**DLS ’07 - Mirages: Behavioral Intercession in a Mirror-based Architecture**

The contributions aren’t very explicitly enumerated in this paper, but they are roughly as follows:

* it reconciles mirror-based systems with behavioural intercession by introducing the concept of *mirages*, base objects whose semantics are defined by implicit mirrors
* it describes and illustrate the integration of this reflective architecture for the distributed object-oriented programming language AmbientTalk
* and it illustrates the utility of mirages by presenting an implementation of future-type message passing based on AmbientTalk mirages

**OOPSLA 2011 - Virtual Values for Language Extension**

(copied verbatim from paper text):

The main contributions of this paper are:

* it virtualizes the entire interface between code and data values, thus providing a general mechanism for value- specific behavioral intercession;
* it clarifies that languages that are not object oriented or only partially object oriented can still enjoy the extensibility benefits of pure object languages;
* it presents an operational semantics for virtual values;
* it illustrate the extensibility benefits of virtual values by implementing six non-trivial language extensions: (1) complex numbers; (2) units of measure; (3) delayed evaluation; (4) taint analysis; (5) contracts; and (6) revocable membranes;
* and it reports on our experience implementing this design in the Firefox browser.

**Proposed OOPSLA 2012 Paper**

The main contributions of this paper are:

* it presents semantic requirements for *object holograms*, virtual objects that behave identically to objects in remote or past executions;
* it illustrates that the use of object holograms improves the accuracy, readability, simplicity and maintainability of remote or post-hoc execution debugging and analysis;
* it demonstrates that mirror-based behavioural intercession, generally studied in the context of dynamic languages, can be efficiently implemented in the statically-typed, pre-compiled Java language on commodity JVMs;
* and it reports on our experience applying object holograms to heap dump analysis and omniscient debugging