

Project Number: IST-2001-33234
Project Acronym: PEPITO
Title: PEer-to-Peer Implementation and TheOry
Progress Report on Languages and Systems
Due date: project month 24
Delivery Date: 2003-12-31
Responsible Partner: EPFL
Participating Partners: EPFL, INRIA, KTH, SICS, UCAM, UCL

Deliverable no:	D3.10
Title:	Progress Report on Languages and Systems
Workpackage no:	WP3
Due date:	month 24
Delivery date:	2003-12-31

Short Description

This deliverable gives a brief report on the activity in WP3 of the Technical Annex.

The task on Mobile Computation [KTH:12] develops a model for strong mobility with concomitant programming abstractions. The model is based on mobile thread and extends the distributed dataflow model underlying Oz and Mozart. A common implementation infrastructure based on thread replication and migration managers has been developed that provides the necessary support for the programming abstractions. A first feasibility study has been conducted by using a common test scenario for mobile applications.

The task on Resource Control [INRIA:34, UCAM:24] mainly consists of the following three contributions: a calculus for safe dynamic access to resources, based on the Join calculus; a calculus for controlling communication even in the presence of mobile agents; a theory of type-safe marshalling and distributed version control in the presence of abstract data types and fancy module systems. The work results on type safe semantics for distributed computing have been implemented in the prototype language Acute, which is based on Ocaml.

The task on Mozart/Oz [SICS:6, UCL:12] mainly intends to extend Mozart/Oz with peer-to-peer abstractions. Additionally, this deliverable reports on the integration of the Distributed Subsystem, developed in WP4, into the Oz language. We designed and implemented a first prototype of a peer-to-peer Oz platform, as a library, called Peer-to-Peer System (P2PS). The P2PS library is developed in Oz on Mozart and it implements DKS and GSChord, distributed hash table based algorithms developed in WP2.

The task on Java/DACE/Scala [EPFL:24] develops a Java interface to the Distributed Subsystem (DSS). We intend to provide the same functionality as the DSS API (the DSS middleware itself is written in C++). The Java API is implemented as a Java library which rely on the Java Native Interface (JNI) to interact with the DSS.

Partner owning:	EPFL
Partner contributing:	EPFL, INRIA, KTH, SICS, UCAM, UCL
Available to:	Public