GPars - Groovy Parallel Systems

Jon Kerridge

Version 1.0, 2015-10-29

Table of Contents

Parallel Processes: Non Deterministic Input 1
Alternative 1

Parallel Processes: Non Deterministic Input

In many systems, there is a requirement to provide feedback from a down stream stage of the process to an upstream one.

The upstream process has no idea when such a piece of feedback information is going to arrive and thus has to be able to accept it at any time. The behaviour of such a process is said to be `non-deterministic` because the arrival of the information cannot be determined when the process is defined. We know that such feedback can arrive but not when.

Similarly, a `process network` may be subject to external interventions that change the parameters of the system. It is known that these interventions will occur but not when. This happens, for example, in medical laboratory equipment where a running check is kept of the range of values for each test that has been produced. Over a given period, it is known that the mean value will lie within known bounds.

If the machine is outside those bounds then it enters an automatic recalibration process.

Alternative

For this purpose, **Alternative** provides the program structuring mechanism. In its simplest form, the **Alternative** manages a number of input channels.

1. On executing an **Alternative**, the state of all the input channels is determined.
2. If none of the channels are ready, the **Alternative** waits until one is ready, reads the input and then obeys the code associated with that input.
3. If one input is ready, then that channel is read and its associated code is obeyed.
4. If more than one channel is ready then one is chosen according to some selection criterion and the channel is read and its associated code obeyed.

Typically, an **Alternative** is incorporated into a looping structure so that the input channels can be repeatedly accessed.

As a first example we shall take the process that generates a sequence of integers, **GNumbers**, previously described in *Section 3.4*. We shall modify it so that it accepts an input which resets the sequence to a number input by the user at any time chosen by the user.

Want to read more of this chapter? Download this chapter's PDF here.
