INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Sardes

System Architecture for Reflective Distributed Computing Environments

Rhône-Alpes

THEME 1A

Activity Report

2003

Table of contents

1. Team

Sardes is a project of Inria Rhône-Alpes and a research team of Imag-LSR (Software, Systems and Networks Laboratory), a joint research unit (UMR 5526) of Centre National de la Recherche Scientifique, Institut National Polytechnique de Grenoble and université Joseph Fourier.

Head of Project-team
Jean-Bernard Stefani [Inria Research Director, on leave from *corps des Télécommunications*]

Administrative Assistant
Valérie Gardès

Inria Personnel
Emmanuel Cecchet [Research Scientist]
Daniel Hagimont [Research Scientist]

Faculty
Fabienne Boyer [associate professor, université Joseph Fourier]
Noël De Palma [associate professor, Institut National Polytechnique de Grenoble]
Sébastien Jean [associate professor, université Pierre Mendès France]
Gilles Kuntz [associate professor, université Joseph Fourier]
Sacha Krakowiak [professor, université Joseph Fourier]
Jacques Mossière [professor, Institut National Polytechnique de Grenoble]

Visiting Scientist
Slim Ben Atallah [École Nationale des Sciences de l'Informatique de Tunis, contract]

Post-doctoral Fellow
Marek Procházka [contract, till 30/9/2003]

Research Associates
Jean-Frédéric Mesnil [till 30/9/2003]
Matthieu Leclercq [from 1/4/2003]
Nicolas Modrzyk [from 1/9/2003]

Ph.D. Students
Takoua Abdellatif [industrial grant (Bull), from 1/10/2003]
Philippe Bidinger [Inria grant (contract)]
Olivier Charra [government grant, till 1/8/2003]
Renaud Lachaize [government grant]
Philippe Laumay [industrial grant (Schlumberger), till 30/11/2003]
Oussama Layaida [Inria grant (contract)]
Simon Nieuviarts [industrial grant (Bull), till 31/10/2003]
Erdem Ozcan [industrial grant (ST Microelectronics), from 1/11/2003]
Vivien Quéma [government grant]
Christophe Rippert [government grant, till 15/10/2003]
Aline Senart [INPG grant (contract), till 15/11/2003]
Vania Marangozova [half-time teaching assistant, till 30/9/2003]

2. Overall Objectives

2.1.1. Objectives.

The objective of the Sardes project is to develop concepts, methods, and tools to build distributed systems (including both infrastructures and applications) that are open, evolvable, and safe. The designers and developers of such systems are facing important challenges, among which scalability, administration, and quality of service. To respond to these challenges, the infrastructures and applications need to be *adaptable*. This may be achieved through reflexivity and component-based design.

The project has three main ambitions:

1. To identify architectural paradigms for the various aspects of adaptable distributed systems (naming, communications, resource management, fault tolerance, mobility); to elaborate the relevant design patterns and software frameworks, as well as programming and meta-programming models.
2. To develop software engineering techniques to build adaptable distributed systems, specially through the use of configurable components, reflexivity, and appropriate construction techniques (e.g. meta-programming and dynamic compilation).
3. To apply the concepts and techniques developed in the project to the prototyping of adaptable software infrastructures; to validate these concepts and techniques in various environments (clusters, large scale networks, embedded systems), and for various application domains (data management, multimedia, real time), with emphasis on scalability, administrability, and quality of service.

2.1.2. Organization and Collaborations.

The project is organized around three main activity domains: models and tools for distributed components; distributed resource management; applications. The distributed resource management theme is taking an increasing importance. In October 2003, we have started a new project-wide activity whose aim is to design and implement an autonomic resource management system on a cluster, in order to ensure prescribed levels of availability and quality of service.

Sardes is involved in several industrial and international collaborations. It is active in the OBJECTWEB consortium (8.2) dedicated to open source middleware. It is a partner of several European projects and networks: Mikado, Ozone, Midas, CaberNet (IST program) and Osmose (ITEA program). It participates in several projects funded by the national research network on software technologies (RNTL): Arcad, Impact, Inside. It collaborates with several industrial partners: Bull, France Telecom, Microsoft, ST MicroElectronics, and has close links with Scalagent, a technology startup created by former members of the project.

3. Scientific Foundations

3.1. Introduction

In this section, we first present the main challenges that face the designers of large scale distributed systems. We then discuss recent advances and open problems in the two main areas covered by Sardes: architectures for adaptable distributed systems, and distributed system administration.

3.2. Challenges of Distributed Systems

The future[1] of information processing applications is envisioned as a range of environments in which processors will be everywhere and will be interconnected by an array of networks, from ad-hoc to the global Internet. Constructing the software base - the *middleware* - for such ubiquitous computing infrastructures poses a number of scientific and technical challenges, which arise from the wide variety of applications and services that need to be supported.

Software will run in a multitude of computing environments ranging from traditional desktops to multiple host, networked systems, to mobile systems, to embedded systems, to wearable computers. Software systems will need to be "aware" of their physical surroundings, taking input from real-world sensors and sending output intended to control the real-world environment. They will need to "understand" their proximity to other computing resources and use this information to guide their behaviour. A number of application scenarios illustrating these new environments have been discussed in a recent report for the IST European program [60].

[1] The text of 3.2 has been adapted from a Research Roadmap Report produced by the Midas IST project (8.4.3), in which Sardes participates.

A fundamental requirement for such environments is for middleware architectures capable of supporting services that are *composable* and *adaptable*. We have singled out four major challenges: scalability, quality of service, manageability and programmability.

1. *Scalability* concerns the ability to scale in several dimensions; scaling in machine forms: from smart labels to server farms to metacomputing network overlays; scaling in numbers: objects, machines, users, locations; scaling in logical and organizational structures: from ad-hoc collaborations networks to federations of multi-domain enterprises.
2. *Quality of service (QoS)* concerns the ability to obtain correctness and service-level guarantees such as timeliness, availability, fault-tolerance, survivability for applications executing in large scale environments. The problem of meeting QoS requirements of applications is made harder in a ubiquitous computing environment where new services and customized services are expected to be added into (existing) applications at an alarming rate.
3. *Manageability* concerns the ability to monitor and to control the operation and evolution of large scale, long-lived distributed applications and services, avoiding manual intervention and centralization (i.e., assumption of one management domain). Distributed applications and services will need to be reconfigured dynamically, for example, to maintain user specified QoS guarantees despite changes in operating conditions (e.g., component failures). Mechanisms are needed to dynamically add, extend, remove or move component services in a dependable and predictable manner.
4. *Programmability* concerns the ability to compose new applications from existing applications and services, to deploy and to maintain them in highly dynamic and heterogeneous computing environments; further, applications are expected to be highly parallel, requiring high-level abstractions necessary for dealing with complexity, with fine-grained resource awareness built according to the end-to-end principle.

Middleware for a ubiquitous computing environment will by necessity be open and standardized, at least at the level of communication protocols, key application programming interfaces and language tools. Furthermore, such middleware will become an economic commodity, i.e. be available at very low or no cost in all manners of equipment and networks to serve as a viable basis for value-added services and applications. In this context, it makes both technical and economical sense to pursue the development of such middleware facilities using an open source model.

In the next two sections, we discuss specific aspects relevant to the current research interests of Sardes.

3.3. Architectures for Adaptable Distributed Systems

A first requirement for adaptable middleware is a modular architecture, in which the interfaces and dependencies between the parts should be clearly identified. The main motivations are to minimize the adaptation induced changes by defining components as adaptation localities, and to use structural modifications (e.g. reconfiguration) as a tool for adaptation. Thus research on adaptable middleware is essentially done in the context of component-based systems. This is the theme of Section 3.3.1.

Models, patterns and frameworks are the basic ingredients of an architectural approach to the design of complex software systems. These aspects are presented in Section 3.3.2.

3.3.1. Adaptable Component-based Middleware

Building an application out of composable parts is an old challenge of software engineering. Despite the apparent simplicity of its requirements, progress towards this goal has been slow. The notions related to components have only been elicited recently [69]. Although commercial infrastructures based on components are now available (e.g. COM+, CCM, EJB, .Net), the rigorous definition of a component model and of its architectural invariants is still an open issue.

Three main approaches are being used to achieve adaptability and separation of concerns in middleware systems: meta-object protocols, aspect-oriented programming, and pragmatic techniques.

Meta-object protocols [63] are the basic device of reflection. A reflective system is one that is able to answer questions about itself and to modify its own behavior, by providing a causally connected representation of itself to a meta-level. In a component infrastructure, the interface available at the meta-level allows the operations related to component lifecycle and composition to be dynamically adapted. A number of research prototypes (OpenORB [52], Flexinet, 2K/DynamicTAO) have been developed to investigate this approach.

The goal of aspect-oriented programming [64] (AOP) is to define methods and tools to better identify and isolate the code related to the various "aspects" present in an application, such as those related to extra-functional properties (security, fault tolerance, performance). This code is usually intertwined with that of the application proper, which is detrimental to easy change.

Pragmatic techniques usually rely on interception (interposing code at the interface between components, while preserving the interface). They are mainly used in commercial middleware.

The price of the increased flexibility brought by these adaptation techniques is paid in performance, due to the additional indirection or interpretation levels that these techniques introduce. Thus optimization techniques for reducing this performance penalty are an active subject of research. They include partial evaluation and program specialization [66], code injection (inlining), dynamic code generation and optimization [67].

3.3.2. Models, Patterns and Frameworks for Distributed Systems Architecture

We single out three aspects in the foundations of distributed systems architecture, in order of decreasing abstraction: defining formal or semi-formal models; identifying generic design patterns; developing the associated software frameworks.

Developing models for distributed systems is notoriously hard, compared to centralized systems. This is due to the increased complexity deriving from the asynchrony of the communication system, from fault tolerance requirements, and from mobility. Two main directions are relevant for our project: modeling software architecture, and specifically component-based systems (a survey of recent work is [65]); defining process calculi, specially for distributed and mobile processes (see survey in [53]). Our work aims at capturing those aspects of distributed systems that are relevant to dynamic adaptation, i.e. modularity, reconfigurability and mobility.

A *design pattern* [58] describes, in an articulated way, the structures, the representations and the techniques used to respond to specific requirements in a given context. A pattern captures in a generic, abstract form the experience acquired in the resolution of a class of related problems. A *software framework* [61] is a program skeleton that may be directly reused, or adapted according to well-defined rules, to solve a family of related problems. A framework usually embodies a pattern or a set of patterns.

Research on patterns for middleware has been very active since 1995, and a number of basic constructs have been identified [54][68]. Most middleware research prototypes are now developed as frameworks, in order to facilitate sharing and reuse.

Our own work is targeted towards global architectural patterns, such as naming and binding, components and reconfiguration, mobility and replication, transactions. These aspects have been less studied than the specific constructs used as building blocks. The results are transferred to the software platforms developed by OBJECTWEB (8.2).

3.4. Distributed System Administration

Administration is the function that aims at maintaining a system's ability to provide its specified services, with a prescribed quality of service. In general terms, administration may be viewed as a *control* activity, involving an event-reaction loop: the administration system detects events that may alter the ability of the administered system to perform its function, and reacts to these events by trying to restore this ability.

The operations performed under system and application administration include configuration and deployment, reconfiguration, resource management, observation and monitoring.

For about 15 years, the approach to system and network administration has been inspired by the manager-agent model, the base of widely used systems such as SNMP and CMIS/CMIP. The limitations of this model are now apparent, specially as regards its ability to scale up and to cope with the increased complexity and the dynamic evolution of the managed systems.

The next sections examine three specific aspects: the new requirements posed by event distribution and resource management, and the trend towards self-managed systems.

3.4.1. Event Channels

Many management functions (monitoring, resource management, fault tolerance) rely on *events*, propagated from the administered components of the system to the management components. In a large Internet-based system, these events may be generated at a very high rate. Thus efficient channels are needed to collect and to filter these events, and to propagate them to their recipients. This problem has emerged as a topic of active research. It is also related to the fast-developing area of sensor networks.

Event propagation usually follows the publish-subscribe pattern [57]. Scalable solutions, based on multicast groups, are known for publish-subscribe based on a fixed (or slowly changing) list of topics. However, designing an efficient publish-subscribe system based on the contents of the propagated messages is still an open problem. For Internet-scale systems, even filtering algorithms running in linear time with respect to the number of subscriptions are considered inefficient, and the goal is to design sub-linear algorithms. The Gryphon [49] and Siena [55] projects have made significant progress towards this goal.

The application of event channels to large scale observation and management is the theme of the Astrolabe [70] project.

3.4.2. Resource Management

The advent of resource-constrained applications, such as multimedia processing and real-time control, raised the need for delegating part of the resource management to the application, while leaving the operating system with the responsibility of fair overall resource sharing between its users. The rationale is that the application is in a better position to know its precise requirements and may dynamically adapt its demands to its needs, thus allowing global resource usage to be optimized, while guaranteeing a better service to each individual application.

The challenge is now to define relevant abstractions for resource management, both for the *resource principals*, i.e. the entities to which resources are allocated, and for the resources themselves. Thus, in the case of cluster computing, new abstractions such as “cluster reserves” and “virtual clusters”, which group a dynamic set of cluster-wide resources, are being investigated. Another challenge is to define what part of resource management is delegated from the operating system to the application, and to specify the relevant interfaces.

Scheduler activations [50] is an example of early work in this respect in the area of multiprocessor scheduling. A more radical approach is illustrated by exo-kernels [56] in which the lower layer exports primitives allowing a kernel or application designer to define his own resource allocation policy. This is done at the expense of additional work, which may be alleviated by the use of appropriate frameworks, as done in THINK [6]. However, since exo-kernels give access to low level system entities, they are prone to security problems.

3.4.3. Autonomic Systems

Up to now, administration tasks have mainly been performed by persons. A great deal of the knowledge needed for administration tasks is not formalized and is part of the administrators’ know-how and experience.

As the size and complexity of the systems and applications are increasing, the costs related to administration are taking a major part of the total information processing budgets, and the difficulty of the administration tasks tends to approach the limits of the administrators’ skills. As a consequence, there is a trend towards automating (at least in part) the functions related to administration. This is the goal of the so-called *autonomic computing* movement [62][59].

Autonomic computing aims at providing systems and applications with self-management capabilities, including self-configuration (automatic configuration according to a specified policy), self-optimization (continuous performance monitoring), self-healing (detecting defects and failures, and taking corrective actions), and self-protection (taking preventive measures and defending against malicious attacks).

Several research projects [51] are active in this area.

4. Application Domains

Key words: *telecommunications*, *multimedia*, *embedded systems*, *electronic commerce*, *systems administration*, *power supply*.

Sardes develops generic tools for distributed applications, in the form of middleware, system kernels, and information servers. These tools are useful in application domains that have one or more of the following properties.

- Cooperation using shared distributed information;
- Mobility of users, information and services;
- Need for dynamic adaptation of infrastructures or applications;
- Use of high performance information servers.

The main actual application domains are:

1. telecommunications : administration of large scale networks, servers and caches for the Web, management of configurable added value services;
2. power supply : administration and monitoring of power supply networked equipment, e.g. uninterrupted power supply units.
3. embedded computing: development of custom made kernels for specific applications (robotics, real time), dynamically reconfigurable kernels;
4. multimedia applications: dynamic adaptation of a videoconferencing system for use by mobile clients;
5. electronic commerce: flexible access to remote services by mobile users, efficient transaction management.

5. Software

5.1. Introduction

Key words: *cluster*, *J2EE*, *benchmark*, *Java thread*, *mobility*.

Software development is an important aspect of the activity of Sardes. This software serves as a testbed to apply, validate and evaluate the methods and tools developed in the project.

Sardes contributes to the development of the ObjectWeb open source code base (see 8.2), which is accessible at http://www.objectweb.org.

5.2. JavaThread : Support for mobility and persistence of Java threads

Contact : Daniel Hagimont.

We have extended the Java Virtual Machine (JVM) with a non-intrusive mechanism for capturing and restoring the complete state of a Java thread. The state may be captured at any execution stage. The capture mechanism does not impose any run time performance penalty. It relies on dynamic type inference techniques for analyzing data on the execution stack, and on dynamic de-optimization of the JIT-compiled Java code. The mechanism has been implemented in the Sun Microsystems JVM.

The *JavaThread* software is distributed in binary form under an Inria license.

See http://sardes.inrialpes.fr/research/JavaThread/

5.3. C-JDBC: Clustered JDBC-accessed Database

Contact : Emmanuel Cecchet.

JDBC™(short for Java DataBase Connectivity), is a widely used Java API for accessing virtually any kind of tabular data. C-JDBC (Clustered Java™DataBase Connectivity) is an open source database cluster middleware that allows any application to transparently access a cluster of databases through JDBC. The database is distributed and possibly replicated among several nodes and C-JDBC load balances the queries between these nodes. The related research activity is described in 6.3.3.

C-JDBC is an OBJECTWEB project distributed under an LGPL license.

See http://c-jdbc.objectweb.org

5.4. Benchmarks for J2EE Systems

Contact : Emmanuel Cecchet.

RUBiS (Rice University Bidding System) is an auction site prototype modeled after `eBay.com`. RUBiS is currently used to evaluate design patterns, application servers and communication layers scalability. RUBiS has been implemented for PHP, Servlets and Enterprise Java Beans (EJB) in 7 different versions. Ongoing activities concern JDO (Java Data Objects) and Microsoft .Net versions of RUBiS. RUBBoS is a bulletin board benchmark modeled after an online news forum. Like RUBiS, RUBBoS was designed to evaluate design patterns, application servers and communication layers scalability.

Both RUBiS and RUBBoS are part of the OBJECTWEB JMOB (Java Middleware Open Benchmarking) project and are distributed under an LGPL license.

See http://jmob.objectweb.org

5.5. JOTM: Java Open Transactional Manager

Contact : Emmanuel Cecchet.

JOTM [26][27][25] is an open source transaction manager implemented in Java. It was originally part of the JONAS Enterprise Java Beans platform. Since 2002, it has been extracted from the JONAS environment and developed as an autonomous project to provide support for any distributed platform that needs transactional facilities.

JOTM implements the Java Transaction API (JTA) specification with full support on both RMI/JRMP and RMI/IIOP. It also implements the Business Transaction Protocol (BTP) to support transactions for Web Services.

JOTM is an OBJECTWEB project distributed under an LGPL license.

See http://jotm.objectweb.org

6. New Results

6.1. Models and Tools for Distributed Components

Our work on models and tools for distributed components has covered two main aspects: advances in theory (6.1.1), and more practically oriented contributions in two areas: optimization techniques for components (6.1.2), and tools for replication management (6.1.3).

6.1.1. Models for Distributed Computing

Participants: Jean-Bernard Stefani, Philippe Bidinger, Sacha Krakowiak.

The work on models has two main objectives: to elicit a model for distributed computing that includes constructs for mobility and dynamic reconfiguration; to provide a basis for the formal specification of the systems developed in the project.

In 2003, we have made progress in two main directions.

- Developing a new calculus for processes, the M-calculus [37][33], which builds on the concepts of the distributed Join calculus and extends them in several directions: dynamic binding, higher order constructs, integration with the λ-calculus. This work is done in association with the IST project Mikado (8.3.2). In his Ph.D. thesis [11], M. Lacoste (France Telecom R&D) has investigated the suitability of the M-calculus as a basis for practical constructs, through the design and partial implementation of a virtual machine based on the M-calculus.
- Developing a calculus with process migration and hierarchical localities, the Kell calculus [34][16], which is intended to provide a foundation for distributed wide-area computing, while allowing efficient implementation. This calculus builds on the ideas of the M-calculus, while retaining the original simplicity of Mobile Ambients, a reference model for mobile computing. The Kell calculus is the foundation of FRACTAL [3][39], a framework for software components, jointly designed by France Telecom R&D and Inria, and distributed by the OBJECTWEB consortium. In contrast with most existing component models, FRACTAL allows hierarchical composition and component sharing, and provides management facilities, including dynamic reconfiguration.

Work is in progress to apply these models to the specification and proof of actual systems, e.g. the THINK framework for kernel development [6]. This is the subject of the ongoing Ph.D. thesis of Ph. Bidinger.

6.1.2. Optimization Techniques for Components

Participants: Daniel Hagimont, Noël De Palma, Fabienne Boyer.

Separation of concerns, which aims at separating different aspects involved in complex applications, is a general trend in software design. It allows a given aspect to be programmed in a more or less isolated manner from the functional code of an application. This trend has been exploited by the AOP (Aspect Oriented Programming) community, through the provision of language support for programming and composing aspects. It has also been experimented in the context of component-based middleware, which usually addresses system-related aspects (e.g., transactions, security, persistence, etc).

In both domains, most implementations of separation of concerns involve indirection objects and extra method calls that incur a non-negligible performance overhead. These indirections are used to integrate aspect code within the functional code of the applications. Therefore, we investigated optimization techniques which aim at avoiding the overhead of these indirection techniques. These techniques rely on code injection: the aspect code is “injected” within the code of the application. We considered two aspects, replication and access control, and we showed how code injection techniques can be used to optimize aspects integration. This led us to the design of an injection pattern called “extended reference” [17], which appears to be well suited to the efficient integration of system aspects.

6.1.3. Tools for Replication

Participants: Daniel Hagimont, Vania Marangozova.

Component-based programming encourages software reuse and (as in aspect-oriented programming) promotes the separation between the implementation of the services provided by the component and the code managing the common system services. One such service, of particular importance to the distributed computing domain, is replication. It is actually used for various purposes e.g. performance (caching), fault-tolerance, availability for mobile users. However, replication management in a component-based environment is still an open issue.

In this context, we proposed an approach to component replication where replication and consistency management can be configured separately from the functional code of the application. With this approach, an application can be configured with different replication strategies and a replication protocol can be reused for different applications. We experimented this approach on the OpenCCM platform (an implementation of the CORBA Component Model) and validated this implementation with several applications.

This work was the subject of the Ph.D. thesis of V. Marangozova [12].

6.2. Distributed Resource Management

We are in the process of developing a framework for distributed resource management. The first task is to accurately monitor a set of distributed resources, by maintaining a knowledge base that registers the available resources and their current state. This may involve a very large set of devices that is spread on a wide geographical scale, and whose composition and allocation state changes dynamically. A first design of a monitoring application is described in 6.2.1.

Each change in the composition of the set of resources and in their allocation state gives rise to an event. A large scale, flexible event propagation system is therefore needed. A framework for the construction of such systems is described in 6.2.2.

6.2.1. Distributed Observation and Monitoring

Participants: Jean-Bernard Stefani, Emmanuel Cecchet, Renaud Lachaize, Philippe Laumay, Vivien Quéma.

A resource monitoring application needs to have the following properties: distribution, as resources are spread all over the world; scalability, as thousands of devices are involved; flexibility, as new devices are joining and leaving the system dynamically; configurability, as the monitoring application must support a wide spectrum of devices; adaptability, as operating conditions may vary; interoperability, as it is necessary to export monitored data to third-party applications.

We have done a preliminary design of a monitoring application that aims at meeting these challenges. The application is composed of three kinds of distributed cooperating tasks: *Collection* tasks are deployed on monitored devices; they produce events. *Processing* tasks are in charge of processing, i.e. filtering, forwarding or aggregating events sent by collection tasks. Finally, *Presentation* tasks make events sent by processing tasks available to the observers. These tasks are built as components executing asynchronously and communicating using a message-oriented middleware (MOM) built with the framework described in the next section.

The forthcoming Ph.D. thesis of Ph. Laumay investigates several aspects of distributed observation and event processing.

6.2.2. A Framework for Event-based Middleware

Participants: Jean-Bernard Stefani, Matthieu Leclercq, Vivien Quéma.

DREAM (Dynamic Reflective Asynchronous Middleware) [31] is a framework dedicated to the construction of asynchronous middleware. It provides a set of components that can be dynamically glued to build different paradigms of asynchronous communication: message-passing, event-reaction, publish-subscribe, etc. As a first experiment, we have developed an implementation of JORAM (a MOM developed in OBJECTWEB) as an assembly of components (message queues, channels, routers, transformers, multiplexers). We have also shown how these components can be differently assembled to build a lightweight version of JORAM, dedicated to devices with limited resources. We are currently developing a set of management tools to deploy, configure and administer middleware built using DREAM.

6.3. Applications

6.3.1. Reconfigurable and Secure Operating Systems Kernels

Participants: Jean-Bernard Stefani, Sacha Krakowiak, Sébastien Jean, Olivier Charra, Christophe Rippert, Aline Senart.

The traditional image of operating systems is changing, since processors are being integrated in a variety of objects and environments: mobile phones, smart cards, sensors, etc. These environments impose specific requirements and constraints: security, availability, fine-grain control of resource usage, flexibility, adaptability.

One approach to meeting these requirements is to use a lightweight framework as a basis for the construction of custom-made systems. THINK [6] is such a framework, initially developed at France Telecom R&D. THINK is inspired by the exo-kernel model (3.4.2), but goes one step further: its interface does not provide any abstraction at all, but it essentially reifies the underlying hardware interface; and it includes a lightweight software framework for developing the basic functions of an operating system (naming, binding and communication).

We have extended the original THINK framework with a lightweight component structure (derived from the FRACTAL component model), which provides capabilities for dynamic composition and reconfiguration. Experimental embedded applications have been developed with THINK in the area of active networks and mobile robots. This work was the subject of the Ph.D. theses of A. Senart [14] and O. Charra (forthcoming).

In order to allow the construction of customized security policies, we have integrated a protection framework into the THINK system. The work was done on four types of resources: processor, RAM, disk storage, and network connections. Protection mechanisms have been developed against the following attacks: service denial by packet flooding or by attack on the disk channel; memory corruption, direct or by insertion of malicious code. This was the subject of the Ph.D. thesis of Ch. Rippert [13]; see also [32].

6.3.2. Managing QoS for Multimedia Applications

Participants: Daniel Hagimont, Oussama Layaida, Slim Ben Atallah.

Multimedia streaming applications, such as videoconferencing, TV streaming or video on demand are increasingly deployed in heterogeneous and mobile environments including workstations, PDAs, mobile phones, etc. These applications are very resource demanding and they usually need to be adapted when executing in constrained environments, including lightweight terminals or low-bandwidth networks. The proxy-based adaptation approach aims at deploying adaptation processes on dedicated intermediate nodes to transcode multimedia streams in real time. This approach is well suited to reducing resource consumption such as CPU, memory and network bandwidth, without modification of end-user applications (since the adaptations take place on proxies). However, the adaptation on the proxy has to be dynamically configured according to the constraints of the execution environment.

To this purpose, we implemented a framework for the dynamic configuration of such proxies [35][24]. This framework relies on a component model for building complex adaptation processes as assemblies of components, and on a configuration language called APSL (Adaptation Proxy Specification Language) for the specification of such architectures.

Using this framework, we implemented adaptation proxies for an existing videoconferencing application and a video on demand application; we thus propose a remedy to the inefficiency of multimedia applications on low capability terminals and wireless networks. In our experience, a CIF-sized video can hardly be streamed and displayed at its original frame rate (25 frames per second) on an iPAQ PDA (the frame rate drops by about 50%). An adequately configured adaptation proxy, which scales down the video size, reduces its quality factor and/or modifies its encoding format, allows us to maintain the original video frame rate by saving on network bandwidth and CPU consumption.

6.3.3. Managing Systems and Applications on Clusters

Clusters of processors are a cost-effective alternative to high performance servers. They were initially introduced for CPU-intensive applications, but their use for data-intensive services is taking an increasing importance. We have investigated three ways of improving the use of clusters for this class of services: replicating databases, load balancing for middle-tier services, system support for high-performance I/O.

6.3.3.1. Clustered Databases

Participants: Emmanuel Cecchet, Nicolas Modrzyk.

We have investigated the use of cluster servers to improve the availability and performance of distributed databases. To that effect, we have introduced the concept of RAIDb (Redundant Array of Inexpensive Databases), the counterpart of RAID (Redundant Array of Inexpensive Disks) for databases. RAIDb [36][22] aims at low cost improvement of performance and fault tolerance over a single database, by combining multiple database instances into an array of databases. Based on this concept, we have implemented C-JDBC (Clustered Java™DataBase Connectivity) [21], an open source database cluster middleware (5.3) that allows any application to transparently access a cluster of databases through the JDBC standard interface. C-JDBC implements database partitioning, full mirroring and partial mirroring. Using standard benchmarks such as TPC-W, we have shown that partial replication can bring substantial scalability improvement (up to 25%) compared to full replication. However, static round robin policies are inefficient and partial replication needs dynamic load balancing algorithms to scale. We are also investigating the performance of SQL query caching at the middleware level.

6.3.3.2. Load Balancing for EJB on Clusters

Participants: Jacques Mossière, Simon Nieuviarts.

We have designed and developed a configurable framework for balancing client requests in a cluster of computers. Using this framework, named CMI (Cluster Method Invocation), programmers may generate a support for server applications replicated in a cluster using a selected load balancing algorithm. This is done very simply, by specifying a few keywords in a file.

CMI uses the same application programming interface as the Java Remote Method Invocation (RMI), to allow easy migration of existing distributed applications to clusters. It is based on Javagroups, a group communication system whose performance has been carefully studied. CMI has been integrated into the OBJECTWEB CAROL project, which wraps several RMI implementations into a single library. CMI has been successfully used in JONAS to provide load balancing mechanisms for EJB applications without needing any additional code from the application developer. Performance benchmarks show a significant increase in the throughput of EJB applications.

6.3.3.3. System Support for Cluster File Systems

Participants: Jacques Mossière, Renaud Lachaize.

The aim of the Proboscis[2] project is to study the sharing of storage devices in clusters, where the storage devices are distributed across the nodes and accessed with efficient networking technologies. For this purpose, we are developing a modular and extensible software infrastructure for remote disk access construction and administration. This framework is intended as a set of building blocks, on which specific storage services such as cluster file systems can be built and deployed. A key feature of Proboscis lies in its use of explicit I/O data paths to facilitate disk access scheduling, reconfiguration at runtime and monitoring.

Current work on Proboscis focuses on dynamic reconfiguration of the data paths to increase the flexibility and availability of clustered data servers. The modularity of the framework is used to allow non-disruptive changes of the low level storage services, such as (remote) disk accesses and software RAID emulation, without any impact on the upper layers of the system (e.g. logical volume managers and cluster file systems). This is beneficial to fault tolerance (data redistribution in case of a server crash), performance (load balancing, replacement of the scheduling algorithm for disk requests), and occasional administration needs (e.g. transparently switching between networks). The current prototype relies on a human administrator for decision making, but the reflexivity features of the framework could lead to a self-manageable version.

This work is the subject of the Ph.D. thesis of R. Lachaize.

The source code of Proboscis is available on request. See http://sardes.imag.fr/~rlachaiz/proboscis.shtml.

7. Contracts and Grants with Industry

7.1. Collaboration with Bull

Participants: Jacques Mossière, Takoua Abdellatif, Simon Nieuviarts.

[2]Proboscis is a joint research effort of Sardes and the DistLab group at DIKU, the Computer Science Laboratory of the University of Copenhagen.

The theme of the collaboration with Bull is the development of system software for exploiting clusters operating under Linux for scientific and data management applications. This collaboration started in October 2000, and also involves the Inria project-team Apache (Jacques Briat, Yves Denneulin).

Our contribution is twofold:

- Using a clustered server for developing an efficient, fault-tolerant version of an *Enterprise Java Beans* (EJB) platform. This is the subject of the forthcoming Ph.D. thesis of S. Nieuviarts (Bull co-funding); see also 6.3.3.
- Developing an administration support system for a cluster. This work started in September 2003 and is the subject of the Ph.D. thesis of T. Abdellatif (Bull co-funding).

7.2. Collaboration with France-Telecom R&D

Participants: Jean-Bernard Stefani, Olivier Charra, Christophe Rippert, Aline Senart, Sacha Krakowiak, Daniel Hagimont, Vania Marangozova.

Sardes maintains an active collaboration with France-Telecom R&D (Norbert Segard Center, Distributed Systems Architecture group):

1. Collaboration within the OBJECTWEB consortium (8.2), on the following aspects: developing the FRACTAL component model [3][39] (J.-B. Stefani); extensions to the THINK software framework [13][14][32] (J.-B. Stefani, O. Charra, Ch. Rippert, A. Senart).
2. Collaboration on distributed process calculi including mobility and distribution (6.1.1), and on the associated virtual machines (J.-B. Stefani). This work is done within the European IST project Mikado (8.3.2) and was the subject of the Ph.D. thesis of M. Lacoste, FT R&D [11], supervised by S. Krakowiak.
3. Contract “JumboBeans”. This contract funded by FT R&D aimed at developing a framework for replication for applications based on Java EJB and CCM components (6.1.3). This was the subject of the Ph.D. thesis of V. Marangozova [12].

7.3. Collaboration with Microsoft

Participants: Emmanuel Cecchet, Daniel Hagimont, Oussama Layaida, Jacques Mossière.

The goal of this contract is to develop techniques for dynamic adaptation of embedded multimedia applications on Microsoft Windows CE .Net. This work is jointly carried out by project-teams Sardes and WAM at Inria Rhône-Alpes (N. Layaida).

We shall extend the adaptation techniques developed in Sardes (6.3.2) for dynamic proxy configuration in order to ensure QoS of multimedia applications using the Microsoft DirectShow middleware framework. We intend to demonstrate that our techniques can be applied in a resource-constrained environment.

7.4. Collaboration with ST-MicroElectronics

Participants: Jacques Mossière, Erdem Ozcan.

The goal of this project, started in November 2003, is to investigate the use of the THINK framework to develop operating systems for on-chip multiprocessors. This is the subject of the Ph.D. thesis of E. Ozcan (ST-MicroElectronics co-funding).

7.5. RNTL Impact Project

Participants: Jean-Bernard Stefani, Sacha Krakowiak, Emmanuel Cecchet, Jean-Frédéric Mesnil, Marek Procházka.

The objective of the Impact project is to contribute to the development of the OBJECTWEB open source platform (8.2) by integrating enhancements coming from various research and development groups in academia and industry. Sardes has contributed on the following aspects.

- Architecture: contribution to the definition of the FRACTAL component model; contribution to the definition of a common transaction management component (JOTM).
- Technical Components: re-engineering of the JORAM message bus; enhancing the EJB JONAS platform for fault tolerance by bean replication.
- Platforms: Contribution to the re-engineering of JONAS: improving communication; developing benchmarks; contribution to the documentation.

See http://www.industrie.gouv.fr/rntl/AAP2001/Fiches_Resume/IMPACT.htm

7.6. RNTL Arcad Project

Participants: Jean-Bernard Stefani, Fabienne Boyer, Daniel Hagimont, Olivier Charra, Noël De Palma, Aline Senart.

Arcad is a 36-month exploratory project (November 2000 to November 2003) funded by the French Ministry of Industry under the RNTL program. Its partners are: Inria (Sardes and Oasis project-teams), France Telecom R&D, École des Mines de Nantes and the I3S laboratory at Nice Sophia-Antipolis (Rainbow group, M. Riveill). The project aims at designing and developing a distributed extensible environment for deploying distributed component-based applications.

See http://arcad.essi.fr/

7.7. RNTL Inside Project

Participants: Jean-Bernard Stefani, Matthieu Leclercq, Vivien Quéma.

Inside is a pre-competitive project funded by the French Ministry of Industry under the RNTL program. Its partners are Inria (Sardes project-team), and three companies: Schneider Electric, OpenSugar and ScalAgent. The goal of the project is to develop a software infrastructure to support Internet-based services (specially distributed equipment monitoring) in the area of power distribution. The infrastructure will be validated in a real-life environment.

See http://www.telecom.gouv.fr/rntl/projet/Posters-PDF/RNTL-Poster-INSIDE.pdf

8. Other Grants and Activities

8.1. National Actions

8.1.1. ARP Network

Sardes is a member of the “Distributed Systems and Applications” group of the national research network on Systems Architecture, Networks and Parallelism (ARP: Architecture, Réseaux et Parallélisme).

See http://www.arp.cnrs.fr.

8.2. ObjectWeb Consortium

Participants: Jean-Bernard Stefani, Sacha Krakowiak, Jean-Frédéric Mesnil, Mathieu Peltier, Marek Procházka.

OBJECTWEB is an open-source software community created at the end of 1999 by France Telecom R&D, Bull and Inria. Its goal is the development of open-source distributed middleware, in the form of flexible and adaptable components. These components range from specific software frameworks and protocols to integrated platforms. OBJECTWEB developments follow a systematic component-based approach.

In 2002, ObjectWeb evolved into an international consortium hosted by Inria. The consortium is an independent non-profit organization open to companies, institutions and individuals.

Sardes contributes to ObjectWeb through its technical involvement in the development of software components and frameworks (see project Impact, 7.5) and through participation in the management structures of the consortium: board (J.-B. Stefani, president), executive committee (S. Krakowiak, member), and college of architects (J.-B. Stefani, E. Cecchet and S. Krakowiak, members).

See http://www.objectweb.org

8.3. Projects Funded by the European Commission

8.3.1. IST Project Ozone

Participants: Daniel Hagimont, Slim Ben Atallah, Oussama Layaida.

The goal of the IST project Ozone is to develop a software environment for ubiquitous computing and ambient intelligence. The project's partners are: Philips (Netherlands), IMEC (Belgium), LEP (France), Epictoid (Netherlands), Eindhoven University (Netherlands), Thomson Multimédia and Inria (project-teams Arles, Moscova and Sardes).

One of the tasks of the project is to develop software frameworks for embedded real-time multiprocessor systems. Our contribution to this task is two-fold:

- we study the application of our results on the management of QoS for multimedia applications (6.3.2) in the context of the Ozone project.
- we study the application of our tools for component replication (6.1.3) in the context of the Ozone project, more precisely in the context of applications based on Web Services.

See http://www.extra.research.philips.com/euprojects/ozone/

8.3.2. IST Project Mikado

Participants: Jean-Bernard Stefani, Philippe Bidinger.

The goal of the IST project Mikado is to develop a model for distributed mobile programming based on a process calculus. Investigation topics include type models, programming language and virtual machines technologies, and specification languages. The project's partners are: Inria (project-teams Mimosa and Sardes), France Telecom R&D (DTL/ASR Laboratory), University of Sussex, UK (M. Hennessy), Università di Firenze, Italy (R. de Nicola), University of Lisbon, Portugal (V. Vasconcelos).

See http://mikado.di.fc.ul.pt/

8.3.3. ITEA Project Osmose

Participants: Jean-Bernard Stefani.

Osmose (Open Source Middleware for Open Systems in Europe) is a project funded by the ITEA program. The project is focused on the development, enhancement, and validation in defined test-beds of a comprehensive adaptable Open Source middleware to be hosted by the ObjectWeb consortium (8.2). The testbeds will be developed in three areas: telecom services, home gateway, and avionics.

The Consortium is built around three sets of partners: 6 large industrial companies (Bull, France Telecom, Philips, Telefonica, Telvent and Thales), 6 SMEs (Bantry Technologies, iTEL, Kelua, Lynx, VICORE and Whitestein Technologies), and 7 academic partners (Charles University, EPFL, Inria, INT, LIFL, Imag-LSR and Universidad Politécnica de Madrid).

See http://www.itea-osmose.org/

8.4. International Networks and Working Groups

8.4.1. CaberNet Network of Excellence (ESPRIT NE 21035)

Sardes is a member of the Network of Excellence *Distributed Computing Systems Architecture*, a.k.a. *CaberNet*. The network provides funding for inter-group collaboration and post-doctoral stays. CaberNet is also organizing ERSADS, the *European Research Seminar on Advances in Distributed Systems*.

See http://www.newcastle.research.ec.org/cabernet/index.html

8.4.2. Artist Network of Excellence (IST-2001-34820)

Sardes is a member of the Network of Excellence *Advanced Real Time Systems* (Artist), whose objective is to coordinate European efforts in advanced real time systems. Sardes is active in the *Adaptative Real-Time Systems For QoS Management* action.

See http://www.systemes-critiques.org/ARTIST/

8.4.3. IST Project Midas (IST-2001-37610)

Participants: Jean-Bernard Stefani, Sacha Krakowiak, Christophe Ney.

Midas is a 12-months project, part of the "accompanying measures" of the IST program funded by the European Commission. Its goal is to set up a research roadmap for developing the next generation of middleware for large scale distributed services. Its partners are: University of Newcastle upon Tyne, Inria (project-team Sardes), École Polytechnique Fédérale de Lausanne, Università La Sapienza (Roma), Hebrew University Jerusalem, HP Labs (Bristol).

See http://www.newcastle.research.ec.org/midas/

8.5. International Bilateral Collaborations

8.5.1. Europe

Sardes maintains long term collaboration with several research groups in Europe:

- University of Newcastle upon Tyne, *Distributed Systems Group* (Prof. Santosh Shrivastava). Collaboration on middleware and distributed programming, specially within project Midas.
- École Polytechnique Fédérale de Lausanne, Distributed Systems Laboratory (Prof. André Schiper) and Distributed Programming Laboratory (Prof. Rachid Guerraoui). Collaboration on fault tolerance through project Midas and through direct contacts.
- University of Lancaster, *Distributed Media Systems* (Prof. Gordon Blair), on adaptable middleware for multimedia communication.
- Trinity College, Dublin, *Distributed Systems Group* (Dr Vinny Cahill), on distributed programming and clusters. A. Senart, a former Sardes Ph.D. student, is staying with TCD for the academic year 2003-2004.
- University of Copenhagen, DIKU Laboratory *Distributed Systems Group* (Prof. Eric Jul), on system support for clustered servers (visits and shared software).

8.5.2. North Africa

Sardes maintains contacts with the Computer Science Department of Université des Sciences et de la Technologie Houari Boumediene, Bab-Ezzouar, Algiers, Algeria (Dr Belkhir), and with École Nationale des Sciences de l'Informatique (ENSI), Université de la Manouba, Tunis, Tunisia. Slim Ben Atallah, associate professor at ENSI, has been staying with Sardes since July 2002 as an invited scientist. Sardes also welcomes ENSI students as interns for last term projects.

9. Dissemination

9.1. Community Service

J.-B. Stefani is a member of the editorial board of the journal *Annales des Télécommunications*, and of the program committees of the *Distributed Applications and Interoperable Systems Conference* (DAIS'03), the *5th International Conference on Distributed Objects and Application* (DOA'03), and the *Middleware'03* Conference.

D. Hagimont is co-program chair of DAIS'03 and a member of the program committees of DOA'03 and of CFSE, the Conference on Operating Systems organized by the French Chapter of ACM SIGOPS.

S. Krakowiak is a member of the steering committee of ERSADS (*European Research Seminar on Advances in Distributed Systems*), the school/workshop organized by the CaberNet Network of Excellence, and a member of the program committees of DAIS'03 and CORE'03. He is also a member (and designated president) of the committee sponsored by *Specif* (the French Computer Science Researchers' and Teachers' Association) to select the best Ph.D. thesis in Computer Science defended each year in France.

E. Cecchet has organized the *Workshop on Middleware Benchmarking* at the OOPSLA 2003 Conference (http://nenya.ms.mff.cuni.cz/projects/corba/oopsla-workshop/). He is in charge of a joint Imag-Inria seminar on "Distributed Systems and Data Management"

Several members of Sardes have contributed to the organization of the 3rd OBJECTWEB Conference (Inria, Rocquencourt, 20-21 November 2003).

9.2. School Organized by the Project

Sardes has organized the ICAR Summer School on Middleware and Distributed Application Construction (*Intergiciel et Construction d'Applications Réparties*). The school took place at Autrans, near Grenoble, from August 25 to 29, 2003, and was attended by about 90 persons from academia and industry. The school was organized by D. Hagimont and N. De Palma. S. Krakowiak gave a course on Patterns and Frameworks for Middleware. E. Cecchet presented case studies on J2EE servers and applications.

See http://sardes.inrialpes.fr/ecole/

9.3. University Teaching

D. Hagimont, F. Boyer, N. De Palma, S. Krakowiak, J. Mossière and J.-B. Stefani have taught several operating systems and distributed systems courses at the M.S. and M.Eng. levels, both at Institut National Polytechnique de Grenoble and at université Joseph Fourier. Most of our Ph.D. students contributed to these courses as teaching assistants.

9.4. Participation in Seminars, Workshops, Conferences

E. Cecchet gave presentations on OBJECTWEB related activities at the Jax 2003, LinuxWorld2003, and Solutions Linux professional conferences.

Several members of Sardes attended various scientific conferences and workshops. See the "communications" section of the Bibliography for details.

10. Bibliography

Major publications by the team in recent years

[1] G. BLAIR, J.-B. STEFANI. *Open Distributed Processing and Multimedia.* Addison-Wesley, 1997.

[2] S. BOUCHENAK. *Making Java Applications Mobile or Persistent.* in « 6th USENIX Conference on Object-Oriented Technologies and Systems », San Antonio, Texas, January, 2001.

[3] É. BRUNETON, T. COUPAYE, J.-B. STEFANI. *Recursive and Dynamic Software Composition with Sharing.* in « Proceedings of the 7th ECOOP International Workshop on Component-Oriented Programming (WCOP'02) », Malaga (Spain), June 10th-14th, 2002.

[4] É. BRUNETON, M. RIVEILL. *An Architecture for Extensible Middleware Platforms.* in « Software – Practice and Experience », volume 31, November, 2001, pages 1237–1264.

[5] E. CECCHET. *SciFS: une mémoire partagée distribuée pour grappes SCI.* in « Technique et Science Informatiques (TSI) », number 5, volume 20, 2001, pages 629–654.

[6] J.-PH. FASSINO, J.-B. STEFANI, J. LAWALL, G. MULLER. *THINK: A Software Framework for Component-based Operating System Kernels.* in « Proceedings of Usenix Annual Technical Conference », Monterey (USA), June 10th-15th, 2002.

[7] D. HAGIMONT, F. BOYER. *A Configurable RMI Mechanism for Sharing Distributed Java Objects.* in « IEEE Internet Computing », number 1, volume 5, 2001, pages 36–44.

[8] S. KRAKOWIAK, S. SHRIVASTAVA, editors, *Recent Advances in Distributed Systems.* series Lecture Notes in Computer Science 1752, Springer, 2000.

[9] PH. LAUMAY, É. BRUNETON, N. DE PALMA, S. KRAKOWIAK. *Preserving Causality in a Scalable Message-Oriented Middleware.* in « Middleware 2001, IFIP/ACM International Conference on Distributed Systems Platforms », Heidelberg, nov, 2001.

[10] É. NAJM, A. NIMOUR, J.-B. STEFANI. *Behavioural Typing for Objects and Process Calculi.* Cambridge University Press, 2001, chapter 13, pages 281–301.

Doctoral dissertations and "Habilitation" theses

[11] M. LACOSTE. *Une machine virtuelle répartie pour la programmation de processus mobiles.* Ph. D. Thesis, université Joseph-Fourier, October, 2003, Prepared at France Telecom R&D, Dept. ASR.

[12] V. MARANGOZOVA. *Duplication et cohérence configurables dans les applications réparties à base de composants.* Ph. D. Thesis, Université Joseph-Fourier, June, 2003.

[13] CH. RIPPERT. *Protection dans les architectures de systèmes flexibles.* Ph. D. Thesis, université Joseph-Fourier, October, 2003.

[14] A. SENART. *Canevas logiciel pour la construction d'infrastructures logicielles dynamiquement adaptables.* Ph. D. Thesis, institut national polytechnique de Grenoble, November, 2003.

Publications in Conferences and Workshops

[15] R. BALTER, L. BELLISSARD, V. QUÉMA. *A Scalable and Flexible Operation Support System for Networked Smart Objects.* in « Proceedings of the 2nd Smart Objects Conference (s0c) », Grenoble, France, May, 2003.

[16] PH. BIDINGER, J.-B. STEFANI. *The Kell calculus: operational semantics and type system.* in « Proceedings 6th IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS 03) », Paris, France, November, 2003.

[17] S. BOUCHENAK, F. BOYER, N. DE PALMA, D. HAGIMONT. *Can Aspects Be Injected? Experience with Replication and Protection.* in « International Symposium on Distributed Objects and Applications (DOA'03) », Catania, Sicily (Italy), November, 2003.

[18] S. BOUCHENAK, D. HAGIMONT, N. DE PALMA. *Efficient Java Thread Serialization.* in « Proceedings of the 2nd ACM International Conference on Principles and Practice of Programming in Java (ACM PPPJ'03) », Kilkenny, Ireland, June, 2003.

[19] S. BOUCHENAK, D. HAGIMONT, N. DE PALMA. *Techniques for Implementing Efficient Java Thread Serialization.* in « Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'03) », Tunis, Tunisia, July, 2003.

[20] E. CECCHET, A. CHANDA, S. ELNIKETY, J. MARGUERITE, W. ZWAENEPOEL. *Performance Comparison of Middleware Architectures for Generating Dynamic Web Content.* in « Proceedings of the 4th ACM/IFIP/USENIX International Middleware Conference (Middleware) », Rio de Janeiro, Brazil, June, 2003, http://middleware2003.inf.puc-rio.br/.

[21] E. CECCHET, J. MARGUERITE. *C-JDBC: Scalability and High Availability of the Database Tier in J2EE environments.* in « the 4th ACM/IFIP/USENIX International Middleware Conference (Middleware), Poster session », Rio de Janeiro, Brazil, June, 2003, http://middleware2003.inf.puc-rio.br/.

[22] E. CECCHET, J. MARGUERITE, W. ZWAENEPOEL. *Partial Replication: Achieving Scalability in Redundant Arrays of Inexpensive Databases.* in « 7th International Conference on Principles of Distributed Systems », La Martinique, France, December, 2003, http://www.math.chalmers.se/~dcs/OPODIS03/.

[23] B. DILLENSEGER, E. CECCHET. *CLIF is a Load Injection Framework.* in « OOPSLA'03 Workshop on Middleware Benchmarking: Approaches, Results, Experiences », Anaheim, CA, USA, October, 2003, http://nenya.ms.mff.cuni.cz/projects/corba/oopsla-workshop/.

[24] O. LAYAIDA. *Configuration Dynamique des Services de Transcodage des Flux Multimédia Continus.* in « Manifestation des Jeunes Chercheurs STIC (MAJESTIC'2003) », Marseilles, France, October, 2003, http://www.math.chalmers.se/~dcs/OPODIS03/.

[25] M. PROCHAZKA, R. ROUVOY, T. COUPAYE. *On Enhancing Component-based Middleware with Transactions.* in « International Symposium on Distributed Objects and Applications (DOA 2003), Poster Session »,

Catania, Italy, November 3-7, 2003.

[26] M. PROCHÁZKA. *A Flexible Framework for Adding Transactions to Components.* in « 8th Int. Workshop on Component-Oriented Programming (WCOP) », Darmstadt, Germany, July 21, 2003.

[27] M. PROCHÁZKA. *Jironde: A Flexible Framework for Making Components Transactional.* in « 4th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS 2003) », Paris, France, November 17-21, 2003, http://fedconf.enst.fr/index-dais.html.

[28] V. QUÉMA, R. BALTER, L. BELLISSARD, D. FÉLIOT, A. FREYSSINET, S. LACOURTE. *Administration and Deployment Tools in a Message-Oriented Middleware.* in « the 4th ACM/IFIP/USENIX International Middleware Conference (Middleware), Poster session », Rio de Janeiro, Brazil, June, 2003, http://middleware2003.inf.puc-rio.br/.

[29] V. QUÉMA, R. BALTER, L. BELLISSARD, D. FÉLIOT, A. FREYSSINET, S. LACOURTE. *Déploiement Asynchrone et Hiérarchique d'Applications Réparties á Composants.* in « 3ème Conférence Française sur les Systèmes d'Exploitation (CFSE-3) », La Colle sur Loup, France, October, 2003.

[30] V. QUÉMA, E. CECCHET. *The Role of Software Architecture in Configuring Middleware: the ScalAgent Experience.* in « Proceedings of the 7th International Conference on Principles of Distributed Systems (OPODIS'2003) », La Martinique, France, December, 2003, http://www.math.chalmers.se/~dcs/OPODIS03/.

[31] V. QUÉMA, R. LACHAIZE, E. CECCHET. *An Asynchronous Middleware for Grid Resource Monitoring.* in « Proceedings of the 1st International Workshop on Middleware for Grid Computing », Rio de Janeiro, Brazil, June, 2003, http://virtual01.lncc.br/mgc2003/.

[32] CH. RIPPERT. *Protection in Flexible Operating System Architectures.* in « ACM SIGOPS Operating Systems Review, volume 37, number 4, pages 8–18 », October, 2003.

[33] A. SCHMITT, J.-B. STEFANI. *The M-calculus: A Higher-Order Distributed Process Calculus.* in « In Proceedings of the 30th Annual ACM Symposium on Principles of Programming Languages (POPL'03) », New Orleans, LA, USA, January 15-17, 2003.

[34] J.-B. STEFANI. *A Calculus of Kells.* in « Proceedings 2nd International Workshop on Foundations of Global Computing », 2003.

[35] S. BEN ATALLAH, O. LAYAIDA, N. DE PALMA, D. HAGIMONT. *Dynamic Configuration of Multimedia Applications.* in « Proceedings of the 6th IFIP/IEEE International Conference on Management of Multimedia Networks and Services(MMNS'03) », Belfast, North Ireland, September 6-10, 2003.

Internal Reports

[36] E. CECCHET, J. MARGUERITE, W. ZWAENEPOEL. *RAIDb: Redundant Array of Inexpensive Databases.* Technical report, number RR-4921, INRIA, September, 2003, http://www.inria.fr/rrrt/rr-4921.html.

[37] J.-B. STEFANI. *A Calculus of Higher-Order Distributed Components.* Technical report, number RR-4692, INRIA, January, 2003, http://www.inria.fr/rrrt/rr-4692.html.

Miscellaneous

[38] T. Abdellatif. *Communications de groupe sur grappes.* Rapport de DEA Informatique : Systèmes et Communication (Master's thesis), École Doctorale Mathématiques et Informatique, Grenoble, June, 2003.

[39] É. Bruneton, T. Coupaye, J.-B. Stefani. *The Fractal Component Model.* ObjectWeb Consortium, September, 2003, http://fractal.objectweb.org/specification/.

[40] H. Huynh Quoc. *Intégration d'aspects non-fonctionnels par insertion de code dans l'environnement Java.* Rapport de DEA Informatique : Systèmes et Communication (Master's thesis), École Doctorale Mathématiques et Informatique, Grenoble, June, 2003.

[41] P. Marguerite. *Déploiement d'un intergiciel en environnement hétérogène à grande échelle.* Rapport de DEA Informatique : Systèmes et Communication (Master's thesis), École Doctorale Mathématiques et Informatique, Grenoble, June, 2003.

[42] V. Quéma. *État de l'art sur la reconfiguration dynamique dans les systèmes répartis.* Contrat RNTL INSIDE, livrable T3.1, October, 2003, http://sardes.inrialpes.fr/papers/reports.shtml.

[43] *Ozone Intermediate Status Report - Software Environment Layer.* Ozone Projet Deliverable D8b, June, 2003, http://www.extra.research.philips.com/euprojects/ozone/.

[44] *Coût et optimisation de la plate-forme.* Contrat RNTL ARCAD, livrable D1.4, September, 2003, http://arcad.essi.fr/.

[45] *Document de mise en œuvre de la plate-forme.* Contrat RNTL ARCAD, livrable D2.3, December, 2003, http://arcad.essi.fr/.

[46] *Synthèse des différentes approches permettant la description d'une configuration logicielle.* Contrat RNTL ARCAD, livrable D4.1, December, 2003, http://arcad.essi.fr/.

[47] *Spécification des différents services pour le déploiement d'une configuration logicielle.* Contrat RNTL ARCAD, livrable D4.2, March, 2003, http://arcad.essi.fr/.

[48] *Démonstrations : supervision et configuration dynamique.* Contrat RNTL ARCAD, livrable D4.3, December, 2003, http://arcad.essi.fr/.

Bibliography in notes

[49] M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, T. D. Chandra. *Matching Events in a Content-Based Subscription System.* in « Symposium on Principles of Distributed Computing (PODC) », pages 53-61, 1999, http://citeseer.nj.nec.com/aguilera99matching.html.

[50] T. E. Anderson, B. N. Bershad, E. D. Lazowska, H. M. Levy. *Scheduler Activations: Effective Kernel Support for the User-Level Management of Parallelism.* in « ACM Transactions on Computer Systems », number 1, volume 10, February, 1992, pages 53–79.

[51] *Research Projects in Autonomic Computing*. IBM Research, 2003, http://www.research.ibm.com/autonomic/research/projects.html.

[52] G. BLAIR, G. COULSON, A. ANDERSEN, L. BLAIR, M. CLARKE, F. COSTA, H. DURAN-LIMON, T. FITZPATRICK, L. JOHNSTON, R. MOREIRA, N. PARLAVANTZAS, K. SAIKOSKI. *The Design and Implementation of OpenORB v2*. in « IEEE Distributed Systems Online, vol. 2 no 6, Special Issue on Reflective Middleware », 2001.

[53] G. BOUDOL, F. GERMAIN, M. LACOSTE. *Analyse des modèles de programmation, plates-formes et langages – Tome I : Document d'analyse des langages et modèles de la mobilité*. Technical report, number RR-3930, INRIA Sophia Antipolis, April, 2000, http://www.inria.fr/rrrt/rr-3930.html, 73 pages.

[54] F. BUSCHMANN, R. MEUNIER, H. ROHNERT, P. SOMMERLAD, M. STAL. *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns*. John Wiley and Sons, 1995, 467 pp..

[55] A. CARZANIGA, D. S. ROSENBLUM, A. L. WOLF. *Design and evaluation of a wide-area event notification service*. in « ACM Transactions on Computer Systems (TOCS) », number 3, volume 19, 2001, pages 332–383, http://doi.acm.org/10.1145/380749.380767.

[56] D. ENGLER, M. KAASHOEK, J. O'TOOLE. *Exokernel: An Operating System Architecture for Application-Level Resource Management*. in « Proceedings of the 1995 ACM Symposium on Operating Systems Principles », pages 251–266, Copper Mountain Resort, CO, USA, December, 1995.

[57] P. TH. EUGSTER, P. FELBER, R. GUERRAOUI, A.-M. KERMARREC. *The Many Faces of Publish/Subscribe*. in « ACM Computing Surveys », number 2, volume 35, June, 2003, pages 114–131.

[58] E. GAMMA, R. HELM, R. JOHNSON, J. VLISSIDES. *Design Patterns: Elements of Reusable Object Oriented Software*. Addison-Wesley, 1994, 416 pp..

[59] A. G. GANEK, T. A. CORBI. *The dawning of the autonomic computing era*. in « IBM Systems Journal », number 1, volume 42, 2003, pages 5–17, Special issue on autonomic computing.

[60] *Report of the IST Advisory Group concerning Software Techologies, Embedded Systems, and Distributed Systems: A European Strategy towards an Ambient Intelligent environment*. Office for Official Publications of the European Communities, June, 2002, http://www.cordis.lu/ist/istag.htm.

[61] R. E. JOHNSON. *Frameworks=(Components+Patterns): How frameworks compare to other object-oriented reuse techniques*. in « Communications of the ACM », number 10, volume 40, October, 1997, pages 39–42.

[62] J. O. KEPHART, D. M. CHESS. *The Vision of Autonomic Computing*. in « Computer - IEEE Computer Magazine », number 1, volume 36, 2003, pages 41–50.

[63] G. KICZALES, J. DES RIVIÈRES, D. BOBROW. *The Art of the Metaobject Protocol*. MIT Press, 1991.

[64] G. KICZALES. *Aspect-Oriented Programming*. in « ACM Computing Surveys », number 4, volume 28, 1996, pages 154, http://doi.acm.org/10.1145/242224.242420.

[65] G. LEAVENS, M. S. (EDS). *Foundations of Component-Based Systems.* Cambridge University Press, 2000.

[66] G. MULLER, R. MARLET, E. VOLANSCHI, C. CONSEL, C. PU, A. GOEL. *Fast, Optimized Sun RPC Using Automatic Program Specialization.* in « Proceedings of the 18th International Conference on Distributed Computing Systems », IEEE Computer Society Press, pages 240–249, Amsterdam, The Netherlands, May, 1998.

[67] I. PIUMARTA, F. RICCARDI. *Optimizing direct threaded code by selective inlining.* in « 1998 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'98) », Montreal, Canada, 1998.

[68] D. C. SCHMIDT, M. STAL, H. ROHNERT, F. BUSCHMANN. *Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects.* John Wiley and Sons, 2000, 666 pp..

[69] C. SZYPERSKI. *Component Software - Beyond Object-Oriented Programming.* Addison-Wesley, 2002, 2nd ed., 589 pp..

[70] R. VAN RENESSE, K. P. BIRMAN, W. VOGELS. *Astrolabe: a Robust and Scalable Technology for Distributed Systems Monitoring, Management, and Data Mining* . in « ACM Transactions on Computer Systems », number 2, volume 21, May, 2003, pages 164–206.