



**PEPITO**  
**IST-2001-33234**  
**PEer-to-Peer Implementation and TheOry**

**Deliverable no: D1.9**

## **Final report on formal models**

REPORT VERSION: first

REPORT PREPARATION DATE: 2005.2.28

CLASSIFICATION: Public

DELIVERABLE NO: D1.9      DUE DATE: Month 38      DELIVERY DATE: Month 38

PROJECT START DATE: 2002.01.01      PROJECT DURATION: 36+4 months

RESPONSIBLE PARTNER: UCAM

PARTICIPATING PARTNERS: EPFL, INRIA, KTH, UCAM, UCL

PROJECT COORDINATOR: Swedish Institute of Computer Science AB

PROJECT PARTNERS: EPFL Lausanne, INRIA Paris, KTH Stockholm, UCL Louvain, University of Cambridge UK



**Project funded by the European Community under the 'Information Society Technologies' Programme (1998–2002)**

Project Number: IST-2001-33234  
Project Acronym: PEPITO  
Title: PEer-to-Peer Implementation and TheOry  
Deliverable No: D1.9  
Final report on formal models  
Due date: project month 38  
Delivery date: 2005.2.28

Responsible Partner: UCAM  
Participating Partners: EPFL, INRIA, KTH, UCAM, UCL

9th March 2005

Editor: Peter Sewell

Authors: Uwe Nestmann, Peter Van Roy, and Peter Sewell.

## **Contents**

**1 Overview**

**2**

# 1 Overview

This deliverable gives an overview of the research carried out in Workpackage 1 during the final period of PEPITO. We confine ourselves here to a brief description, leaving a more discursive treatment of the work on formal models carried out throughout the project to Deliverable D1.2, *Survey of formal challenges and solutions for peer-to-peer computation*.

The Technical Annex<sup>1</sup> contains two tasks in this workpackage:

▷ **Transactions and the Semantics of Failure**

1. A main goal here was to establish an accurate failure semantics for low-level interaction. As we stated in the project proposal:

“We will develop rigorous semantic models of the low-level interactions and failures that may occur in a peer-to-peer system — not in the rather abstract styles of traditional distributed algorithm theory or process calculi, but for the actual networking APIs used.”

The final period of PEPITO has seen this achieved. We have produced a behavioural specification of the networking *Sockets API* and the underlying TCP and UDP protocols that exactly characterises behaviour in the presence of message loss, malicious attack with arbitrary messages, and arbitrary use and misuse of the API. It is mathematically rigorous, detailed, and has been shown to be accurate by an experimental validation process. The specification and an extended discussion document have been made available; one paper has been submitted for publication and more will follow in the future. The work establishes practical techniques for rigorous description and automated testing that can also be used for future protocol designs, at either the transport or peer-to-peer overlay levels.

- [1] Steven Bishop, Matthew Fairbairn, Michael Norrish, Peter Sewell, Michael Smith, and Keith Wansbrough. Rigorous specification and conformance testing techniques for network protocols, as applied to TCP, UDP, and Sockets. Submitted for publication. Available at <http://www.cl.cam.ac.uk/users/pes20/Netsem/paper.pdf>, February 2005. 15pp.
- [2] Steven Bishop, Matthew Fairbairn, Michael Norrish, Peter Sewell, Michael Smith, and Keith Wansbrough. TCP, UDP, and Sockets: rigorous and experimentally-validated behavioural specification. Volume 1: Overview. Available at <http://www.cl.cam.ac.uk/users/pes20/Netsem/tr.pdf>, February 2005. vi+96pp.
- [3] Steven Bishop, Matthew Fairbairn, Michael Norrish, Peter Sewell, Michael Smith, and Keith Wansbrough. TCP, UDP, and Sockets: rigorous and experimentally-validated behavioural specification. Volume 2: The specification. Available at <http://www.cl.cam.ac.uk/users/pes20/Netsem/alldoc.pdf>, February 2005. xxv+380pp.

We refer to the reader to Deliverable D1.4 *Study of possible semantics of failures* for further details of this work.

2. At a higher level, programs that deal with failure must often use *transactional* idioms. In this period we have proposed a system for executing transactions on top of structured peer-to-peer (P2P) networks, described in more detail in D1.2. Our system guarantees most properties usually expected from transactions, namely atomicity, independence, and consistency. The system is based

---

<sup>1</sup>(as revised 2003-07-09, at which point some material was moved to WP3)

on distributed locking, with a fully decentralized management. It avoids deadlocks and starvation by assigning priorities to transactions.

A complete technical report is available:

- 4 Valentin Mesaros, Raphaël Collet, Kevin Glynn, and Peter Van Roy. A Transactional System for Structured Overlay Network Research Report RR2005-01, Université catholique de Louvain. Available at <ftp://ftp.info.ucl.ac.be/pub/reports/2005/rr2005-01.pdf>.

#### ▷ Models of Peer-to-Peer Group Collaboration

1. Work such as the above is needed to characterise the existing failure semantics, but it is also necessary to establish practical techniques for reasoning about systems above that semantics. In particular, we aimed to make it possible to reason about *executable* descriptions of distributed algorithms, rather than idealised pseudocode or automata, in a fully formal manner. We have demonstrated that this is possible, reasoning within the Isabelle proof assistant about programs written in an executable language (a fragment both of OCaml and of our Acute language developed in WP3) above our earlier UDP/Sockets semantics. Fully-formal reasoning about any executable code is very challenging, so we have attempted only modest examples to date, not a full-fledged P2P algorithm, but the following paper addresses many of the key difficulties.

[5] Michael Compton. Stenning's protocol implemented in UDP and verified in Isabelle. In *Proceedings of The Australasian Theory Symposium, 2005*. Available at <http://www.cl.cam.ac.uk/users/pes20/Netsem/stenning.pdf>.

2. Complementing the above, we have conducted a verification of a P2P algorithm—a static overlay network based on the DKS of WP2—in a more idealised model. Structured peer-to-peer overlay networks can be seen as a class of algorithms that provide efficient message routing for distributed applications using a sparsely connected communication network. In this work, we formally verified a typical application running on a fixed set of nodes. This is intended as a foundation for studies of more dynamic systems.

We identified a value and expression language for a value-passing CCS (*Calculus for Communicating Systems*) that allows us to formally model a distributed hash table implemented over a static DKS overlay network. We then provided a specification of the lookup operation in the same language, allowing us to formally verify the correctness of the system in terms of observational equivalence between implementation and specification. For the proof, we employed an abstract notation for reachable states that allows us to work conveniently up to structural congruence, thus drastically reducing the number and shape of states to consider. The structure and techniques of the correctness proof are reusable for other overlay networks.

Recently, we found that our proof techniques can be interpreted via an extension of the so-called *Cones and Foci* framework of Fokkink et al, so this verification task also feeds nicely back into the Concurrency Theory community. We are currently working on a journal version where this feedback is explained abstractly and then applied to the static DKS algorithm, but it will not be finished during the course of the PEPITO funding period.

[6] Johannes Borgström, Uwe Nestmann, Luc Onana Alima, and Dilian Gurov. Verifying a structured peer-to-peer overlay network: The static case. In Corrado Priami and Paola Quaglia, editors, *Proceedings of Global Computing 2004*, volume 3267 of *Lecture Notes in Computer*

*Science*, pages 251–266. Springer-Verlag, 2005. Available from <http://lampwww.epfl.ch/~uwe/doc/borgstroem.etal:gc-2004.pdf>.

- [7] Johannes Borgström, Uwe Nestmann, Luc Onana Alima, and Dilian Gurov. Verifying a structured peer-to-peer overlay network: The static case. Technical Report IC/2004/76, EPFL, September 2004. Available from [http://ic2.epfl.ch/publications/documents/IC\\_TECH\\_REPORT\\_200476.pdf](http://ic2.epfl.ch/publications/documents/IC_TECH_REPORT_200476.pdf).
3. At a still higher level of abstraction, we reported in D1.8 on an analysis of the properties of peer-to-peer systems for connection-based anonymity. A journal paper on this work has been accepted for publication:
- [8] Andrei Serjantov and Peter Sewell. Passive attack analysis for connection-based anonymity systems. *International Journal of Information Security (IJIS)*, 2005. To appear.
4. Finally, two position papers have been presented drawing on the above experience and that of Workpackage 3:
- [9] James Leifer, Michael Norrish, Peter Sewell, and Keith Wansbrough. Acute and TCP: specifying and developing abstractions for global computation. In *Proceedings of the APPSEM II Workshop, Tallinn. 2pp*, April 2004. Available from <http://www.cl.cam.ac.uk/users/pes20/appsem-tallinn.ps>.
- [9] Peter Sewell and Keith Wansbrough. Applied semantics: Specifying and developing abstractions for distributed computation (grand challenge discussion paper – GC2, GC4, and GC6). Position paper for Grand Challenge meeting, Newcastle. 5pp, 2004. Available from <http://www.cl.cam.ac.uk/users/pes20/grandchallenge2004.pdf>.