Project Number: IST-2001-33234
Project Acronym: PEPITO
Title: PEer-to-Peer Implementation and TheOry
Requirements, type systems, and semantics for peer-to-peer versioning support
Due date: project month 38
Delivery Date: 2005.2.28
Responsible Partner: UCAM
Participating Partners: INRIA, UCAM
Available to: Public

Deliverable no:	D3.12
Title:	Requirements, type systems, and semantics for peer-to-peer versioning support
Workpackage no:	WP3
Due date:	38
Delivery date:	2005-2-28

This deliverable's goal is to address fundamental language design problems that arise from versioning in peer-to-peer computing, in which the *construction of systems is carried out in decentralised ways* and for which the parts of these systems *evolve over time* and yet need to be able to *communicate amongst themselves*.
To that end, we have:

- designed a calculus for analysing the problem of managing *multiple versions* of the same library or object that have been dynamically loading into a single run-time, studying the disciplines and expressivity required to extend smoothly Java-like languages with support for multiple versions of objects.
- created a language, Acute, that extends an OCaml core to support distributed development, deployment, and execution, allowing type-safe interaction between separately-built programs. It is expressive enough to enable a wide variety of distributed infrastructure layers to be written as simple library code above the byte-string network and persistent store APIs, disentangling the language runtime from communication. Acute has a fully formal semantics and a prototype implementation.
- constructed a core calculus for dynamic software updating in C-like languages that is flexible, safe, and predictable. It supports dynamic updates to functions (even active ones), to named types and to data, allowing on-line evolution to match source-code evolution as we have observed it in practice.