

Functional Reactive Programming

Paul Hudak

Yale University, New Haven, CT 06518, USA

paul.hudak@yale.edu

www.cs.yale.edu/users/hudak.html

Abstract. *Functional reactive programming*, or FRP, is a style of programming based on two key ideas: continuous time-varying *behaviors*, and event-based *reactivity*. FRP is the essence of *Fran* [1,2], a domain-specific language for functional reactive graphics and animation, and has recently been used in the design of *Frob* [3,4], a domain-specific language for functional vision and robotics. In general, FRP can be viewed as an interesting language for describing *hybrid systems*, which are systems comprised of both analog (continuous) and digital (discrete) subsystems. Continuous behaviors can be thought of simply as functions from time to some value: **Behavior** $a = \text{Time} \rightarrow a$. For example: an image behavior may represent an animation; a Cartesian-point behavior may be a mouse; a velocity-vector behavior may be the control vector for a robot; and a tuple-of-distances behavior may be the input from a robot's sonar array. Both continuous behaviors and event-based reactivity have interesting properties worthy of independent study, but their integration is particularly interesting. At the core of the issue is that events are intended to cause discrete shifts in *declarative* behavior; i.e. not just shifts in the state of reactivity. Being declarative, the natural desire is for everything to be first-class and higher-order. But this causes interesting clashes in frames of reference, especially when time and space transformations are applied. In this talk the fundamental ideas behind FRP are presented, along with a discussion of various issues in its formal semantics. This is joint work with Conal Elliott at Microsoft Research, and John Peterson at Yale.

References

1. Conal Elliott. Modeling interactive 3D and multimedia animation with an embedded language. In *Proceedings of the first conference on Domain-Specific Languages*, pages 285–296. USENIX, October 1997.
2. Conal Elliott and Paul Hudak. Functional reactive animation. In *International Conference on Functional Programming*, pages 163–173, June 1997.
3. John Peterson, Paul Hudak, and Conal Elliott. Lambda in motion: Controlling robots with haskell. In *First International Workshop on Practical Aspects of Declarative Languages*. SIGPLAN, Jan 1999.
4. A. Reid, J. Peterson, G. Hager, and P. Hudak. Prototyping real-time vision systems: An experiment in DSL design. To appear Proc. Int. Conference on Software Engineering, May 1999.