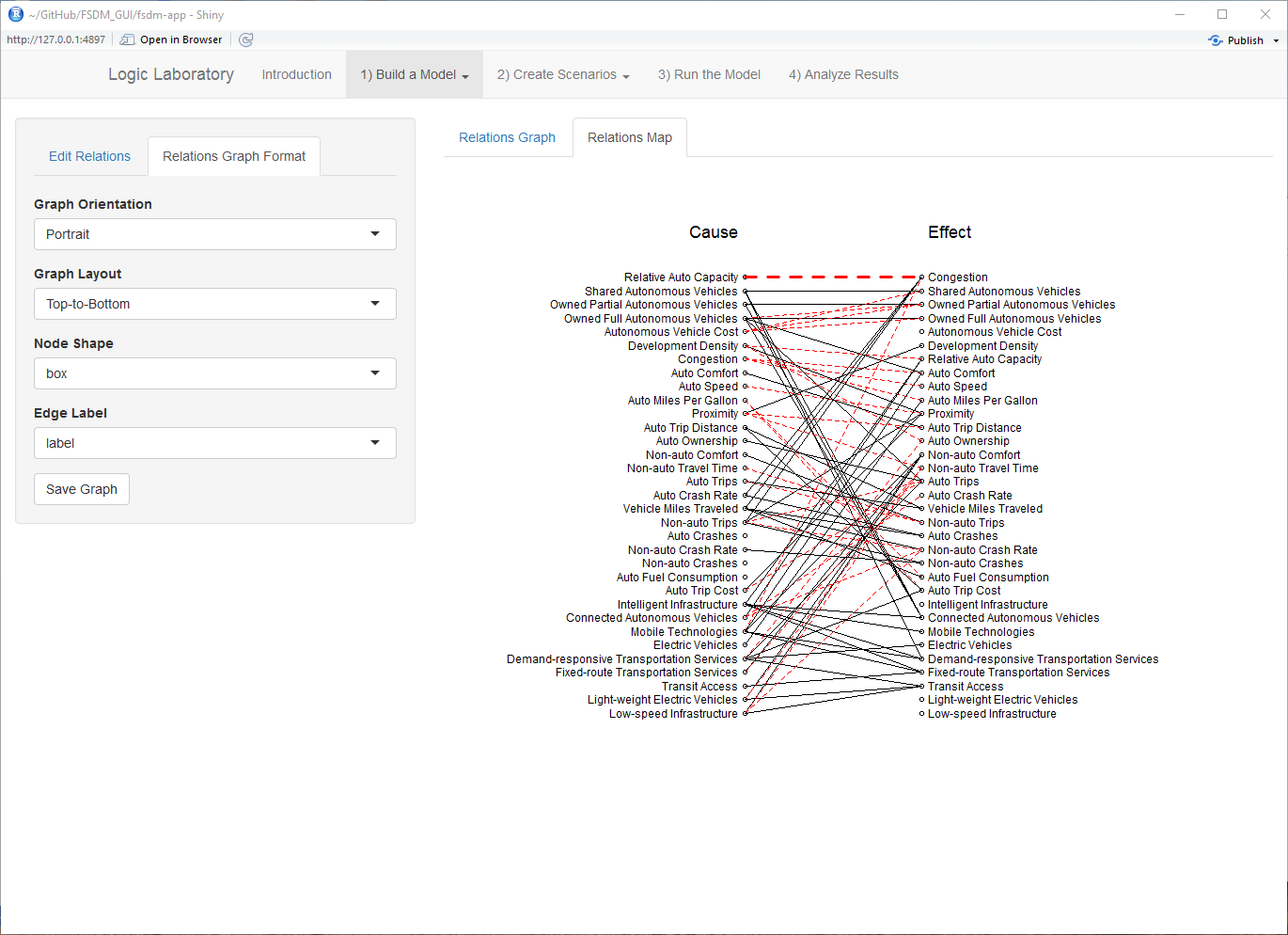
The **Relations Map** displays relations as connections between two lists of concepts with the causal concepts shown on the left-hand side and the affected concepts shown on the right-hand side. Relationships are show as lines connecting causal concepts to affected concepts. The colors of the lines have the same meanings as with the Relations Graph. One advantage of the relations map is that a selected causal concept and the concepts that it affects are shown at the top of the lists and the lines connecting them are highlighted. This makes it easy to see the causal connections.

The **Relations Graph** and **Relations Map** respond to the choices for **Causal Group** and **Affected Group** selections in the **Edit Relations** tab. This can be helpful to simplify the relations graph and relations map to better understand interactions. For example, Figure 10 shows what the relations graph for the Futures2 model looks like when the causal group and affected group are selected to be “Technologies & Services”. Following is a listing of the input fields shown in the **Edit Relations** tab. Almost all of these are drop-down list selectors because the allowable values are predetermined.

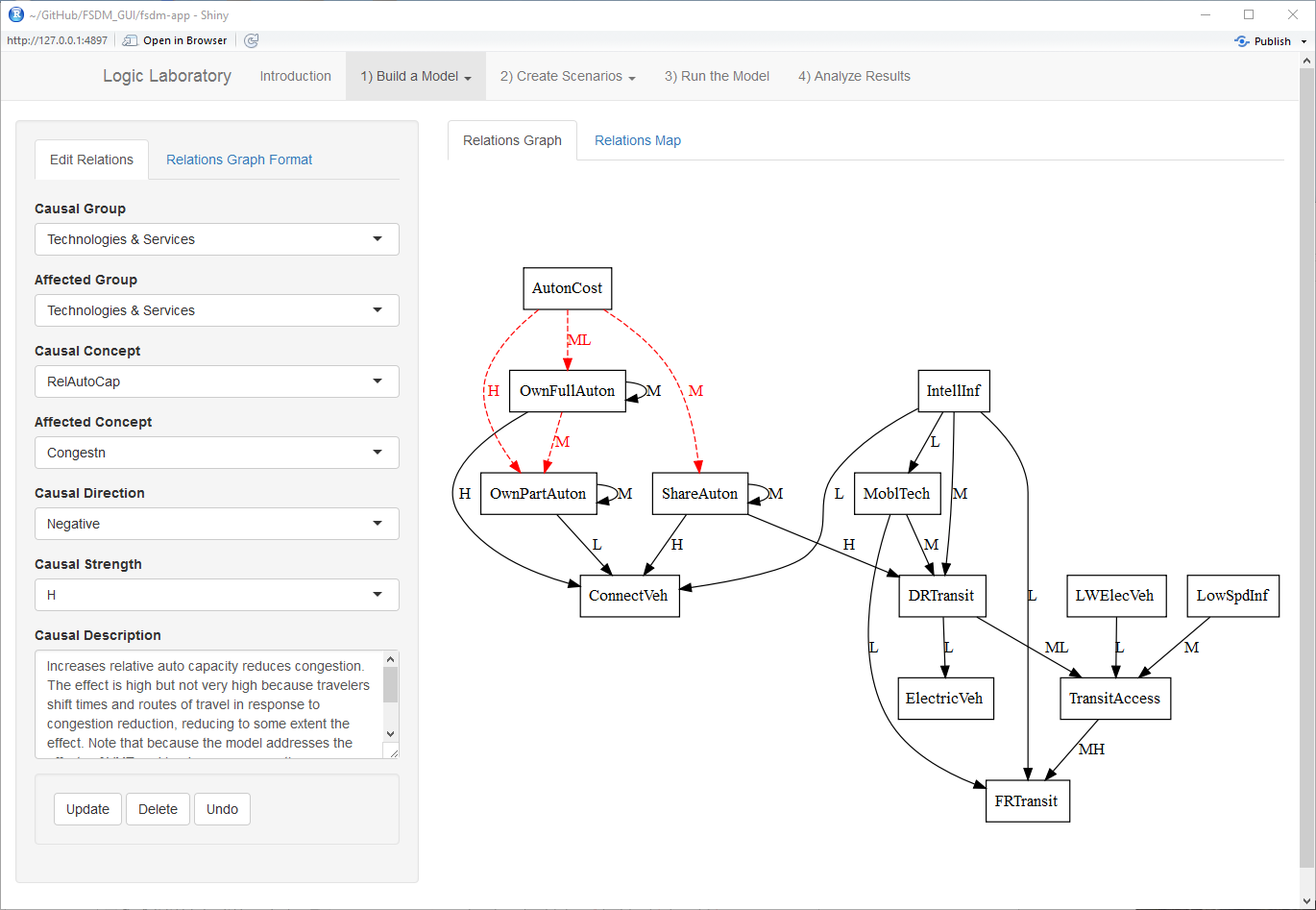
* **Causal Group**: This drop-down list allows you to select a concept group name to use for filtering the causal nodes to show in the relations graph and relations map.
* **Affected Group**: This drop-down list allows you to select a concept group name to use for filtering the affected nodes to show in the relations graph and relations map.
* **Causal Concept**: This drop-down list allows you to select a concept that is a causal concept in a causal relationship.
* **Affected Concept:** This drop-down list allows you to select a concept that is an affected concept in a causal relationship.
* **Causal Direction:** This drop-down list allow you to select whether the causal direction is positive or negative. A positive direction means that causal and affected concepts change in the same directions (increase causes increase and decrease causes decrease). A negative direction means that causal and affected concepts change in opposite directions (increase causes decrease and decrease causes increase).
* **Causal Strength:** This drop-down list allows you to select the strength of the causal relationship using the codes identified above.
* **Causal Direction:** This text box is where you document the causal relationship. The documentation should include an explanation of why the relationship is asserted, why the asserted relationship is positive or negative, and why the magnitude category was chosen.

The buttons at the bottom of the **Edit Relations** tab enable you to register changes in relationships to the model. You will notice that unlike concepts editing, there is no **New** button. To create a new relationship you just select the causal and affected concepts using the drop-down lists and make choices regarding the type of relationship and its strength. Once the relationship values have been filled in, the **Update** button must be clicked in order for the change to be incorporated into the model. Edits to existing relationships are done in the same way; select the causal and affected concepts from the drop-down lists and then edit the other values. The highlighted selected relationship may be deleted by clicking on the **Delete** button. Finally, an edited or deleted relationship can be restored to its previous state by clicking on the **Undo** button. It should be noted that only the last update or deletion can be undone.

**Figure 8. Edit Relations Screen with the Edit Relations and Relations Graph Tabs Selected**



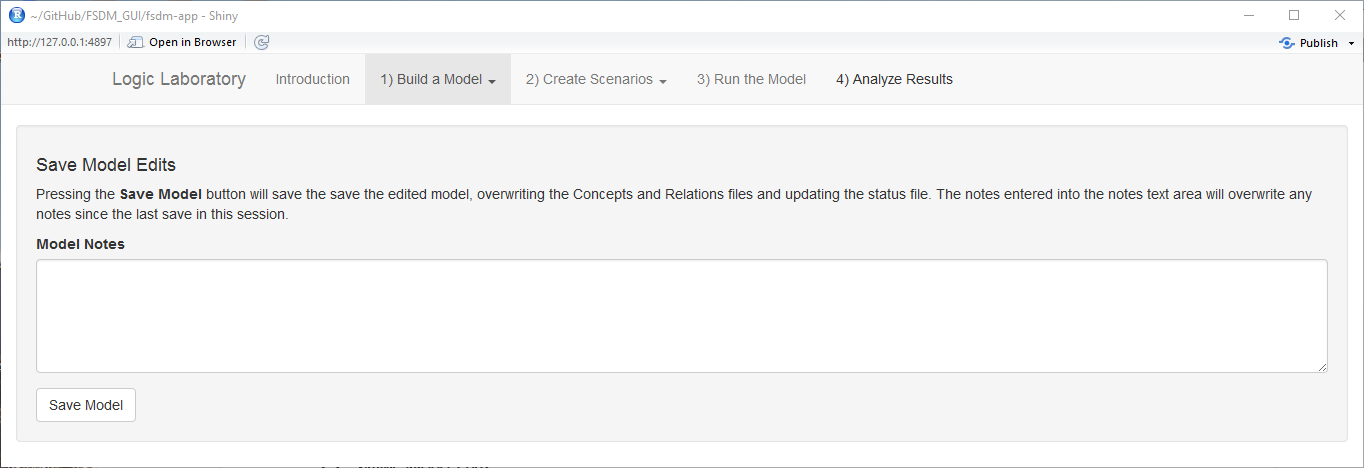
**Figure 9. Edit Relations Screen with the Relations Graph Format and Relations Map Tabs Selected**



**Figure 10. Showing a Portion of the Relations Graph by Selecting Causal Group and Affected Group**

## Saving Model Edits

The **Save Model** screen (Figure 11) is used to save the model edits to files. Clicking on the **Save Model** button overwrites the existing concepts.json, relations.json, and status.json with updated information. The **Model Notes** text input box is there so that you can document the nature of the edits being made. This information is appended to existing notes in the status.json file to keep a running documentation of model edits. It is not required that you make notes but it is good practice to do so.



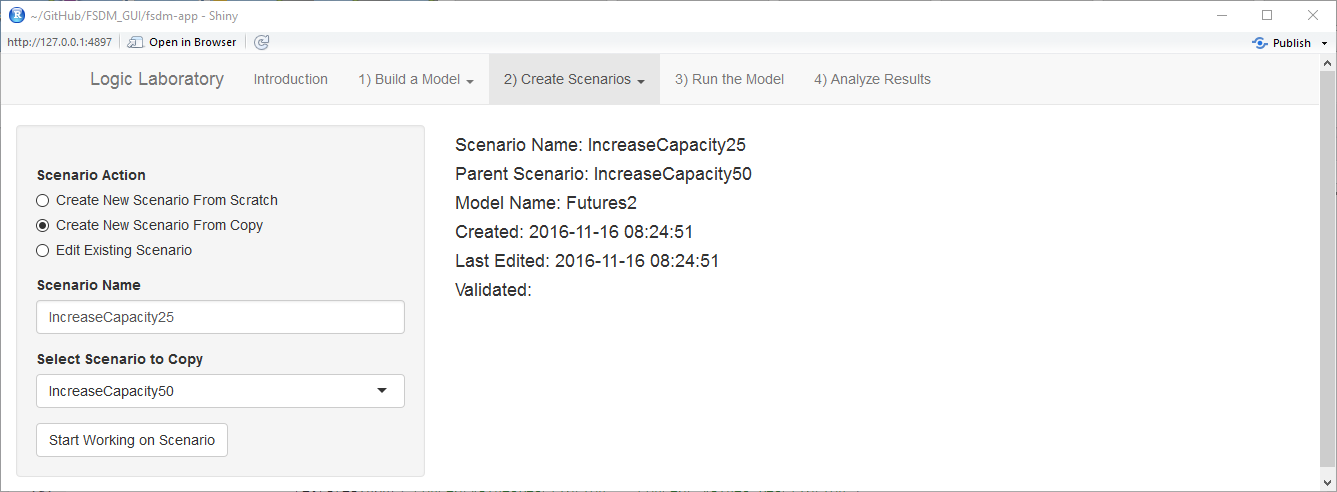
**Figure 11. The Save Model Edits Screen**

# Create Scenarios

The **Create Scenarios** menu has two menu choices, **Select Scenarios** and **Edit Scenario Values**.

## Select Scenarios

Figure 12 shows the **Select Scenarios** screen. **Scenario Action** options include **Create New Scenario From Scratch**, **Create New Scenario From Copy**, and **Edit Existing Scenario**. The figure shows what the screen looks like if the **Create New Scenario From Copy** choice is selected. The user types in the name of the new scenario in the **Scenario Name** text box. The scenario to be copied is chosen in the **Select Scenario to Copy** drop-down list box. Clicking on the **Start Working on Scenario** button initializes or loads the scenario so that it is ready for editing. Status information about the scenario is displayed the right-hand side of the screen. The **Validated** line is blank because the scenario still needs to be validated. Scenario validation is discussed in the last section.



**Figure 12. The Select Scenarios Screen**

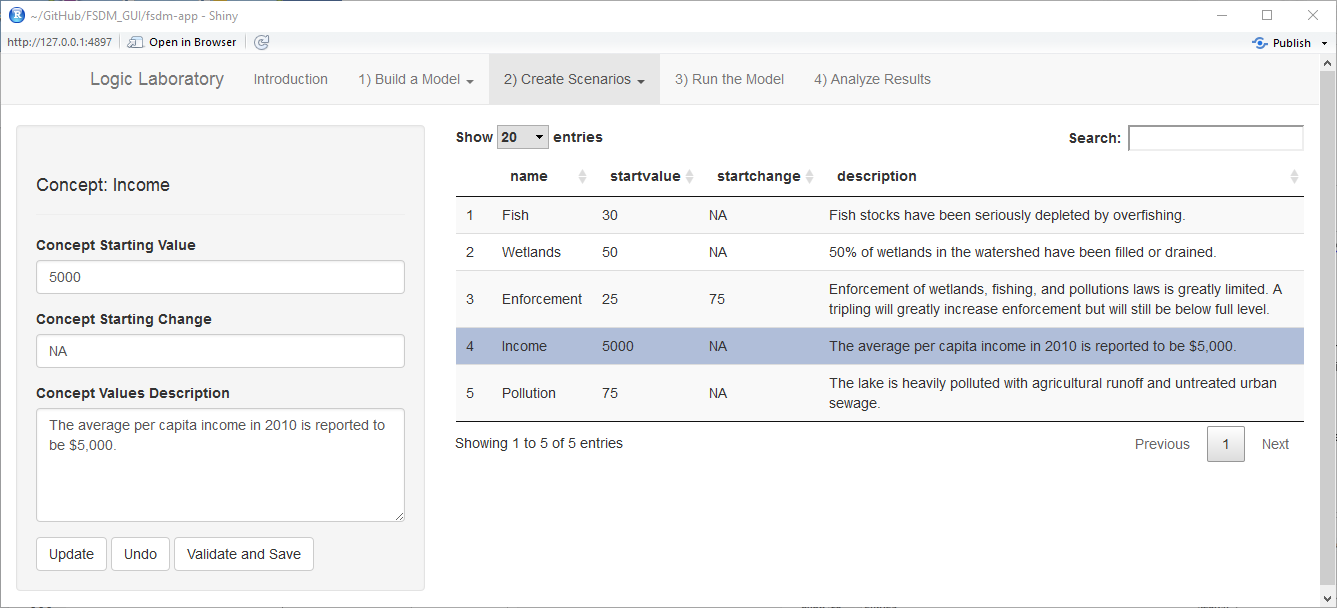
There are some differences in the appearance of the screen if either of the other scenario actions is chosen. If the **Create New Scenario From Scratch** option is chosen, the **Select Scenario to Copy** drop-down list will not be present. If the **Edit Existing Scenario** option is chosen, the **Scenario Name** text box will not be present and the drop-down list will be labeled **Edit Existing Scenario**.

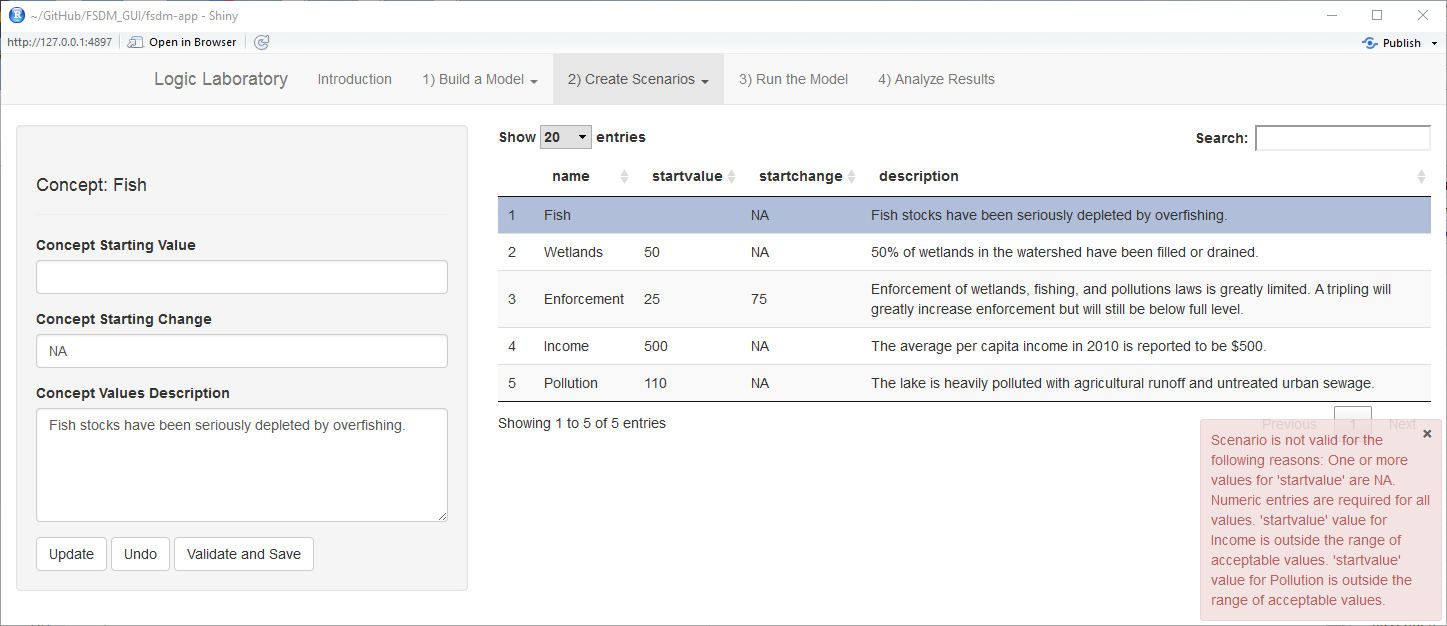
## Edit Scenario Values

Figure 13 shows the **Edit Scenario Values** screen. The table on the right-hand side of the screen shows the scenario values. The selected concept is highlighted in the table. You change which concept is selected by clicking on the row for that concept. The name of the selected concept is shown at the top of the panel on the left-hand side of the screen. The scenario values may be edited in the text boxes below. These are as follows:

* **Concept Starting Value:** This is the value that the concept starts with. The value must be in the same units that were defined for the concept. In addition, the value must be within the range of values defined for the concept. For example, Figure 13 shows the starting value for the “Income” concept to be 5000 is consistent with the defined measurement units (dollars per capita) and the defined range of values (1000 to 15000). Every concept in the model must have a starting value.
* **Concept Starting Change:** This is the value that a concept will be changed to start a model run. For example, Figure 13 shows a scenario where the value of the “Enforcement” concept changes from 25 to 75. When this scenario is run, the model will calculate what the values of all other concepts will be when the “Enforcement” concept reaches its final value. If a concept does not have a numeric starting change value, it must have a value of “NA”. Typically only one or a couple of concepts will have starting change values. That’s because the primary purpose of the Logic Laboratory is not to forecast the future, it is to assist users in reasoning about systems. Specifying a number of starting changes at the same time makes it more difficult to understand how the scenario inputs result in the model outputs.
* **Concept Values Description:** This is where you document the starting value and the starting change for the concept.

At the bottom of the window are several buttons. The **Update** and **Undo** buttons have the same meanings as their uses on other screens. The **Validate and Save** is used to validate the scenario against the model and to save the scenario if it validates. The validation process checks that scenario values are consistent with the value ranges for the model concepts. The user is notified whether the validation succeeded or failed with an alert at the bottom right-hand corner of the screen. The alert will be red if there are validation errors and the errors will be enumerated in the alert. Figure 14 shows an example of what a validation error alert looks like. Note that if the model or scenario has been edited after the last time the scenario was validated, the scenario must be revalidated before it can be run.

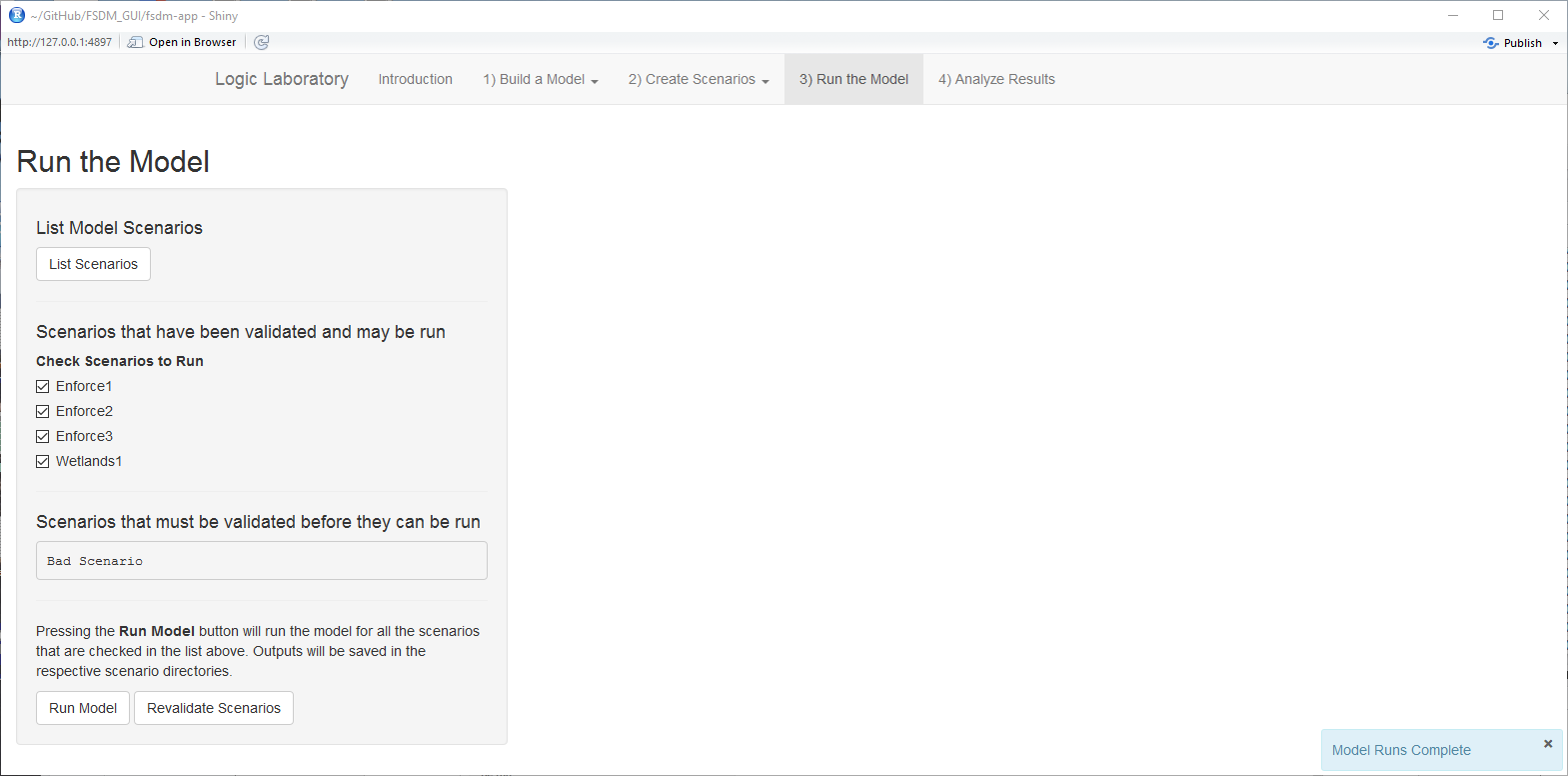


**Figure 13. The Edit Scenario Values Screen**

**Figure 14. Example of Scenario Validation Results**

# Run the Model

Figure 15 shows what the **Run the Model** screen looks like. At the top of the left-hand panel is the **List Scenarios** button. Clicking on this button refreshes the inventory all the model scenarios and determines which have been validated and which have not. This button needs to be clicked whenever new scenarios are created or a different model is selected in order to update the listing. Scenarios that have been validated are shown with checkboxes that allow them to be selected to be run. Scenarios that have not been validated are shown in the text box. The **Revalidate Scenarios** button enables you to revalidate all of the model scenarios. This is not a substitute for validating scenarios when they are created. Rather it provides a shortcut for revalidating scenarios when a change is made to a model that is unlikely to affect scenario validation, such as when a relationship between concepts is changed. Selected scenarios will be run when the **Run Model** button is clicked. An alert pops up in the lower right-hand corner of the screen to show model run progress and announces the completion of the model runs. Figure 15 shows and example.



**Figure 15. The Run the Model Screen**

# Analyze Results

Figure 16 shows the **Analyze Results** screen with the analysis results for two scenarios. The **Analyze Results** screen enables you to view the results of a single scenario or compare the results of two scenarios. The scenarios to display are selected using the **Select Scenario 1** and **Select Scenario 2** drop-down lists. If the same scenario is selected in both drop-down lists, a single graph is presented with the scenario results. If two different scenarios are selected, two side-by-side graphs are presented as shown in Figure 16. If a new scenario is created and run, click on the **Update Scenario Selection Set** button to register the new scenario in the drop-down lists.

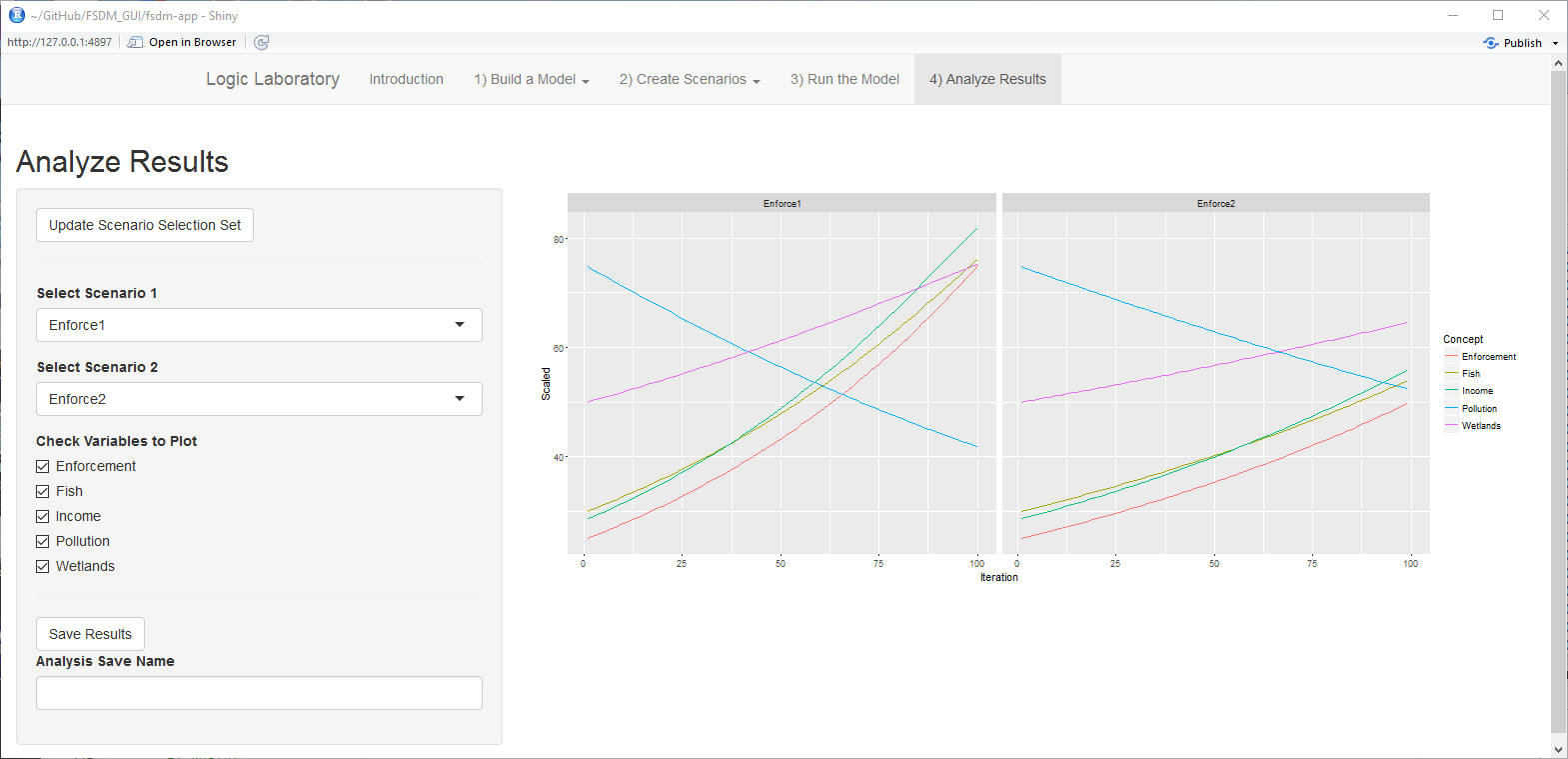
Below the scenario selection drop-down lists is a checkbox listing of all the model concepts. You select which concepts to show in the graphs by clicking on the concept checkboxes. Clicking on an unchecked checkbox selects it and clicking on a checked checkbox unselects it. Graphs will be plotted on the right side of the screen when two or more concepts are selected.

The graphs show the relative values of the concepts. The horizontal axis can be thought of as a relative time axis. As the model runs a scenario, the concept (or concepts) that is specified as having a starting change is incremented in small steps between the starting value and the starting change value. With each increment the model calculates how the value of every other concept changes as a result. Since each iteration can be thought of as occurring over some period of time, the horizontal axis gives an indication of the relative passage of time. The vertical axis indicates the relative value of each concept on a scale of 1 to 100. The values of concepts are shown on the same relative scale so that they can be more easily compared.

The analysis results may be saved by entering the name of a folder to save the results to and clicking on the **Save Results** button. Two files are saved in the folder:

* **data.csv**: a comma-separated values text file which contains the scaled and unscaled values for all of the selected concepts for the selected scenarios (see Appendix A for information on the file structure); and,
* **plot.png:** a portable network graphics image file of the graphs.

You should note that the Logic Laboratory will not protect you from overwriting existing files if you use an **Analysis Save Name** that already exists.



**Figure 16. The Analyze Results Screen**

# References

1. Vogt, R., Wang, H., Gregor, B., & Bettinardi, A. (2015). Potential Changes to Travel Behaviors & Patterns: A Fuzzy Cognitive Map Modeling Approach, Special Issue on “Capturing, measuring and responding to changes that influence travel behavior” in Transportation (Springer), DOI: 10.1007/s11116-015-9657-3.

2. Gregor, B., Wang, H., Bettinardi, A. (2016). Modeling The Potential Consequences of New Transportation Technologies and Services Using a Fuzzy Cognitive Map Based Model, paper presented at Transportation Research Board 6th Innovations in Travel Modeling conference, Denver.

# Appendix A: Model Output File Formats

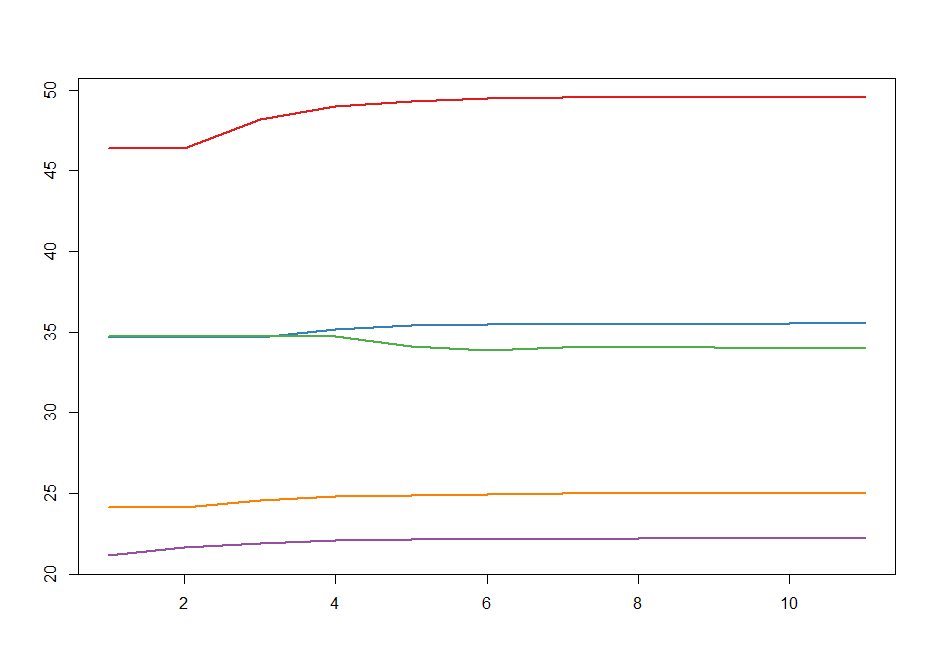
When a scenario is modeled, the results of the model run are saved in a file named **Outputs\_ls.RData** in the scenario folder. Another data file is saved the user chooses to save the results of an analysis. This file is named **data.csv** and is saved in the analysis folder that was created for saving the results. The following sections describe these files in more detail.

## Outputs\_ls.RData File

The **Outputs\_ls.RData** file is an R binary file. As such, it must be loaded into R to view or manipulate its contents. When loaded into R, the file creates an R list object that has the following 3 named components:

* **ScaledSummary:** is a matrix of all the concept values for each iteration on the relative scale of 1 to 100. The rows of the matrix represent iterations and are named with the iteration number. The columns of the matrix represent the concepts.
* **RescaledSummary:** like the ScaledSummary is a matrix of the concept values by iteration, but the values are in the units in which the concepts are defined.
* **ScaledFull:** is a list which contains the results of calculations that occurring during each iteration. Each component of the list is a matrix which contains the interim calculations that are made during an iteration.

The **ScaledFull** list shows the results of all of the calculations that are made as a model is run. There are many more calculations than there are iterations, the consequence of each increment cascade across the network. The calculation of these cascading effects needs to be continued until the concept values stabilize. Figure 17 shows an example of what the intermediate results look like for one iteration.



**Figure 17. Example of Intermediate Model Results for One Model Iteration**

Examination of the **ScaledFull** data can be useful if the model results look problematical in one or more respects. The problems may show up as a lack of convergence in the intermediate results for a concept or the values for a concept abruptly changing to the maximum or minimum value. These problems would indicate problems in how the model is designed.

## data.csv File

The **data.csv** file is a comma-separated values text file which stores the data from an analysis. The file can be easily imported into a spreadsheet for further analysis. Table 1 shows an example of what the file looks like when it is loaded into a spreadsheet. The first column identifies the concept. The second column identifies the model iteration. (Note that the data for most of the iterations have been removed from this example.) The third column shows the scaled value for the concept for the iteration and scenario identified in the row. The fourth column shows the rescaled values (in the defined measurement units). The last column identifies the scenario.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Concept** | **Iteration** | **Scaled** | **Rescaled** | **Scenario** |
| Enforcement | 1 | 25.0 | 25.0 | Enforce1 |
| Enforcement | 2 | 26.5 | 26.5 | Enforce1 |
| Fish | 1 | 30.0 | 30.0 | Enforce1 |
| Fish | 2 | 31.5 | 31.5 | Enforce1 |
| Income | 1 | 28.6 | 5000 | Enforce1 |
| Income | 2 | 29.9 | 5192 | Enforce1 |
| Pollution | 1 | 74.7 | 75.0 | Enforce1 |
| Pollution | 2 | 72.4 | 72.6 | Enforce1 |
| Wetlands | 1 | 50.0 | 50.0 | Enforce1 |
| Wetlands | 2 | 51.1 | 51.1 | Enforce1 |
| Enforcement | 1 | 25.0 | 25.0 | Wetlands1 |
| Enforcement | 2 | 26.5 | 26.5 | Wetlands1 |
| Fish | 1 | 30.0 | 30.0 | Wetlands1 |
| Fish | 2 | 32.2 | 32.2 | Wetlands1 |
| Income | 1 | 28.6 | 5000 | Wetlands1 |
| Income | 2 | 31.0 | 5341 | Wetlands1 |
| Pollution | 1 | 74.7 | 75.0 | Wetlands1 |
| Pollution | 2 | 72.0 | 72.2 | Wetlands1 |
| Wetlands | 1 | 50.0 | 50.0 | Wetlands1 |
| Wetlands | 2 | 52.3 | 52.3 | Wetlands1 |

**Table 1. Example of data.csv File**

# Appendix B: Paper Presented at the Transportation Research Board 6th Innovations in Travel Modeling conference, Denver.

## MODELING THE POTENTIAL CONSEQUENCES OF NEW TRANSPORTATION TECHNOLOGIES AND SERVICES USING A FUZZY COGNITIVE MAP BASED MODEL

**Brian Gregor, P.E.** (corresponding author)

Oregon Systems Analytics LLC

PO Box 2211, Salem, OR 97308

Tel: 503-877-0349 Fax: 503-385-0352; Email: gregor@or-analytics.com

**Haizhong Wang, Ph.D., Assistant Professor**

Oregon State University, School of Civil and Construction Engineering

101 Kearney Hall, Oregon State University, Corvallis, OR 97331

Tel: 541-737- 8538 Fax: 541-737-3052; Email: Haizhong.Wang@oregonstate.edu

**Alex Bettinardi, P. E.**

Oregon Department of Transportation

555 13th Street NE, Suite 2, Salem, OR 97301

Tel: 503-986-4104 Fax: 503-986-4174; Email: Alexander.O.BETTINARDI@odot.state.or.us

Word count: 4726 words text + 9 tables/figures x 250 words (each) = 6976 words

**ABSTRACT**

The next several decades look to be times of great flux for transportation. Our transportation systems are likely to be greatly altered by new transportation technologies and services such as intelligent infrastructure, connected vehicles, automated vehicles, and the expansion of demand-responsive transit services. It is important that transportation planners and decision makers consider the potential implications of these emerging technologies and services when making long-range transportation decisions because they could substantially affect the assumptions and analysis used in making those decisions. Unfortunately, most transportation models are not useful for these purposes because they are designed to analyze the effects of incremental changes to transportation systems, not the effects of disruptive changes. This paper describes a new method for modeling the potential consequences of new technologies and services using a variant of the fuzzy cognitive map (FCM) approach, which enables problems involving imprecise and uncertain information to be modeled. Significant modifications to the standard FCM approach were made to address deficiencies found in applying the standard approach. The new approach, named the Fuzzy Systems Dynamics Model (FSDM) retains some basic FCM characteristics, but it deviates substantially in a number of ways as well. It has been found that this produces models having well-behaved dynamics, that can be explained in common-sense terms, be easily configured, run many scenarios quickly, and used to analyze scenarios of disruptive change.

## OBJECTIVES AND MOTIVATIONS

Transportation modelers have developed very sophisticated and complex models of the relationships of different aspects of travel demand to the characteristics of the populations being modeled, the places where they live, and the transportation facilities and services in those places. Although these models are outstanding technical achievements, their complexity substantially limits the domain of transportation policy and planning questions to which they may be applied. The models are built primarily to evaluate the effects of specific transportation system changes and/or land developments, and are very difficult to adapt for broader purposes because of their complexity. Moreover, their complexity and data intensiveness makes them impractical to use for analyzing policy questions having any amount of ambiguity because long model run times and requirements for extensive and detailed inputs severely limit the number of scenarios that may be modeled. Consequently, when these models are used, ambiguity tends to be assumed away and considerations that aren’t addressed are set aside.

Until recently, assuming away ambiguity has been fairly easy to do. Although the urban landscape and transportation system have changed dramatically in past decades, the type, direction and pace of change has been fairly predictable. Urban areas have been spreading outward and single-family development, auto ownership, and vehicle-miles traveled (VMT) have all been increasing. Until recently, it looked like these trends would continue into the future and, although some questioned the wisdom of continuing to accommodate them, there were no apparent limits to their continuation or reasons why they would not continue.

Now however, a number of major changes are occurring that make it unwise to assume away the ambiguity that now surrounds most transportation decisions. VMT growth has stagnated in recent years and VMT per capita has been declining. Although the “Great Recession” undoubtedly contributed to stagnating VMT growth, the longer-term trend in declining VMT per capita suggests that other forces are at work as well such as changing perceptions about the benefits and liabilities of ever increasing automobility. Equally important in disrupting how people think about the future is the awareness of environmental constraints, most notably a constraint on the amount of fossil carbon that can be released into the atmosphere without jeopardizing the future living conditions for hundreds of millions of people. Finally, major changes to transportation technologies and services are occurring that may substantially change travel trends.

The next several decades look to be times of great flux for the transportation system. The combination of intelligent infrastructure, connected vehicles, and autonomous vehicles could substantially reduce accident rates, congestion and driving stress. It might also increase average driving speeds, the distances that people drive, and per capita VMT. Autonomous vehicles might also substantially reduce the operating cost of demand-responsive and fixed-route transit services and might help solve the “last mile” problem, enabling many households to reduce or eliminate car ownership. On the other hand, this might enable more people to travel in single-occupant vehicles who wouldn’t otherwise be able to own or operate one, and might compete with fixed-route transit services for riders.

Transportation decision-makers will be pushed by the public to consider the implications of these potential changes regardless of whether computer models are available to assist in doing so. Lacking computer models to assist them, the decision-makers will rely on their own mental models, political consensus, and perhaps other analysis and expert judgement, to guide them. This would be unfortunate because although computer models are not the be-all and end-all of transportation analysis, they do help maintain logical consistency in the analysis of complex systems. This is arguably the most important role of models, and it is a role that can be carried out even when dealing with decision domains that involve considerable amounts of ambiguity. However it will require different sorts of models than the complex travel demand models that have been developed primarily to analyze transportation projects and other specific transportation system and land use changes.

The objective of this paper is to describe one such method for modeling transportation issues that involve imprecise and uncertain information. The method is a variant of the fuzzy cognitive map (FCM) approach which comes from the field of “soft computing”. Substantial alterations have been made to the standard FCM approach to address deficiencies in using the standard approach for public policy questions. The new approach is named a fuzzy systems dynamics model (FSDM) for reasons that are explained below. A FSDM has been developed to model metropolitan area transportation systems and the potential consequences of new transportation technologies and services. The model and how it works is described and several applications are demonstrated. These results come from research that has been partially funded by the Oregon Department of Transportation (ODOT) in collaboration with Oregon State University (OSU) and Oregon System Analytics (OSA) to develop a fuzzy cognitive map model that can be used to evaluate the potential effects of new transportation technologies and services on transportation systems to assist in prioritizing future research efforts. Earlier results from that research have been published (*1*). The methods and results presented below include significant changes from the earlier published research.

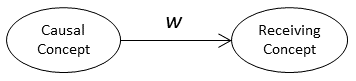
## INTRODUCTION TO FUZZY COGNITIVE MAPS AND THE FUZZY SYSTEMS DYNAMICS MODEL

The approach used in this research is derived from fuzzy cognitive mapping, a modeling approach developed in the field of soft computing (*2*). Fuzzy cognitive maps are models that are structured as directed graphs (digraphs) where each node represents a concept and each edge represents a causal linkage (3). This use of digraphs to express causal relationships is shared by a number of modeling methods such as path modeling (*4*), Bayesian networks (*5-6*), and structural equation modeling (SEM) (*7*).

The inspiration for fuzzy cognitive maps (FCM) came from the cognitive mapping technique used by Axelrod to evaluate systems thinking in political domains (*8*). However, whereas Axelrod’s cognitive maps represented causal effects in simple bipolar terms (+1 vs. -1), FCMs represent causal effects in degrees (e.g. a little, some, a lot). Some FCMs do this using a fuzzy rule-based approach but most follow the approach developed initially by Kosko for representing causal effects as continuous numeric values (*3*).

**Concepts of Fuzzy Cognitive Maps**

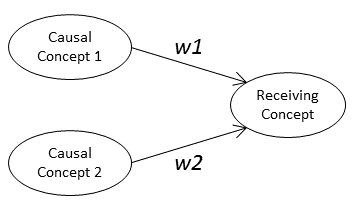
FCMs show causal relationships using digraphs. The effect of a causal concept on a receiving concept is represented by an arrow, and the magnitude of effect by a weight (w) as illustrated in Figure 1. Weights are typically in the range of -1 to 1 where positive values are meant to represent causal increases while negative values are meant as causal decreases. The absolute value of the weight indicates the relative strength of the effect where 0 means no effect and 1 means a very strong effect.



**Figure 1. Simple Two-Node Fuzzy Cognitive Map**

In the standard FCM approach, the value of the receiving concept after a change in a causal concept is equal to the changed causal value multiplied by the weight. If *c* and *c’* are the initial and subsequent values of the causal concept and likewise *r* and *r’* are the initial and subsequent values of the receiving concept, then in the standard FCM the value of *r’* is calculated as follows:

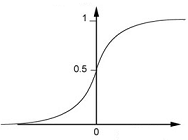
A receiving concept may be affected by more than one causal concept as shown in Figure 2.



**Figure 2. Fuzzy Cognitive Map with Two Causal Nodes**

The value of the receiving concept *r’* is the sum of the effects of the causal concepts.

Concept values in standard FCMs are constrained to the range of 0 to 1 and may be thought of as the proportion of some actual base values. Since the value of *r’* as computed above may exceed 1, the standard FCM uses an “activation function” to scale the result to that range. A variety of activation functions may be used, but most are “S” shaped sigmoid functions like the logistic function shown in Figure 3.



**Figure 3. Logistic Activation Function**

The standard FCM formulation is so similar to the formulation of neural networks that FCMs are often referred to as neural networks with feedback.

Because a FCM can include cycles (e.g. Figure 5) it is necessary to iterate the calculations until the concept values achieve a new stable state or until it is evident that a stable state can’t be achieved.

**New FCM Formulation**

While the approach used by fuzzy cognitive maps of showing causal relationships in directed graphs is useful, the standard mathematical formulation has some limitations when used to model social systems. Most importantly, the mathematics of standard FCMs are not consistent with the meanings most commonly attributed to causal weights. The common view of causality is that a change in the value of a causal concept results in a change in the value of a receiving concept; for example an increase in development density results an increase in fixed-route transit service. This is how weights in FCMs are commonly described. But the FCM mathematics do not represent relationships between changes in concept values. Instead, they represent relationships between the values. For example, in the simple FCM above, the value of the receiving concept is the value of the causal concept times the weight.

This disconnection between meaning and mathematics causes other problems. Negative weights become a significant problem because rather than meaning that an increase in a concept causes a decrease in the concept it affects, they mean that if the value of the causal concept is positive, the value of the affected concept will be negative. This of course is nonsensical since concepts can’t have negative values. This problem may be eliminated in a causal map by changing the meaning of a concept to enable inverting the directionality of the concept (*3*). For example, instead of showing congestion as having a negative relationship to fuel economy (miles per gallon), it could be shown as a positive relationship to fuel diseconomy (gallons per mile). However, changing all negative value to positive values in a complex system model may be challenging.

A new FCM formulation has been developed to overcome this problem. The mathematics of this new FCM have been revised to be consistent with the definition of weights as causal relationships between concepts. Weights in the new FCM are defined as the ratio of the proportional changes of the receiving and causal concepts as follows.

Weights as defined above are similar to elasticities. The combined effect of multiple causal concepts, rather than being the sum of the individual effects, is the product of the individual proportional effects. Thus, if we call the proportional change in a causal concept Δ*c*, the value of the proportional change in the receiving concept Δ*r* is calculated as follows:

A second problem with standard FCM formulation for social system modeling is that it is difficult to attribute a real world meaning to the activation function which is used to constrain concept values to the range of 0 to 1. While this approach may have practical values in making a FCM work, it has no apparent theoretical justifications.

The alternative FCM formulation replaces the activation function with causal and receiving sensitivity functions which modify the effects of a causal weight based on the respective concept values. If *sc* is the causal sensitivity given the value of the causal node and *sr* is the receiving sensitivity given the value of the receiving node, then the change in the receiving node, Δ*r*, due to a change in the causal node, Δ*c*, is calculated as follows:

The forms of the causal and receiving sensitivity functions are shown in Figure 4. It can be seen that these functions are mirror images of one another. The causal sensitivity is 0 when the value of the causal concept is 0 and rises to 1 when the concept is at 50 percent of its maximum value and stays at that level as the concept value increases further. The sensitivity of the receiving node to change is 1 when the value of the concept is less than 50 percent and decreases thereafter, falling to 0 when the receiving concept reaches 100 percent of its maximum. If the starting concept values in a simple two-node system (Figure 1) are small, the combined sensitivity of the system follows an inverted “U” shape as the causal concept is increased to 100 percent of its maximum value.

Sensitivity

**Figure 4. Causal and Receiving Sensitivities and Their Joint Effect**

Although these asserted sensitivity functions have not been demonstrated to be correct through empirical evidence or theoretical proof, the forms do make sense. It is expected that the effect of a change in a causal concept on a receiving concept will be small when the causal concept is small and will grow as the causal concept increases. For example, if the number of bicycling trips is very small (say 1% of all trips), increasing their number by 10% would have little effect on the number of automobile trips. However, if the number of bicycling trips is large (say 50% of all trips), then increasing their number by 10% would have a substantial effect on the number of automobile trips. In the case of a receiving concept, it makes sense that it would be most sensitive to change when its value is small and it would become less sensitive at high values. For example, it was easier to increase the market share of cell phones by 50% when they were 10% of the market, than when they had 60% market share.

A final difference between the new method and the standard FCM approach is in how the value of causal concept that is assumed to change in a scenario is changed from its initial value to its final value. In the case of the standard FCM, the causal concept is set to its final value and the model iterates (assuming it has cycles) until a new stable state is achieved (assuming that there is one). In the case of the new method, a causal concept is changed gradually in small increments and the model is iterated to closure with each increment. This is consistent with the nature of weights as elasticities.

The new method has been given the working name of Fuzzy Systems Dynamics Model (FSDM) because like system dynamics models it calculates the change in concepts as the accumulation of small differences as the model iterates. However, unlike system dynamics models the difference calculations use simple elasticities and sensitivity functions, rather than difference equations, and time is not explicit in the FSDM model.

**Comparison of FSDM and Other Methods**

Although there are superficial similarities between FSDM and other modeling methods that used directed graphs, the approaches are very different. Contrasting the FSDM with the method that transportation modelers are most likely to have some familiarity, structural equation modeling (SEM), may be useful to help readers better understand the purpose and methods of the FSDM. The similarity between the FSDM and SEM starts and ends with the use of a directed graph that is asserted by the model developer to represent the structure of a model. However SEM is a multivariate linear regression method which accounts for direct and indirect effects of variables on one another. Simultaneous linear equations are used to estimate the coefficients of the model. SEM models are often used to incorporate latent (unobserved) variables into a model. The primary purpose of SEM is to determine the relative strength of relationships between variables given the asserted model structure. The estimated coefficients, relate the values of variables.

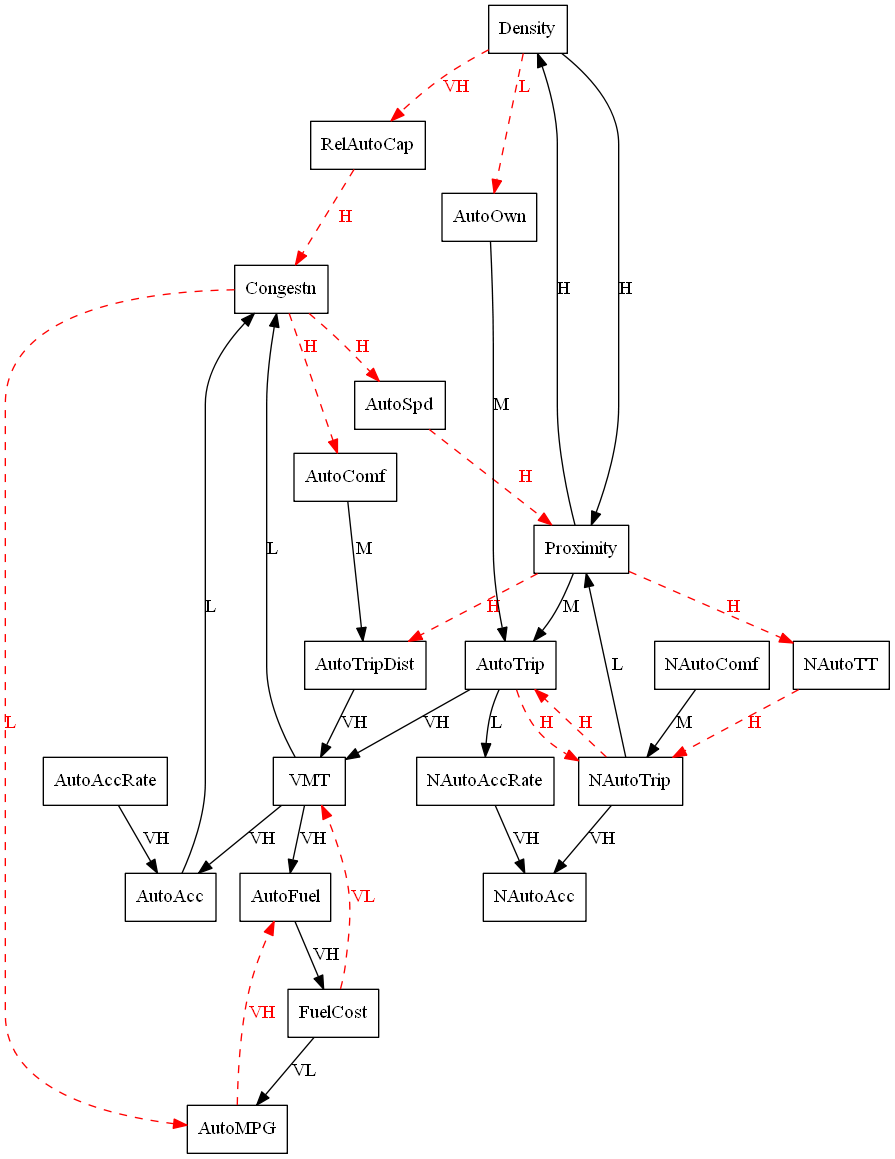
The FSDM method is not a statistical modeling method. Its purpose is to develop logical models to help analyze problem domains where there is significant ambiguity and it is not possible to estimate statistical relationships. The specifications of a FSDM model and the relationships between the concepts being modeled are developed through expert judgement and group processes. This is a method often used in the development of fuzzy cognitive maps and fuzzy logic systems. In the case of the model presented below, the development of the specifications relied on the judgements of the research team supported by a technical advisory committee of transportation analysts and planners from ODOT and Oregon MPOs, and the FHWA.

The FSDM method also differs from SEM with respect to the meaning of the relationships between concepts (variables). In SEM models, as with standard FCMs, the parameters establish the relationship between the values of the connected variables similar to how parameters in multivariate regression models relate the value of the independent variable with those of the dependent variables. In FSDM models, the edge weights are fuzzy elasticities that describe how the change in the value of a causal concept affects the change in the value of the receiving concept. The effects of edge weights are modified by causal and receiving sensitivity functions as described above.

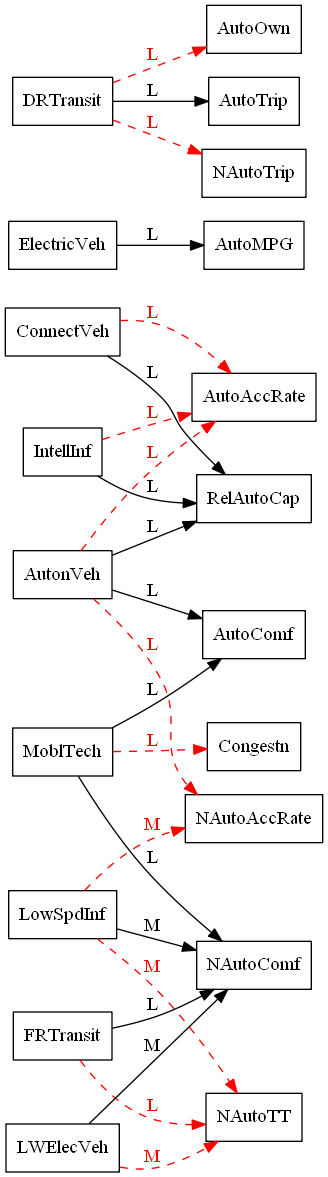
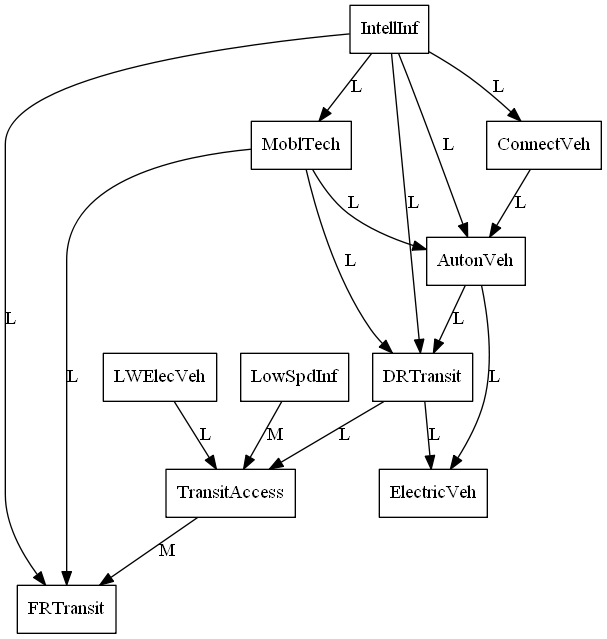
**A FSDM OF METROPOLITAN AREA TRANSPORTATION SYSTEM INTERACTIONS WITH NEW TRANSPORTATION TECHNOLOGIES AND SERVICES**

A cognitive map of the effects of new transportation technologies and services has been built in two components with connections between them. Approaching the problem as two separate components enabled the team to simplify the conceptualization, design and testing of the overall cognitive map. One of the components depicts the relationships between travel behavior and outcomes. This is shown in Figure 5. The concept of behavior as depicted in this graph is broad, representing system behavior. In addition, this cognitive map includes development density and relative roadway capacity because of the importance of those concepts to the understanding of system behavior relationships. All concepts are specified in per capita or proportional terms (e.g. VMT per capita) so that population growth is factored out of the model. In addition, this model assumes no change in real income. The solid black arrows in the graph show positive causal relationships and the red dashed arrows show negative causal relationships. The labels for the arrows indicate the strength of the relationships: VL = very low, L = low, ML = moderately low, M = moderate, MH = moderately high, H = high, VH = very high. The other component is a cognitive map of transportation technologies and services. This portion of the cognitive map and the connections of it to the travel behaviors and outcomes portion are shown in Figure 6.

The meanings of the concepts shown in these figures are described in Table 1.



**Figure 5. Travel Behavior and Outcomes Sub-model**



**Figure 6. Transportation Technologies and Services Sub-model and Connections with Travel Behaviors and Outcomes Sub-model**

**Table 1. Description of Model Concepts**



The concepts in the cognitive map were operationalized and data were gathered for as many of the measures as could be found from readily available sources for 93 metropolitan areas in the US. Data sources included the Texas Transportation Institute’s Urban Mobility Study, the US Census Bureau, the FHWA’s Highway Statistics reports, the 2009 National Household Travel Survey (NHTS), and a report by the American Automobile Association on metropolitan traffic fatalities and injuries. These data were used to establish value ranges for the operationalized concepts, values to use for a test area to check model sensitivity, and values to use to compare model results to.

## RESULTS

The FSDM representation of the effects of new transportation technologies and services has been tested with several scenarios for a hypothetical metropolitan area which has characteristics similar to the Portland-Vancouver metropolitan area for the operationalized concepts. The scenarios included:

1. Doubling the population-weighted population density of the metropolitan area;
2. Halving the population-weighted population density of the metropolitan area;
3. Substantially increasing the relative road capacity of the metropolitan area (to be equivalent to that of Houston, TX);
4. Growing autonomous vehicles to be 50% of the vehicle fleet; and,
5. Increasing intelligent infrastructure to cover 80% of the roadway system.

The model appears to respond realistically to the density and relative roadway capacity tests. Figure 7 shows the results of the doubling-density test. It compares the effects of doubling density on congestion (percentage of VMT occurring in congestion), vehicle travel (daily vehicle miles of travel per person), average speed (ratio of congested to uncongested speed), and auto ownership (percentage of households having one or fewer vehicles). Along with the trajectory for the test metropolitan area, the graphs show the values collected for 47 metropolitan areas that are identified in the 2009 NHTS data sets. (Data points for the New York-Newark NY-NJ-CT are not shown because they are outliers relative to the data points for other metropolitan areas.) The directions of change for the test metropolitan area are consistent with expectations and the relative magnitudes of changes are consistent with the observed metropolitan area values. This was also found to be the case for the halving density and increasing relative capacity scenarios.

The results for all of the scenarios are shown in Table 2. The table shows the starting values for the concepts for the test metropolitan area and the percentage changes in the values of the concepts for the five scenarios. The directions and relative magnitudes of the changes all appear to be reasonable. It is very important to note, however, that there are many uncertainties about the magnitudes of the causal effects of new transportation technologies and services. A key purpose of this modeling techniques is that it allows analysts to evaluate the consequences of making different assumptions about these effects. Some assumptions are likely to have greater effect on the outcomes than other assumptions. This will help in the assessment of where research, refinement and better understanding is most needed.

Double_Density_Comparison

**Figure 7. Comparison of Results of Doubling Density Test with the Values of Selected Concepts for 47 Metropolitan Areas Identified in the 2009 NHTS**

**Table 2. Concept Starting Values and Percentage Changes in Values with Test Scenarios**



## IMPLICATIONS FOR THE PRACTICE OF TRAVEL MODELING

Although more work needs to be done, the results so far demonstrate that the FSDM approach will be useful for analyzing transportation technology and service change scenarios. The travel behavior and outcomes portion of the model can produce models of metropolitan area travel behavior and outcomes that respond sensibly to changes in population density and roadway capacity. Although the transportation technology and services portion of the model, and its connections to the travel behaviors and outcomes needs more work, the preliminary results indicate that the model can be completed and put into practice. Once completed, the model will enable transportation agencies to test out various scenarios of technological and service changes to analyze the potential consequences. Tests can be made of the sensitivity of model results to changes in model parameters. Specifically, the Transportation Planning Analysis Unit of the Oregon Department of Transportation hopes to use the model being refined in this work to better evaluate and prioritize what aspects of future transportation technology and behavior changes have the greatest potential to impact the performance measures, information, and results used in projects and by decision makers. Those future conditions that have the greatest potential to impact project outcomes and decisions would then get further research attention and priority in developing methods to address those conditions in Oregon’s transportation modeling and analysis work.

At a more general level, the FSDM approach is useful for modeling the logical consequences of policies and scenarios in problem domains where information is imprecise and uncertain. The approach produces models that are open to examination and can be explained in a straightforward manner using causal diagrams. Moreover, the strengths of causal relationships are expressed in simple terms. Granted, Figures 5 and 6 are complicated, but the complexity reflects the reality of the transportation system. However, anyone who is willing to spend the time to work through the diagrams will be able to understand the key relationships which drive system behavior and outcomes. The use of this modeling technique will enable transportation modelers and others to develop logical modelers to assist decision-makers when making decisions regarding public policy matters that involve significant amounts of ambiguity.

The mathematics of the model is simple and explainable, and produces sensible model behavior. In addition, this simplicity enables the model to run very quickly. It is possible to run the model on a real time basis and to run thousands of scenarios in a short amount of time. This is very useful for strategic planning applications where the purpose of modeling and analysis is to aid reasoning about decisions that affect and will be affected by future conditions, and not to forecast what the future will be.

Finally, all of the model codes to implement FSDM models and data supporting this research are available under an open-source license. They can be examined, critiqued, and improved upon by all who are interested. They are available at https://github.com/gregorbj/FSDM. This project is actively under development and the repository will be updated with additions. Future extensions will include a graphical user interface that will make it easier for users to create and run FSDM models.

**ACKNOWLEDGEMENTS**

The work described in this paper was partially supported through funding from the Oregon Department of Transportation and the Federal Highway Administration, U.S. Department of Transportation.

## REFERENCES

1. Vogt, R., Wang, H., Gregor, B., & Bettinardi, A. (2015). Potential Changes to Travel Behaviors & Patterns: A Fuzzy Cognitive Map Modeling Approach, Special Issue on “Capturing, measuring and responding to changes that influence travel behavior” in Transportation (Springer), DOI: 10.1007/s11116-015-9657-3.

2. León, M., Rodriguez, C.,  García, M., Bello, R., Vanhoof, K. (2010) Fuzzy Cognitive Maps for Modeling Complex Systems, Chapter [Advances in Artificial Intelligence](http://link.springer.com/book/10.1007/978-3-642-16761-4) Volume 6437 of the series [Lecture Notes in Computer Science](http://link.springer.com/bookseries/558) pp 166-174.

3. Kosko, B. (1986) Fuzzy Cognitive Maps, Int. J. Man-Machine Studies, 24, 65-75.

4. Asher, H. B. (1983). Causal modeling (No. 3). Sage. pp. 36-37.

5. Sedki, K., Louis Bonneau de Beaufort. (2012) Cognitive Maps and Bayesian Networks for Knowledge Representation and Reasoning. 24th International Conference on Tools with Artificial Intelligence, 2012, Greece. pp.1035-1040.

6. Douali, N., Csaba, H., Roo, J., Papageorgiouc, E., Jaulent, M-C. (2014) Diagnosis Support System based on clinical guidelines: comparison between Case-Based Fuzzy Cognitive Maps and Bayesian Networks. computer methods and programs in biomedicine 113, pp. 133–143.

7. Huang, S-C., Lo, S-L., Lin, Y-C. (2013) Application of a fuzzy cognitive map based on a structural equation model for the identification of limitations to the development of wind power, Energy Policy 63 (2013) 851–861.

8. Axelrod, R. (1976) Structure of Decision: The Cognitive Maps of Political Elites, Princeton University Press, Princeton.

1. https://en.wikipedia.org/wiki/Directed\_graph [↑](#footnote-ref-1)