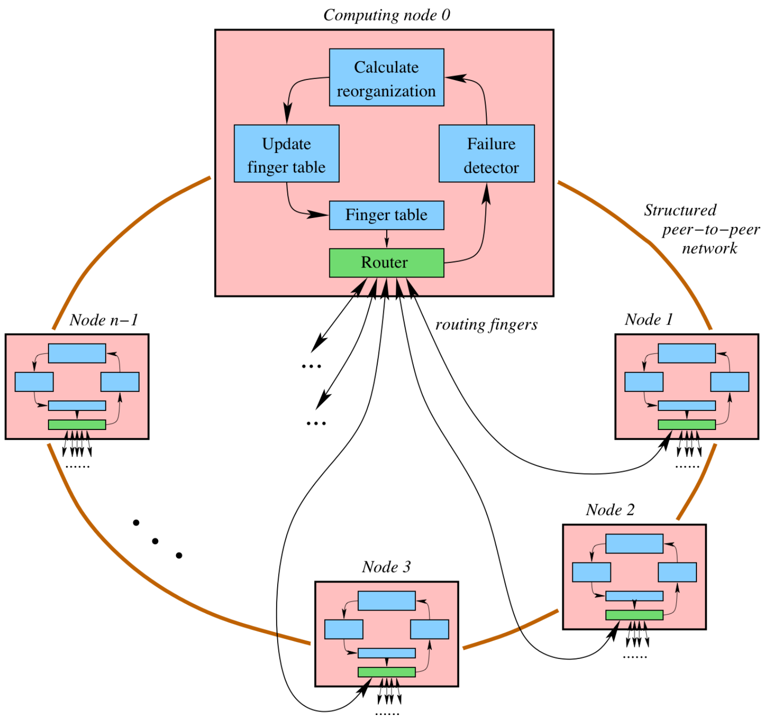

Project no. 034084
Project acronym: SELFMAN
Project title: *Self Management for Large-Scale Distributed Systems based on Structured Overlay Networks and Components*

European Sixth Framework Programme
Priority 2, Information Society Technologies

Deliverable reference number and title:	PAR Periodic Activity Report
Due date of deliverable:	July 15, 2007
Actual submission date:	July 15, 2007
Start date of project:	June 1, 2006
Duration:	36 months
Organisation name of lead contractor for this deliverable:	UCL
Revision:	1
Dissemination level:	PP

Contents

1 Executive summary **1**

2 Project objectives and major achievements **2**
2.1 Comparison with state of the art 2
2.2 Progress for each objective 3

3 Workpackage progress of the period **4**
3.1 Workpackage 1: Structured overlay network and basic mechanisms . 4
3.2 Workpackage 2: Service architecture and component model 4
3.3 Workpackage 3: Self-managing storage and transactions 5
3.4 Workpackage 4: Self-management services 6
3.5 Workpackage 5: Application requirements and evaluations 6
3.6 Workpackage 6: Management, dissemination, and exploitation . . . 7
3.6.1 Project Wiki 7
3.6.2 Advertorial in Parliament Magazine 7

4 Consortium management **9**
4.1 Consortium update 9
4.2 Coordination activities 9
4.3 Project meetings 9

5 Papers and publications **11**

A Plan for using and disseminating the knowledge **13**
A.1 Almende Summer School 13
A.2 Formal Methods for Components and Objects (FMCO 2007) 13

1 Executive summary

The SELFMAN project has just finished its first year of work addressing the challenges of building large distributed systems. The ultimate goal of SELFMAN is to solve this problem by building these systems as self-managing from the start. We are doing this by combining the technologies of structured overlay networks and advanced component models. Overlay networks already solve the self-management problem for the lower levels of the system, such as connectivity and routing. We will extend this to the higher levels, services and applications, by means of mechanisms that are available in advanced component models.

During the first year of the project, we worked on four main areas: (1) building robust structured overlay networks with the mechanisms for self management, (2) designing a programming model and component architecture that can express all the self management operations, (3) defining a transaction protocol that works over a structured overlay network, and (4) collecting relevant application scenarios.

We have now successfully built overlay networks that work for Internet-style failures and for network partitioning. We are exploring general techniques for building self managing systems, based on feedback loops and concurrent event-based architectures. We have designed a component model that allows us to express all the operations we need, such as distribution, reconfiguration, monitoring, passivation. The component model is a process calculus with a simple formal semantics, so there will be no unpleasant surprises in its behavior. We have designed an atomic commit protocol for implementing transactions on top of an overlay network, based on a version of the Paxos consensus protocol.

Finally, we have collected three application scenarios from our industrial partners to ensure the relevance of our work. All three applications address real industry needs. The first is a machine-to-machine data transportation application that ensures reliable connectivity between end users and data providers, with a distributed infrastructure. The second is a distributed database system that can host a distributed version of a wikipedia, which can be accessed and updated in distributed fashion. The third is a peer-to-peer video streaming application that can be used for large-scale content distribution applications.

In the second year of the project, we will build the basic self-management services that will enable us to build these applications. We will also complete our work on the component model so that it is ready to support the applications. We will then start building full-fledged self-managing applications based on the scenarios of our industrial partners.

2 Project objectives and major achievements

The objective of the SELFMAN project is to address directly the challenge of building large distributed systems. “Abnormal” events such as local failures become normal and even common. We will address this challenge by building self managing applications. These applications will be able to reconfigure themselves to handle changes without human intervention but according to management policies. SELFMAN will achieve this by combining two advanced software technologies in which Europe is playing a major role: peer-to-peer networks and software components. This will enable the development of many more Internet applications and Internet-based companies. We will build several self-managing applications according to industrial needs and we will develop a set of tools and guidelines for building self-managing applications.

During the first year of the project, we worked on four main areas: (1) building robust structured overlay networks with the mechanisms for self management, (2) designing a programming model and component architecture that can express all the self management operations, (3) defining a transaction protocol that works over a structured overlay network, and (4) collecting relevant application scenarios. We first compare our progress in these areas with the state of the art. Then we explain in more detail what we have done. For full information, please look at the deliverables themselves.

2.1 Comparison with state of the art

Detailed comparisons of this work to the state of the art are given in the relevant deliverables. We summarize this work here.

- Regarding the structured overlay networks (SON), we have extended the state of the art in two directions that have previously been unsolved in practice: network partitioning and imperfect failure detection. We show how a SON can heal itself in the case of network partitioning. We show how a SON can maintain its structure in the case of imperfect failure detection. Both of these properties are essential for the practical use of SON in applications. It was important that we solve both problems quickly in the project.

- Regarding the programming and component models for self managing systems, we have extended the state of the art in two directions. First, we have focused attention on self management as a general approach for soft design. Other work in this area is limited to addressing specific problems. Second, we have designed an advanced component model that can express all the operations we need for self management (monitoring, reconfiguration, passivation, distribution, etc.). This component model is a process calculus, Oz/K, that is carefully designed to show the important concepts.

- Regarding the transaction protocol, we have designed an atomic commitment protocol that works correctly over a structured overlay network with replicated data. It uses a modified version of the Paxos consensus protocol. This is the first application of this protocol to a structured overlay network that we know of.

- Regarding the application scenarios, we have collected scenarios from the industrial partners regarding applications that are important for their business. This will ensure the relevance of our solutions.

2.2 Progress for each objective

For objective (1), we have designed two SONs: DKS: a Java application that uses synchronous failure detection and locking, and P2PS: an Oz application that uses asynchronous failure detection and no locking. We have working prototypes of both DKS and P2PS. We have extended DKS to handle network partitioning and extended P2PS to handle imperfect failure detection. Network partitioning and imperfect failure detection are common on the Internet; it was imperative that we find solutions for them.

For objective (2), we have worked in two main directions: first, the computation model, and second, how to program self-managing systems:

- We designed the Oz/K computation model, which is a process calculus that provides a powerful and flexible way to compute with components as first-class entities. Oz/K exists on top of the Oz kernel language, a powerful and flexible process calculus that is designed to be used by programmers. We expect that Oz/K is too rich for the purposes of SELFMAN; we intend to implement the subset of this model that we need for self management.

- We started an exploration on how to program with feedback loops. Self-managing systems consist of a large set of interacting feedback loops. We must learn how to program these systems so that their global properties can be designed in easily, without complicated analyses. As part of this work, we have reformulated the SON algorithms in terms of interacting feedback loops.

For objective (3), we designed an atomic commit algorithm that uses the Paxos uniform consensus algorithm to do commit in the case of majority correctness. This algorithm will work if used on replicated data (the majority of the replicas must be correct) with Internet-style failures. The algorithm requires no failure detector.

For objective (4), we gathered four realistic application scenarios, according to the real needs of four partners: three industrial partners France Telecom, PeerTV, and Bull, and ZIB. These scenarios illustrate quite a variety of applications. Later in the project we will decide which of these scenarios we will implement. The main problem was that E-Plus left the project in Fall 2006. As a result, we decided that we should not be limited to a single application scenario.

3 Workpackage progress of the period

3.1 Workpackage 1: Structured overlay network and basic mechanisms

Objectives The first-year objective is to build a structured overlay network that is a foundation for the rest of the project and that has the basic primitives and hooks for implementing self management.

Progress UCL built a "relaxed ring" version of P2PS that maintains connectivity despite Internet-style failures. KTH built a version of DKS that is able to tolerate network partitioning and merge together again when the partition disappears. ZIB explored a different topology for structured overlay networks in its SONAR ring, which uses a multi-dimensional torus. We plan to use the P2PS and DKS rings in the continuation of the project. The SONAR ring is important for understanding skewed distributions, which may occur in the application scenarios that we will implement. NUS made an extensive literature study of structured overlay networks concentrating on the security issues. NUS has started working on monitoring mechanisms (including continuous monitoring), which are needed for self tuning, self configuration, and self protection.

Deviations and corrective actions UCL needed more man-power than expected to build its robust ring, but we needed urgently to solve this problem since the robust ring is essential for the rest of the project. As corrective action, UCL will compensate by using a university-funded teaching assistant to do part of the work in the last two years of the project. ZIB is getting more involved in workpackage 1 and will move 10 pm to workpackage 1. ZIB will reduce its involvement in workpackage 2 and workpackage 4 as a result. NUS had delays hiring researchers for its security work. This delayed the work itself; the other partners compensated by organizing a workshop on Collective Intelligence to start the design of the security architecture.

Deliverables D1.1 (Report on low-level self-management primitives for structured overlay networks), D1.3a (First report on security in structured overlay networks).

Milestones M12 (Understand how to do structured overlay network with component model). This milestone is not yet completely achieved. We have successfully implemented the overlay networks that will be used in the rest of the project, but we have not yet completely reformulated them using a component model.

3.2 Workpackage 2: Service architecture and component model

Objectives To build a distributed component architecture that has the basic primitives needed for self management.

Progress Progress has focused on how to program self-managing systems, their component-based event-driven architecture, and the formal calculi that express their behavior. UCL has studied how to program self-managing systems, as graphs of interacting feedback loops, and applied this to the construction of structured overlay networks. The relaxed ring architecture is formulated in this way. UCL has also designed a new failure-handling model for Oz that is asynchronous and implemented it in the Mozart system. INRIA has defined the Oz/K process calculus in which distribution and components can be expressed in a uniform way in the Oz kernel language. KTH has developed a component-based event-driven architecture for building distributed applications. This architecture uses the asynchronous formalism that KTH presented in the mini-course it organized. FT R&D has worked on the Fractal component model, adding key functionality for handling components as first-class entities.

Deviations and corrective actions UCL needed more man-power than expected to study programming with feedback loops, to build the relaxed ring, and to build the new failure-handling model. We will solve the man-power problem in the same way as for workpackage 1, with which this work is closely connected, ZIB will reduce its involvement in this workpackage to get more involved in workpackage 1.

Deliverables D2.1a (Report on basic computation model), D2.2a (Report on architectural framework specification), D2.3a (Report on formal operational semantics).

Milestones M12 (Understand architectural framework with component model using SON). This milestone is achieved.

3.3 Workpackage 3: Self-managing storage and transactions

Objectives To design and build a self-managing storage service that provides replication and the ability to perform transactions. In the first year, this objective is to have a formal model for transactions over a structured overlay network.

Progress ZIB and KTH have designed a transaction protocol for implementing transactions on a structured overlay network with replication and partial failure. The protocol is based on a Paxos consensus algorithm.

Deviations and corrective actions No deviations.

Deliverable : D3.1a (First report on formal models for transactions over structured overlay networks).

Milestones M12 (Understand how to do transactions over structured overlay network). This milestone is achieved.

3.4 Workpackage 4: Self-management services

Objectives To design and implement the self-management services needed by the applications we will develop.

Progress No work was done for this workpackage during the first period. The purpose of this workpackage is to design and implement the self-management services needed for our applications. This work will begin in the second year, after we have the component model and the structured overlay network that goes with it.

Deviations and corrective actions ZIB will reduce its future involvement in this workpackage to get more involved in workpackage 1.

Deliverables No deliverables at M12.

Milestones No milestone at M12.

3.5 Workpackage 5: Application requirements and evaluations

Objectives To build the applications according to the application scenarios that are interesting for our industrial partners.

Progress This deliverable presents three application scenarios: a M2M use case (FT R&D), a distributed database use case (ZIB), and a peer-to-peer content distribution use case (KTH, for the new partner Stakk AB). A fourth use case regarding a J2EE application server proposed by Bull could not be integrated in the time frame of the deliverable, but will be available later. In the second year of the project, we will decide which of these scenarios will be implemented. Instead of a multi-tier application server as explained in the DoW, our approach will let us be more in line with the actual needs of industry.

Deviations and corrective actions The industrial partner E-Plus left the project; there is no application scenario for them. Instead, we have contacted Bull (France, through FT R&D) and Stakk AB (Sweden, through KTH) who will join the project in the Fall. Currently, they will provide us with application scenarios (of which one is added to this deliverable).

Deliverable D5.1 (Report on user requirements for application servers)

Milestones M12 (Understand requirements for applications to be developed). This milestone was achieved.

3.6 Workpackage 6: Management, dissemination, and exploitation

Objectives To manage the project, disseminate results, and collaborate with other projects.

Progress We have created a project website, `www.ist-selfman.org`, as a Wiki, a project mailing list, `selfman-tech@googlegroups.com`, and a project document repository in SVN, `https://savane.info.ucl.ac.be/svn/selfman`. We have organized five project meetings, to enable collaboration between partners and to create a common technical language between them. We have placed an advertorial in the special issue of Parliament magazine on ICT. More information on this work is given below.

Deviations and corrective actions We did not organize a project workshop in which third parties would be invited. Rather, we spent our effort in improving the collaboration between the partners by organizing five project meetings (see Section 4.3), of which two were mini-courses (on reliable distributed programming and on collective intelligence) given by experts. We will also participate in the Almende Summer School (giving a lecture, see Section A.1) and in the Formal Methods for Components and Objects workshop (giving a session, see Section A.2).

Deliverables D6.1a (Project website and Wiki), D6.1b (First project workshop), D6.5a (First progress and assessment report with lessons learned).

Milestones M12 (Project making good progress (all deadlines respected)). This milestone is achieved, with minor changes to deadlines.

3.6.1 Project Wiki

The project has a Wiki at `www.ist-selfman.org`. Creating this Wiki turned out to be a good idea. It provides a well-presented and structured website. Partners can update it by adding pages and uploading files on their own. This has worked well.

We had one problem in the Wiki, that of "robot pirates". This consisted of pirate activity that corrupted the WIki. We solved the problem by requiring a simple login procedure to modify the Wiki. In addition, part of the Wiki contains internal project documents and is only readable to project partners, again using a simple login procedure.

3.6.2 Advertorial in Parliament Magazine

We published a half-page advertorial in the Nov. 13, 2006 issue of Parliament Magazine to publicize the goals of the SELFMAN project. The publication was done at a special rate for European project. This magazine is distributed fortnightly to members of Parliament and their associates. This was an ICT special issue devoted to ICT, which was also handed out to the participants of the IST 2006 conferenct in Helsinki. The following text was published:

> The next major challenge in Internet applications has arrived. The Internet is now ready to support large-scale applications because of its increased scale, maturity, and bandwidth. But the task of developing and maintaining these applications is overwhelming and is slowing down their development. "Abnormal" events such as software updates, failures, security threats, and performance hotspots are becoming normal and even frequent.
>
> The SELFMAN project is tackling this challenge head on by making self managing applications. These applications will be able to reconfigure themselves to handle changes without human intervention but according to management policies. SELFMAN will achieve this by combining two advanced software technologies in which Europe is playing a major role: peer-to-peer networks and software components. This will enable the development of many more Internet applications and Internet-based companies.
>
> SELFMAN will support open source software development by contributing to two European open-source consortia: the ObjectWeb Consortium (a major developer of industrial-quality distributed middleware) and the Mozart Consortium (supporting an advanced programming platform being used in research and education worldwide).

4 Consortium management

4.1 Consortium update

The partner E-Plus left the project soon after its beginning, due to an internal reorganization. We have found two new industrial partners to replace E-Plus: Stakk AB (formerly PeerTV) (Sweden) and Bull (France). These industrial partners, including also partner France Telecom, will provide us with the application scenarios that we will implement in the project.

We also had a minor problem with NUS, which had a 6-month delay to find researchers to fund its security work. This has slightly slowed that work, which the other partners compensated by proactively organizing a mini-course on Collective Intelligence (see below).

4.2 Coordination activities

We have the following coordination activities at the level of the SELFMAN project (in addition to local collaborations done by individual partners):

- We started collaboration with the Grid4All project. We held both projects' kickoff meetings back to back.
- We are collaborating through the CoreGRID Network of Excellence, which provides an environment for discussion and exchanging information.
- Several project partners are part of the EVERGROW Integrated Project. This has provided an excellent starting environment for SELFMAN (we can say that without this environment, SELFMAN would probably not have existed).
- We will participate in the Almende Summer School on self organization on Aug. 26-28, 2007 (see Appendix A).
- We will participate in the FMCO 2007 coordination activity on Oct. 24-26, 2007, with the projects Artist, Sensoria, Modelware, and Credo (see Appendeix A).

4.3 Project meetings

We had a series of project meetings at each of the major partners' sites to jump-start the project. The detailed programs of these meetings, along with much presentation material, are available on the project Wiki at `www.ist-selfman.org`.

- SELFMAN Kickoff Workshop, KTH in Stockholm, June 8-9, 2006. Participants: UCL (Peter Van Roy, Raphaël Collet, Yves Jaradin), INRIA (Jean-Bernard Stefani, Noel De Palma), KTH (Seif Haridi, Per Brand, Vlad Vlassov, Konstantin Popov), ZIB (Alexander Reinefeld, Thorsten Schütt, NUS (Roland Yap).

- SELFMAN project meeting, UCL at Louvain-la-Neuve, September 26-27, 2006. During this meeting much of the collaborative work on the initial tasks started.
- Mini-course on reliable distributed programming, given by Seif Haridi at ZIB in Berlin, October 12-13, 2006. This course was based on the book "Introduction to Reliable Distributed Programming" by Rachid Guerraoui and Luis Rodrigues (Springer, 2006).
- Project meeting, at Grenoble at INRIA and France Telecom, November 20-21, 2006 This meeting discussed possible application scenarios of France Telecom based on the ODIS platform for M2M applications and the CLIF load injection framework. It also discussed INRIA's Fractal component model for self management and the formal basis of this component model.
- Collective intelligence: a mini-course on systems with collective behavior, given by Mohammed El-Beltagy of Optomatica at ZIB in Berlin, February 15-16, 2007. This course covered game theory, agoric systems, and the theory of collectives, and its implication for SELFMAN. The page

  `www.natural-computation.com/selfman`

  contains the course material.

5 Papers and publications

The following papers were published during the first year of the project. Most of these papers are available as appendices to the deliverables.

- C. Arad, R. Roverso, A. Ghodsi, and S. Haridi. *Middleware for Building Internet-scale, Dynamic, Distributed Applications*. KTH Technical Report, June 2007.

- E. Bruneton, T. Coupaye, M. Leclerc, V. Quema, and J-B. Stefani. *An Open Component Model and Its Support in Java*. Published in Software Practice & Experience Journal–Special Issue on Auto-adaptive and Reconfigurable Systems, 36(11-12), 2006.

- Raphaël Collet. *The Limits of Network Transparency in a Distributed Programming Language*. Ph. D. dissertation, Dept. of Computing Science and Engineering, Université catholique de Louvain (Expected completion Nov. 2007).

- Raphaël Collet and Peter Van Roy. *Failure Handling in a Network-Transparent Distributed Programming Language*. In Advanced Topics in Exception Handling Techniques, volume 4119 of Lecture Notes in Computer Science. Springer, 2006.

- T. Coupaye and J-B. Stefani. *Fractal Component-Based Software Engineering - Report of Fractal CBSE Workshop at ECOOP'06*. Published in 20th European Conference on Object-Oriented Programming (ECOOP 2006) Workshop Reader, LNCS 4379, 2007.

- Donatien Grolaux, Boris Mejías, and Peter Van Roy. *PEPINO: PEer-to-Peer network INspectOr*. In *proceedings of The Seventh IEEE International Conference on Peer-to-Peer Computing* (to appear).

- M. Kessis, P. Déchamboux, C. Roncancio, T. Coupaye, and A. Lefebvre. *Towards a Flexible Middleware for Autonomous Integrated Management Applications*. In the International Multi-Conference on Computing in the Global Information Technology (ICCGI'06), August 2006.

- M. Leger, T. Coupaye, and T. Ledoux. *Reliability of Dynamic Reconfigurations in Component-Based Systems*. France Telecom Technical Report, February 2007.

- M. Lienhardt, A. Schmitt, and J.B. Stefani. *Oz/K: A Kernel Language for Component-Based Open Programming*. Technical Report RR-6202, Institut National de Recherche en Informatique et Automatique (INRIA), France, 2007.

- M. Lienhardt, A. Schmitt, and J.B. Stefani. *Oz/K: A Kernel Language for Component-Based Open Programming*. In 6th ACM International Conference on Generative Programming and Component Engineering (GPCE). ACM Press, 2007 (to appear).

- Boris Mejías, Donatien Grolaux, and Peter Van Roy. *Improving the Peer-to-Peer Ring for Building Fault-Tolerant Grids*. In CoreGRID Workshop on Grid-* and P2P-*, Heraklion, Greece, June 2007.

- Boris Mejías and Jorge Vallejos. *Implementing Self-Adaptability in Context-Aware Systems*. In Workshop on Multiparadigm Programming with Object-Oriented Languages, part of ECOOP 2007, July 31, 2007.

- Boris Mejías and Peter Van Roy. *A Relaxed Ring for Self-Organising and Fault-Tolerant Peer-to-Peer Networks*. In XXVI International Conference of the Chilean Computer Science Society (SCCC 2007), Nov. 2007.

- Monika Moser and Seif Haridi. *Atomic Commitment in Transactional DHTs*. In CoreGRID Workshop on Grid-* and P2P-*, Heraklion, Greece, June 2007.

- S. Plantikow, A. Reinefeld, and F. Schintke. *Distributed Wikis on Structured Overlays*, In CoreGRID Workshop on Grid-* and P2P-*, Heraklion, Greece, June 2007.

- Rajiv Ramnath, Sufatrio, and Roland H. C. Yap. *WinResMon: A Tool for Discovering Software Dependencies, Configuration, and Requirements in Microsoft Windows*. In LISA'06: 20th Large Installation System Administration Conference, 2006, pp. 175–186.

- Thorsten Schütt, Florian Schintke, and Alexander Reinefeld. *A Structured Overlay for Multi-Dimensional Range Queries*. In Euro-Par 2007 (to appear).

- Tallat M. Shafaat, Ali Ghodsi, and Seif Haridi. *Handling Network Partitions and Mergers in Structured Overlay Networks*. In The Seventh IEEE International Conference on Peer-to-Peer Computing (to appear).

- Peter Van Roy. *Self Management and the Future of Software Design*. In Third International Workshop on Formal Aspects of Component Software (FACS '06), Springer ENTCS 182, Sept. 2006.

The following papers were submitted for publication during the first year of the project.

- A. Diaconescu and B. Dillenseger. *Composite Probes: a Generic Monitoring Framework for Hiearchical Management of Heterogeneous Data*. Submitted for publication by France Telecom in April 2007.

- A. Harbaoui, B. Dillenseger, and J.-M. Vincent. *Performance Characterization of Black Boxes with Self-Controlled load injection for Simulation-based Sizing*. Submitted for publication by France Telecom in May 2007.

- N. Jayaprakash, T. Coupaye, C. Collet, P.-C. David. *Flexible Reactive Capabilities in Component-Based Autonomic Systems*. Submitted for publication by France Telecom in May 2007.

A Plan for using and disseminating the knowledge

We have done the following actions for using and disseminating knowledge in SELFMAN.

- The SELFMAN Wiki at `www.ist-selfman.org` has a Google PageRank of 5/10, which is quite good for a technical site. The ObjectWeb Consortium site (`www.objectweb.org`) and the Mozart Consortium site (`www.mozart-oz.org`) both have a PageRank of 7/10.
- Peter Van Roy gave the talk *Self Management and the Future of Software Design*, on programming self-managing systems with feedback loops, invited by Jos Fernandez-Villacanas at the European Commission, Directorate General INFSO, on Oct. 25, 2006.
- SELFMAN participates in the ObjectWeb Consortium and the Mozart Consortium, two European consortia that release open-source software. In this regard, Boris Mejias gave a lightning talk on the Mozart system at the open-source conference FOSDEM 2007 on Feb. 25, 2007.
- SELFMAN will participate in two events, Almende Summer School and FMCO 2007, which are described below.

A.1 Almende Summer School

SELFMAN will participate in the Almende Summer School to be held Aug. 28-30, 2007, in Rotterdam, NL. P. Van Roy is invited lecturer. The summer school offers graduate students, PhD students and postdoctoral fellows in mathematics, physics and social-technical sciences an intensive three-day course of lectures and workshops on self-organizing networks. It is open to all interested participants. Its Web site is `summerschool.almende.com`.

The summer school is organized by Almende BV (`www.almende.com`), a company localized in Rotterdam, NL, and focused on self organization, adaptive company managements and innovation. It provides assistance for helping organizations become self organizing and self managing, by maximizing the synergy between people and software. The company delivers intelligent multi-agent systems for all aspects of organizations (including in particular marketing and primary processes).

A.2 Formal Methods for Components and Objects (FMCO 2007)

The FMCO series of workshops brings together researchers in formal methods, components, and object technology. SELFMAN will participate in FMCO 2007, to be held Oct. 24-26, 2007. This event will be more substantial than previous events. In addition to the refereed papers, it will invite the following projects:

- Artist (Cluster Real-Time Components).

  `www.artist-embedded.org/artist`

- Selfman (Self Management for Large-Scale Distributed Systems based on Structured Overlay Networks and Components).

  `www.ist-selfman.org`

- Sensoria (Software Engineering for Service-Oriented Overlay Computers).

  `sensoria.fast.de`

- Modelware (Model Driven Development).

  `www.modelware-ist.org/index.php`

- Credo (Modeling and analysis of evolutionary structures for distributed services).

  `credo.cwi.nl`

SELFMAN will give four talks at this event:

- *Handling imperfect failure detection in a structured overlay network: the relaxed ring approach*, by Boris Mejias, Donatien Grolaux, Peter Van Roy.
- *Handling network partitioning in a structured overlay network*, by Tallat Shafaat, Ali Ghodsi, Seif Haridi.
- *Atomic commitment in a transactional DHT built on a structured overlay network*, by Monika Moser, Seif Haridi.
- *Oz/K: A kernel language for component-based open programming*, by Michael Lienhardt, Alan Schmitt, Jean-Bernard Stefani.