



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Project-Team oasis*

*Objets Actifs, Sémantique, Internet et  
Sécurité*

*Sophia Antipolis - Méditerranée*

THEME COM

*Activity*  
*R* *eport*

2007



## Table of contents

<b>1. Team</b>	<b>1</b>
<b>2. Overall Objectives</b>	<b>2</b>
2.1. Presentation	2
2.2. Highlights	2
<b>3. Scientific Foundations</b>	<b>2</b>
3.1. Programming with Distributed Objects	2
3.2. Formal models for distributed objects	3
3.3. Static Analysis and Verification	3
<b>4. Application Domains</b>	<b>3</b>
4.1. Grid Computing	3
4.2. Service Oriented Architectures (SOA)	4
4.3. Electronic Business	5
<b>5. Software</b>	<b>5</b>
5.1. ProActive	5
5.2. Vercors platform	7
<b>6. New Results</b>	<b>7</b>
6.1. Distributed Programming Models	7
6.1.1. Theory of Distributed Objects	7
6.1.2. SPMD Programming for the Grid	8
6.1.3. Grid Branch and Bound Framework	8
6.1.4. Skeleton-based parallel programming	8
6.1.5. Component-oriented Grid Programming	9
6.1.6. Behavioural Specification for Reliable Distributed Components	10
6.2. Grid Middleware and Applications	11
6.2.1. First-class Futures and their Support in Fault-tolerance	11
6.2.2. Distributed Garbage Collector	11
6.2.3. Peer-to-Peer Computing Infrastructure	12
6.2.4. Standardisation of Deployment of Components and Objects	12
6.2.5. Grid Computing for Computational Finance	13
<b>7. Contracts and Grants with Industry</b>	<b>13</b>
7.1. Introduction	13
7.2. ITEA project S4ALL	13
7.3. AGOS	14
7.4. MobiTools	14
7.5. Thales Avionics Electrical Systems S.A.	14
<b>8. Other Grants and Activities</b>	<b>14</b>
8.1. National Collaborations	14
8.1.1. ACI Fiacre	14
8.1.2. ANR DiscoGrid	15
8.1.3. ANR GCPMF : Grille de Calcul pour les Mathematiques Financieres	15
8.1.4. ARC Automan	15
8.1.5. Contrat Plan État Région Grille et Calcul Pair-à-Pair	16
8.2. European Collaborations	16
8.2.1. NOE CoreGRID	16
8.2.2. IP Bionets	17
8.2.3. Strep GridComp	17
8.2.4. SSA NESSI-Grid	18
8.2.5. SSA EchoGrid	18
8.3. International Collaborations	18

8.3.1.	Stic-Amsud ReSeCo	18
8.3.2.	Collaboration with University of Sydney	18
8.3.3.	Stic Asia	18
<b>9.</b>	<b>Dissemination</b> .....	<b>19</b>
9.1.	Leadership within the Scientific Community	19
9.1.1.	Committees	19
9.1.2.	Miscellaneous	19
9.2.	Short Visits	19
9.3.	Teaching	19
9.3.1.	Teaching	19
9.3.2.	Theses	19
9.3.3.	Internships	20
9.4.	Participation in conferences and workshops	20
9.4.1.	Invited Talks	20
9.4.2.	Talks in workshops and Conferences	21
9.4.3.	Tutorials	21
9.4.4.	Miscellaneous	22
<b>10.</b>	<b>Bibliography</b> .....	<b>22</b>

# 1. Team

OASIS is an INRIA joint project with CNRS and University of Nice Sophia Antipolis, via the laboratory I3s (UMR 6070).

## Head of the team

Denis Caromel [ Prof. UNSA, delegation INRIA, HdR ]

## Vice-Head of the team

Eric Madelaine [ CR INRIA ]

## Administrative assistants

Patricia Lachaume [ I3S ]

Patricia Maleyran [ INRIA, since aug 10 ]

Christiane Gallerie [ INRIA, until aug 10 ]

## Research Scientist

Fabrice Huet [ MdC UNSA ]

Françoise Baude [ MdC UNSA, HdR ]

Ludovic Henrio [ CR CNRS ]

## PhD students

Alexandre Di Costanzo [ MESR, until oct 31 ]

Antonio Cansado [ INRIA, Conicyt ]

Brian Amedro [ SCS-Agos, since oct 01 ]

Elton Mathias [ INRIA-Cordi, since oct 01 ]

Guillaume Chazarain [ MESR ]

Imen Filali [ SCS-Agos / FP7 SOA4All, since sep 15 ]

Marcela Rivera [ INRIA, Conicyt ]

Mario Leyton [ INRIA, Conicyt ]

Muhammad Khan [ STIC-Asia, since oct 01 ]

Paul Naoumenko [ INRIA/Région PACA ]

Viet Dung Doan [ FP6 Bionets ]

## Post-doctoral fellows

Ian Stokes-Rees [ INRIA-Cordi, until nov 30 ]

Regis Gascon [ INRIA-Cordi, since oct 01 ]

## Student interns

Jean-Luc Scheefer [ master 2 ISI, since oct 01 ]

Johann Fradj [ master 2 ISI, since oct 07 ]

## Technical staff

Abhijeet Gaikwad [ since nov 15, ANR GCPMF ]

Arnaud Contes [ FP6 GridComp, since feb 01, until sep 30 ]

Cédric Dalmaso [ FP6 GridComp ]

Christian Delbé [ FP6 GridComp, until dec 31 ]

Virginie Legrand Contes [ FP6 S4All, FP6 NESSI-Grid, until sep 30 ]

Clément Mathieu [ IA-INRIA ]

Fabien Viale [ FP6 S4All, FP6 NESSI-Grid, until dec 31 ]

Germain Sigety [ INRIA, since sep 17 ]

Guillaume Laurent [ FP6 GridComp, since apr 02, until dec 31 ]

Romain Quilici [ since nov 01 ]

Vincent Cavé [ ANR DiscoGrid, since feb 19 ]

Yu Feng [ ETSI, FP6 EchoGrid ]

## Visiting Scientist

Nadia Ranaldo [ Univ. Sannio - CoreGrid, since sep 01, until sep 30 ]

Wasseim Al Zouabi [ Univ. Stuttgart - CoreGrid, until mar 31 ]

## 2. Overall Objectives

### 2.1. Presentation

The team focuses its activities on distributed (Grid) computing and more specifically on the development of secure and reliable systems using distributed asynchronous objects (active objects - OA of OASIS). From this central point of focus, other research fields are considered in the project:

- Semantics (first S of OASIS): formal specification of active objects with the definition of ASP (Asynchronous Sequential Processes) and the study of conditions under which this calculus becomes deterministic.
- Internet (I of OASIS): Grid computing with distributed and hierarchical components.
- Security (last S of OASIS): analysis and verification of programs written in such asynchronous models.

With these objectives, our approach is:

- theoretical: we study and define models and object-oriented languages (semantical definitions, equivalences, analysis);
- applicative: we start from concrete and current problems, for which we propose technical solutions;
- pragmatic: we validate the models and solutions with full-scale experiments.

Internet clearly changed the meaning of notions like mobility and security. We believe that we have the skills to be significantly fruitful in this major application domain; more specifically, we aim at producing interesting results for embedded applications for mobile users, Grid computing, peer-to-peer intranet, electronic trade and collaborative applications.

### 2.2. Highlights

- Creation of the ActiveEon startup – prize winner of the National Competition of Company Creation (CNCE 2007)s.
- Undergoing process of standardization of a Grid Component Model and its deployment: A format for the description of deployment over a Grid is finalized and has been submitted for standardization to ETSI.

## 3. Scientific Foundations

### 3.1. Programming with Distributed Objects

The paradigm of object-oriented programming, although not very recent, got a new momentum with the introduction of the Java language. The concept of object, despite its universal denotation, is clearly still not properly defined and implemented: notions like inheritance, sub-typing or overloading have as many definitions as there are different object languages. The introduction of concurrency into objects also increases the complexity. It appeared that standard Java constituents such as RMI (Remote Method Invocation) do not help building, in a transparent way, sequential, multi-threaded, or distributed applications. Indeed allowing, as RMI does, the execution of the same application to proceed on a shared-memory multiprocessors architecture as well as on a network of workstations (intranet, Internet), or on any hierarchical combination of both, is not sufficient for providing a convenient and reliable programming environment.

The question is thus: how to ease the construction, deployment and evolution of distributed applications ?

We have developed competencies in both theoretical and applicative side fields, such as distribution, fault-tolerance, and the construction of a Java library dedicated to parallel, distributed, and concurrent computing.

## 3.2. Formal models for distributed objects

In distributed object systems where concurrent processes co-exist, being able to prove properties such as confluence and determinism is a big step towards establishing functional correctness of those systems. It also greatly simplifies the reasoning about them.

A few years ago, we designed the ASP calculus for modeling distributed objects. It remains to this date one of our major scientific foundations. ASP is a calculus for distributed objects interacting using asynchronous method calls with generalized futures. Those futures naturally come with a transparent and automatic synchronization called wait-by-necessity. In large-scale systems, our approach provides both a good structure and a strong decoupling between threads, and thus scalability. Our work on ASP provides very generic results on expressiveness and determinism, and the potential of this approach has been further demonstrated by its capacity to cope with advanced issues, such as mobility, group communications, and components [8].

ASP thus provides confluence and determinism properties for distributed objects. Such results should allow one to program parallel and distributed applications that behave in a deterministic manner, even if they are distributed over local or wide area networks.

The ASP calculus is a model for the ProActive library. An extension of ASP models distributed asynchronous components.

## 3.3. Static Analysis and Verification

Even with the help of high-level libraries, distributed systems are more difficult to program than classical applications. The complexity of interactions and synchronizations between remote parts of a system increases the difficulty of analyzing their behaviors. Consequently, safety, security, or liveness properties are particularly difficult to ensure for these applications.

While research on formal verification of software systems has been active for a long time, its impact on the development methodology and tools has been slower than in the domain of hardware and circuits. This is true both at a theoretical and at a practical level, from the definition of adequate models representing programs, the mastering of state complexity through abstraction techniques or through new algorithmic approaches, to the design of software tools that hide to the final user the complexity of the underlying theory.

In the context of distributed component systems, not only do we get better descriptions of the structure of the system, making the analysis more tractable, but we also find out new interesting problems. For instance, we contributed to a better analysis of the interplay between the functional definition of a component and its possible runtime transformations, expressed by the various management controllers of the component system.

Our approach is bi-directional: from models to program, or back. We use techniques of static analysis and abstract interpretation to extract models from the code of distributed applications [3]. On the other hand, we generate “safe by construction” code skeletons, from high level specifications; this guarantees the behavioural properties of the components. We then use generic tools from the verification community to check properties of these models. We concentrate on behavioural properties, expressed in terms of temporal logics (safety, liveness), of security, of adequacy of an implementation to its specification and of correct composition of software components.

# 4. Application Domains

## 4.1. Grid Computing

**Keywords:** *Grid, Telecommunications, distribution, fault tolerance, group communication, mobile object systems, peer-to-peer, security, synchronization.*

As distributed systems are becoming ubiquitous, Grid computing is emerging as one of the major challenge for computer science: seamless access and use of large-scale computing resources, world-wide. The word "Grid" is chosen by analogy with the electric power grid, which provides pervasive access to power and has had a dramatic impact on human capabilities and society. It is believed that by providing pervasive, dependable, consistent and inexpensive access to advanced computational capabilities, computational grids will have a similar transforming effect, allowing new classes of applications to emerge.

Another challenge is to use, for a given computation, unused CPU cycles of desktop computers in a Local Area Network. This is intranet Computational Peer-To-Peer.

There is a need for models and infrastructures for grid and peer-to-peer computing, and we promote a programming model based on communicating and mobile objects and components.

## 4.2. Service Oriented Architectures (SOA)

**Keywords:** *Autonomic Management, Distributed Enterprise Service Bus, Distributed Service Composition, Large-Scale deployment and monitoring.*

Service Oriented Architectures aim at the integration of distributed services at the level of the Enterprise, or as proposed recently, of the whole the Internet. The Oasis team offers solutions to some of the problems encountered here:

- Deployment of a service on the service bus: as services depend upon other services, deployment and runtime management can be eased if these dependencies are made explicit. Indeed, services required for another service to work can be instantiated or discovered more easily if the dependencies are known. The recently defined Service Component Architecture (SCA) model is gaining popularity. We are starting to conduct research efforts to promote the Grid Component Model as an alternative or a complement to SCA. Indeed, we think that GCM is by essence well equipped for supporting services that are widely distributed, and may need to be invoked in an asynchronous manner. Additionally, GCM supports interoperability between component models like we succeeded to achieve it for CCA and GCM [39].
- Interoperability between services: the uniform usage of web services can provide a simple interoperability between them, unfortunately web-services only address interoperability in the case the service invocation model is remote procedure call (otherwise, more complex protocol translations would be needed). As GCM components can be exposed as web services [62] they are already ready to be deployed as services on an ESB (Enterprise Service Bus).
- large-scale deployment and monitoring of a set of (similar) services on the bus, and/or on a large set of machines or virtual machines (like we succeeded to do it for e.g. OSGi gateways [32]): such capability will really make SOA ready for the Internet scale.
- making the service bus large-scale and distributed: this is needed if the non-functional services that are needed for the bus to operate (e.g. discovery service) must be replicated and distributed. We intend to use GCM components for building the bus itself, giving it large-scale capabilities (this subject will be addressed through the SOA4ALL IP FP7 project to which the team is involved, around the OW2 Petals ESB itself based on Fractal and Fractal RMI distributed components)
- self-management of the SOA infrastructure: this pertains to autonomic and self-management of the ESB or more specifically the Enterprise Server itself. We highlighted during the work conducted in the ARC Automan the need for the management system to be itself more large-scale, grid aware. Again the use of GCM components instead of Fractal-RMI components whenever needed can be a solution to the scalability problem

Finally, a very challenging problem in the SOA field is posed by the need to have dynamic, adaptable and even autonomic service compositions, and consequently adequate choreography and orchestration tools. The capability of GCM components to be composed dynamically as composite components is our starting point for an alternative to standard workflow models: GCM openness wrt non-functional features will be the basis to equip GCM services with autonomic, bio-inspired composition strategies. This is particularly addressed within the BIONETS IP project.



### 4.3. Electronic Business

**Keywords:** *formal methods, mobility, program analysis, proofs, security, telecommunications.*

Here, “electronic business” stands for distributed applications over the Internet that require safety and security. Without strong guarantees on confidentiality, privacy, integrity, authentication and availability, such applications cannot be deployed, because of too high a risk

We give examples of such applications:

- Secure electronic commerce: such applications distributed over networks may contain very complex behaviors. This may lead to deadlocks, starvation, and many other kinds of reachability or liveness problems. It is necessary to propose methods for specifying the application behaviour, its demands, and tools to check the implementation against those requirements. On the other hand, protection of communications and data is a requirement for the development of commercial applications. These security requirements have to be expressed in a security policy agreed by all partners, including customers.
- Secure collaborative applications: a multi-site enterprise may want to use Internet for the communication between different services and the collaborative building of a particular task, leading to specific problems of election, synchronization, load balancing, etc.
- Mobility for enterprise applications: a mobile worker should be able to run professional applications from anywhere, using heterogeneous network, and any device (desktop, laptop, PDA, board computer) in a transparent and a secure manner.

## 5. Software

### 5.1. ProActive

**Participants:** F. Baude, D. Caromel, G. Chazarain, A. di Costanzo, C. Delbé, F. Huet, V. Legrand, M. Leyton, B. Amedro, C. Mathieu, F. Viale, C. Dalmaso, A. Contes, G. Laurent.

ProActive is a Java library (Source code under LGPL license) for parallel, distributed, and concurrent computing, also featuring mobility and security in a uniform framework. With a reduced set of simple primitives, ProActive provides a comprehensive API allowing to simplify the programming of applications that are distributed on a Local Area Network (LAN), on cluster of workstations, or on Internet Grids.

The library is based on an Active Object pattern that is a uniform way to encapsulate:

- a remotely accessible object,
- a thread,
- an actor with its own script,
- a server of incoming requests,
- a mobile and potentially secure agent.

and has an architecture to inter-operate with (de facto) standards such as:

- Web Service exportation,
- HTTP transport,
- ssh, rsh, RMI/ssh tunneling,
- Globus: GT2, GT3, GT4, gsi, gLite, Unicore, ARC (Nordugrid)
- LSF, PBS, Sun Grid Engine, OAS.

ProActive is only made of standard Java classes, and requires **no changes to the Java Virtual Machine**, no preprocessing or compiler modification; programmers write standard Java code. Based on a simple Meta-Object Protocol, the library is itself extensible, making the system open for adaptations and optimizations. ProActive currently uses the RMI Java standard library as default portable transport layer, but others such as Ibis or HTTP can be used instead, in an adaptive way.

ProActive is particularly well-adapted for the development of applications distributed over the Internet, thanks to reuse of sequential code, through polymorphism, automatic future-based synchronizations, migration of activities from one virtual machine to another. The underlying programming model is thus innovative compared to, for instance, the well established MPI programming model.

In order to cope with the requirements of large-scale distributed and heterogeneous systems like the Grid, many features have been incorporated into ProActive such as:

- The deployment infrastructure that supports almost all Grid/cluster protocols: LSF, PBS, SGE, ssh, Globus, Unicore ...;
- The communication layer that can rely on RMI or HTTP or IBIS, or SOAP or RMI/ssh. This last protocol allows one to cross firewalls in many cases;
- The component framework which implements the ObjectWeb Fractal hierarchical component model is now mature, and is being extended with collective interfaces for targeting (parallel) grid components;
- The graphical user interface IC2D offers many other views of an application, for instance the Job monitor view, that allows better control and monitoring;
- The ability to exploit the migration capability of active objects, in network and system management;
- A computational P2P infrastructure;
- Object Oriented SPMD programming model with its API;
- Distributed and Non-Functional Exceptions handling;
- Fault-Tolerance and Checkpointing mechanisms;
- File Transfer capabilities over the Grid;
- A generic task scheduler;
- ProActive connectors for remote JMX-based operations and an OSGi compliant version of the ProActive library, this involved the development of a “bundled” version of the library.

We have demonstrated on a set of applications the advantages of the ProActive library, and among other we are particularly proud of the following results, showing that portable and transparent Java code can compete with specific optimized approaches:

- NQueen challenge, where we equaled the world record  $n=24$  (227 514 171 973 736 solutions) in 17 days based on ProActive’s P2P infrastructure (300 machines).
- NQueen challenge, where we get the world record  $n=25$  (2 207 893 435 808 352 solutions) in 6 month based on ProActive’s P2P infrastructure using free cycles of 260 PCs.

ProActive is a project of the ObjectWeb Consortium. ObjectWeb is an international consortium fostering the development of open-source middleware for cutting-edge applications: EAI, e-business, clustering, grid computing, managed services and more. For more information, refer to [10] and to the web pages <http://www.objectweb.org> and <http://www.inria.fr/oasis/proactive> which list a lot of white papers.

#### **Distributed Management Tool Based on ProActive**

In the RNRT PISE project, we designed and implemented a remote management tool for OSGi, based on ProActive and on the JMX (Java Management eXtension) management standard relying on typed groups. Our objective is to go further in the “bundlelisation” of the ProActive environment: besides the capability to host a ProActive runtime on an OSGi gateway and run classic distributed active-object oriented applications, the focus is to provision, run and monitor component-oriented ProActive/GCM applications.

## 5.2. Vercors platform

**Participants:** A. Cansado, E. Madelaine, E. Salageanu, S. Ahumada.

Vercors is a verification platform for distributed components. It comprises several tools covering the whole process of verification. Rather than creating a new model-checker, our model-generator produces input for state-of-art tools to check component specifications.

The Vercors tools include front-ends for specifying the architecture and behaviour of components in the form of UML diagrams. We translate these high-level specifications, into behavioural models in various formats, and we also transform these models using abstractions. In a final step, abstract models are translated into the input format for various verification toolsets. Currently we mainly use the various analysis modules of the CADP toolset.

We propose two alternative flows from specification to model-checking: in the first case, we start with UML-2.1 diagram editors, describing the architecture (component structures) and the behaviour (state-machines); from these diagrams we produce internal forms of the TTool software<sup>1</sup>, and we use TTool as a back-end to connect to the CADP model-checking modules. The first prototype was built by Emil Salageanu and Solange Ahumada, and published in [25]. In the same paper, we describe ideas for extending this work by including specific features of the GCM component model. A second version has been achieved recently, that is completely integrated as a set of Eclipse plugins [58]. This gives us a better integration of the code, allowing much better reporting of errors to the developer, and will be the base for future expansion of the specification language.

In the second case, we start at a lower level of specification, using directly the Fractal-ADL as a description of the architecture, and Lotos as a language for expressing the behaviour of primitive components. From these, the ADL2N tool creates two files: a pNet in hierarchical Parameterized FC2 format expressing the synchronisations between components and an instantiation file defining the domain abstractions. We have enhanced ADL2N this year with basic functionalities for the abstraction of user data types [59].

## 6. New Results

### 6.1. Distributed Programming Models

#### 6.1.1. Theory of Distributed Objects

**Participant:** L. Henrio.

This work extends results published in [8]. The  $\zeta$ -calculus, and its semantics were published by Abadi and Cardelli [60]. In collaboration with Florian Kammüller (Technische Universität Berlin) we formalized the  $\zeta$ -calculus, and its reduction semantics in the Isabelle theorem prover. Then we proved the confluence property of this calculus, and published the results in [38]. We also proved the type-safety property of  $\zeta$ -calculus in Isabelle/HOL.

This provided us with the right framework to start designing a distributed functional calculus, based on ASP:  $ASP_{fun}$ .  $ASP_{fun}$  features distributed functional objects interacting through asynchronous method calls and futures. They evaluate concurrently the requests they receive, and return results completely or partially evaluated. We designed a type system for  $ASP_{fun}$ , and established for it the generic “progress property”, which ensures the *absence of deadlock*. We think this result could be used in the design of service-oriented (distributed) architectures communicating with requests-futures. These results have been first published as a research report [48].

All these results have been formalized and proved in the Isabelle theorem prover.

---

<sup>1</sup>from Ludovic Aprvile, LabSoc, ENST Sophia-Antipolis, <http://labsoc.omelec.enst.fr/turtle/>

### 6.1.2. SPMD Programming for the Grid

**Participants:** B. Amedro, D. Caromel.

The SPMD (Single Program, Multiple Data) programming model is a common way to develop parallel applications. An SPMD program is characterized by a single program which is written and executed onto each node of a cluster, and runs independently. Then, each copy of the program runs concurrently on different sets of data. All these copies can be view as processes which are members of a same group: an *SPMD group*.

By overlapping communication and computation as we can see with the *ForgetOnSend* contract, and parallelizing independent tasks as proposed with the *synchronization schema*, the work presented in [43] proposes some solutions to simplify the development of an SPMD distributed application. To evaluate the performance impact of those proposals, we have performed some experiments on the national Grid5000 grid. Also, some of those solutions, such as the *ForgetOnSend* contract can be extended to a larger use case than the SPMD programming model.

### 6.1.3. Grid Branch and Bound Framework

**Participants:** A. di Costanzo, D. Caromel.

In [34], we present *Grid'BnB*, a parallel branch and bound framework for grids. Branch and bound (B&B) algorithms find optimal solutions of search problems and NP-hard optimization problems. Because of the size of problems and their NP-hardness, it is usually difficult to find optimal solutions and it may be impossible with standard resources such as desktops and clusters. On the other hand, grids provide a large amount of computational resources. Thereby we propose a new B&B framework for grids to solve optimization problems.

*Grid'BnB* is a Java framework that helps programmers to distribute problems over grids by hiding distribution issues. It is built over a master-worker approach and provides a transparent communication system among tasks. This work also introduces a new mechanism to localize computational nodes on the deployed grid. With this mechanism, we can determine if two nodes are on the same cluster. This mechanism is used in *Grid'BnB* to reduce inter-cluster communications. We ran experiments on a nationwide grid. With this testbed, we analyzed the behavior of a highly communicating application deployed on a large-scale grid. This application solves the flow-shop problem.

### 6.1.4. Skeleton-based parallel programming

**Participants:** D. Caromel, L. Henrio, M. Leyton.

The structured parallelism approach (skeletons) takes advantage of common patterns used in parallel and distributed applications. The skeleton paradigm separates concerns: the distribution aspect can be considered separately from the functional aspects of an application.

Specifications that exhibit structured patterns can benefit from libraries or from programming languages that support skeletons. The goal here is that some day, the skeleton libraries will be able to handle the complex attributes of Grid programming: heterogeneity, dynamicity, adaptability, etc.

In 2006 we began designing Calcium, a framework for programming with algorithmic skeletons. Calcium has been implemented in Java, and provides skeleton programming as a Java library. The patterns currently available in Calcium are:

- **farm** Also known as master-slave is used for task replication.
- **pipe** Is used for sequenced computation, where different stages of computation must be executed one after the other.
- **if** Represents dynamic conditional evaluation.
- **while** Represents iteration computation combined with conditional evaluation.
- **for**, Represents iteration computation.
- **d&c**, The divide and conquer patterns corresponds to data parallelism, where a task is subdivided into smaller problems, the problems are solved, and then conquered to achieve the results.
- **map**. Also corresponds to data parallelism, and represents a particular case of d&c, where the division and conquer is only performed once.

An important feature of the Calcium framework is that these patterns can be nested to solve more complex applications.

The Calcium framework is built on top of the ProActive middleware, in order to feature distributed computation. Therefore, Calcium takes advantage of ProActive's deployment framework for performing resource acquisition on the Grid, and uses ProActive's active object model for communication.

This year, we made the Calcium framework progress in several directions. We proposed a mechanism to perform fine tuning of algorithmic skeletons' muscle code. The approach extends previous performance diagnosis techniques that take advantage on pattern knowledge by: taking into consideration nestable skeleton patterns, and relating the performance inefficiency causes with the skeleton's responsible muscle code. This is necessary because skeletons provide a higher-level programming model, and as such, low level causes of performance inefficiencies have no meaning to the programmer. The proposed approach can be applied to fine tune applications that are composed of nestable skeleton patterns. The relation of inefficiency causes with the responsible muscle code is found by taking advantage of the skeleton structure, which implicitly informs the role of each muscle code. The results of these proposal were presented at the international Euro-par 2007 conference [36].

This year, we also defined a type system for algorithmic skeletons. We tackled this problem from both a theoretical and a practical approach. On the theoretical side we contributed by: formally specifying a type system for algorithmic skeletons, and proving that this type system guarantees that types are preserved by reduction. Type preservation guarantees that *skeletons can be used to transmit types between muscle functions*.

On the practical side, we have implemented the type system using Java and "generics" feature. The type enforcements are ensured by the Java type system, and reflect the typing rules introduced in the theoretical section. Globally, this ensures the correct composition of the skeletons. As a result, we have shown that: no further type-casts are imposed by the skeleton language inside muscle functions; and most importantly, *type errors can be detected when composing the skeleton program*.

The results of the type system for algorithmic skeletons will be presented at the international Euromicro-PDP 2008 conference [35].

### 6.1.5. Component-oriented Grid Programming

**Participants:** F. Baude, D. Caromel, A. Cansado, C. Dalmaso, L. Henrio, E. Madelaine, E. Mathias, P. Naoumenko, A. Cansado, M. Rivera.

Here, our objective is to simplify the design, programming and evolution (adaptation) of distributed Grid applications. In particular, we have defined parallel and hierarchical distributed components starting from the Fractal component model developed by INRIA and France-Telecom [64]. We are involved in the design of the Grid Component Model (GCM) [65], which is one of the major results produced by the CoreGrid European Network of Excellence. The GCM is intended to become a standard for Grid components, and most of our research on component models are related to it. The GCM is an extension of the Fractal model. On the practical side, ProActive/GCM is a prototype implementation of the GCM above the ProActive library; currently, not all GCM features are implemented in ProActive yet. ProActive/GCM is intended to become the reference implementation of the GCM, as is the goal of the European project GridComp. In 2007, we made a lot of efforts consolidating and improving the usability of the implementation of the GCM in ProActive.

#### Designing Non-Functional Concerns as Components

As part of the design of the GCM, we started research on the componentization of the component membranes (a membrane encapsulates the component control). This consists in adopting a component view of the non-functional and control aspects, in the same way the component model structures the functional concerns. This contribution should result in a powerful model for the design and adaptation of components control. We built a prototype introducing components inside the membranes of Fractal/GCM components. The advantages of this approach are a better structuring of non-functional aspects, and better reconfiguration possibilities. Two publications resulted from this work [31], [30]. The reference implementation with ProActive is well advanced. The BIONETS European projects aims at building autonomous services inspired by biology. In this context,

the work on componentized membranes is used for dynamic composition and evolution of services. Indeed, a plan of composition and several evolution strategies can be designed as component systems inside the non-functional part of services. The GCM component model also proposes to define autonomous components, having inside their membrane complex strategies to self-adapt to changes of the context. We plan in the future to implement several such examples of applications with autonomous entities, dynamic composition and evolution.

### **Stopping and Reconfiguring Distributed Components**

As part of her thesis, Marcela Rivera designed and implemented an algorithm to stop a component system. This algorithm is decomposed in two phases. The first phase consists in preparing the component system for the shutdown process. The second phase consists in (recursively) synchronizing the hierarchical stop all subcomponents and, then, the system is stopped in a bottom-up fashion, starting by the primitive components. The conditions required for this algorithm are fairness, and absence of dead-locks. One of the interesting properties of this algorithm is that it is able to handle the presence of futures.

More generally, this work is set in the context of non-functional aspects and reconfigurations in a component system. In the future, we plan to generalize the first results obtained this year in order to:

- study the issues related to asynchrony in the component model frameworks, and more precisely issues related to the active object model, and to the sequentiality in each component.
- identify the deadlocks, inconsistency, or impossibility, and find solutions to perform any kind of reconfiguration.
- design a set of high-level reconfiguration primitives allowing to achieve complex operations, but also to trigger such operations on specific events.

### **SPMD Components**

In the context of the DiscoGRID ANR funded project, we have started to use those collective invocations in order to design a MPI-like hierarchical SPMD programming model [40]. Indeed, when programming a Grid, it appears that, for good performance purposes, it would be useful to take into consideration the physical topology, which is in fact a hierarchy of multi-processors, multi-clusters on multi-sites and possibly on multiple grids. This work is done in collaboration with applied mathematicians (namely the CAIMAN and SMASH teams, partners of the DiscoGRID project). Indeed, they represent a community where programmers are used to the standard SPMD message-passing based model, and that is quite reluctant to adopt another model. Nonetheless, they are ready to design and program their parallel algorithms in a way that takes the physical hierarchy into consideration. The solution we are exploring within DiscoGRID is to rely on a GCM-based support in order to organize the MPI application in a hierarchical way, but in a transparent way for programmers.

## **6.1.6. Behavioural Specification for Reliable Distributed Components**

**Participants:** A. Cansado, L. Henrio, E. Madelaine, R. Gascon, E. Salageanu, S. Ahumada.

We have continued our work on the specification and verification of distributed components, both at the formal and methodological level, that we describe here, and at the practical level, that is described in the Software section 5.2.

At the level of specification languages for distributed components, we made two main contributions this year: The first is the JDC (Java Distributed Components) specification language, defined in the context of A. Cansado PhD work, and that was used for the CoCoME case-study [19]. The second, in collaboration with Ludovic Apvrille from Eurecom, builds on our work on UML diagrams for Fractal components; in [25] we have described new versions of this UML-based formalism, both for synchronous Fractal components, and for asynchronous GCM components. Those graphical formalisms maps to subsets of the JDC language.

The CoCoME case-study has been performed in the context of a European team collaboration, aiming at comparing a number of modeling approaches for component systems. It is a realistic and large application for the management of distributed cash-desk lines. We have applied both JDC and UML diagrams to specify the CoCoME, and used the Vercors environment to check its behavioural properties [19].

The ADL2N tool is our central module for the generation of behavioural models for Fractal/GCM components. This year we have formalised ([59]) the structure and the algorithms required for the generation of Fractal controllers in these models, leading to a new new definition of the pNets API, and first implementation of the controllers generation in the tools.

From the same architectural and behavioural descriptions, we have started defining methods for the generation of GCM/ProActive code. More specifically, all control parts of the code (the request service policy and all remote communications for the component) can be generated from the specification, leaving only the business methods code to be written by the developer. This lead to the guarantee that the properties proven on the model will be valid on the implementation. First reports on this approach are reported also in [19].

Last we have started a longer term research, aiming at the integration of novel model-checking engines in our framework. The idea is to used so-called “infinite system” methods, when this is required, to check properties of parameterized systems without using finite abstractions. This should apply typically to properties involving integer counters, or regular structures like fifo queues.

## 6.2. Grid Middleware and Applications

### 6.2.1. *First-class Futures and their Support in Fault-tolerance*

**Participants:** D. Caromel, M. U. Khan, L. Henrio, C. Delbé.

We analyzed the automatic continuation mechanism (a mechanism for updating first class futures) of ProActive with the aim of integrating it with the CIC protocol for fault tolerance that is already implemented. We identified some limitations in the existing Automatic continuation mechanism and proposed solutions to overcome them. We identified problems like Early Reply, duplicate Continuations, etc. We implemented these modifications and performed benchmarks for evaluating their performance using the Timit API for benchmarking ProActive applications. CIC protocol was also studied and we proposed a number of modifications to enable an integration of Automatic continuation and CIC protocol for fault tolerance [49].

Currently our focus is on defining an easy to use formal abstraction for Automatic Continuation in ProActive. This would help in better understanding automatic continuations. In addition, we aim to carry out different experiments for a comparative analysis of different automatic continuation. Our long term goal is the integration of automatic continuation with CIC protocol and incorporating support for non-fifo services in the protocol.

### 6.2.2. *Distributed Garbage Collector*

**Participants:** D. Caromel, G. Chazarain, L. Henrio.

With the increasing use of active objects, manual termination of these objects is becoming a burden. Distributed garbage collectors are a solution to this problem. Currently existing garbage collectors have at least one of these limitations:

- need of a central server (not peer to peer)
- unbounded message size
- only acyclic garbage collection

Guillaume Chazarain designed and implemented a distributed garbage collector solving all of these limitations. It is based on a heartbeat between an active object and its children in the object references graph. The heartbeat message is  $O(1)$  in size and the locally computed reply to this message is used in the algorithm.

Acyclic garbage collection is simple, an active object knows it will receive a heartbeat message at a specified frequency from its referencers. So, if no message is received during a certain amount of time, the active object considers itself to be garbage and terminates.



Cyclic garbage collection is where the complexity is. Unlike local garbage collectors and most distributed garbage collectors, we don't attempt to prove that a cycle is unreachable. Instead we try to find a strongly connected component in the active objects graph where every active object is waiting for a request. This component is clearly unreachable garbage although we didn't use its unreachability property to find it. Once such a component is found, one of its member terminates itself. The outcome is either some easily collectable acyclic garbage, or a smaller strongly connected component where the algorithm can restart.

Strongly connected components are found by making a consensus on the last activity in the component. The last activity is maintained as a Lamport clock with the owner associated. It is incremented on a:

- State change (Busy  $\rightarrow$  Idle)  $\Rightarrow$  it may be the last activity
- Loss of a child  $\Rightarrow$  it may be the parent in the spanning tree
- Loss of a parent  $\Rightarrow$  it may be the owner of the last activity

The last activity is propagated using the heartbeat messages from the parents to their children. The children use the reply to the heartbeat message to make a consensus with their parents about the last activity in the component. When such a consensus is made, the owner of the activity terminates itself. The reply to the heartbeat message is in fact a traversal of a spanning tree covering the strongly connected component.

The distributed garbage collector has been implemented and successfully tested on small scale configurations. Then, large scale experiments have been made to validate the approach. The result was published in [33].

As part of this development, some insight was needed on the topologies of object references in applications. So a plugin for IC2D was made to extract the object references graph from active objects and graphically represent it.

### 6.2.3. Peer-to-Peer Computing Infrastructure

**Participants:** I. Filali, F. Huet.

We designed a P2P infrastructure implemented on top of the ProActive Java library, allowing the provision of computational nodes for distributed applications. Computational nodes are indeed Java Virtual Machines (JVM), which are located on LANs and clusters. This infrastructure is totally self-organized and fully configurable. The creation and the maintenance of this network of JVMs is based on exchange of messages between peers. Each peer of the infrastructure is an active object, which has a list of nodes. These nodes are acquired by the applications that use the P2P network. According to the power of the machine, a peer can provide one or several nodes.

We are currently working on the infrastructure to improve its performance and usability. The first step is to redesign the communication layer to use a push-pull model. Peers offering resources advertise them on the network and this information is stored in caches. Using these caches, we can reduce the number of messages necessary to find available resources. An article presenting this work has been submitted for publication. In a second step, we will study the resource description to have a better description of resources, including static (e.g. CPU frequency) and dynamic (e.g. load) parameters.

### 6.2.4. Standardisation of Deployment of Components and Objects

**Participants:** D. Caromel, C. Matthieu, G. Laurent.

The existence of several different grid middleware platforms or job scheduler calls for a standardisation effort in the description of the application being deployed and the grid structure it is deployed on. Based on our previous experience with the Proactive deployment descriptor, a complete redesign of ProActive's deployment framework has been started. A deployment description is now split over two files, one describing the grid itself, and the other describing the application. This allows the reuse of the grid description for any other application using the same grid. The format for the grid description file is finalized and has been submitted for standardisation to ETSI. The implementation of the new deployment framework is nearing completion, and will be part of the next release of ProActive.



### 6.2.5. Grid Computing for Computational Finance

**Participants:** F. Baude, V. D. Doan, I. Stokes-Rees.

Computation in financial services includes the over-night calculation and time-critical computations during the daily trading hour. Academic research and industrial technical reports have largely focused on over-night computing tasks and the application of parallel or distributed computing techniques. This work has instead focused on time-critical computations required during trading hours, in particular Monte Carlo simulation for option pricing and other derivative products. We have designed and implemented a software system called PicsouGrid which utilizes the ProActive library to parallelize and distribute various option pricing algorithms. Currently, PicsouGrid has been deployed on various grid systems to evaluate its scalability and performance in European option pricing. We also developed several European option pricing algorithms such as standard, barrier, basket options to experiment in PicsouGrid. A part of this work was presented in [63] and [26]. In the second half of 2007 several American option pricing algorithms have been implemented (Longstaff and Schwartz, Ibanez and Zapatero, and Picazo), including those which price options on several assets simultaneously (called basket options). Due to the terms of American options the algorithms have a much higher computational demand, and therefore complicated strategies are employed to improve the efficiency of the option pricing estimate, which in turn complicates the implementation of a parallelization strategy. Our work has focused on finding efficient parallelization strategies which can be used for a range of pricing algorithms. The objective is to allow algorithm designers to focus on an efficient serial implementation without concern for the parallelization, and for the model to be used to automatically or semi-automatically provide a load-balanced (for heterogeneous compute resources) parallel implementation. To date we have only exercised the Ibanez-Zapatero algorithm using PicsouGrid, using the OASIS desktop cluster to get the performance benchmarks. Further experimentation will be done on the remaining algorithms and at a larger scale (greater than 30 workers). The extended abstract of this work can be found in [37]. The need for parallel Monte Carlo simulators has driven the need to evaluate high quality parallel random number generators in Java, a subject we are currently investigating. We have a strong collaboration with Mireille Bossy in the TOSCA team in the aspect of mathematics and numerical simulations. We also investigate the Quasi Monte Carlo methods in option pricing. Objective is toward to a distributed Monte Carlo framework in computational finance field.

## 7. Contracts and Grants with Industry

### 7.1. Introduction

By nature, several research contract we are involved in and that are detailed in Sections 8.1 and 8.2 involve industrial partners, we focus in this section on projects where industrial partners play significantly a major role relatively to academic ones.

First, as a small collaboration involving an industrial partner, the thesis of Paul Naoumenko is situated in the context of a collaboration with France Telecom, interactions mainly occur on the domain of the M2M (Machine to Machine architecture).

### 7.2. ITEA project S4ALL

S4ALL (Services for All) is an ITEA project, leaded by Alcatel CIT Research and Innovation. INRIA is also member of this project, through the involvement of several of its teams or hosted teams: ObjectWeb, SARDES, JACQUARD and OASIS. The global aim of the project is to explore the technical solutions, and consolidate the existing ones that may appear suitable for building the following vision: *A world of user-centric services that are easy to create, share and use.* Our contribution in this project is mainly on the usage of the ProActive platform, that should be helpful for programming and deploying distributed services, and publish them on the service bus. Moreover, as services may be deployed everywhere, including on scarce-resource devices, we target not only standard JVMs, but also OSGi platforms.

The project is built around three sets of partners: 6 large industrial companies (Alcatel CIT Research and Innovation, Bull, Nokia, Schneider Electric, Thales and Vodafone), 3 SMEs (Capricode, mCentric, Xquark), and 6 academic partners (Fraunhofer Fokus, Helsinki Institute for Information Technology, INRIA, INT, Univ. Joseph Fourier IMAG, Univ. Politecnica de Madrid).

Our involvement within this ITEA project contributed to the definition of a generic tool for the deployment, execution and management of OSGi service based applications and OSGi gateways. As these operations must scale we focussed on the use of the ProActive Grid programming model, specifically relying on its typed group communication: ProActive serves as the remote entry point for management operations and consequently relies on an interface with the JMX monitoring standard. This ProActive based JMX connector has proven also very valuable for introducing better scalability in the IC2D ProActive monitoring application.

This project started in July 2005, for 24 months, for a total of 129 kEuros.

### 7.3. AGOS

“Architecture de Grille Orientée Services”; labeled by the *pôle de compétitivité* SCS (“Solutions Communicantes Sécurisées”), and financed *FCE Ministère de l’Industrie* (from October 2007 to March 2010).

The AGOS project aims at building a service oriented Grid architecture. The objective is to design an automated and integrated managing environment for business and computing infrastructure of companies. This environment is based on the secured collaboration of software components and distributed resources. It fits with the needs of the companies that face a constantly evolving environment. The project will use the Open-source middleware ProActive which implements the European Grid Component Model (GCM).

The partners of the project are: HP, OASIS (INRIA-UNSA-CNRS), the startup: ActiveEon issued from OASIS, Oracle, Amadeus, Canal de Provence.

### 7.4. MobiTools

The MobiTools project, labeled by the *pôle de compétitivité* Solutions Communicantes Sécurisées is a 2 years project ending Dec. 2008, involving SMEs from the PACA region, and INRIA funded by a regional grant. In this context, MobileDistillery and OASIS are working on an integrated solution for ProActive active objects and objects running on mobile devices with only J2ME to be able to communicate. Furthermore, we aim to emulate the mobility of active objects from a ProActive runtime to a J2ME support and vice versa. Such integration will permit to devise distributed and mobile Java applications in a seamless manner whatever the underlying runtime supports. We have just started the above described research.

### 7.5. Thales Avionics Electrical Systems S.A.

Thales Avionics Electrical Systems designs and produces electrical power generation systems for aircraft and is a world leader for both commercial and military applications. They are using softwares (i.e. Mathworks’ Matlab/Simulink, Synopsys’ SABER) in order to model and simulate electrical power systems in different critical conditions. The contract aims at using ObjectWeb ProActive in order to speed up simulations that can last several hours.

This contract started in September 2007 for 12 months, and has a budget of 120 kEuros.

## 8. Other Grants and Activities

### 8.1. National Collaborations

#### 8.1.1. ACI Fiacre

Fiacre stands for “Flabilité des Assemblages de Composants REpartis: Modèles et outils pour l’analyse de propriétés de sécurité et de sureté”; Fiacre is a software project of the French *Action Concertée Incitative* (ACI) *Sécurité Informatique* of the Ministry of Research. This project started in September 2004, for 3 years, involving 97 kEuros.

Gathering teams specialized in behavioural specifications of components, languages and models for distributed, mobile, and the programming of communicating application, and compositional verification, the goal of FIACRE is to design methods and tools for specification, model extraction, and verification of distributed, hierarchical, and communicating components. We would like the collaboration to result in a software prototype applicable to real applications.

Members of this project are: INRIA Oasis (coordinator) and Vasy teams, Fera/SVF, and ENST/ILR.

Our contribution is centered on the Vercors platform, including our new UML-based specification environment, and methods and tools for building parameterized models of distributed components, [25], [19].

The Fiacre ACI was finished in sept'07, and its results reported at the Colloque STIC, Paris, nov'07.

### 8.1.2. ANR DiscoGrid

This ANR funded project gather partners that are applied mathematicians (OMEGA/NACHOS and SMASH teams) and computer scientists researching in distributed and grid programming environments (OASIS, PARIS, LaBRI SoD, MOASIS).

The DiscoGRID project aims at defining a new SPMD programming model, suited to High Performance Computing on Computational Grids. Grids are hierarchical in nature (multi-CPU machines, interconnected within clusters, themselves interconnected as grids), so the incurred latency for inter processes communication can vary greatly depending on the effective location of the processes. The challenge is to define a programming model that should allow programmers to exploit this hierarchy, as easily and efficiently as possible. As the MPI SPMD message-passing model is very popular in High Performance Computing we are defining a hierarchical extension of MPI. To address Grid hierarchy and high dynamicity, this new MPI implementation will rely on GCM components, organized hierarchically as composite components in order to reflect the effective deployment of the MPI application on the grid [61].

This project started in January 2006, for 36 months, involving 110 kEuros.

### 8.1.3. ANR GCPMF : Grille de Calcul pour les Mathematiques Financieres

We collaborate with the OMEGA/TOSCA team within the ANR project entitled "GCPMF" funded by the ANR Research Program "Calcul Intensif et Grilles de Calcul 2005".

Financial applications require to solve large size computations, that are so huge that they can not be tackled by conventional PCs. A typical example addresses risk analysis evaluation as periodically run by financial institutions (like VaR – Value at Risk –, and also market risks: greeks, duration, beta, ...). Parallelism is already applied in this financial context, but usage of Computing Grids is far from being mastered.

The aim of this ANR program is to highlight the potential of parallel techniques applied to mathematical finance computing on Grid infrastructures. The consortium that conducts this project includes ten participants from academic laboratories in computer science and mathematics, banks, and IT companies.

This year, in collaboration with Mireille Bossy from TOSCA, and with Stéphane Vialle and his team from Supélec, we have continued the design and implementation of a Grid software architecture based on ProActive, named PicsouGrid [63], and in particular on typed groups. Besides parallel algorithms for the pricing of European options by Monte Carlo Methods, we have put a specific effort to also integrate some parallel algorithms of American pricing methods that we aim at evaluating for being good candidates for gridification [37].

This project started in January 2006, for 36 months, and a total budget of 44 kEuros.

### 8.1.4. ARC Automan

The AutoMan project aims at devising a new breed of autonomic protocols to attain high scalable and available multi-tier J2EE enterprise server replication deployed on grid systems. Grid technology will provide the infrastructure to exploit a pool of available servers to attain self-provisioning. Autonomic management will continuously tune the system to maximize performance and availability without involving human administrators.

The members of this project are:

- INRIA Rhône-Alpes – SARDES research group,
- INRIA Sophia Antipolis – OASIS research group,
- and Technical University of Madrid – LSD research group.

The AutoMan project started in March 2006 for a length of 2 years, with a budget managed by the Sardes team.

The objective of the Oasis team in the Automan project is to explore the possibility to apply the ProActive technology to the Gridification of duplicated and self-administrated J2EE enterprise servers. The JADE tool, developed at INRIA SARDES already addresses the autonomic management of clustered J2EE application servers. Our aim is to go further by being able to recruit dynamically grid nodes in order to run replica of the server. First and encouraging results can be found in [57].

### 8.1.5. *Contrat Plan État Région Grille et Calcul Pair-à-Pair*

This contract aims at building a regional computing platform. This is achieved by mixing desktop machines with dedicated ones like clusters. Users willing to submit a job will do so by accessing a webpage and uploading their program. It will then be scheduled and executed on a free machine. The scheduler is currently under development.

In the first part of the project, the access to the platform will be restricted to Inria members. Once most of the tools have been developed, the access will be open to industrial partners.

A convention has been signed with Microsoft to provide a specific cluster with Microsoft Compute Cluster Server.

The members of this project are the Inria and the Eurecom institute (Télécom Paris Ecole Polytechnique Fédérale de Lausanne).

The total budget for this project is 500kEuros for Inria and 100kEuros for Eurecom.

## 8.2. European Collaborations

### 8.2.1. *NOE CoreGRID*

CoreGRID is an European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies.

The CoreGRID Network of Excellence (NoE) aims at strengthening and advancing scientific and technological excellence in the area of Grid and Peer-to-Peer technologies. To achieve this objective, the Network brings together a critical mass of well-established researchers (119 permanent researchers and 165 PhD students) from forty-two institutions who have constructed an ambitious joint programme of activities. This joint programme of activity is structured around six complementary research areas that have been selected on the basis of their strategic importance, their research challenges and the recognized European expertise to develop next generation Grid middleware.

Besides the involvement of OASIS in the management and dissemination activities, the team is involved in three virtual institutes of the NoE.

- **Programming Model:** we are leading the Task dedicated to *Components and Hierarchical Composition*; our involvement here is to guide the design of a component model for the Grid (named GCM: Grid Component Model) at the European level. Besides, we are also involved in the Task dedicated to the study of *Basic Programming Models* for which we promote our approach of distributed and active object programming, extended with group communications and an innovative OO-SPMD approach, and in the Task dedicated to *Advanced Programming Models* in which we promote our tools for ensuring and verifying the correct behaviour of components.

- System Architecture: thanks to our experience in transparent checkpointing and recovery, we contribute to the research and integration work around Dependability in GRIDs.
- Problem Solving Environments, tools and GRID systems: thanks to our practical experience in developing the ProActive platform, we contribute in the collective study and effort to yield a generic, interoperable, portable, high-level grid toolkit, platform and environment.

This project started in September 2004, for 48 months, involving 62 kEuros in 2006.

### 8.2.2. *IP Bionets*

The OASIS team is involved in the European project called BIONETS (BIOlogically-inspired autonomic NETworks and Services)

The motivation for BIONETS comes from emerging trends towards pervasive computing and communication environments, where myriads of networked devices with very different features will enhance our five senses, our communication and tool manipulation capabilities. The complexity of such environments will not be far from that of biological organisms, ecosystems, and socio-economic communities. Traditional communication approaches are ineffective in this context, since they fail to address several new features: a huge number of nodes including low-cost sensing/identifying devices, a wide heterogeneity in node capabilities, high node mobility, the management complexity, and the possibility of exploiting spare node resources. BIONETS aims at a novel approach able to address these challenges. BIONETS overcomes device heterogeneity and achieves scalability via an autonomic and localized peer-to-peer communication paradigm. Services in BIONETS are also autonomic, and evolve to adapt to the surrounding environment, like living organisms evolve by natural selection. Biologically-inspired concepts permeate the network and its services, blending them together, so that the network moulds itself to the services it runs, and services, in turn, become a mirror image of the social networks of users they serve.

The team is involved in work packages 3.1 (Requirement Analysis and Architecture), 3.2 (Autonomic Service Life-Cycle and Service Ecosystems) and 3.4 (Probes for Service Framework).

The project started in 2006, for 48 months, for a total budget of 127 kEuros.

### 8.2.3. *Strep GridComp*

GridCOMP is a Strep project under leadership of ERCIM. Denis Caromel is the scientific coordinator. The European partners are university of Pisa and CNR in Pisa, university of Westminster on the academic side, and GridSystems (Spain), IBM Zurich (Switzerland), ATOS Origin (Spain) on the industrial side. Additionally there are 3 partners outside Europe, namely from university of Tsinghua (Beijing, China), university of Melbourne (Australia) and university of Chile (Santiago, Chile).

GridCOMP main goal is the design and implementation of a component based framework suitable to support the development of efficient grid applications. The framework will implement the "invisible grid" concept: abstract away grid related implementation details (hardware, OS, authorization and security, load, failure, etc.) that usually require high programming efforts to be dealt with.

The GCM implementation provided by OASIS in the GridCOMP EU project is based on ProActive. The design of this implementation follows these main objectives:

- Follow the GCM specification.
- Base the implementation on the concept of Active Objects. Components in this framework are implemented as active objects, and as a consequence benefit from the properties of the active object model.
- Leverage the ProActive library by proposing a new programming model which may be used to assemble and deploy active objects. Components in the ProActive library therefore also benefit from the underlying features of the library.
- Provide a customizable framework, which may be adapted by the addition of non functional controllers and interceptors for specific needs, and where the activity of the components is also customizable.

The first evaluation of GridComp occurred this year and was very positive: “The technical work done so far is very good and the review panel would like to congratulate the consortium. The consortium is supported with very good management from the coordinator and this is well reflected in the quality of the deliverables and presentations.”

The project has started in July 2006, for a duration of 30 months, with an overall budget of 634 kEuros.

In this context we have funded a short visit of Pr. T. Barros from UDP in January 07, and participated in the yearly Reseco workshop, this year in Montévidéo (Uruguay).

#### **8.2.4. SSA NESSI-Grid**

The NESSI-Grid SSA focuses on next generation Grid computing techniques that will allow to structure services coming from information technologies in order to provide them to industrial users.

NESSI-Grid aims at, through a strong cooperation and coordination of the projects inside the NESSI initiative, providing both a structure and resources in order to achieve this goal. NESSI-Grid started in May 2006 and will finish in November 2008.

Our contribution has been on the NESSI-Grid Strategic Research Agenda, in which we provided a comprehensive state-of-the-art study on scientific and business grids.

#### **8.2.5. SSA EchoGrid**

EchoGrid aims at developing the collaboration between Europe and China in the domain of research and technologies for Grid computing. Its objectives are to: design a global vision between Chinese and European researchers and industrials, exchange experiences and choose the best standards to build up a Standard Quality Assurance Process, and establish long-term partnerships between different actors.

EchoGRID started in January 2007 and will finish in December 2008.

### **8.3. International Collaborations**

#### **8.3.1. Stic-Amsud ReSeCo**

ReSeCo ( *Reliability and Security of Distributed Software Components*) is a collaboration of INRIA with partners of the south American CONESUD, namely Un. of Cordoba (Argentina), Un. of Montevideo (Uruguay), Un. Diego Portales and Un. de Chili (Chili). The two complementary themes of this project are the Specification and Verification of Component Systems on one side, and Security through Verifiable Evidence (proof Carrying Code) on the other hand. It started in November 2006 for a duration of 3 years, and will fund researcher visits and organization of workshops.

#### **8.3.2. Collaboration with University of Sydney**

This collaboration is entitled “Study of ProActive and Standardization for Remote instruments”. Its objective is to establish remote access for a network of structure determination and analysis instruments, ultimately providing the basis for developing a Grid enabled network linkable to other instrument, data and computation grids; national and international. The Project will entail research into the use of ProActive middleware to facilitate the integration of scientific instruments into a Grid computing and storage environment.

This collaboration started in November 2007 and will finish in March 2008.

#### **8.3.3. Stic Asia**

Stic Asia is a multilateral call project with universities of BUPT (Beijing, China), Tsinghua (Beijing, China), SCUT (Guangzhou, China), and NUST (Pakistan).

## 9. Dissemination

### 9.1. Leadership within the Scientific Community

#### 9.1.1. Committees

- **Eric Madelaine**  
Program committee chair and steering committee member of FACS'07,  
Program committee member of SCCC'07 and Provecs'07.
- **Françoise Baude**  
Program committee member of "CoreGRID Workshop on Grid Programming Model, Grid and P2P Systems Architecture, Grid Systems, Tools and Environments" and its post-proceeding review process; of the Workshop on Middleware for Grid Computing – MGC 2007 with the Middleware conference; of the Joint HPC-GECO/CompFrame'07 Workshop with the OOPSLA conference; of the IASTED PDCN'2007 conference; LAGrid workshop at CCGrid'07
- **Denis Caromel**  
chair of the committee for the AITO Dahl-Nygaard Awards 2007.
- **Fabrice Huet**  
Program committee member of HiPerGRID'07, PCGrid'07, HPDC'08, CCGrid'08, and PCGrid'08.

#### 9.1.2. Miscellaneous

Denis Caromel won the prize of the Concours National de Création d'Entreprise de Technologies (CNCE 2007), Ministère de la Recherche.

### 9.2. Short Visits

- **Florian Kammüller** from Technische Universität Berlin visited us for 1 week in September 2007. We worked on the formalization of ASP in Isabelle/HOL.
- **Rodolfo Toledo** from Universidad de Chile (CoreGrid Partner) visited us during 2 weeks in September. We worked on: Handling non-functional aspects of skeleton programming.

### 9.3. Teaching

#### 9.3.1. Teaching

- **Eric Madelaine**  
gives a course on Formal Techniques, within the Master 2 RSD (UNSA).
- **Françoise Baude**  
Course convenor in Operating System Design and Concurrent Programming (Licence 3, IUT Informatique 2, and Master TIM 2), and numerous exercise labs involvement (in XML and Web technologies, Databases, Java programming, parallel programming, ...)
- **Fabrice Huet**  
Coordinator of the 1st year of Master of Computer Science,  
Course convenor of Advanced System Programming (Master 1),  
Distributed Systems (Master 1) and Network Game Programming (Licence 3)

#### 9.3.2. Theses

The following theses have been defended this year:

- Christian Delbé: "Tolérance aux pannes pour objects actifs asynchrones : Protocole, Modèle et Expérimentations" defended the 24th January 2007.
- Alexandre di Costanzo: "Branch-and-Bound with Peer-to-Peer for Large-Scale Grids"

The following theses are in preparation:

- Muhammad Uzair Khan: “Supporting First Class Futures in a Fault-Tolerant Java Middleware” (Since Oct 2007).
- Marcela Rivera: “Reconfiguration and Life-cycle of Distributed Components: Asynchrony, Coherence and Verification” (since Dec 2006).
- Paul Naoumenko: “A Component Oriented Approach for Autonomous Services - Application to future generation communication networks, based on dynamic interactions between mobile devices” (Since Oct 2006).
- Antonio Cansado: “Verification and Generation of Safe Distributed Components” (Since Oct 2005)
- Viet Dung Doan: “Adequation of grid computing to computation intensive calculations in the financial domain” (Since Oct 2006).
- Imen Filali: “Peer to Peer computational Grids with reservation in service oriented architectures” (Since Oct 2007).
- Elton Mathias: “Hierarchical Grid Programming based upon a component-oriented approach” (Since Sept 2007).

### 9.3.3. Internships

- Elton Mathias: “Hierarchical Message Passing through Grid Components and the ProActive Platform”
- Muhammad Uzair Khan: “A Fault-tolerance Mechanism for Non-FIFO Services and Future Updates”
- Imen Filali: “P2P resource discovery in a Grid environment”
- Brian Amedro: “Efficient and Flexible OO SPMD for numerical Applications”
- Christophe Vergoni: “Dynamic clustering of an unstructured P2P network”
- Johan Fradh, Jonathan Martin, Jean-Luc Scheefer: “High level Scheduler for Grids ”
- Vasile Jureschi: “Applying Scientific and Technical Writing to the ProActive Library”
- Jean-Michael Legait (apprentice): “Monitoring of Grid Applications”
- Vladimir Bodnartchouk (apprentice): “Tools for Performance Analysis of Distributed Applications in ProActive”
- Solange Ahumada: “A UML Specification tool for distributed components”
- Emil Salageanu: “Tools for the behavioural modeling of distributed components”

## 9.4. Participation in conferences and workshops

### 9.4.1. Invited Talks

- **Ludovic Henrio**  
Invited speaker at the Autonomics’07 conference: “A Component Platform for Experimenting with Autonomic Composition”
- **Denis Caromel**  
Invited speaker at PCGrid 07 workshop in conjunction with IPDPS 07, ‘Towards Deployment Contracts in Large Scale Clusters & Desktop Grids’  
“ProActive professional Solutions”, presentation et discussions, January 18, Airbus Industry, Toulouse.  
“Open Source Middleware for the Grid: ObjectWeb ProActive Towards Standardization and Interoperability ”, invited presentation in “Journées Grilles de calcul”, January 19, Université Toulouse 1, Manufacture des Tabacs, Toulouse.



“Overview of GCM (Grid Component Model) and the GridCOMP EU Project”, January 29th, Melbourne University, Australia.

“Open Source Middleware for the Grid: ObjectWeb ProActive”, Keynote presentation, Feb. 1st, ACSW 2007 in 5th Australasian Symposium on Grid Computing and e-Research (AusGrid 2007), and 30th Australasian Computer Science Conference, Ballarat, Australia.

“Report on Grid Plugtests” and “ProActive GCM Overview”, EchoGRID First Strategic Workshop, February 2007, Beihang University (BUAA), Beijing, China.

“ProActive for Java Fault-Tolerance”, invited presentation in QosCosGrid EU project meeting, March 2007, Paris.

“Overview of ProActive for Multimedia”, Mikros Image, March 2007, Paris.

“Parallel Processing and Middleware for Enterprise Grid”, presentation to the CADENCE enterprise, March 2007, Sophia Antipolis.

“GCM Deployment standard proposal”, Manchester, UK, May 2007, TC Grid Meeting and OGF.

“From Theory to Practice in Distributed Component Systems”, Amsterdam, CWI, October 2007, Formal Methods for Components and Objects (FMCO’2007)

“ProActive GCM Middleware presentation”, ECHO Grid Meeting, IV GRIDs@work, Beijing, October 28, 2007

“ProActive and GCM: Status and future directions”, IV ProActive and GCM User Group Wednesday October 31 st, 2007, Beijing, China

“Evolution of the GCM standard”, ETSI TC Grid meeting, CNIC, Beijing, Nov. 2007.

“ProActive: Parallel Computing from Multi-Cores to Enterprise Grids”, invited talk at Tsinghua University, Nov. 2007.

#### 9.4.2. Talks in workshops and Conferences

- **Ludovic Henrio**  
FMOODS’07: “A Mechanized Model of the Theory of Objects”
- **Eric Madelaine**  
Colloque STIC 07: “ACI Sécurité FIACRE: Fiabilité des Assemblages de Composants REpartis”
- **Mario Leyton**  
Euro-par’07: “Fine Tuning Algorithmic Skeletons”
- **Christian Delbé**  
PPOPP’07: “Promised Messages: Recovering from Inconsistent Global States”
- **Ian Stokes-Rees**  
ISGC’07: “PicsouGrid - A Framework for Financial Computations on the Grid”
- **Paul Naoumenko**  
CoreGRID Workshop: “A flexible model and implementation of component controllers”
- **Antonio Cansado**  
Presentation at the *GI-Dagstuhl Research Seminar Seminar*: “A Specification Language for Distributed Components implemented in GCM/ProActive”
- **Viet Dung Doan**  
MCM’2007: “Comparison of parallel and distributed American option pricing: Through Continuation Values Classification versus Optimal Exercise Boundary Computation”
- **Virginie Legrand Contes**  
ICAS’07: “Large-Scale Service Deployment–Application to OSGi”

#### 9.4.3. Tutorials

- Clement Mathieu and Denis Caromel. “ProActive Tutorial Parallelism, Distribution, and Grid”. 1st EchoGRID International Conference. Beijing, China, 26-27 Apr. 2007

- Elton Mathias and Matthieu Morel. “Open Source Middleware for the GRIDs: ObjectWeb ProActive”. CCGRID’07. Rio de Janeiro, Brazil, May 14th 2007
- Elton Mathias and Jean-Michael Legait. “ProActive and Grid Component Model (GCM)”. CoreGrid Summer School 2007. Budapest, Hungary, Sept. 6-7th 2007
- Cedric Dalmasso and Antonio Cansado. “ProActive/GCM Tutorial and Hands-On Grid Programming”. IV Grids@Work. Tsinghua University, Beijing, China, Nov. 1st, 2007
- Fabrice Huet. Tutorial at the ASCI Course A14: Advanced Grid Programming Models: “The ProActive grid programming environment”

#### 9.4.4. Miscellaneous

- **Ludovic Henrio**  
Presentation at the *CoreGrid Scientific Advisory Board*: “The Grid Component Model: an Overview”  
Presentation at the *Reseco* workshop: “ASPfun: A Distributed Object Calculus and its Formalization in Isabelle”
- **Eric Madelaine**  
Presentation at the *GridComp workshop*: “Tools for the Specification and Verification of GCM Components”  
Presentation at the *Reseco* workshop: “Generation of Safe GCM Components”
- **Antonio Cansado**  
Presentation at the *GridCOMP* workshop: “Specifying GCM component with UML”
- **Ian Stokes-Rees**  
seminar in Oxford on “ProActive”; booth at OGF20/EGEE Users Conference in Manchester
- **Vincent Cavé and Brian Amedro**  
Demonstration of ProActive at the INRIA booth at SuperComputing 2007 conference, USA

## 10. Bibliography

### Major publications by the team in recent years

- [1] I. ATTALI, D. CAROMEL, S. O. EHMETY. *A Natural Semantics for Eiffel Dynamic Binding*, in "ACM Transactions on Programming Languages and Systems (TOPLAS)", vol. 18, n<sup>o</sup> 5, November 1996.
- [2] I. ATTALI, D. CAROMEL, S. O. EHMETY, S. LIPPI. *Semantic-based visualization for parallel object-oriented programming*, in "Proceedings of OOPSLA’96, San Jose, CA", ACM Sigplan Notices, vol. 31, 10, ACM Press, October 1996.
- [3] T. BARROS, R. BOULIFA, E. MADELAINE. *Parameterized Models for Distributed Java Objects*, in "Forte’04 conference, Madrid", LNCS 3235, Springer Verlag, 2004, <http://hal.inria.fr/inria-00087222>.
- [4] F. BAUDE, D. CAROMEL, F. HUET, L. MESTRE, J. VAYSSIÈRE. *Interactive and Descriptor-based Deployment of Object-Oriented Grid Applications*, in "11th IEEE International Symposium on High Performance Distributed Computing", 2002, p. 93–102.
- [5] F. BAUDE, D. CAROMEL, M. MOREL. *From Distributed Objects to Hierarchical Grid Components*, in "International Symposium on Distributed Objects and Applications (DOA), Catania, Sicily, Italy, 3-7 November, Springer Verlag", vol. 2888, Lecture Notes in Computer Science, LNCS, 2003, p. 1226–1242.

- [6] F. BAUDE, D. CAROMEL, D. SAGNOL. *Distributed Objects for Parallel Numerical Applications*, in "Mathematical Modelling and Numerical Analysis Modelisation, special issue on Programming tools for Numerical Analysis, EDP Sciences, SMAI", vol. 36, n<sup>o</sup> 5, 2002, p. 837–861.
- [7] D. CAROMEL. *Towards a Method of Object-Oriented Concurrent Programming*, in "Communications of the ACM", vol. 36, n<sup>o</sup> 9, September 1993, p. 90–102.
- [8] D. CAROMEL, L. HENRIO. *A Theory of Distributed Object*, Springer-Verlag, 2005.
- [9] D. CAROMEL, L. HENRIO, B. SERPETTE. *Asynchronous and Deterministic Objects*, in "Proceedings of the 31st ACM Symposium on Principles of Programming Languages", ACM Press, 2004, p. 123–134.
- [10] D. CAROMEL, W. KLAUSER, J. VAYSSIÈRE. *Towards Seamless Computing and Metacomputing in Java*, in "Concurrency Practice and Experience", vol. 10, n<sup>o</sup> 11–13, November 1998, p. 1043–1061.
- [11] D. CAROMEL, L. MATEU, E. TANTER. *Sequential Object Monitors*, in "Proceedings of the 18th European Conference on Object-Oriented Programming (ECOOP 2004), Oslo, Norway", LNCS, n<sup>o</sup> 3086, jun 2004, p. 316–340.
- [12] F. HUET, D. CAROMEL, H. E. BAL. *A High Performance Java Middleware with a Real Application.*, in "Proceedings of the ACM/IEEE SC2004 Conference on High Performance Networking and Computing, 6-12 November 2004, Pittsburgh, PA, USA", 2004.

## Year Publications

### Books and Monographs

- [13] D. CAROMEL, S. CHAUMETTE, G. FOX, P. GRAHAM (editors). *JaC PDC: Java and Components for Parallelism, Distribution and Concurrency*, Satellite workshop of IEEE IPDPS'2007, IEEE CS Press, Long Beach, march 2007.

### Doctoral dissertations and Habilitation theses

- [14] C. DELBÉ. *Tolérance aux pannes pour objets actifs asynchrones - protocole, modèle et expérimentations*, Ph. D. Thesis, Université de Nice-Sophia Antipolis, January 2007.
- [15] A. DI COSTANZO. *Branch-and-Bound with Peer-to-Peer for Large-Scale Grids*, Ph. D. Thesis, Université de Nice Sophia-Antipolis, October 2007.

### Articles in refereed journals and book chapters

- [16] I. ATTALI, D. CAROMEL, L. HENRIO, F. L. D. AGUILA. *Secured Information Flow for Asynchronous Sequential Processes*, in "Electronic Notes in Theoretical Computer Science", Proceedings of the International Workshop on Security and Concurrency (SecCo 2005), vol. 180, Jun 2007, p. 17–34, <http://www.sciencedirect.com/science/article/B75H1-4NWWPWK-3/2/7d9b65efb765e19b7671edc744c5eeb5>.
- [17] L. BADUEL, F. BAUDE, D. CAROMEL. *Asynchronous Typed Object Groups for Grid Programming*, in "International Journal of Parallel Programming", vol. 35, n<sup>o</sup> 6, 2007, p. 573–614.

- [18] F. BAUDE, D. CAROMEL, M. LEYTON. *File transfer in Grid applications at Deployment, Execution and Retrieval*, in "Multiagent and Grid Systems", vol. 3, n<sup>o</sup> 4, 2007, p. 1–11.
- [19] A. CANSADO, D. CAROMEL, L. HENRIO, E. MADELAINE, M. RIVERA, E. SALAGEANU. *A Specification Language for Distributed Components implemented in GCM/ProActive*, in "The Common Component Modeling Example: Comparing Software Component Models", to appear, 2007.
- [20] D. CAROMEL. *From Theory to Practice in Distributed Component Systems*, in "Electronic Notes in Theoretical Computer Science (ENTCS)", Keynote talk in Formal Aspects of Component Software (FACS'06), 2007.
- [21] D. CAROMEL, A. DI COSTANZO, C. DELBÉ. *Peer-to-Peer and Fault-tolerance: Towards Deployment-based Technical Services*, in "Future Generation Computer Systems", To appear, 2007.
- [22] D. CAROMEL, A. DI COSTANZO, C. MATHIEU. *Peer-to-Peer for Computational Grids: Mixing Clusters and Desktop Machines*, in "Parallel Computing", vol. 33, n<sup>o</sup> 4-5, 2007, p. 275–288.
- [23] D. CAROMEL, L. MATEU, E. TANTER. *Parallel object monitors*, in "Concurrency and Computation: Practice and Experience", Dr. Geoffrey Fox Editor, 2007.
- [24] N. PARLAVANTZAS, V. GETOV, M. MOREL, F. BAUDE, F. HUET, D. CAROMEL. *Componentising a Scientific Application for the Grid*, in "Achievements in European Research on Grid Systems. Invited papers from the 2nd CoreGRID Integration Workshop, Cracow, October 2006", CoreGRID series, ISBN: 978-0-387-27935-0. Also as the CoreGrid TR003 report, Springer, 2007, <http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0003.pdf>.

### Publications in Conferences and Workshops

- [25] S. AHUMADA, L. APVILLÉ, T. BARROS, A. CANSADO, E. MADELAINE, E. SALAGEANU. *Specifying Fractal and GCM Components With UML*, in "proc. of the XXVI International Conference of the Chilean Computer Science Society (SCCC'07), Iquique, Chile", IEEE, November 2007, <http://www-sop.inria.fr/oasis/Vercors/slides/2007-SCCC-SAhumada.pdf>.
- [26] F. BAUDE, M. BOSSY, V. DOAN, I. STOKES-REES. *PicsouGrid - A Framework for Financial Computations on the Grid*, in "International Symposium of Grid Computing, Taiwan, 2007", Abstract, march 2007.
- [27] F. BAUDE, D. CAROMEL, C. DELBÉ, L. HENRIO. *Promised Messages: Recovering from Inconsistent Global States*, in "Proceedings of the ACM SIGPLAN 2007 Symposium on Principles and Practice of Parallel Programming", Short paper, ACM Press, March 2007, p. 154–155.
- [28] F. BAUDE, D. CAROMEL, A. DI COSTANZO, C. DELBÉ, M. LEYTON. *Towards Deployment Contracts in Large Scale Clusters & Desktop Grids*, in "Int. Workshop on Large-Scale, volatile Desktop Grids, in conjunction with the IEEE IPDPS conference, Denver, Colorado, USA", invited paper, April 2007.
- [29] F. BAUDE, D. CAROMEL, L. HENRIO, M. MOREL. *Collective Interfaces for Distributed Components*, in "CCGrid 2007: IEEE International Symposium on Cluster Computing and the Grid", May 2007, p. 599-610.
- [30] F. BAUDE, D. CAROMEL, L. HENRIO, P. NAOUMENKO. *A Flexible Model and Implementation of Component Controllers*, in "CoreGRID Workshop on Grid Programming Model, Grid and P2P Systems Architecture, Grid

Systems, Tools and Environments", CoreGRID TR-0080 technical report, june 2007, p. 12–23, <http://www.coregrid.net/mambo/content/view/365/320/>.

- [31] F. BAUDE, L. HENRIO, P. NAOUMENKO. *A Component Platform for Experimenting with Autonomic Composition*, in "First International Conference on Autonomic Computing and Communication Systems (Autonomics 2007)", Invited Paper, ACM Digital Library, Oct 2007.
- [32] F. BAUDE, V. LEGRAND CONTESS, V. LESTIDEAU. *Large-Scale Service Deployment—Application to OSGi*, in "Proceedings of The Third International Conference on Autonomic and Autonomous Systems: ICAS 2007, Los Alamitos, CA, USA", IEEE Computer Society, 2007, p. 19–26.
- [33] D. CAROMEL, G. CHAZARAIN, L. HENRIO. *Garbage Collecting the Grid: a Complete DGC for Activities*, in "Proceedings of the 8th ACM/IFIP/USENIX International Middleware Conference, Newport Beach, CA", nov 2007.
- [34] D. CAROMEL, A. DI COSTANZO, L. BADUEL, S. MATSUOKA. *Grid'BnB: A Parallel Branch & Bound Framework for Grids*, in "Proceedings of the HiPC'07 - International Conference on High Performance Computing", To appear, 2007.
- [35] D. CAROMEL, L. HENRIO, M. LEYTON. *Type Safe Algorithmic Skeletons*, in "Proceedings of the 16th Euro-micro International Conference on Parallel, Distributed and network-based Processing, Toulouse, France", To appear, February 2008.
- [36] D. CAROMEL, M. LEYTON. *Fine Tuning Algorithmic Skeletons*, in "13th International Euro-par Conference: Parallel Processing", Lecture Notes in Computer Science, vol. 4641, Springer-Verlag, 2007, p. 72–81.
- [37] V. DOAN, M. BOSSY, F. BAUDE, I. STOKES-REES. *Comparison of parallel distributed American option pricing: Through Continuation Values Classification Versus Optimal Exercise Boundary Computation*, in "Sixth IMACS Seminar on Monte Carlo Methods (MCM 2007)", Extended Abstract, July 2007.
- [38] L. HENRIO, F. KAMMÜLLER. *A Mechanized Model of the Theory of Objects*, in "9th IFIP International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS)", LNCS, Springer, June 2007.
- [39] M. MALAWSKI, M. BUBAK, F. BAUDE, D. CAROMEL, L. HENRIO, M. MOREL. *Interoperability of grid component models: GCM and CCA case study*, in "Towards Next Generation Grids, proceedings of the CoreGRID Symposium in conjunction with Euro-Par 2007", CoreGRID series, Springer Verlag, august 2007, p. 95–106.
- [40] E. MATHIAS, F. BAUDE, V. CAVÉ, N. MAILLARD. *A Component-Oriented Support for Hierarchical MPI Programming on Multi-Cluster Grid Environments*, in "SBAC-PAD 2007: 19th International Symposium on Computer Architecture and High Performance Computing, Gramado, Brazil", IEEE Computer Society Press, Oct 2007, p. 135-142.
- [41] N. PARLAVANTZAS, V. GETOV, M. MOREL, F. BAUDE, D. CAROMEL. *Design Support for Componentising and Grid-enabling Scientific Applications*, in "Workshop HPC Grid programming Environments and Components and Component and Framework Technology in High-Performance and Scientific Computing (HPC-GECO+COMPFRAME)", in conjunction with ACM SIGPLAN ooPSLA 2007", ACM Digital Library, 2007, p. 31–18.

- [42] N. PARLAVANTZAS, M. MOREL, V. GETOV, F. BAUDE, D. CAROMEL. *Performance and Scalability of a Component-Based Grid Application*, in "9th Int. Workshop on Java for Parallel and Distributed Computing, in conjunction with the IEEE IPDPS conference", April 2007.

### Internal Reports

- [43] B. AMEDRO. *Efficient and flexible OOSPMD for numerical applications*, MSc Report, June 2007.
- [44] R. BADIA, F. BAUDE, V. GETOV, T. KIELMANN, I. TAYLOR. *CoreGRID Workshop on Grid Systems, Tools and Environments, 1st Dec. 2006 (Proceedings)*, Technical report, n<sup>o</sup> TR-0103, CoreGRID Institute on Grid Systems, Tools and Environments, Sep 2007, <http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0103.pdf>.
- [45] F. BAUDE, L. HENRIO, ET AL. *Specification of Service Evolution*, Technical report, BIONETS IP Project Deliverable from the Autonomic Service Life-Cycle And Service Ecosystems WP (3.2), Aug 2007, [http://www.bionets.eu/docs/BIONETS\\_D3\\_2\\_2.pdf](http://www.bionets.eu/docs/BIONETS_D3_2_2.pdf).
- [46] F. BAUDE, ET AL. *Refinement of Service Architecture and Requirements*, Technical report, BIONETS IP Project Deliverable from the Requirements Analysis And Architecture WP (3.1), Aug 2007, [http://www.bionets.eu/docs/BIONETS\\_D3\\_1\\_2.pdf](http://www.bionets.eu/docs/BIONETS_D3_1_2.pdf).
- [47] Y. FENG, E. MADELAINE, I. STOKES-REES, ET AL. *Grid Vision and Strategic Research Agenda*, Technical report, NESSI-Grid: Networked European Software and Services Initiative - Grid, October 2007.
- [48] L. HENRIO, F. KAMMÜLLER, H. SUDHOF. *ASPFun: A Functional and Distributed Object Calculus Semantics, Type-system, and Formalization*, Research Report, n<sup>o</sup> 6353, INRIA, 11 2007, <http://hal.inria.fr/inria-00186963/fr/>.
- [49] U. KHAN. *Study on First Class Futures and incorporating them in CIC Protocol*, Rapport de stage de Master 2, Technical report, 2007.
- [50] E. MATHIAS. *Hierarchical Message Passing through Grid Components and the ProActive Platform*, Rapport de stage Master RSD, Master Thesis, UNSA, June 2007.
- [51] OASIS TEAM. *CFI early prototype*, Deliverable D.CFI.02, Technical report, March 2007.
- [52] OASIS TEAM. *Standardization Strategy*, Deliverable D.DIS.02, Technical report, June 2007.
- [53] OASIS TEAM, ET AL. *Innovative Features of GCM (with sample case studies): a Technical Survey*, Deliverable D.PM.07, Technical report, Sep. 2007.
- [54] OASIS TEAM, GRIDCOMP PARTNERS. *Architectural design of the component framework*, Deliverable D.CFI.03, Technical report, June 2007.
- [55] OASIS TEAM, GRIDCOMP PARTNERS. *Methods and Techniques for legacy Code Wrapping as Grid Components*, Deliverable D.CFI.04, Technical report, December 2007.

- [56] OASIS TEAM, GRIDCOMP PARTNERS. *Proceedings of the first GridCOMP workshop*, Deliverable D.DIS.03, Technical report, December 2007.
- [57] C. RUZ, F. BAUDE, V. LEGRAND CONTES. *Towards Grid Monitoring and deployment in Jade, using ProActive*, Research Report, n° 6340, INRIA, 10 2007, <https://hal.inria.fr/inria-00182554>.
- [58] E. SALAGEANU. *Environment for the Specification of Distributed Components*, Rapport de stage de Master 2, Technical report, Université de Nice-Sophia Antipolis, September 2007.
- [59] E. SALAGEANU. *Tools for the behavioural modeling of distributed components*, Rapport de projet de fin d'études de Master 2, Technical report, Université de Nice-Sophia Antipolis, June 2007.

## References in notes

- [60] M. ABADI, L. CARDELLI. *A Theory of Objects*, Springer-Verlag, New York, 1996.
- [61] F. BAUDE, D. CAROMEL, N. MAILLARD, E. MATHIAS. *Hierarchical MPI-like Programming using GCM Components as Implementation Support*, Selected and presented at the CoreGRID workshop: Grid Systems, Tools and Environments, Grids@work II conference, Dec. 2006.
- [62] F. BAUDE, V. LEGRAND CONTES, N. PARLAVANTZAS, J. DÜNNWEBER, S. GORLATCH. *Towards Automatic Creation of Web Services for Grid Component Composition*, in "1st CoreGRID integration workshop", "Also selected and presented at the CoreGrid workshop: Grid Systems, Tools and Environments, Oct. 12th 2005, Grids@work conference", Nov 2005.
- [63] S. BEZINNE, V. GALTIER, S. VIALLE, F. BAUDE, M. BOSSY, V. D. DOAN, L. HENRIO. *A Fault Tolerant and Multi-Paradigm Grid Architecture for Time Constrained Problems. Application to Option Pricing in Finance*, in "2nd IEEE International Conference on e-Science and Grid Computing", IEEE, December 2006.
- [64] E. BRUNETON, T. COUPAYE, M. LECLERCP, V. QUEMA, J. STEFANI. *An Open Component Model and Its Support in Java.*, in "7th Int. Symp. on Component-Based Software Engineering (CBSE-7)", LNCS 3054, may 2004.
- [65] COREGRID, PROGRAMMING MODEL INSTITUTE. *Basic Features of the Grid Component Model (assessed)*, 2006, <http://www.coregrid.net/mambo/images/stories/Deliverables/d.pm.04.pdf>.