

A new interface for OPTIMA

Samuel Gélineau

OPTIMA is a transaction framework with strong fault-tolerant guarantees. The framework uses *open multithreaded transactions* as its transaction model, which allows many threads to participate in a transaction. As a comparison, most competing models allow concurrent transactions, but assume that each one is controlled by a single thread.

Prototypes of the framework in Java and Ada have been developed by Jörg Kienzle, who also designed the transaction model. Recently, we attempted to introduce a new optimization called *look-ahead* to the Java version. While implementing it, I often stumbled upon features which could easily be implemented by using OPTIMA itself. Moreover, the features I needed did not create cyclic dependencies with the features I was implementing, yet I was unable to bootstrap because the interface only allowed the full set of OPTIMA's guarantees to be unleashed upon a program, rather than the subset I needed.

In my presentation, I will describe my attempts as I redesigned OPTIMA's interface to make it more modular.