This material is based upon work supported in part by the National Science Foundation under Grant No. 0639876. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

On Serializing and Deserializing FRP-style Interactive Programs

Yoshiki Ohshima

February 15, 2013

1 Introduction

Our group has been developing a GUI framework called KSWorld [7] that draws upon Functional Reactive Programming (FRP) [3]. The core idea is to describe the application logic in terms of the graph of data dependency in a time-aware manner. As in other GUI frameworks, there is a type of basic graphical object, which is called the "`Box`". A `Box` has fields that represent its characteristics such as its location, its graphical appearance (called a shape) etc., and also its dynamic actions. What separates the KSWorld from more conventional GUI framework is that these fields are nodes, or *streams*, in the dependency graph.

The KSWorld is designed to support end-user authoring in the spirit of Etoys [1]. The user can add and remove `Box`es and streams into his project at any time while the system/application is running. When the user is satisfied, a set of `Box`es that the user is interested in will be serialized to be stored or sent over the network, and deserialized at the other end of the pipe. The deserialized project may be merged into the environment that may be quite different from where it was created.

Upon serializing a project made in this manner, which elements in the project to export and how to import them is an interesting question. (But, here is the short answer: "behaviors" should be saved with their current values while the "events" should not be saved with their current values. The behaviors should be re-evaluated upon loading.)

In the following sections, we explain the idea more deeply.

2 Events and Behaviors

In FRP, there are two types of time-varying variables. One is called the "behavior", which represents a continuous value over time whose time domain stretches from $-\infty$ to ∞. In other words, a behavior has *always* a value at any given time, including at the "current time", which denotes the logical time where the system is. (The value of the behavior at the current time is "current value".) Another type is called the "event", which represents a discrete sequence of values

VPRI Memo M-2013-001