General Help

Robot Control

Overview

The purpose of the Robot Control applet is to simulate a simple autonomous agent in a world, and to modify the agent's controllers to produce novel behaviour. To assist the user in this task, the controller code is represented as CILog statements, which is structured as defined in the book "Computational Intelligence: A Logical Approach" by Poole, Mackworth, and Goebel. The environment, high layer, and middle layer controllers are implemented in this way, and can be modified by changing their Prolog code.

Menu Help

The File Menu

The File Menu has options to create graphs and load files, as well as quitting the program. The application version of the applet can also save files.

- Create New Graph - clears the currently loaded map. All changes will be lost.
- Load Sample Graph - allows the user to load from a selection of pregenerated examples.
- Load From File - [Available only if running as an application] Load a graph from the local disk.
- Load From URL - Load a graph by specifying a url where the graph is located.
- Save- [Available only if running as an application] Saves a the XML representation of the current graph to the local disk, which can then be reloaded later. The file created includes the world representation, the robot properties, the current plan, and the High, Middle and Environment control code.
- Print - Prints the canvas as displayed in Solve mode.
- Quit - Kills the applet.

The Edit Menu

The Edit menu allows the user to view a text representation of the robot world, and to inspect and modify the Prolog code of the controllers and environment.

- View/Edit Prolog Code for the High Layer Controller - Displays the Prolog code used for the High Layer Controller, and allows the user to modify the code.
- View/Edit Prolog Code for the Middle Layer Controller - Displays the Prolog code used for the Middle Layer Controller, and allows the user to modify the code.
- View/Edit Prolog Code for the Environment - Displays the Prolog code used for the Environment, and allows the user to modify the code.
- View/Edit Prolog Code for the Built-in Functions - Displays the Prolog code for some built-in functions, and allows the user to modify values that modify the functions' behaviour.
- View/Edit World Representation - Displays a text representation of the currently loaded robot world. The user can modify this representation manually.
- View/Edit XML Representation - Displays an XML representation of the currently loaded

robot world and controllers. The user can modify this representation manually. The DTD is defined here.

The View Menu

The Options Menu allows the user to modify the appearance of the applet, and to change the properties of the robot and the location plans.

- Font Size- changes the font size of the canvas display.
- Line Width- changes the line thickness of the canvas display.
- Autoscale- Adjust the graph to be fitted in canvas.
- Pan - moves the graph around the canvas manually. This is particularly useful when the graph has exceeded the bounds of the scrollbars. This allows the user to drag the graph into view. To pan the canvas, select the Pan checkbox and click and hold the right mouse button. Now drag the mouse in the direction you want the graph to move. A bounding box will appear to show the user the relative position of the drawn graph.
- Zoom - increases the view area of the canvas. To zoom, select the zoom checkbox (this is on by default) and click and hold the right mouse button. Now drag the mouse up to zoom in, or drag the mouse down to zoom out. A bounding box will appear to show the user the relative size of the drawn graph.
- Show Message Panel- Show or hide the message prompts above the main canvas.
- Show Button Text- Show or hide the text on the buttons in the toolbar.
- Show Buttons- Show or hide the removeable toolbar buttons.
- Show Log Panel - Show or hide the controller log panel below the main canvas. Default is ON.

Help Menu

- Legend for Nodes/Edges - Opens a dialog with a legend for describing what the graph shapes/colours mean.

- Help - Opens this web page, which provides general help on the Decision Tree Learning applet.

- Tutorials - Opens up the Tutorial web page. Tutorials walk through how to use the applet.

- About CIspace - Provides brief information about the CIspace project.

- About this Applet - Identifies the applet version and names of developers.

The Create Toolbar

The 'Create Toolbar' provides ways for the user to create graphs(maps) manually.

- Create Location - creates a location. Select this button and then click on the canvas to create a location. A dialog will pop up that will appear where the user can change the name of the location to be created. There are also fields to define the position of the location on the canvas, but it is advised that the user should allow the applet to set this automatically.

  *Note:*The plan is automatically created as the locations are placed. The plan is the same as the order that locations are placed.

- Create Wall - creates a wall. Select this button and then click on the canvas to define the start point of the wall. Clicking on the canvas again defines the end point of the wall.

- Select Entity - With this button selected you can select and move locations and walls by clicking and dragging entities.

- Delete Entity - deletes an entity. Select this button and then click on the entity that you want to delete from the map.

- View/Modify Plan - pops up a dialog that allows the user to create a list of locations that will be visited by the robot in order. To add a location to the plan, select the location from the 'available locations' list and click the "Add --->" button. To remove a location from the plan, select the location from the "Current Plan" list and click the "<--- Remove" button. To update the plan, click the "Update" button. To return to the applet without changing the plan, click "Cancel".

  Note: The plan cannot be altered while the robot is running, or if the robot has been run but has not been reset.

- View/Modify Robot Properties - pops up a dialog that allows the user to modify the robot's properties.

  - Maximum Time Step to Run - the number of time steps that is executed when the "Run Robot" button is pressed.
  - dt - the time interval for each step. This should be an integer number.
  - Number of Pixels per Unit - defines the number of pixels that the robot takes per step. This also has the effect of increasing the size of the whisker sensor.<
  - Show Trace - determines whether or not the robot leaves a trace as it moves.

  Note: The robot properties cannot be altered while the robot is running, or if the robot has been run but has not been reset.

The Solve Toolbar

The 'Solve Toolbar' provides options to run the robot.

- Run Robot - runs the robot for a number of time intervals, set in the View/Modify Robot Properties dialog (default is 300).

- Step Robot - runs the robot for one time step.

- Stop Robot - when the robot is running, this button stops the execution.

- Clear Trace - clears the trail that the robot draws on the canvas.

- Reset Robot - clears the controllers and resets the plan. This initializes the robot for another execution.

- View/Modify Plan - pops up a dialog that allows the user to create a list of locations that will be visited by the robot in order. To add a location to the plan, select the location from the 'available locations' list and click the "Add --->" button. To remove a location from the plan, select the location from the "Current Plan" list and click the "<--- Remove" button. To update the plan, click the "Update" button. To return to the applet without changing the plan, click "Cancel".

  Note: The plan cannot be altered while the robot is running, or if the robot has been run but has not been reset.

- View/Modify Robot Properties - pops up a dialog that allows the user to modify the robot's properties.
    - Maximum Time Step to Run - the number of time steps that is executed when the "Run Robot" button is pressed.
    - dt - the time interval for each step. This should be an integer number.
    - Number of Pixels per Unit - defines the number of pixels that the robot takes per step. This also has the effect of increasing the size of the whisker sensor.
    - Show Trace - determines whether or not the robot leaves a trace as it moves.

  Note: The robot properties cannot be altered while the robot is running, or if the robot has been run but has not been reset.

Modifying the Controllers

To modify the controllers, go to the Edit Menu and choose which controller you need to modify (high layer, medium layer, or environment). This will display a text area that shows the current controller code. This code is in Prolog. To add a new Prolog statement, just type it into the window. The user should be careful to input correct code into the controller. The applet will warn the user if the code is invalid, but sometimes errors (especially logical errors) can only be pinpointed when the robot is run.

Click "Update" to load the new controller code, or click "Default" to reset the controller code to the original. Clicking "Cancel" invalidates any modifications the user did since the last update.

Note: The controllers cannot be altered while the robot is running, or if the robot has been run but has not been reset.

Built-in Functions

There are some built-in functions that can be used in the controllers without being defined.

- add1(Prev, Next) increments a value by 1. This is not reversible, and should be treated as the IS operation Next IS Prev + 1.

- sub1(Prev, Next) decrements a value by 1. This is not reversible, and should be treated as the IS operation Next IS Prev - 1.

- positive(N) is true if N is positive.

- negative(N) is true if N is negative.

- zero(N) is true if N is zero.

DTD Definition

```
<!DOCTYPE ROBOTXML [
        <!ELEMENT ROBOTXML ( WORLD, HIGHCONTROLLER, MIDDLECONTROLLER, ENVIRONMENT ) >
                <!ATTLIST ROBOTXML VERSION NMTOKEN #REQUIRED >
        <!ELEMENT WORLD ( LOCATION+, WALL+, ROBOT, PLAN ) >
        <!ELEMENT LOCATION ( NAME, PROPERTY ) >
        <!ELEMENT NAME ( #PCDATA ) >
```

```
          <!ELEMENT PROPERTY ( #PCDATA ) >
          <!ELEMENT WALL ( START, END ) >
          <!ELEMENT START ( #PCDATA ) >
          <!ELEMENT END ( #PCDATA ) >
          <!ELEMENT ROBOT ( PROPERTY, DIRECTION ) >
          <!ELEMENT DIRECTION ( #PCDATA ) >
          <!ELEMENT PLAN ( LOC+ ) >
          <!ELEMENT LOC ( #PCDATA ) >
          <!ELEMENT HIGHCONTROLLER ( LogicProgram ) >
          <!ELEMENT LogicProgram ( #PCDATA ) >
          <!ELEMENT ENVIRONMENT ( LogicProgram ) >
          <!ELEMENT MIDDLECONTROLLER ( LogicProgram ) >
]>
```