Software Technologies

SELFMAN: Self management of large-scale Internet applications

THE ADVENTURES OF SELFMAN

As Internet applications become larger and more complex, the task of managing them becomes overwhelming. "Abnormal" events such as software updates, faults, attacks, and hotspots become frequent. SELFMAN will show how to handle these events automatically by making the application self managing.

At A Glance: *SELFMAN*

Self Management for Large-Scale Distributed Systems based on Structured Overlay Networks and Components

Project Coordinator

Peter Van Roy
Université catholique de Louvain (UCL)
Tel: +32 (10) 47 31 50
Fax: +32 (10) 45 03 45
pvr@info.ucl.ac.be
http://www.ist-selfman.org

Partners: Université catholique de Louvain (BE), Kungliga Tekniska Högskolan (SE), INRIA (FR), France Telecom (FR), Zuse Institut Berlin (DE), E-Plus Mobilfunk (DE), National University of Singapore (SG)

Duration: 6, 2006 – 5, 2009

Project funding (EC/tot.): €1.96m/€2.57m

Further Information

- **IST Research:** ***Software Technologies***
  DG Information Society & Media
  Unit D3
  Tel: *+32 (2) 29 54 308*
  Fax: *+32 (2) 29 67 018*
  infso-st@cec.eu.int
  http://cordis.europa.eu/ist/st/
- **Europe's Information Society: Thematic Portal:**
  http://europa.eu.int/information_society/

More and more applications are written to take advantage of the Internet. The reliability and bandwidth of the Internet have become good enough to support them. The result is that application complexity is increasing rapidly. Furthermore, many small devices (cellphones, PDAs, etc.) are joining the Internet, adding to the problem. SELFMAN will solve this complexity problem by making applications self managing. This will enable the development of many more Internet applications and Internet-based companies. As proof of concept, SELFMAN will build a scalable self-managing three-tier business application.

Approach and Expertise

SELFMAN will provide self management by combining two advanced technologies in which Europe has played a major role, namely structured overlay networks and component models. Structured overlay networks developed out of peer-to-peer and provide scalability, communication guarantees, and efficiency. Component models provide the framework to extend the self-managing properties of structured overlay networks over the whole application.

SELFMAN will build a large-scale three-tier application with a self-managing database and evaluate it using real trace data

SELFMAN partners have technological expertise and know-how in structured overlay networks (KTH, UCL), component models (INRIA, France Telecom, KTH, UCL), cluster management (INRIA, ZIB), storage services (ZIB, INRIA), large-scale distributed telecom applications (France Telecom, E-Plus Mobilfunk), and security (NUS, KTH). SELFMAN will encourage the development of Open Source software in the ObjectWeb Consortium and the Mozart Consortium.

Self Management in Software Development

A major goal of SELFMAN is to show how to program large-scale software systems with a desired global behavior (such as stability, performance, fault tolerance, and security). We will take ideas from system theory and apply them to software development. We consider self management as a general technique that can be applied to all of software development and not just to reducing human management costs.

Self Management

The goal of SELFMAN is to make large-scale distributed applications that are *self managing*. Self management, as a general concept, means that the system should be able to reconfigure itself to handle changes in its environment or requirements without human intervention but according to high-level management policies. Management is one of the key obstacles to deploying large-scale applications running on networks such as the Internet or company intranets. Currently many specialized personnel are needed to keep large Internet applications running. SELFMAN will contribute to removing this obstacle and thus enable the development of many more Internet applications and Internet-based companies that depend on such applications. In the context of SELFMAN, we define self management along four axes:

- *self configuration:* systems configure themselves according to high-level management policies,
- *self healing:* systems automatically handle faults and repair them,
- *self tuning:* systems continuously monitor their performance and adjust their behavior to optimize resource usage and meet service level agreements, and
- *self protection:* systems protect themselves against security attacks.

Combining Peer-to-Peer with Components

We will provide self management by combining a component model with a structured overlay network. The component model will support dynamic configuration, the ability of part of the system to reconfigure other parts at run-time, which is the key property that underlies the self-management abilities. Basing the system on a structured overlay network lets SELFMAN extend the abilities of the component model to large-scale distributed systems. Structured overlay networks have made much progress since their origins in peer-to-peer file-sharing applications. In contrast to these applications, structured overlay networks provide guarantees for efficient communication and reorganization in case of failures. These are already low-level self-management properties. Combining this with the component model, SELFMAN will build high-level self-management properties on top of these low-level properties.

Both Foundational and Applied

SELFMAN will do both foundational and applied research. The foundational research will design a distributed service architecture that combines structured overlay networks (for communication and basic self management) with component models (for the higher self-management primitives). We will do implementation work in two directions: first, to explore how an industrial standard platform (J2EE) can be made self managing, and second, to push self management as far as we can, in terms of fundamental programming language research (Mozart), without being restrained by existing tools. The interplay between these two implementations will benefit both.

Three-Tier Application Scenario

SELFMAN is organized around a key application scenario, namely a three-tier distributed business application. The application will be built on top of a self managing distributed storage service that will be developed in the project. We will use industrial trace data to measure the effectiveness of our self managing architecture. The industrial partners will use the results of SELFMAN to guide their strategic decisions for distributed systems development.

Open Source Dissemination

SELFMAN will disseminate its results through publication, Web publication (e.g., Wikis and blogs), and public events (e.g., workshops). In addition, SELFMAN will disseminate its software as open source whenever possible. In this context, the goals of SELFMAN are to achieve the first of the three phases of open source development, namely seed software of sufficient quality, and to initiate the second phase, namely community building. SELFMAN partners are major players in two European open source consortia, namely the ObjectWeb consortium (http://www.objectweb.org, hosted by INRIA) and the Mozart consortium (http://www.mozart-oz.org, hosted by UCL). The third phase, active community, exists for these two consortia and we plan to leverage these communities. One of the flagship products of ObjectWeb is the JOnAS J2EE application server. The results of SELFMAN will directly benefit the JOnAS code base. The Mozart consortium focuses on the development of the Mozart Programming System, a multi-paradigm distributed programming platform that supports rapid prototyping of advanced applications. SELFMAN will result in novel abilities for Mozart.