EPIC QuickStart (Mac)
7/18/04 - Kieras & Hornof

This is an introduction to running a model using EpicApp, a standalone executable file that runs an EPIC model for a specific simulated device.

What you need to start
• An EpicApp that will run on your operating system
• A production rule (prs) file for the task simulation supported by that EpicApp

To start EPIC and run a model

1. Start EPIC.
Start the "EpicApp" application.

You should see a Menu bar that says "EpicApp File Edit Control", and seven windows with the following seven titles:

Physical Vision, Sensory Vision, Perceptual Vision,
Physical Audition, Sensory Audition, Perceptual Audition,
Normal Output

The three "Vision" windows each have two concentric circles in them. The circles are 1° and 10° of visual angle around the point where the eyes are looking.

2. Load the production rules file.
Open the "Control" menu and select "Compile ..."; you should get a file opening dialog titled "Open production rule file". Navigate to the the directory containing this file and choose a file with a ".prs" extension as the production rule file. In the "Normal Output" window, you should see a message giving the full file path, ending in "<filename>.prs compiled."

3. Run the model.
Open the "Control" menu and select "Run". Objects should appear and disappear in the visual windows. The Normal Output window should have a lot of text in it, including a listing of all parameter settings for that run of the model.

Repeatedly select Control/Run (or use the shortcut key) until the model finishes running and "Simulation Done!" appears in the Normal Output window.

4. Study the results.
EpicApp runs the production rules to simulate accomplishing a task on a device, and to output performance data from the task execution. Each run of a model generally produces data that you can study to evaluate how the task was executed. The simulated device plays the role of the experimental apparatus, and records the behavior of the simulated human and outputs some statistics. The summary data are typically displayed in the "Normal Output" window at the end of a simulation.

To control model execution

To run one cycle at a time:
Open the "Control" menu and select "Go one cycle" (or use the shortcut key).

To adjust the number of cycles executed with each “Run” command:
1. Open the "Control" menu and select "Run Controls ..."
2. Change the number in the “Cycles/Run command” field.
3. Select “OK”.

To pause execution after a particular rule fires:
1. Open the "Control" menu and select "Run Controls ..."
2. Select “Breaks Enabled for Rules in Red”.
3. Scroll through the list of rules and select the rules on which you want to pause.
4. Select “OK”.
Note: Later, when the model pauses, continue with another “Run” or "Go" command. If you disable the Breaks, the breaks on the individual rules will remain, and you can just re-enable them again later. Recompiling the rules clears all of the breaks, however.

To zoom in or out in the visual displays

1. Open the "Control" menu and select "Display Controls ..."
2. Change “Spatial scale pixels/degree”. Bigger numbers zoom in. Smaller numbers zoom out.
Note: You may need to adjust the size of the windows accordingly.

Note: To adjust the displayed size of objects, but not zoom in or out, adjust the “Spatial object magnification” in the same dialog box.

To watch the PPS execution and to see the contents of WM after each cycle

1. Open the "Control" menu and select "Display Controls ..."
2. Select the first two check boxes, “PPS Memory Contents” and “PPS Run Messages.” (Note: The other three check boxes, “PPS Compiler Messages”, “Compiler Details”, and “Run Details”, provide additional information not generally needed when building a model.)
3. Select “OK”.

The PPS execution output will now appear in the “Normal Output” window. You may need to control the number of cycles executed (discussed above) to quickly find the part that is of interest to you (such as when you expect a rule to fire, but it does not fire).

Note: The same PPS execution output will also be saved to a file if you open the "Control" menu, select "Logging ...", and open a file.

To modify the production rules.

1. Open the production rule (.prs) file with a text editor.
2. Make and save your edits. Be sure to save as plain text (not RTF or any other format).
3. In EpicApp, reload the .prs file with the “Compile...” command (or use its shortcut key).
4. Run the model and see how its execution has changed.

(Mac OS 9 users will need to change the .prs extension to .txt, open and save the file using TextEdit, and then change the extension back to .prs.)

To interpret the PPS execution output:

Here is output from the execution trace [with annotations]. In the annotations, "WM" refers to working memory, "PRs" are production rules, "LHS" refers to the left-hand "If" side of a rule, and "POO" refers to the EPIC Principles of Operation document.

```
...
3350:Cycle 66:                          [This starts the report for Cycle #66,
                                        which occurred after 3350 ms elapsed.]
Goal                                  [These headings organize the WM contents.]
  (Goal Do Visual_search)           [The task goal is the first WM item listed.]
Step
  (Step Look_at Random_Object)            [This is the next step in the task.]
Tag                                    [Tags are WM items created by the PRs.]
  (Tag 61 Target_Label)               [61 has been saved as the target label.]
  ...
  (Tag Vpsychobj7 fixation_object)          [Vpsychobj7 the unique identifier
                                                 for a psychological object.]
Visual
  (Visual Vpsychobj1 Above Vpsychobj2)           [These are the visual objects
  (Visual Vpsychobj1 Eccentricity Periphery)      and features in WM.  They
  (Visual Vpsychobj1 Status Visible)              correspond with the contents
  (Visual Vpsychobj1 Text 61)                     of the "Perceptual Vision"
  ...                                             window, the third of the
  (Visual Vpsychobj7 Color Yellow)                three visual windows.
  (Visual Vpsychobj7 Eccentricity Fovea)          Eccentricity is relative to
  (Visual Vpsychobj7 Shape Triangle)              the current gaze position.
  (Visual Vpsychobj7 Size Medium)
  (Visual Vpsychobj7 Status Visible)
  (Visual Vpsychobj7 Text 39)
Motor
  (Motor Manual Execution Free)                  [These are the statuses of the
  (Motor Manual Modality Free)                    three motor processors.  Each
  (Motor Manual Preparation Free)                 status is always Busy or Free.
  (Motor Manual Processor Free)                   Execution Busy, for example,
  ...                                             means the processor is not
  (Motor Ocular Execution Free)                   available to start a new
  ...                                             action.  See the POO for more
  (Motor Vocal Execution Free)                    details.]
  ...
*** Rules fired:                                  [The PRs that fired on this
Fire: Continue_random_search                       cycle (#66) are listed here.
  Bindings: ((?label:39)                           The bindings show the symbols
  (?nominee:Vpsychobj5)                            in WM that were assigned to
  (?object:Vpsychobj7)                             the variables to match the
  (?target_label:61))                              LHS of a rule.]
  Add: (Tag Vpsychobj5 checked)                   [These are the modifications
  Add: (Tag Vpsychobj5 fixation_object)            to WM after the rule fires,
  Delete: (Tag Vpsychobj7 fixation_object)         seen on the next cycle.]
  Motor command:
  (Ocular Perform Move Vpsychobj5)             [This motor command is executed.]
3400:Cycle 67:                                     [The next cycle starts here.]
...
5300:Cycle 105: No rules fired                  [WM is not displayed for cycles
5350:Cycle 106: No rules fired                   that do not modify WM.]
...
```

To study the flow of control among processors in more detail

1. Open the "Control" menu and select "Tracing ..."
2. Select the processors you want to trace.
3. Select “OK”.

A “Trace Output” window will appear and display a textual representation of the flow of information among the processors. Below is an [annotated] output when the “Manual” and “Device” processors are traced:

```
...
[The first four lines indicate that 4499 msec into the simulation, the human’s
manual motor processor receives a command and starts its execution.  No
preparation time is needed because the new action is the same as the previous.
The “Human1” alludes to the fact that multiple users are possible.]

4499:Human1:Human:Manual:Motor_Command_Action_event received.
4499:Human1:Human:Manual:Preparation started for Manual_Punch_action
4499:Human1:Human:Manual:Preparation complete for Manual_Punch_action
4499:Human1:Human:Manual:Execution started for Manual_Punch_action

[The next line shows that 75 msec later, the device received the event.  It is
called “Williams67” after the device that is being simulated.]

4574 Williams67:Device_Keystroke_event received

[The next two lines shows that 175 msec later, the manual motor processor
completed the action.]

4749:Human1:Human:Manual:Motor_Movement_Complete_event received.
4749:Human1:Human:Manual:Execution complete for Manual_Punch_action
...
```

To send the model output to a file

To save the parameter settings that appear at the start of a run, the results of the model run, and the PPS execution output (when the execution is viewed as described above):

1. Open the "Control" menu, select "Logging ...", and open a file.
2. Select “OK”.
3. To stop logging, either
   (a) quit EpicApp or
   (b) open the "Control" menu, select "Logging ..." and close the file.
4. View the contents of your saved file using a text editor.

Note that this will not save the output of the “Trace Output” window.

To change a parameter in the simulated device

EpicApp simulates one device at a time. Sometimes the device simulation is designed to take a parameter, such as the number of trials to run in an experiment. To change this parameter:

1. Determine what how the parameter string is used for a device. This should be provided with the documentation provided for a specific EpicApp. (The Williams 67 parameter string, for example, sets the number of trials in an experiment and the number of items that appear in the display. For example, the string “10 5” indicates ten trials and five objects.)
2. Open the "Control" menu, select "Run Controls ...".
3. Set the “Device Parameter String”.
4. Select “OK”.
5. Verify there are no error messages in the “Normal Output” window.