



Extended Example

Transition Events

FLEA experimental user interface

Defining transition events

The FLEA experimental interaction window for definition of transition events is shown below, with the choices to define the event MoreLicensesThanUsers already made. Note that the transition event is defined in terms of relations, not events. As the values in these relations change, the monitor is continually checking to see whether the MoreLicensesThanUsers event has occurred. As with other event definitions, to become notified of occurrences of those events it is necessary to select one or more notification styles - click the Notify button on the control panel to make such choices.

The screenshot displays the FLEA experimental user interface. The main window is titled "Define Event: event occurs on TRANSITION". It features a text field for the event name, currently set to "MoreLicensesThanUsers". Below this are two input fields for conditions A and B, both containing "[COUNTLICENSES =)" and "[COUNTUSERS =)" respectively. There are two groups of radio buttons: "Idiom" with options "start Exists A", "start A > B" (selected), and "start A"; and "Parameters" with options "Don't Care", "Shared" (selected), and "Different". A note states: "NOTE - many of the definitions this form constructs require some manual editing to correctly declare parameters". Below the note is a text area containing the following code:

```
[ defevent MoreLicensesThanUsers :transition
  [ ( ts ) s.t.
  [ start ( > ( COUNTLICENSES = ) ( COUNTUSERS = ) ) ] ] ]
```

 To the right of the text area are buttons for "HELP", "Clear", "OK", and "Exit". On the far right, a panel titled "Defined Events, Relations and Predefine" shows a list of events and relations, including "COUNTLICENSES(ANON-REL1991)" and "COUNTUSERS(ANON-REL1991)". At the bottom right, a "Monitor Control Panel 3.0" is visible, featuring a grid of buttons: "Define", "Notify", "Set", "Query", "Exit", and "HELP", along with two green circular indicators.