

# Modules as Values in a Persistent Object Store<sup>1</sup>

Gilad Bracha

*Horizon Technologies of New York, Inc.*

`gilad@cs.utah.edu`

Charles F. Clark, Gary Lindstrom<sup>2</sup> and Douglas B. Orr

*University of Utah*

`{clark,lindstrom,dbo}@cs.utah.edu`

## ABSTRACT

We report on an object manager (OM) providing persistent implementations for C++ classes. Our OM generalizes this problem to that of managing persistent *modules*, where the *module* concept is an abstract data type (ADT). This approach permits a powerful suite of module manipulation operations to be applied uniformly to modules of many provenances, including non-class based entities such as conventional object files, application libraries, and shared system libraries. OMOS, a generalized linker and loader, plays a central role in our OM. Class implementations are represented by OMOS modules, which in turn are constructed from OMOS meta-objects encapsulating linkage *blueprints*. We cleanly solve the problems of (i) logically (but not physically) including executable object files in our OM, (ii) reconciling class *inheritance history* and *linkage history*, and (iii) supporting alternative implementations of a class, for client interoperability or version control.

**Key words:** Persistence, object stores, modules, dossiers, class implementations, dynamic linking and loading, functional interposition.

---

<sup>1</sup> This research was sponsored by the Defense Advanced Research Projects Agency (DOD), monitored by the Department of the Navy, Office of the Chief of Naval Research, under Grant number N00014-91-J-4046. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

<sup>2</sup> Primary contact author; Computer Science - 3190 MEB, Salt Lake City, UT 84112 USA, phone +1-801-581-5586, fax +1-801-581-5843.