ABSTRACT

MIT Haystack's Atmospheric Sciences Group has for nearly 50 years operated the Millstone Hill upper atmospheric radar, focused on studies of the near-Earth space environment. The radar is a complex system, which can pose a challenge for experiment design. The aim of this project was to create self-made software that affords the flexibility to design complex systems, which can pose a challenge for experiment design. The state chart drawing program was created from scratch.

The state chart drawing program supports such functions as moving, cutting, pasting, saving and opening files, undoing actions, redosing actions, and editing state and transition attributes, and choosing a state chart version previously entered data. The software also performs a number of tasks automatically that ensure a drawing is consistent. While open source software exists to translate code into state chart drawings, no low-cost readily available software could be found to translate state chart drawings into code. For this reason, and for the flexibility and control self-made software affords, the state chart drawing program was created from scratch.

PROGRAM FLOW

- SAMPLE DRAWING

- SAMPLE PARAMETER FILE

- USE CASE: MOVING AN OBJECT

1. The parent fields of states and the start and end fields of transitions are sensed and automatically recorded upon any creation, move, or paste of a state or event. Note the parent of StateID_23 and start and end attributes of EventID_25 before the move.

2. Select Groups of objects using the Select Groups button, which creates a rectangular drag tool. Single objects may be selected with the "Select Object" button, by clicking within a delta distance of an object's perimeter.

3. Selected objects are highlighted and can then be dragged.

4. The mouse button has not yet been released. The arrow representing EventID_25 overlaps the circle representing StateID_19.

5. The mouse button has been released and the program has automatically corrected the length and attachment point of EventID_25, has deleted the start arrow pointing to StateID_23 since its super-state, StateID_1, is hierarchical, and has updated the parent attribute of StateID_23 and the start and end attributes of EventID_25.

FEATURE ALGORITHM

- Saving and opening files

- Pickling: Python functions that turn objects into byte streams

- Saving start arrows consistent

- Find the smallest surrounding rectangle for each parent type: hierarchical or concurrent, and update arrows accordingly to type. Check for consistency every time an object is moved, pasted, or erased. If parent is concurrent, check that each child object has one start arrow and make changes as necessary. If hierarchical: look for a start arrow within the rectangle. Keep the tree arrow found and erase other start arrows, else if no start arrow was found, assign a start arrow to the first object found.

- Real-time drawing of arrows

- Change coordinates of arrowhead to current cursor location and continually redraw the screen

- Real-time drawing of rectangles and ability to drag a rectangle in any direction

- Calculate the vector that starts at the start click and ends at the current mouse location; create 45 degree lines in each direction from the rectangle. The uppermost, leftmost, point on the rectangle and its width and height are the input to a GTK rectangle drawing function.

- Snap-to-fit arrows

- If the point clicked is within a delta distance of a certain object’s perimeter, create a vector from the shape's corner to the point clicked; if the object is a circle, scale the vector by the radius of the circle to obtain the snap to point on the circle. If the object is a rectangle, determine which edge will be crossed by the vector and find the intersection of the vector and the line segment of the rectangle's edge to find the snap-to point.

- Highlighting Selecting objects

- Is the click within a delta distance of any object’s perimeter? Arrow: does the point lie within the rectangle created by delta on either side and on either side of the line segment? Circle: does the point satisfy the equation of the circle, accounting for delta? Rectangle: does the point lie within any of the four rectangles created by delta on either side and on either side of each line segment?

- Ensuring consistency of parent fields of states and start and end fields of transitions

- Basic: find the smallest surrounding rectangle to find an object’s parent. Currently the code accounts for a number of specific conditions individually. For example, if a rectangle is moved or erased, make the parent field of all objects that previously had that rectangle as a parent the moved rectangle’s former parent. This approach does not waste time or space making unnecessary updates but is more complex than an approach which keeps all objects with every move, create, paste, and erase action.

- Undo/Redo

- Keep a set of super-lists to hold the current values of all variables needed to recreate the current drawing. Appended new values to each of these lists any time an action is taken. Undo an action by moving backwards through the set of lists; redo by moving forward.

- Moving objects

- For all highlighted objects, draw vectors from the original point clicked to all object parameters needed to draw the object (for example, radius, for a circle). As the cursor moves, update the object’s parameters to the current location of the mouse minus the vector corresponding to the parameter.

- Printing status so that each parent-state is printed before its child

- Use a depth-first search: add the base state to a list. While the list is not empty, pop the last state from the list, write that state’s attributes to the file, and add its children to the list. Print all circles afterward.

FUTURE WORK

- Print and print preview buttons

- Capability to represent events that are turned into messages and then back into events to facilitate distributed computing

- Ability to draw loops at any location on a shape

- Rotation of event names so that they appear at the same angles as the arrows to which they correspond

- Ability to extend or contract lengths of arrows when objects or groups of objects are moved, if keeping all objects attached is desired

- Possible standardization of all shapes as rectangles

- Way to show movement that does not cause the screen to blink