INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team oasis

Objets Actifs, Sémantique, Internet et Sécurité

Sophia Antipolis - Méditerranée

THEME COM

Activity Report

2008

Table of contents

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

1. Team

Research Scientist

Denis Caromel [Team Leader, Prof. UNS, détachement INRIA, HdR]
Eric Madelaine [CR INRIA]
Ludovic Henrio [CR CNRS]

Faculty Member

Fabrice Huet [MdC UNS]
Françoise Baude [MdC UNS, HdR]

External Collaborator

Florian Kammuller [U.T. Berlin]

Technical Staff

Abhijeet Gaikwad [ANR GCPMF, to nov 08]
Bastien Sauvan [GridiCOMP, to sep 08]
Bastien Sauvan [ADT Glaxy, since oct 08]
Cédric Dalmasso [FP6 GridCOMP, to nov 08]
Clément Mathieu [IA-INRIA, to nov 08]
Elaine Isnard [AGOS, since jan 08]
Fabien Viale [FP6 S4All, FP6 NESSI-Grid, Thales]
Fabrice Fontenoy [AGOS, since dec 08]
Florin Bratu [IA-INRIA, since oct 08]
Franca Perrina [AGOS, since jan 08]
Germain Sigety [INRIA]
Guillaume Laurent [FP6 GridCOMP, to dec 08]
Oleg Smirnov [Microsoft, since oct 08]
Romain Quilici [Convention Un. Sydney, to mar 08]
Vincent Cavé [ANR DiscoGrid, to dec 08]
Yu Feng [ETSI, FP6 EchoGrid, to June 09]

PhD Student

Antonio Cansado [INRIA, Conicyt, to oct 08]
Brian Amedro [SCS-Agos]
Cristian Ruz [INRIA, Conicyt, since feb 08]
Elton Mathias [INRIA-Cordi]
Guilherme Peretti-Pezzi [CIFRE SCP, since oct 08]
Guillaume Chazarain [MESR, to jan 08]
Imen Filali [SCS-Agos / FP7 SOA4All]
Marcela Rivera [INRIA, Conicyt]
Mario Leyton [INRIA, Conicyt, to oct 08]
Muhammad Khan [Pakistan fellowship, STIC-Asia]
Paul Naoumenko [INRIA/Région PACA]
Viet Dung Doan [FP6 Bionets]
Virginie Contes [SOA4ALL, since mar 08]

Post-Doctoral Fellow

Regis Gascon [INRIA-Cordi]

Visiting Scientist

Guilherme Peretti-Pezzi [ERCIM, since jan 08, to aug 08]

Nadia Ranaldo [Univ. Sannio - CoreGrid, to mar 08]

Administrative Assistant

Patricia Maleyran [INRIA]
Sandra Devauchelle [I3S]

Other

Jean-Luc Scheefer [Apprentice, master ISI, to sep 08]
Johann Fradj [Apprentice, master ISI, to sep 08]
Etienne Valette d'Osia [Apprentice, master ISI, since oct 08]
Laurent Vanni [Apprentice, master ISI, since oct 08]

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

- Undergoing process of standardisation of a Grid Component Model and its deployment:
  - 2 official ETSI TC Grid standards officially approved and published by the project: "GCM Interoperability Deployment standard", and "GCM Application Interoperability Description"
  - and 2 new work items have been approved towards standardisation: "GCM Fractal ADL", and "GCM Management API"
- Organisation of the Grids@work conference, at Sophia-Antipolis, including a Grid plugtest, and for the first time co-located with the FMCO conference (an event gathering several European projects on formal methods for components and objects).

3. Scientific Foundations

3.1. Programming with Distributed Objects and Components

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 ?

One of the answers we suggest relies on component-oriented programming. In particular, we have defined parallel and hierarchical distributed components starting from the Fractal component model developed by INRIA and France-Telecom [54]. We have been involved in the design of the Grid Component Model (GCM) [55], 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.

For providing a better programming and runtime environment for object and component 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

A few years ago, we designed the ASP calculus for modelling 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 generalised futures. Those futures naturally come with a transparent and automatic synchronisation 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 [6].

ASP 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. A functional fragment of ASP has been modelled in the Isabelle theorem prover.

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 synchronisations between remote parts of a system increases the difficulty of analysing their behaviours. Consequently, safety, security, or liveness properties are particularly difficult to ensure for these applications. Formal verification of software systems has been active for a long time, but 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.

We concentrate on the area of distributed component systems, where 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 [2]. 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 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*, *synchronisation*.

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 mobile objects and components.

In this domain, the OASIS team strongly contributed to the design standardisation and implementation of a Grid-oriented component model called GCM (Grid component model).

4.2. Service Oriented Architectures (SOA)

Keywords: *Autonomic Management*, *Distributed Service Composition*, *Distributed Service Infrastructure*, *Large-Scale deployment and monitoring*, *Peer-to-Peer service registration and lookup*.

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 seeks solutions to the problems encountered here, with the underlying motivation to demonstrate the usefulness of a grid programming approach like ProActive and GCM in this area :

- Deployment of a service on the service infrastructure: 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 conducting research to promote the Grid Component Model as 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, still participating in a global SCA-based SOA. We thus pursue works to make SCA and GCM become interoperable models.

- Interoperability between services: the uniform usage of web services can provide a simple interoperability between them. GCM components can be exposed as web services [57], and we have started some research and development to permit a GCM component to invoke an external web service through a client interface, and thus to have GCM/SCA components be integrated in SCA-based applications relying on SCA bindings configured as web services.
- Large-scale deployment and monitoring of a set of (similar) services on a possibly large set of machines from e.g. a computing grid, a cloud of machines, etc.: such capability will really make SOA ready for the Internet scale, and we are designing some grid services, accessible as web services, in order to leverage the required functionalities for Grid/cloud deployment of components/services and monitoring of the resulting runtime infrastructure.
- Distributed and Scalable service bus: this is needed if the non-functional services that are needed for a service bus to operate (e.g. registry, discovery, monitoring services) must be replicated and distributed. We intend to use GCM components for building the bus itself, giving it large-scale capabilities (this subject is specifically addressed in the context of the SOA4ALL IP FP7 project the team is involved in, closely contributing to the design of a large-scale distribution of the OW2 Petals ESB, itself based on Fractal and Fractal RMI distributed components)
- Peer-to-peer based service registry and service lookup protocols: in an Internet-based world hosting possibly billions of services, the registration and subsequent lookup of services can only be addressed along a semantic-based approach, and should allow a robust and scalable way to store and query for service descriptions. In the context of the SOA4ALL IP FP7 project, we have started research to contribute to the design of a semantic space where services will be stored and looked upon based on their semantic description. For scalability purposes, the space specification is organised as a peer-to-peer network, further implemented in a distributed, scalable way relying on a grid middleware as e.g. the ProActive technology.
- Self-management of the SOA infrastructure and SOA applications: this pertains to autonomic and self-management of the service infrastructure, but also of the component assemblies that constitute the Service Oriented Application. Again the use of GCM components instead of Fractal-RMI components whenever needed can be a solution to the scalability problem. For service compositions represented as component assemblies, we are exploring the use of control components put in the component membranes, acting as sensors or actuators, that can drive the self-management of composite services, e.g. according to a negotiated Service Level Agreement.
- Distributed and agile workflow enactment: as BPMN and BPEL are the standard ways to define a service orchestration, we are considering how such a *composition in time* approach can be mapped into an architectural-based view involving (SCA) components. Besides, efficient and secured orchestration of such service compositions can benefit from distribution and parallelism. In this aim, we investigate how GCM can be successfully used to design a parallel, distributed, yet flexible orchestration engine handling a BPEL workflow description previously decomposed into sub-workflows. Deployment and management of the decomposition can also be addressed easily by having the distributed workflow relying on GCM components.

5. Software

5.1. ProActive

Participants: B. Amedro, F. Baude, D. Caromel, V. Cavé, C. Dalmasso, V.-D. Doan, J. Fradj, A. Gaikwad, F. Huet, E. Isnard, M. Khan, V. Legrand, M. Leyton, G. Laurent, C. Mathieu, P. Naoumenko, F. Perrina, M. Rivera, J.-L. Scheefer, G. Sigety, F. Viale.

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, Clouds 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 tunnelling,
- Globus: GT2, GT3, GT4, gsi, gLite, Unicore, ARC (NorduGrid)
- LSF, PBS, Sun Grid Engine, OAR, GLite, Load Leveler

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 optimisations. 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 synchronisations, 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 new deployment framework has been standardised by the ETSI. It allows to deploy ProActive, native and MPI application on almost all Grid/cluster protocol: Windows CCS, GLite, Sun Grid Engine, LSF, OAR, PBS, SSH, RSH etc.
- 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;
- GCM component support;
- 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 others we are particularly proud of the following results, showing that portable and transparent Java code can compete with specific optimised approaches:

- NQueen challenge, where we equalled 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 months 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 [8] and to the web pages http://www.objectweb.org and http://www.inria.fr/oasis/proactive which list several white papers.

The following new features have been developed in 2008:

- GCM Deployment support
- A new API to write distributed Monte-Carlo simulations
- Added a Remote Objects layer. It provides remote passive objects
- Prototype of forget-on-send to improve performances of one-way communications
- Prototype of a distributed debugger

5.2. Vercors platform

Participants: A. Cansado, E. Madelaine, P. Valenzuela, V. Maisonneuve, K. Nirski, M. Baranowski.

Vercors is a verification platform for distributed components covering the whole process of verification.

The Vercors tools (http://www-sop.inria.fr/oasis/Vercors) 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 have continued our work on diagram editors for high-level specifications of the architecture and behaviour of components. The architecture editor is now able to read and write ADL files for the GCM, giving a first level of interoperability with other GCM tools. It also addresses new constructs specific of the GCM, namely collective interfaces and membranes. This new version is completely integrated as a set of Eclipse plugins [39]. 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.

We have also started experiments with new verification engines for so-called "infinite systems". We have a prototype for verifying reachability properties of systems containing finite state machines (FSM) connected by unbound Fifo queues. Challenges are now to demonstrate the practical feasibility of this approach, and to integrate this kind of specific engines with the rest of the platform.

6. New Results

6.1. Distributed Programming Models

6.1.1. Theory of Distributed Objects

Participant: L. Henrio.

This work extends results published in [6]. The ς-calculus, and its semantics were published by Abadi and Cardelli [50]. In collaboration with Florian Kammüller (Technische Universität Berlin) we previously modelled the ς-calculus and a distributed functional calculus, based on ASP – ASP_{fun} – in the Isabelle/HOL theorem prover.

This year, we have consolidated our proofs of type safety and progress for ASP_{fun}. Properties have been extended, and we proved that no cycle of future can be created in this calculus We think this result could be used in the design of service-oriented (distributed) architectures communicating with requests-futures.

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

Further work include formalisation of the imperative ASP, and of confluence properties for ASP_{fun}.

6.1.2. 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.

In 2008 we 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* (muscle functions are basic blocks composed by skeletons, they encapsulate the business core).

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 were presented at the international Euromicro-PDP 2008 conference [20].

Additionally, in 2008 we also provided a transparent file support for algorithmic skeletons. A file data model for algorithmic skeletons was proposed, focusing on transparency and efficiency. Transparency is achieved using a workspace factory abstraction and the proxy pattern to intercept calls on `File` type objects. Thus allowing programmers to continue using their accustomed programming libraries, without having the burden of explicitly introducing non-functional code to deal with the distribution aspects of their data. A *hybrid* file fetching strategy is proposed (instead of *lazy* or *eager*), that takes advantage of annotated functions and pipelined multithreaded interpreters to transfer files in-advance or on-demand. Experimentally, using a BLAST skeleton application, it is shown that the *hybrid* strategy provides a good tradeoff between bandwidth usage and CPU idle time.

The results of the transparent file support for skeletons were published at the international IPDPS 2008 conference [21].

6.2. Component-oriented Grid Programming

6.2.1. Designing Non-Functional Concerns as Components

Participants: F. Baude, L. Henrio, P. Naoumenko.

As part of the design of the GCM, we progressed on the research concerning the componentisation of component membranes [52] (a membrane encapsulates the component control and supervision of the functional part). 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. By taking advantage of the component-oriented approach, components inside the membrane (or managers) give powerful means to design and make evolve autonomic strategies for GCM components. Indeed, GCM components can be considered as autonomic entities that change their behaviour according to changes in their environment in order to maintain an equilibrium with respect to the environment. The advantages of this approach are a better structuring of non-functional aspects, and better reconfiguration possibilities. The new structure of the membrane allows the design of distributed and hierarchical autonomic decisions, since managers inside the membrane can connect to managers situated either in membranes of inner functional components or in membranes of external components. We built a prototype and implemented an API introducing non-functional (NF) components inside the membranes of Fractal/GCM components. It is now possible to create and manage NF components programmatically.

The BIONETS European projects aims at building autonomous services inspired by biology. In this context, the work on componentised membranes is used for dynamic composition and evolution of services. Indeed, a plan of composition and several autonomic and evolution strategies can be designed as component systems inside the non-functional part of BIONETS services. We successfully implemented an example of an application with autonomous entities. This work is presented in [43]. In the future, we plan to improve the ADL (Architecture Description Language) used to describe the component architecture by introducing componentised membranes at the level of the architecture description. We also plan to use this new ADL to describe and implement new examples of autonomic systems with GCM components.

6.2.2. Reconfiguring Distributed Components

Participants: L. Henrio, M. Rivera.

As second part of her thesis, Marcela Rivera defined a framework to support distributed reconfiguration of component systems. This work extends an existing scripting language to enable remote interpretation of reconfiguration procedures. To support this extension of the language, we had to extend components with a non-functional ability: the interpretation of reconfiguration scripts. Thanks to this approach, reconfiguration scripts can now be evaluated in a distributed, i.e., non-centralised, manner. The resulting interpreter, together with the script extension have been implemented and experimented using the ProActive implementation of the Grid Component Model.

To reach this goal, the existing component frameworks has be extended with two features:

- A controller, i.e. a non-functional port, localised in several (possibly all) components that is able to interpret reconfiguration orders.
- An extension of an existing scripting language for reconfiguration, adding primitives for distributed interpretation: remote execution of a reconfiguration script, and evaluation of script expressions to improve the passing of parameters between different execution contexts.

We showed the adequacy of our contribution by providing an implementation of those features in the context of the Grid Component Model (GCM) [14], and of its reference implementation above the ProActive middleware. We extend the FScript [56] reconfiguration language, that was designed for the Fractal [54] component model, and adapt it to distributed component systems.

We plan to verify the coherence in the component states along reconfigurations. Indeed, suppose for example that two consecutive requests (on the same binding) should necessarily be addressed to the same destination component (for example, one request sends additional information necessary to fulfil another one). Then, between those two requests, no reconfiguration can occur if it involves the binding used for the requests.

As a consequence, we plan to design a way of specifying synchronisation between reconfiguration steps and the application, this should be the main interaction between functional and non-functional aspects, and should be studied carefully in order to maintain the "good separation of aspects" that exists in Fractal and GCM.

6.2.3. SPMD and Components

Participants: F. Baude, V. Cavé, E. Mathias.

In the context of the DiscoGRID ANR funded project, we have continued to promote the usage of a GCM-based infrastructure to support an MPI-like hierarchical SPMD programming model (Section 8.1.2). The idea of this infrastructure is to wrap MPI processes within primitive components and use the components and their inherent features to efficiently support point-to-point and collective communication whenever the MPI communications are not possible [58].

This work is done in collaboration with applied mathematicians (namely the CAIMAN and SMASH teams, partners of the DiscoGRID project). As a matter of fact, they represent a community where programmers are used to the standard SPMD message-passing based model, and that are 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. A key point to the success of this partnership is the development and usage of solutions in a completely transparent way for programmers.

In order to support DiscoGrid primitives and MPI collective communications we have improved GCM collective interfaces and extended the proposed specification of GCM collective interfaces. These improvements come from the need to perform more advanced collective communication at interface level and include the possibility to define partial-multicast and partial-gathercast invocations along with the configuration of communication semantics, aggregation, distribution and reduction policies.

Besides these improvements, we have defined a complex collective interface called gather-multicast interface, which is, in fact, a concatenation of a gather and a multicast interface. This interface is capable of performing true many-to-many communication (also known as MxN) among primitive component, inserted in the context of composite components that hold this kind of interface. The gather-multicast interface is being used in the DiscoGrid runtime to perform a high-level operation called *'update'*, which uses neighbourhood information to update data shared among process distributed on the grid.

As expected, the usage of such an interface has introduced scalability and performance issues due to bottlenecks in aggregation and distribution policies. Our approach to tackle this problem has been the definition of direct bindings among components to bypass the bottlenecks and somehow distribute the gather-multicast communication semantics. The definition and usage of the gather-multicast interfaces to perform MxN communication, as well as optimisations though direct bindings were published in [26] and have shown the usefulness and the importance of introduced interfaces and their optimisations.

6.2.4. Monitoring Components

Participants: F. Baude, C. Ruz.

Monitoring components is important not only for optimising or fine-tuning an application. Being able to collect on-line monitoring data can help to make runtime decisions over the configuration of a system in order, for example, to ensure a previously agreed QoS.

We are designing a monitoring component-based system that will reside (at least partly, some complementary aspects being for instance deployed as probes at the level of the service bus) in the membrane of GCM components. This monitoring system will be able to track the runtime path of a request through a running component application. The information of the components involved in serving a request and the time it took to serve it in each of them will help to determine weak points in the application.

The monitoring scheme must be general enough to be used in the architecture of a GCM application, and allow for scalability considering that no single component must need to keep all the information of the system; instead the components in charge of monitoring can interact with external components by using non functional interfaces to obtain the desired information in a pull mode. In a similar way, inner components of a composite can send notifications to their parent in a push mode, also using non functional interfaces.

6.2.5. Mapping Workflows on Components

Participants: F. Baude, V. Legrand Contes.

The need of a scalable, agile, and adaptable workflow engine is gaining more and more interest in the SOA community. This kind of workflow engine will provide non functional features to service compositions. Those non-functional features could include fault tolerance, agility, and adaptability according to the execution context or service availability.

In the context of the SOA4All project and Galaxy ADT we aim at demonstrating the interest of mapping a service composition (i.e. workflow) into a GCM component; a workflow is thus represented by a hierarchical and distributed component. Thus, it benefits from the Fractal properties, and in particular dynamic reconfiguration.

We are currently exploring different ways to model a service composition with GCM components and identify the best way to use GCM at different steps of the workflow life-cycle: deployment, dynamic and distributed workflow execution, and monitoring of the execution.

6.3. Behavioural Specification for Reliable Distributed Components

6.3.1. A specification Language for Distributed Components: JDC

Participants: A. Cansado, R. Gascon, L. Henrio, E. Madelaine.

In the context of the Grid Component Model (GCM), we have developed a new specification language for dealing with its specificities. Even if there are many specification languages in the literature, none fits well in the context of distributed components, particularly because they do not focus on asynchronous communications and complex synchronisations used in the component model.

To this aim, we have developed a specification language close to Java called Java Distributed Components (JDC). The language includes both the architecture and the behaviour definitions of components. This language is built on top of our behavioural model that was the main result of T. Barros's PhD thesis, and provides a powerful high-level abstraction of the system. The benefits are twofold: (i) we can interface with verification tools, so we are able to verify various kinds of properties; and (ii), the specification is complete enough to generate code-skeletons defining the control part of the components.

We have opted for a Java-like language for several reasons: (i) it is close to the target expertise of engineers, using common syntax such as method calls and data classes; (ii) it allows part of the specification to be embedded within the code skeletons; (iii) it uses the same datatypes as in the implementation, guaranteeing that operations on the datatypes are directly useful without modification.

These results have been published in [30], [19].

We are also developing a graphical version of the specification language, corresponding to a static subset of JDC. This is implemented using model-driven methods, in the form of diagram editors within the Vercors software platform.

6.3.2. *Infinite Models*

Participants: R. Gascon, E. Madelaine, V. Maisonneuve.

In Vercors, the verification of safety properties on the formal models generated by the platform (pNets) uses finite state model-checkers. As a consequence, this operation relies on the instantiation of some parameters and the finite abstraction or abstract interpretation of others. In order to improve this procedure, we have investigated the field of infinite state model-checking. This work follows three directions depending on the characteristics of pNets:

- manipulation of “infinite data” (data in an infinite set),
- representation of unbounded queues,
- representation of parameterised topologies.

Each of these points is a source of infiniteness since it introduces an unbounded parameter. The goal of infinite-state model-checking methods is to consider directly these unbounded parameters without any abstraction. But the combination of these different parameters makes the verification problem very difficult: from a theoretical point of view, any of the parameters mentioned above can imply undecidability of standard reachability analyses.

However, there are several techniques using semi-algorithm or restrictions of the general problems that allow tackling some problems in practise. We have studied some of these techniques with the objective to find a way to apply them to pNets. For instance, some representations using finite state automata can be used to represent (infinite) set of states in systems manipulating integers, queues or pointers. Regular expressions can also be used to describe the evolution of a parameterised or dynamic network during an execution. These are a few examples of the different possibilities that we have been studying. Now, an important open question is to understand how these techniques can be mixed together to verify safety properties as precisely as possible. Indeed, none of them is able to treat all the infinite aspects of Vercors formal model.

As experimentation, we have implemented a prototype to explore the set of configurations that can be reached in a network of finite state machines communicating with unbounded FIFO queues. We want to add mechanisms able to check liveness; we would also like to define an input language to express the safety properties that can be checked using all our material. Of course, the other unbounded parameters linked to the data and the topology have to be taken into account. For the moment, such tools extensions are difficult to design since we do not exactly know how to mix the techniques needed with the current implementation. However, it is already possible to introduce some infinite data, with the condition that the set of their values can be finitely partitioned w.r.t. the kind of properties to verify.

6.3.3. *Modelling Futures*

Participants: A. Cansado, L. Henrio, E. Madelaine.

We studied the behavioural modellings for *transparent first-class* futures, and their use within distributed components. Futures are *transparent* if the result is automatically created and implicitly awaited upon the first access to the value; and futures are *first-class* if they can be transmitted between components as usual objects. Thus, because of the difficulty to identify future objects, analysing the behaviour of components using first-class transparent futures is challenging. Our contribution consists in first a static representation for futures, second a means to detect local deadlocks in a component system with first class futures, and finally extensions to interface definitions in order to avoid such deadlocks.

The main contribution here is to provide behavioural proxies that expresses the flow of future references and future values. It extends our previous works by giving behavioural models for transparent first-class futures, relying heavily on the properties proved in the ASP-calculus. We showed that this model can be used to detect deadlocks.

These results have been published in [18].

6.4. Grid Middleware and Applications

6.4.1. Optimising Distributed Object Computing

Participants: B. Amedro, D. Caromel, L. Henrio, M. Khan.

Sterile Requests differentiation

ProActive rendez-vous is a synchronisation time which occurs each time a request is sent. This time is necessary to ensure *causal order* of requests. However, in some cases and for performance purpose, we can perform this synchronisation in parallel with computing.

Our work proposes to distinguish a sub category of ProActive requests: *sterile requests*. A request is known as *sterile* if it does not have any descendant, i.e. if during its service it does not send new requests, except to itself or to the activity which sent the request it is serving (its parent). Assuming this definition, a *sterile request* can be sent and its rendez-vous can be delegated to a concurrent thread if the parameters of the request are not modified after the sending. Such a request is invoked using the primitive *ForgetOnSend*.

A Study of Future Update Strategies

Futures enable an efficient and easy to use programming paradigm for distributed applications. In ProActive, an active object is analogous to a process, having its own thread and a message queue for storing incoming requests. Futures, as used in ASP and ProActive, represent the result of an asynchronous invocation and can be safely transmitted between processes. As references to futures disseminate, a strategy is necessary to propagate the computed result of each future to the processes that need it.

Our work addresses the problem of the efficient transmission of those computed results. It presents three main strategies for updating futures. These include two eager strategies: *Eager forward-based*, and *Eager message-based*, and one lazy strategy, *Lazy message-based*. The two eager strategies update the futures as soon as the results are available, while the lazy strategy is a on-demand strategy, resolving the future only when the value is strictly needed. We focussed on providing a semi-formal description which allows us to perform preliminary cost analysis. To verify our cost analysis, we carried out some experiments to determine the efficiency of each strategy under different conditions. A publication has been written from this contribution and is currently under review.

In near future, we will focus on the non-trivial problem of garbage collection of the computed results. Another challenging problem is developing (and formally proving) a protocol for cancellation of requests in an active object environment. A sub-problem is to allow cancellation of only specific future updates, resulting in improved performance, for example in the case of workflow-based scenarios.

6.4.2. Peer-to-Peer Infrastructure

Participants: I. Filali, F. Huet.

Researches on P2P networks have focused not only on the network architecture but also on the semantic of the stored data, moving from simple keywords to more sophisticated RDF-based data model. In the context of the SOA4ALL project, we are working on the design and the implementation of a distributed semantic space infrastructure. We have proposed a multi-layers architecture based on DHTs overlays. The infrastructure aims at fully distributing data among participating peers. In the second part of the project, the infrastructure will be used to store semantic description of services such as monitoring service. We are exploring on how to improve P2P information retrieval mechanisms in order to efficiently query the stored RDF services.

6.4.3. Grid Computing for Computational Finance

Participants: F. Baude, V. D. Doan, A. Gaikwad.

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 utilises the ProActive library to parallelise 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 previously developed several European option pricing algorithms such as standard, barrier, basket options to experiment in PicsouGrid [53], [59]. Then several Bermudan American (BA) option pricing algorithms have been implemented (i.e. Longstaff-Schwartz and Ibanez-Zapatero). Due to the terms of BA 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 parallelisation strategy. Our work is thus focused on finding efficient parallelisation 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 parallelisation, and for the model to be used to automatically or semi-automatically provide a load-balanced (for heterogeneous computing resources) parallel implementation.

In 2008, we specifically investigated the parallelisation of the Picazo algorithm for pricing very high dimensional BA options (i.e upto 40 assets) and performed experiments on the Grid'5000 test-bed, up to 250 nodes on multi-sites (i.e. Sophia, Bordeaux, Nancy etc.). We also investigated some additional machine learning techniques: Support Vector Machine and Boosting to compare them to the one proposed in the original version of the Picazo algorithm. The results were reported in the INRIA technical report [38] and later published in the Workshop on High Performance Computational Finance at the Supercomputing Conference [22]. We expect to further investigate the use of different classification algorithms because they influence the predictive accuracy and the computational time.

As part of the Grids@Work conference, we defined the fifth Grid Plugtest for finance - Super Quant Monte Carlo Challenge 2008 with Mireille Bossy and Frédéric Abergel from the MAS laboratory of Ecole Centrale de Paris. Based on the Master/Slave API of the ProActive library, we designed and implemented an API specially for parallel handling of Monte Carlo simulations. During the contest, the participants were successfully experimented their applications using this API for both pricing and hedging of high dimensional European options (i.e. up to 100 assets) such as Vanilla and certain flavours of barrier options using up to thousand computational cores. One publication is under preparation and we already submitted an abstract for a conference in mathematical simulation.

A. Gaikwad also exhibited ProActive middleware and PicsouGrid in the JavaOne conference. All this research work is regularly presented as part of GCPMF project's meetings.

6.4.4. Standardisation of the Grid Component Model

Participants: D. Caromel, C. Dalmasso, G. Laurent, L. Henrio, E. Madelaine, C. Matthieu, B. Sauvan.

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. With the support of our GridCOMP partners, we have been working on the standardisation of various aspect of the Grid Component model (GCM), within the GRID technical committee of ETSI. These standards come in 4 parts:

- GCM Application Description: adopted, July 2008
- GCM Interoperability Deployment: adopted, July 2008
- GCM Fractal ADL (Architecture Description Language): stable draft, Dec 2008
- GCM Application Programming Interface: first draft, Dec 2008

7. Contracts and Grants with Industry

7.1. Introduction

By nature, several research contracts 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. AGOS

"Architecture de Grille Orientée Services" is a project labelled by the *pôle de compétitivité* SCS ("Solutions Communicantes Sécurisées"), and financed by *FCE Ministère de l'Industrie* (from October 2007 to March 2010).

AGOS is a development project integrating and standardising a scientific approach (INRIA) and an industrial approach (HP and Oracle) of two innovative technologies: Grid computing and service oriented architecture. AGOS defines such a generic functional integration architecture. AGOS delivers also a secured software platform providing the following:

- A library of services based on standards;
- A set of tools to build comprehensive applications both Grid and SOA compliant, with their associated operational and business process monitoring in real time;
- A methodology expertise to build on or migrate to this architecture.

Thus, it will be proposed to the partner companies an automated and integrated management environment of the applications activity based on an existing infrastructure with communication as the main paradigm insuring secured cooperation of application components and distributed hardware. This project is clearly based on an industrial process and doesn't constitute an academic exercise targeting at integrating Grid Computing, Service Oriented Architecture and Business Intelligence. Instead, its goal is to deliver a concrete implementation for companies using local area networks up to intra-networks. AGOS will be instantiated for the SCP and Amadeus Use Cases.

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

7.3. MobiTools

The MobiTools project, labelled by the *pôle de compétitivité* Solutions Communicantes Securisées 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 have worked (see [48], [49] on an integrated solution for ProActive active objects and objects running on mobile devices with only J2ME to be able to communicate. Furthermore, we succeeded to emulate the mobility of active objects from a ProActive runtime to a J2ME support and vice versa. Such integration permits to devise distributed and mobile Java applications in a seamless manner whatever the underlying runtime supports.

7.4. 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.

Feasibility Study
The feasibility study focused on the options available for accelerating simulations of electricity generation in airplanes. The simulations are defined using Simulink, a Matlab toolkit for describing and running simulations. The study concluded that simulations were highly tightly coupled and thus very difficult to parallelise. A parametric sweep approach could allow running several simulations in parallel and an implementation was feasible in the project time frame. Other approaches discovered were uncertain in terms of efficiency gain and would require an extension of the project in order to be implemented.

ProActive Matlab toolkit
A Matlab toolkit for integrating a connection to the ProActive Scheduler within Matlab was designed and implemented. This connection brings the ability to run Matlab processes in parallel and seamlessly retrieve results. The toolkit has been integrated within the ProActive Scheduler release and is operational.

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

7.5. Microsoft Compute Cluster Server

Microsoft provides a cluster solution called Compute Cluster Server (now HPC2008). This software provides access to a cluster through high level tools. This contract aims at interfacing Objectweb ProActive with Microsoft Compute Cluster Server so that it is possible to access CCS nodes through the ProActive API.

8. Other Grants and Activities

8.1. National Collaborations

8.1.1. ADT Galaxy (INRIA transversal action)

The ADT Galaxy contributes to make INRIA a value-added player in the SOA arena, by designing and developing an Open Framework for Agile and Dynamic Software Architecture. This ADT will work for INRIA and INRIA's research project-teams direct benefit, and aims at pre-assembling technological bricks from various teams, projects and preparing them to be transferred through the open source software channel.

The ADT aims at providing an IT agile platform, built on dynamic software architecture principles, and fitting for flexibility, dynamical reconfiguration, adaptability, continuity and autonomic computing. Fractal, SCA-Tinfi and GCM/ProActive are the major technologies which will be the technological drivers of this ADT. The different usage scenarios as well as the different tools which will be developed at infrastructure, application and business levels will demonstrate that this platform is able to support the design, modelling, deployment, and execution of business processes. In the same time, the ADT will target the definition of a new common language to manipulate dynamically adaptive distributed SOA-based systems, encompassing application, and middleware layers. This common language will take different forms, inherited from works done by several project-teams with their distinct skills, and illustrates a new kind of collaboration between teams, coupling research and development works.

Contributors to this ADT are mainly research project-teams, including OASIS, ADAM (Lille), ECOO (Nancy), ASCOLA (Rennes), ObjectWeb/TUVALU (Grenoble), SARDES (Grenoble) and TRISKELL (Rennes), and the ADT Galaxy is led and managed by the TUVALU team.

The duration of this ADT is over 28 months : the kickoff meeting has been held on July 3rd, 2008 and the project is planned by end of October, 2010.

8.1.2. ANR DiscoGrid

This ANR funded project gathers 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 allows 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.

The DiscoGrid project is developed upon four main axis: the definition and implementation of a grid-aware mesh partitioner; the specification of a high-level MPI-based programming interface; the design and development of a runtime supporting the proposed extensions and inter-cluster communication and the development of real-size simulation software based on partial differential equations (PDEs).

Initially, the Oasis group was involved in the specification of the programming interface with the definition of the main concepts of hierarchical SPMD communication. In a second phase, the group developed the programming interface (C/C++ and Fortran) and the infrastructure needed for native code wrapping as well as a prototype based on Active Objects and adaptation of existing numerical applications, namely the Poisson3D solver and BHE [51].

Currently, to address Grid hierarchy and high dynamicity, the group is involved in the development of a more advanced GCM-based runtime, organised hierarchically, in order to reflect the effective deployment of the MPI application on the grid [26]. We also develop a more complex application, the CEM, based on electromagnetic wave propagation with the usage of irregular mesh partitioning.

Additionally, this project brought important contributions to the GCM, these contributions include: better definition of collective interfaces; optimisation techniques for collective interfaces; and wrapping and deployment of native applications. Current, yet unpublished, results are very promising as they show the importance of a grid-oriented approach on the development of non-embarrassingly parallel applications.

This project started in January 2006, initially for 36 months, involving 110 kEuros. The project was extended for 6 months upon a demand of project partners.

8.1.3. ANR GCPMF : Grille de Calcul pour les Mathématiques Financières

We collaborate with the TOSCA team within the ANR project entitled “GCPMF” funded by the ANR Research Program "Calcul Intensif et Grilles de Calcul 2005", co-leading the PhD thesis of Viet Dung Doan.

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

8.1.4. 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 web-page 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. IP SOA4All

Service Oriented Architectures for All (SOA4All) is a Large-Scale Integrating Project funded by the European Seventh Framework Programme, under the Service and Software Architectures, Infrastructures and Engineering research area.

Computer science is entering a new generation. The emerging generation starts by abstracting from software and sees all resources as services in a service-oriented architecture (SOA). In a world of services, it is the service that counts for a customer and not the software or hardware components which implement the service. Service-oriented architectures are rapidly becoming the dominants computing paradigm. However, current SOA solutions are still restricted in their application context to being in-house solutions of companies. A service Web will have billions of services. While service orientation is widely acknowledged for its potential to revolutionize the world of computing by abstracting from the underlying hardware and software layers, its success depends on resolving a number of fundamental challenges that SOA does not address today.

SOA4All will help to realize a world where billions of parties are exposing and consuming services via advanced Web technology: the main objective of the project is to provide a comprehensive framework and infrastructure that integrates complementary and evolutionary technical advances (i.e., SOA, context management, Web principles, Web 2.0 and Semantic Web) into a coherent and domain-independent service delivery platform.

OASIS is involved in work packages 1 (SOA4All Runtime), 2 (Service Deployment and Use) and 6 (Service Construction).

We strongly collaborate with the ObjectWeb/TUVALU EPI and also ADAM, from which Philippe Merle is co-leading with Françoise Baude the PhD thesis of Virginie Legrand-Contes.

8.2.2. NOE CoreGRID

CoreGRID is a 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 recognised 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.
- Grid Systems, Tools and Environments: 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.

8.2.3. 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 localised 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.

8.2.4. 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 implement the "invisible grid" concept: abstract away grid related implementation details (hardware, OS, authorisation 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. Therefore, components in the ProActive library also benefit from the underlying features of the library.
- Provide a customisable 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 customisable.

After the first positive evaluation, the second evaluation of GridCOMP occurred this year and was very positive: “The reviewers are impressed by the fact that already two standards have been approved. The scientific work is excellent, the technical work is very good and so are the dissemination and exploitation activities. The review panel would like to congratulate the consortium.”

The project has started in July 2006, for a duration of 33 months, with an overall budget of 674 kEuros.

8.2.5. 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.

8.2.6. 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, to exchange experiences and choose the best standards to build up a Standard Quality Assurance Process, and to establish long-term partnerships between different actors.

Through EchoGRID’s roadmaps for the next 3, 5, and 10 years, the achievements of EchoGRID in connecting actors from EU and China through a series of events in both regions to deliberate top-level challenges for new computing paradigms and define research priorities moving forward.

Oasis currently hosts Zhihui Dai from CNIC as EchoGrid Research Fellowship, working in Inria from 1st April to 31st Dec. 2008, and Rui Zhi from NUDT, as EchoGrid staff exchange fellowship, visiting Inria from 10th Nov- 12th Dec 08.

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 (Urugay), 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 organisation of workshops.

8.3.2. Collaboration with University of Sydney

This collaboration is entitled “Study of ProActive and Standardisation 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 entailed 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 finished in March 2008.

8.3.3. Stic Asia

Stic Asia is a multilateral project with universities of BUPT (Beijing, China), Tsinghua (Beijing, China), SCUT (Ghuanzhou, China), and NUST (Pakistan). *Experiments and Dissemination on Grid Standard: ProActive GCM*, is a collaborative research and academic exchanges project on Grid standard between Inria and the partners from Asia. It is partially funded by French ministry of Foreign affairs starting from July 2007 and will finish in June 2009.

The main objective of this project is to foster international scientific cooperation in Grid research between France and Asian partners, share experiments and disseminate ProActive and Grid Component Model (GCM) standard for Grid Middleware and applications interoperability. Furthermore, it is intended to support and establish partnership from mobility programs in a short and long term.

9. Dissemination

9.1. Program Committees and Conference Organisation

All the team was involved in the organization the Grid PlugTests and Contest and the associated Grids@work conference and ProActive Users Group Meeting in October at INRIA Sophia Antipolis. During this events, the team gave several talks, organized tutorials, and the ProActive User Group Meeting. This year, the FMCO'08, a conference gathering European projects on formal methods for components and objects occurred during the Grids@work week, allowing a better involvement of the community on formal methods.

- **Eric Madelaine**
  Program committee chair and steering committee member of FACS'08,
  Chair of FMCO'08, Program committee member of SAVBCS'08, Qosa'08, SERA'08, and of the journal l'Objet.
- **Françoise Baude**
  Program committee member of *Workshop on Middleware for Grid Computing - MGC* joint with ACM/IFIP/USENIX International Middleware Conference 2008, IASTED Parallel and Distributed Computing and Networks conference (PDCN 2009), *Workshop for Component-Based High-Performance Computing (CBHPC2008)* with ACM CompArch 2008, ECOOP (European Conference on Object-Oriented Programming) 2009.
  Reviewer for *Concurrency and Computation:Practice and Experience, Wiley*.
  Reviewer for an ANR project submitted to the call “Domaines émergents: Nouveaux défis scientifiques et technologiques”, 2008; Internal Reviewer for the D.6.1.1 SOA4ALL deliverable, and for 3 GridCOMP deliverables (D.NFCF.03, D.NFCF.04, D.COL.03).
- **Ludovic Henrio**
  Program Committee member of FESCA'09. Internal Reviewer for the D 1.1.3/D3.1.3 BIONETS deliverable
- **Fabrice Huet**
  Program committee member of HPDC'08, CCGrid'08, HiPerGrid08 and PCGrid08.

9.2. Short Visits

- Rui Zhi from NUDT, as EchoGrid staff exchange fellowship, visiting inria from 10th Nov- 12th Dec 08.
- in 2008, we have hosted several research visitors from China and Pakistan partners, including Mr. Yulai Yuan from Tsinghua Univ, China who stayed from April to July 2008, and Mr. Kamran Qadir from NUST, Pakistan, who stayed from July to December 2008.

9.3. Teaching

9.3.1. Teaching

- **Françoise Baude**
  From September 2008, co-director of the Département d'Informatique de l'UFR Sciences, University of Nice-Sophia Antipolis; Course convenor in Parallel and Distributed Programming (Master 1 IFI with the involvement of Fabrice Huet); in Web Services and Service Oriented Architectures (Master 2 MIAGE, Nouvelles Technologies et Direction de Projets); Operating System Design and Concurrent Programming (Licence 3, IUT Informatique 2 with the involvement of Elton Mathias), and numerous exercice labs involvement (in XML and Web technologies, Java programming, multi-threaded programming, ...)
- **Denis Caromel**
  Course convenor of Distributed Programming and Multi-Tiers Architectures (Master 1) with the involvement of Mario Leyton.
- **Fabrice Huet**
  Coordinator of the 1st year of Master of Computer Science,
  Course convenor of Advanced Operating System (Master 1),
  Distributed Systems (Master 1) and Network Game Programming (Licence 3)

9.3.2. Theses

The following theses have been defended this year:

- Antonio Cansado: "Verification and Generation of Safe Distributed Components" (Dec 2008), director Eric Madelaine
- Mario Leyton: "Advanced Features for Algorithmic Skeleton Programming" (Oct 2008), director Denis Caromel

The following theses are in preparation:

- Brian Amedro: " title" (Since Oct 2007), director Denis Caromel.
- Muhammad Uzair Khan: "Supporting First Class Futures in a Fault-Tolerant Java Middleware" (Since Oct 2007), director Ludovic Henrio.
- Marcela Rivera: "Reconfiguration and Life-cycle of Distributed Components: Asynchrony, Coherence and Verification" (since Dec 2006), director Ludovic Henrio.
- 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), director Françoise Baude and Ludovic Henrio.
- Viet Dung Doan: "Adequation of grid computing to computation intensive calculations in the financial domain" (Since Oct 2006), director Françoise Baude and Mireille Bossy.
- Imen Filali: "Peer to Peer computational Grids with reservation in service oriented architectures" (Since Oct 2007), director Fabrice Huet.

- Elton Mathias: “Hierarchical Grid Programming based upon a component-oriented approach” (Since Sept 2007), director Françoise Baude.
- Virginie Legrand-Contes: “Large Scale and Distributed Services Orchestration” (Since Mar 2008), director Françoise Baude and Philippe Merle.
- Cristian Ruz: “Autonomic Service Deployment and Management of grid-based enterprise services” (Since Mar 2008), director Françoise Baude.
- Guilherme Peretti-Pezzi: “ProActive Parallel Hydraulic Simulations for Grid and SOA Environments” (Since October 2008), director Denis Caromel, CIFRE funding with the “Canal de Provence” company.

9.3.3. Internships

- Boutheina Bannour: “Reconfiguration of distributed components”
- Baptiste de Stefano: “Middleware permettant d’intégrer des applications pour téléphones mobiles dans des applications réparties de type ProActive”
- Nicolas Dodelin: “Intégration de la technologie ProActive dans le logiciel Celsius”
- Pablo Valenzuela: “An Eclipse editor for the Java Distributed Component Description Language”
- Sona Djohou: “Outils pour la preuve formelle de propriétés ASP”
- Vasile Jureschi: “Applying scientific and technical writting to the ProActive library”
- Julian Krzeminski: “Exploitation of Microsoft CCS (Compute Cluster Server) with ProActive GCM ETSI Standard”
- Abhishek Gupta: “Parallélisation de script Matlab/Simulink et Scilab/Sicos avec ProActive”
- Tomasz Dobek: “Computational Peer-To-Peer System on Destok Machines”
- Vivien Maisonneuve: “Etude et implantation d’un algorithme de vérification pour les systèmes distribués avec canaux Fifo non bornés
- Maxime Menant: “Step by Step Debug for a Large-Scale Distributed System”
- Jean-Michel Guillaume: “Virtualization and scheduling for Parallel and Distributed Tasks with Fault Tolerance (Analyse, Conception, Prototype and tests)”
- Nirski Krzysztof: “Generation of behavioural models for GCM components : formalisation and implementation”
- Mikolaj Baranowski: “Generation of behavioural models for GCM components : formalisation and implementation”
- Ankush Kumar KAPUR: “Data Space for Grid and SOA : Accessing and Managing Remotes Files and Data to be carried out within the Open Source middleware ProActive”
- Florin-Alexandru Bratu: “Analysis, Design and implementation of ProActive integration with JPEE platforms using annotations ant other techniques”
- Etienne Vallette d’Osia: “Step by Step Debug for a Large-Scale Distributed System”
- Samuel Bauer: “Développer ProActive, un middleware Java pour le calcul distribué et parallèle”
- Jonathan Martin: “Hierarchical Scheduling and resource Management”
- Ali Fawaz: “Hierarchical Scheduling and resource Management”
- Laurent Vanni: “Step by Step Debug for a Large-Scale Distributed System”

9.3.4. Member of thesis Committees

- **Françoise Baude**
  • Reviewer and member of PhD thesis committee of Jeremy Dubus (10/10/2008), *Une démarche orientée modèle pour le déploiement de systèmes en environnements ouverts distribués*. Directors Jean-Marc Geib and Philippe Merle, ADAM EPI.
  • Reviewer and member of PhD thesis committee of Jakub Kornas (23/10/2008), *Contributions to software deployment in a component-based reflective architecture*. Directors Jean-Bernard Stefani and Olivier Gruber, SARDES EPI.
- **Denis Caromel**
  • Reviewer and member of PhD thesis committee of Maciej Malawski, *Component-based Methodology for Programming and Running Scientific Applications on the Grid*, october 2008. Supervisor: Prof. Jacak Kitowski and Prof. Marian Bubak.
  • Reviewer and member of HDR committee of Frédéric Guidec, *Déploiement et support à l'exécution de services communicants dans les environnements d'informatique ambiante*, 27 juin 2008, Valoria, Université de Bretagne Sud.

9.4. Participation in conferences and workshops

9.4.1. Invited Talks

- **Francoise Baude**
  Invited talk at INRIA Rhône Alpes/IMAG: "The Grid Component Model and its applications", April 8, 2008
- **Ludovic Henrio**
  Invited talk at Technische Universität Berlin: "ASP: Asynchronous Sequential Processes"
- **Denis Caromel**
  • "Parallel Code Performance Boost: A Java Toolkit for Retrofitting Parallel and Grid Capabilities", invited seminar at Harvard Medical School, January 2008, Boston.
  • "From Components to Services to Utilities", Contribute to the European roadmap for Software & Software Services, Brussels, March 4, EU FP8 Preparation Meeting.

  • "Bridging Distributed and Multi-Core Computing: ProActive for Parallelism and Grids" IBM Research Zurich, Invited Seminar, April 21st, Zurich.
  • "Strong Programming Model for Strong Weak Mobility: The ProActive Parallel Suite" MDM 2008, Invited Keynote, April 29, Beijing, China.
  • "SOA + GRIDs with the AGOS project and ProActive", invited talk at EMEA SOA Oracle & BEA Architect Seminar, June 5th 2008, Nice, France.

  •"ProActive Parallel Suite: Bridging Distributed and Multi-Core Computing", DAPSYS 2008, 7th International Conference on Distributed and Parallel Systems, Invited Keynote, Debrecen, Hungary, September 4, 2008.

  •"The GridCOMP EU project", invited seminar at XtreemOS plenary meeting, october 8th, Ljubljana, Slovania.

  • "Grid Component Model and ProActive Parallel Suite for Science", CGW'08, Cracow Grid Workshops, Invited Keynote, October.

  • "ProActive and GCM: architecture and overview of new features", invited Seminar, 5th ProActive and GCM User Group, in GRIDs@Work, October 21, 2008, Sophia, France.

  •"GridCOMP: An Advanced Component Platform for an Effective Invisible Grid", invited Seminar, Technical Concertation meeting, in GRIDs@Work, October 22, 2008, Sophia, France.

  • "GridCOMP Overview and results of the 5th Grid Plugtests Contest", invited presentation in EU-China Cooperation workshop, 28 October 2008, Beijing, China.

  • "Java for HPC Now: MPI Fortran vs. ProActive Java", invited presentation at Sun Microsystems HPC Consortium, November 17, 2008, co-located with SC'08, Austin, USA.

- **Eric Madelaine**
  Invited talk at Charles University, Prague, nov 2008: “Specification and Verification Tools for Grid Component-based Applications”
- **Fabrice Huet**
  Invited talk at the 1st EDGeS Grid training workshop & 2nd AlmereGrid Grid Experience workshop, November 27, 2008, Almere, The Nederlands : “The ProActive Parallele Suite”

9.4.2. Talks in Workshops and Conferences

- **Ludovic Henrio**
  FMCO’08 Symposium: “Distributed Components and Futures: Models and Challenges”
  FACS’08: “Transparent First-class Futures and Distributed Components”
- **Eric Madelaine**
  FMCO’08 Symposium: “Specification and Verification Tools for Grid Component-based Applications”
- **Mario Leyton**
  PDP Euromicro’08: “Type safe algorithmic skeletons”
  IPDPS 2008: “A Transparent non-Invasive File Data Model for Algorithmic Skeletons”
  JavaPDC at IPDPS 2008: “Java ProActive vs. Fortran MPI: looking at the Future of Parallel Java ”
  DPA at Euro-par 2008: “ProActive Parallel Suite: From Active Objects-Skeletons-Components to Environment & Deployment”
- **Antonio Cansado**
  FACS’08: “Unifying Architectural and Behavioural Specifications of Distributed Components”
- **Viet Dung Doan**
  Workshop on Finance at SuperComputing 2008: “Gridifying” Classification-Monte Carlo algorithm for pricing high-dimensional American options
- **Marcela Rivera**
  CBHPC’08: “An algorithm for safely stopping a component system”
- **Elton Mathias**
  CBHPC’08:“A GCM-Based Runtime Support for Parallel Grid Applications”

9.4.3. Tutorials

- Guilherme Peretti Pezzi, Cristian Ruz and Elaine Isnard. “ProActive Parallel Suite and GCM Components”. CoreGrid Summer School 2008. Dortmund, Germany, Jul. 10th 2008
- Cristian Ruz. Tutorial on ProActive Parallel Suite, GCM Components, and Hands-On Grid Programming. Universidad de Talca, Curicó, Chile, Nov. 19-20th 2008
- Fabrice Huet. Tutorial at the ASCI Course A14: Advanced Grid Programming Models: “The ProActive grid programming environment”
- Cédric Dalmasso and Bastien Sauvan. “ProActive/GCM Hands-On Tutorial on component”. V Grids@Work. INRIA, Sophia Antipolis, France, Oct. 23rd, 2008

9.4.4. Miscellaneous

- **Ludovic Henrio**
  • Presentation at Bionets *review meeting*: “Graph-based Service Individual specification: Creation and Representation”
  • Scientific presentation at the *CoreGrid Final review*: “Recent results in the GCM”
  • Presentation at the *CoreGrid* WP3 and WP7 meeting: “A Component Platform for Experimenting with Autonomic Composition” and “Implementing MxN Pattern in GCM/ProActive”

- **Eric Madelaine**
  Presentation at the *Reseco* workshop, dec 2008: “Specification and Verification Tools for Grid Component-based Applications”
- **Denis Caromel**
  “EU-China Standardization and Quality Assurance”, presentation at EU EchoGRID Review, WP 3: Brussels, March 12 2008.
- **Viet Dung Doan**
  Annual GCPMF meeting in Calyon Paris: “Grid based Bermudian Ð American option pricing algorithms : Analysis of two methods”
- **Vincent Cavé and Elton Mathias**
  Annual ANR DiscoGRID meeting: “WP2/WP4 status and next steps”, June 2008
- **Abhijeet Gaikwad**
  GCPMF meeting “Distributed and Grid Computing in Computational Finance” held within the V Grids@work conference: High Dimensional American Option Pricing with ProActive, October 2008.

10. Bibliography

Major publications by the team in recent years

[1] L. Baduel, F. Baude, D. Caromel. *Asynchronous Typed Object Groups for Grid Programming*, in "International Journal of Parallel Programming", vol. 35, n^{o} 6, 2007, p. 573-614.

[2] 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.

[3] 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.

[4] 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.

[5] D. Caromel. *Towards a Method of Object-Oriented Concurrent Programming*, in "Communications of the ACM", vol. 36, n^{o} 9, September 1993, p. 90–102.

[6] D. Caromel, L. Henrio. *A Theory of Distributed Object*, Springer-Verlag, 2005.

[7] 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.

[8] D. Caromel, W. Klauser, J. Vayssière. *Towards Seamless Computing and Metacomputing in Java*, in "Concurrency Practice and Experience", vol. 10, n^{o} 11–13, November 1998, p. 1043–1061.

[9] 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^{o} 3086, jun 2004, p. 316-340.

[10] 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

Doctoral Dissertations and Habilitation Theses

[11] A. Cansado. *Formal Specification and Verification of Distributed Component Systems*, Ph. D. Thesis, Université de Nice - Sophia Antipolis – UFR Sciences, December 2008.

[12] M. Leyton. *Advanced Features for Algorithmic Skeleton Programming*, Ph. D. Thesis, Université de Nice - Sophia Antipolis – UFR Sciences, October 2008.

Articles in International Peer-Reviewed Journal

[13] T. Barros, R. Boulifa, A. Cansado, L. Henrio, E. Madelaine. *Behavioural Models for Distributed Fractal Components*, in "Annals of Telecommunications", to appear, 2008, https://hal.inria.fr/inria-00268965.

[14] F. Baude, D. Caromel, C. Dalmasso, M. Danelutto, V. Getov, L. Henrio, C. Pérez. *GCM: A Grid Extension to Fractal for Autonomous Distributed Components*, in "Annals of Telecommunications", to appear, 2008.

Invited Conferences

[15] D. Caromel, L. Henrio, E. Madelaine. *Active Objects and Distributed Components: Theory and Implementation*, in "FMCO 2007, Berlin Heidelberg", F. de Boer, M. Bonsangue (editors), LNCS, n^{o} 5382, Springer-Verlag, 2008, p. 179–199.

[16] D. Caromel, M. Leyton. *ProActive Parallel Suite: From Active Objects-Skeletons-Components to Environment & Deployment*, in "Euro-Par 2008 Workshops: Parallel Processing, Las Palmas de Gran Canaria, Spain", Lecture Notes in Computer Science, To appear, Springer-Verlag, August 2008.

International Peer-Reviewed Conference/Proceedings

[17] B. Amedro, V. Bodnartchouk, D. Caromel, F. Huet. *Java ProActive vs. Fortran MPI: Looking at the Future of Parallel Java*, in "Workshop in IEEE IPDPS'2008 Proceedings, IEEE CS Press, Miami", IPDPSP'08, april 2008.

[18] A. Cansado, L. Henrio, E. Madelaine. *Transparent First-class Futures and Distributed Component*, in "International Workshop on Formal Aspects of Component Software (FACS'08), Malaga", Electronic Notes in Theoretical Computer Science (ENTCS), Sept 2008.

[19] A. Cansado, L. Henrio, E. Madelaine, P. Valenzuela. *Unifying Architectural and Behavioural Specifications of Distributed Components*, in "International Workshop on Formal Aspects of Component Software (FACS'08), Malaga", Electronic Notes in Theoretical Computer Science (ENTCS), Sept 2008.

[20] D. Caromel, L. Henrio, M. Leyton. *Type Safe Algorithmic Skeletons*, in "PDP '08: Proceedings of the 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2008), Toulouse, France", Selected for the Best Paper Session, IEEE Computer Society, February 2008, p. 45–53.

[21] D. CAROMEL, M. LEYTON. *A Transparent non-Invasive File Data Model for Algorithmic Skeletons*, in "22nd International Parallel and Distributed Processing Symposium (IPDPS), Miami, USA", IEEE Computer Society, March 2008, p. 1-8.

[22] V. D. DOAN, A. GAIKWAD, M. BOSSY, F. BAUDE. *"Gridifying" Classification-Monte Carlo algorithm for pricing high-dimensional American options*, in "Workshop on High Performance Computational Finance, IEEE/ACM Supercomputing conference 2008", 2008.

[23] I. FILALI, F. HUET, C. VERGONI. *A Simple Cache Based Mechanism for Peer to Peer Resource Discovery in Grid Environments*, in "CCGRID '08: Proceedings of the 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID), Washington, DC, USA", IEEE Computer Society, 2008, p. 602–608.

[24] R. GASCON. *An Automata-based Approach for CTL* with Constraints*, in "Proceedings of the 10th International Workshop on Verification of Infinite State Systems (INFINITY'08), Toronto, Canada", P. HABERMEHL, T. VOJNAR (editors), Electronic Notes in Theoretical Computer Science, To appear, Elsevier Science Publishers, 2008.

[25] L. HENRIO, M. RIVERA. *Stopping Safely Hierarchical Distributed Components*, in "Proceedings of the Workshop on Component-Based High Performance Computing (CBHPC'08) in conjunction with ACM SIGPLAN CompArch 2008", October 2008.

[26] E. MATHIAS, F. BAUDE, V. CAVÉ. *A GCM-Based Runtime Support for Parallel Grid Applications*, in "Proceedings of the Workshop on Component-Based High Performance Computing (CBHPC'08) in conjunction with ACM SIGPLAN CompArch 2008", October 2008.

[27] E. TEJEDOR, R. M. BADIA, P. NAOUMENKO, M. RIVERA, C. DALMASSO. *Orchestrating a safe functional suspension of GCM components*, in "CoreGRID Integration Workshop. Integrated Research in Grid Computing", April 2008.

Scientific Books (or Scientific Book chapters)

[28] F. BAUDE, D. CAROMEL, L. HENRIO, P. NAOUMENKO. *A Flexible Model and Implementation of Component Controllers*, CoreGRID, Springer, 2008.

[29] J. BUSTOS-JIMENEZ, D. CAROMEL, M. LEYTON, J. M. PIQUER. *Load Information Sharing Policies in Communication-Intensive Parallel Applications*, T. PRIOL, M. VANNESCHI (editors), Springer-Verlag, 2008, p. 111-122.

[30] 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", Lecture Notes in Computer Science, http://agrausch.informatik.uni-kl.de/CoCoME, vol. 5153, Springer, 2008.

[31] M. MALAWSKI, T. GUBALA, M. KARZTELNIK, T. BARTYNSKI, M. BUBAK, F. BAUDE, L. HENRIO. *High-level scripting approach for building component-based applications on the grid*, CoreGRID, Springer, 2008, p. 307–320.

Books or Proceedings Editing

[32] E. MADELAINE, M. LUMPE (editors). *Proceedings of the 4th International Workshop on Formal Aspects of Component Software*, vol. vol 215, ENTCS, 2008, http://hal.inria.fr/inria-00337975/en/.

Research Reports

[33] M. ALDINUCCI, S. CAMPA, M. COPPOLA, M. DANELUTTO, G. ZOPPI, A. BASSO, A. BOLOTOV, F. BAUDE, H. BOUZIANE, D. CAROMEL, L. HENRIO, C. PÉREZ, J. CUNHA, C. MICHAEL, P. CLASSEN, C. LENGAUER, J. COHEN, S. MC GOUGH, N. CURRLE-LINDE, P. DAZZI, N. TONELLOTTO, J. DÜNNWEBBER, S. GORLATCH, P. KILPATRICK, N. RANALDO, E. ZIMEO. *Proceedings of the Programming Model Institute Technical meeting 2008*, Technical report, n^{O} TR-0138, Institute of Programming Model, CoreGRID - Network of Excellence, May 2008, http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0138.pdf.

[34] B. AMEDRO, V. BODNARTCHOUK, D. CAROMEL, C. DELBÉ, F. HUET, G. TABOADA. *Current State of Java for HPC*, RT-0353, Rapport Technique, 2008, http://hal.inria.fr/inria-00312039/en/.

[35] T. BARROS, R. BOULIFA, A. CANSADO, L. HENRIO, E. MADELAINE. *Behavioural Models for Distributed Fractal Components*, Research Report, n^{O} 6491, INRIA, 04 2008, http://hal.inria.fr/inria-00268965.

[36] F. BAUDE, V. CAVÉ, E. MATHIAS. *Distributed objects and parallel component implementation*, Technical report, ANR DiscoGrid project, Delivrable from the WP 2: Distributed objects and parallel components communications, Jan 2008.

[37] F. BAUDE, L. HENRIO, P. NAOUMENKO, H. PFEFFER. *Graph-Based Service Individual specification: Creation and Representation*, http://www.bionets.eu/docs/BIONETS_D3_2_3.pdf, Technical report, BIONETS IP Project Delivrable from the Autonomic Service Life-Cycle And Service Ecosystems WP (3.2), Jan, revised June 2008.

[38] M. BOSSY, F. BAUDE, V. D. DOAN, A. GAIKWAD, I. STOKES-REES. *Parallel Pricing Algorithms for Multi–Dimensional Bermudan/American Options using Monte Carlo methods*, Research Report, n^{O} 6530, INRIA, 05 2008, https://hal.inria.fr/inria-00278514.

[39] A. CANSADO, E. MADELAINE, P. VALENZUELA. *VCE: A Graphical Tool for Architectural Definitions of GCM Components*, Technical report, n^{O} ITI-08-07, Sep 2008.

[40] L. HENRIO, D. CAROMEL, F. BAUDE. *A component platform for experimenting with autonomic composition*, CoreGRID Technical Report TR-138, Technical report, 2008.

[41] L. HENRIO, M. RIVERA. *An algorithm for safely stopping a component system*, RR-6444, Rapport de recherche, 2008.

[42] J. LAHTI, L. HENRIO, K. VILLE, F. LAURA, D. MIORANDI, D. LINNER, H. PFEFFER, F. BAUDE. *Advanced Service Life-Cycle and Integration*, Technical report, n^{O} D.3.2.4, BIONETS IP Project Delivrable from the Autonomic Service Life-Cycle And Service Ecosystems WP (3.2), June 2008.

[43] D. LINNER, H. PFEFFER, S. STEGLICH, F. BAUDE, L. HENRIO, P. NAOUMENKO. *Service Probes Implementation and Evaluation*, Technical report, n^{O} D.3.4.1, BIONETS IP Project Delivrable from the Service Probes WP (3.4), June 2008.

[44] NESSI-GRID WP1 PARTNERS. *Grid Vision and Strategic Agenda, Deliverable D.1.5*, final version, Technical report, NESSI-Grid SSA EU project, Nov 2008.

[45] N. RANALDO, G. TRETOLA, E. ZIMEO, N. CURRLE-LINDE, W. A. ZOUAB, O. MANGOLD, M. RESCH, D. CAROMEL, A. DI COSTANZO, C. DELBÉ, J. FRADJ, M. LEYTON, J.-L. SCHEEFER. *A Scheduler for a Multi-paradigm Grid Environment*, Technical report, n^{o} 155, CoreGrid, July 2008.

Other Publications

[46] B. BANNOUR. *Langage de Reconfiguration pour Composants Distribués*, Mémoire de recherche, Masters thesis, Université de Bordeaux 1, june 2008.

[47] B. BANNOUR. *Langage de Reconfiguration pour Composants Distribués*, Rapport de stage, Masters thesis, Ecole Nationale Supérieure d'Electronique, Informatique et Radiocommunication, September 2008.

[48] N. DODELIN. *Intégration de la technologie ProActive dans le logiciel Celsius*, Masters thesis, Polytech'Nice, Département d'Informatique, September 2008.

[49] B. D. STEFANO. *Middleware permettant d'intégrer des applications pour téléphones mobiles dans des applications réparties de type ProActive*, Masters thesis, Polytech'Nice, Département d'Informatique, September 2008.

References in notes

[50] M. ABADI, L. CARDELLI. *A Theory of Objects*, Springer-Verlag, New York, 1996.

[51] 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.

[52] 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.

[53] S. BEZINNE, V. GALTIER, S. VIALLE, F. BAUDE, M. BOSSY, VIET DUNG. 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.

[54] E. BRUNETON, T. COUPAYE, M. LECLERCP, V. QUEMA, JEAN-BERNARD. 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.

[55] 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.

[56] P. DAVID, T. LEDOUX. *Safe Dynamic Reconfigurations of Fractal Architectures with FScript*, in "Proceeding of Fractal CBSE Workshop, ECOOP'06, Nantes, France", 2006.

[57] J. DÜNNWEBBER, F. BAUDE, V. LEGRAND, N. PARLAVANTZAS, S. GORLATCH. *Towards Automatic Creation of Web Services for Grid Component Composition*, in "Integrated Research in GRID computing - Invited papers from the 1st CoreGRID Integration Workshop 2005", ISBN: 978-0-387-47656-8. Also as the CoreGrid TR003 report, http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0003.pdf, Springer, 2006.

[58] 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.

[59] I. STOKES-REES, F. BAUDE, VIET DUNG. DOAN, M. BOSSY. *Managing parallel and distributed Monte Carlo simulations for computational finance in a grid environment*, in "Proceedings of the International Symposium on Grid Computing 2007", S. LIN (editor), Springer Verlag, 2008.