GPars - Groovy Parallel Systems

Jon Kerridge

Version 1.0, 2015-10-29

Table of Contents

Graphical User Interfaces: Brownian Motion

Previously, in `Chapter 3`, a simple user interface was developed that enabled easier interpretation of the output from networks of `groovyPlugAndPlay` processes. This chapter explores more complex user interfaces in conjunction with a relatively simple graphical application based upon particle movement.

The **JCSP** package contains an active implementation of the java AWT (`Abstract Windows Toolkit`).

The term **active** is here used to mean that each **AWT** component, for example, button, scrollbar and canvas, has been wrapped in a process so that component events and configuration are undertaken by channel communications. This means that the active components can be connected to any process. Furthermore, the programmer does not have to write any event handling or listener methods as these are contained within the active process wrapper. The active components inherit capabilities from the basic **AWT** components, thus the methods and fields associated with the component can be reused and active and ordinary non-active components can be used in the same interface.

The primary benefit of the active **AWT** components is that processes that access the user interface can utilise their non-deterministic capabilities, thereby reflecting the unpredictable behaviour of user interfaces. The user interface has no knowledge of when, for example, a button is going to be pressed and thus either a channel communication or an alternative provides a simple method for capturing that non- deterministic behaviour.

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