

Deductive and Object Data Languages: A Quest for Integration *

Michael Kifer

Department of Computer Science
University at Stony Brook
Stony Brook, NY 11794, U.S.A.
kifer@cs.sunysb.edu

Abstract. According to rumors, the early hybrids of object-oriented and deductive languages were mutants that escaped from secret Government AI labs. Whether this is true or not, the fact is that by mid-80's, database and logic programming communities began to take notice. The temptation was hard to resist: the object-oriented paradigm provides a better way of manipulating structured objects, while logic and deduction offer the power and flexibility of ad hoc querying and reasoning. Thus, hybrid languages have the potential for becoming an ideal turf for cultivating the next generation of information systems.

The approaches to integration of the two paradigms range from logic-based languages with unified declarative semantics, to message-passing prologs, to Prolog/C++ cocktails.

In the past eight years, my colleagues and I have been developing a unified object-based logic intended to capture most of the essentials of the object-oriented paradigm. The overall plot here is that once the fundamentals are in place, the actual hybrid programming languages can be carved out of the logic the same way as deductive data languages have been carved out of the classical logic.

There are two distinct aspects in object-orientation: structural (the statics) and procedural (the dynamics). The static aspect was somewhat easier to capture in logic and after taking a few wrong turns F-logic was developed [17]. Taming the dynamics was harder, because there was no widely accepted solution to the problem of state changes even for traditional deductive languages. Our recent work on Transaction Logic [8, 7] appears to provide an adequate framework for specifying the dynamics in classical deductive languages.

This paper first reviews some core aspects of F-logic and Transaction Logic, and then shows how the two can be combined in a natural way to yield a unified foundation for future work on deductive object-oriented languages both in theory and practice. At the end, we discuss areas that still remain to be explored.

* Work supported in part by the NSF grant IRI-9404629.