

Lucian : Dataflow and Object-orientation

Dominic Orchard

There are a multitude of programming languages in existence. Why? Because no single language or paradigm can be all things to all people. Whilst one class of programs may be succinctly written, easily reasoned about, and efficiently compiled within one language another class of programs may be impenetrably intractable for the human and the compiler. Language interoperation and multi-paradigm languages are the means by which programmers can get the best of both worlds. This talk introduces the Lucian programming language, a cross-paradigm derivative of the Lucid dataflow language, that interoperates declarative dataflow and object-orientation. Programs that are dynamic or reactive can be succinctly expressed within Lucid. However, not all parts of such a program are necessarily easy to express in dataflow form. Lucian provides an escape to an imperative object-oriented language where subparts of a program may be more easily written and compiled. Conversely Lucian provides a way to write programs using objects in Lucid's dataflow equation style. This talk introduces Lucid for the uninitiated viewer and proceeds to introduce the central constructs of Lucian. A comparison of the underlying computational models of Lucid and object-orientation is given to explain the appropriateness of this interoperation.

Dominic Orchard

University of Cambridge, Computer Laboratory, William Gates Building, 15
JJ Thomson Avenue, Cambridge CB3 0FD, UK
dominic.orchard@cl.cam.ac.uk

BCTCS 2009 - Talk abstracts