

A Reactive Programming Model for Global Computing

G erard Boudol

Inria Sophia Antipolis, France

Abstract

In this talk I introduce a programming model for the mobile code, and more generally for programming in a global computing perspective. I first present some requirements for this style of programming, arising from the features and new observables of the global computing context, and I briefly discuss some of the models and programming languages that have been proposed - Obliq, pi-based and Linda-based models, Ambients. I then present a model based on the ideas of "synchronous" programming, that is based on suspension and preemption primitives associated with locally broadcast events. This programming style, providing a notion of reaction and time-out, looks appropriate to address the unreliable character of accessing resources in a global computing context, and to deal with the various kinds of failures - such as unpredictable delays, transient disconnections, congestion, etc. - that arise in a global network.

The proposed model, called ULM, combines a standard imperative and functional style, such as the one of ML or Scheme, with some construct for "reactive" programming. The model also proposes constructs for programming mobile agents, that move together with their state, made of a control stack and a store. This makes the access to references also potentially suspensive, and this is one of the main novelties of the model. The focus of this work is on giving a precise semantics for a small, yet expressive core language. Some examples of this expressiveness are given in the talk.