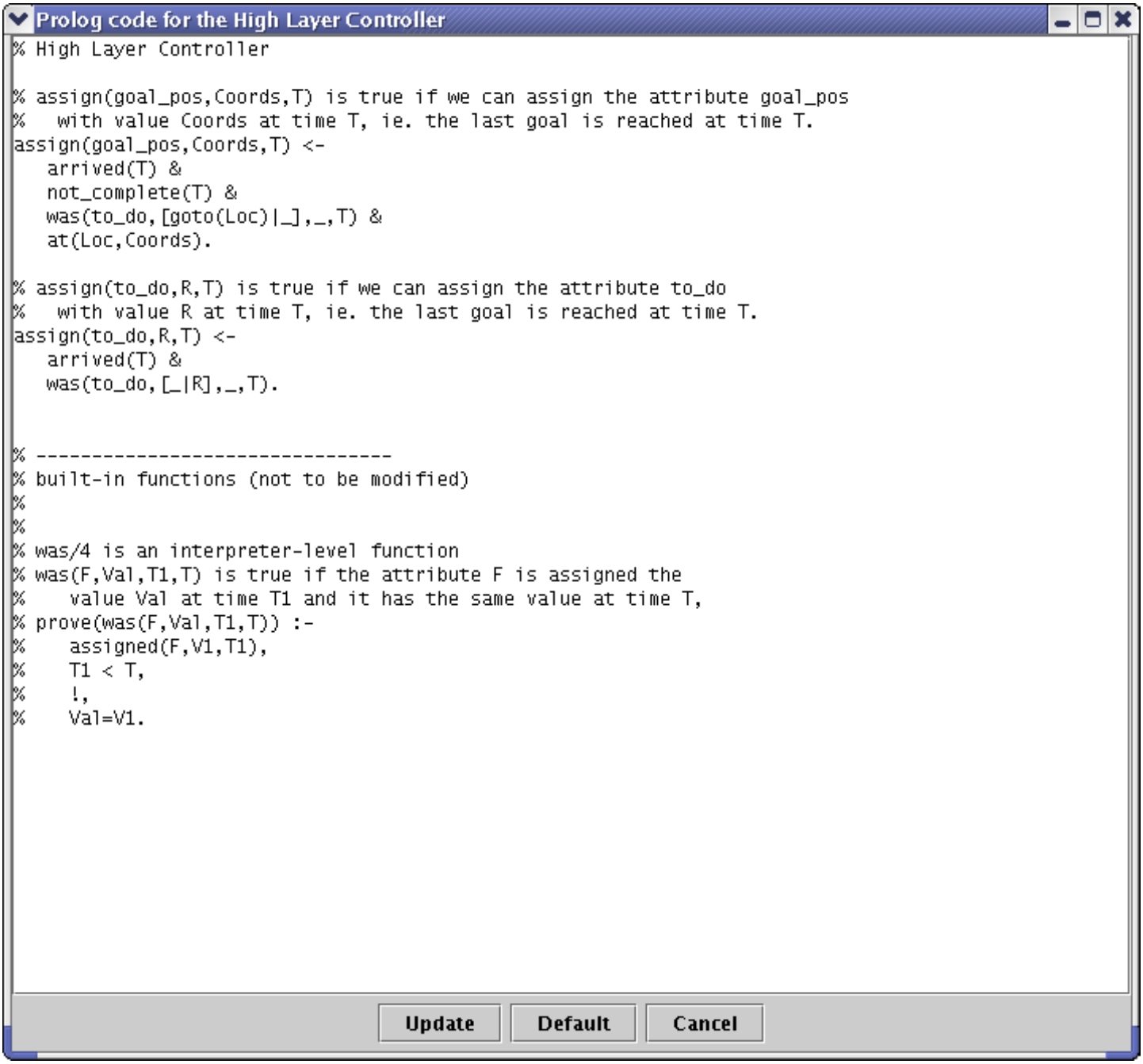
Tutorials

Robot Control

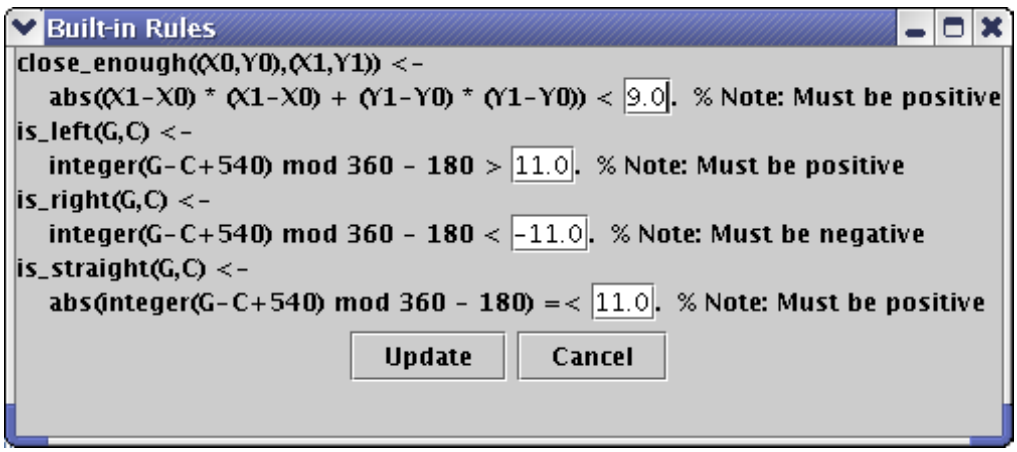
Tutorial Three: Controllers

The robot uses a system of hierarchical controllers to determine its actions. Since these controllers are written in prolog, it is possible to modify them, and see how the robot performs differently. Is it possible to write a controller that would not allow the robot to crash? The high level controller manages the plan and assigns goals to the middle level controller, which is to achieve them individually. The middle level controller is the one that determines *how* goals are to be achieved and in which direction the robot should go to achieve them. It handles steering, in general terms, but the actual interaction with the environment is handled by the lowest level controller. This controller defines the details of the robot's movement and does not have to know anything about the overall plan. It simply has to know what to do when (for example) it hits an obstacle.

To view or change the controller code, go to the 'Edit' menu. You will see options to view and modify the high and medium level controllers as well as the environment and built-in functions. The built-in functions define the numerical threshholds for what it means to be "close to" or "left of" an object. The edit windows for each controller look quite similar. Below is the window allowing you to edit the high level controller. The windows for editing the middle level and environment level controllers are quite similar.



Notice that there are certain built-in functions listed in the controller code that cannot be modified. These functions are part of the prolog interpreter. The other built-in functions can be modified to some extent. The numerical threshholds that they define can be changed, but nothing more.



As a final note, the syntax accepted by this applet follows the conventions for CIlog syntax. CIlog

rules are of the form *head* <- *body*, and the atoms in the bodies of rules are joined by the "&" symbol instead of by commas.