

## ABSTRACT

---

This thesis describes the way in which different reflective technologies can be used to develop a dynamically adaptable and extensible computational system, without any dependency of a concrete programming language, built over a heterogeneous computing platform.

An abstract machine with a reduced instruction set is built as the root computation system's engine; it offers the programmer basic reflection computation primitives. Its reduced size and its introspective capabilities, make it easy to be deployed in heterogeneous computational systems, becoming a platform-independent computational system.

By using the reflective features offered by the abstract machine, the programming abstraction level can be extended. This would be programmed on its own language, without needing to modify the virtual machine's source code and, therefore, without losing code portability. As an example of its extensiveness, the programming environment is programmed (using the abstract machine programming language) achieving facilities such as persistence, distribution, thread scheduling or garbage collection. All this new abstractions are adaptable at runtime to any system application.

A generic processing language system is designed, making the whole system independent of the language the programmer may select. Any application can interact with each other, by using the abstract machine's computational model, whatever the programming language the application has been codified in.

The dynamic flexibility the system achieves is greater –both in expressiveness and in the adaptable computational-features set– than any other existing one. At runtime, the application structure and its programming language specification (lexical, syntactic and semantic features) are completely customisable; it can be adapted by the own application as well as by any other program –no matter the language they have been programmed in. The reflective system developed has no restriction at all: it can adapt any computational trait without any expressiveness restriction –it may use the whole abstract machine programming language. This flexibility offered is dynamic: there is no need to stop the application execution in order to adapt it to any runtime requirement, unexpected at design time.