



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team oasis

*Objets Actifs, Sémantique, Internet et
Sécurité*

Sophia Antipolis - Méditerranée

Theme : Distributed Systems and Services

Activity
R *eport*

2009

Table of contents

1. Team	1
2. Overall Objectives	2
2.1. Presentation	2
2.2. Highlights	2
3. Scientific Foundations	2
3.1. Programming with Distributed Objects and Components	2
3.2. Formal models for distributed objects	3
3.3. Static Analysis and Verification	3
4. Application Domains	4
4.1. Grid Computing	4
4.2. Service Oriented Architectures (SOA)	4
5. Software	5
5.1. ProActive	5
5.2. Vercors platform	7
6. New Results	8
6.1. Distributed Programming Models	8
6.1.1. Theory of Distributed Objects	8
6.1.2. Skeleton-based parallel programming	8
6.2. Component-oriented Grid Programming	8
6.2.1. Mechanised formalisation of a Distributed Component Model	8
6.2.2. Designing Non-Functional Concerns as Components	9
6.2.3. Reconfiguring Distributed Components	10
6.2.4. SPMD and Components	10
6.2.5. Monitoring Components	11
6.2.6. Mapping Workflows on Components	11
6.3. Behavioural Specification for Reliable Distributed Components	12
6.4. Grid Middleware and Applications	12
6.4.1. Optimising Distributed Object Computing	12
6.4.2. Peer-to-Peer Infrastructure	13
6.4.3. Grid Computing for Computational Finance	13
6.4.4. Federating DSBs at Internet Scale Upon a Component-Based Approach	14
6.4.5. Standardisation of the Grid Component Model	15
7. Contracts and Grants with Industry	15
7.1. Introduction	15
7.2. AGOS	15
7.3. Thales Avionics Electrical Systems S.A.	16
7.4. Microsoft Compute Cluster Server	16
8. Other Grants and Activities	16
8.1. National Collaborations	16
8.1.1. ADT Galaxy (INRIA transversal action)	16
8.1.2. ANR DiscoGrid	17
8.1.3. ANR OMD2	17
8.1.4. Contrat Plan État Région Grille et Calcul Pair-à-Pair	18
8.2. European Collaborations	18
8.2.1. IP SOA4All	18
8.2.2. IP Bionets	18
8.2.3. Strep GridCOMP	19
8.3. International Collaborations	20
8.3.1. Stic-Amsud ReSeCo	20

8.3.2. Stic Asia	20
9. Dissemination	20
9.1. Program Committees and Conference Organisation	20
9.2. Short Visits and external collaborators	21
9.3. Teaching	21
9.3.1. Teaching	21
9.3.2. Theses	21
9.3.3. Internships	22
9.3.4. Member of thesis Committees	22
9.4. Participation in conferences and workshops	22
9.4.1. Invited Talks	22
9.4.2. Talks in Workshops and Conferences	23
9.4.3. Tutorials	23
9.4.4. Miscellaneous	23
10. Bibliography	24

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

1. Team

Research Scientist

Eric Madelaine [CR INRIA]
Ludovic Henrio [CR CNRS]

Faculty Member

Denis Caromel [Team Leader, Prof. UNSA, HdR]
Françoise Baude [MdC UNSA, HdR]
Fabrice Huet [MdC UNSA]

Technical Staff

Bastien Sauvan [ADT Galaxy]
Elaine Isnard [AGOS]
Fabien Viale [FP6 S4All, FP6 NESSI-Grid, Thales]
Fabrice Fontenoy [AGOS]
Florin-Alexandru Bratu [IA-INRIA]
Franca Perrina [AGOS]
Germain Sigety [INRIA, to sep 09]
Oleg Smirnov [Microsoft, Thales]
Yu Feng [ETSI, FP6 EchoGrid]

PhD Student

Brian Amedro [SCS-Agos]
Cristian Ruz [INRIA, Conicyt]
Elton Mathias [INRIA-Cordi]
Francesco Bongiovanni [INRIA-PACA, since sep 09]
Guilherme Peretti Pezzi [SCP]
Imen Filali [SOA4All]
Marcela Rivera [INRIA, Conicyt]
Muhammad Khan [STIC-Asia]
Paul Naoumenko [INRIA/Région PACA, to sep 09]
Sheheryar Malik [alcatel selfnets, since oct 09]
Viet Dung Doan [FP6 Bionets, ANR-GCPMF]
Virginie Legrand Contes [SOA4ALL]
Antonio Cansado [INRIA, Conicyt, to feb 09]

Post-Doctoral Fellow

Regis Gascon [INRIA-Cordi, to apr 09]

Visiting Scientist

Dou Sun [Stic Asie, since oct 09]

Administrative Assistant

Sandra Devauchelle [I3S]
Sylvie Lelaidier [INRIA, since jun 09]
Patricia Maleyran [INRIA, to may 09]

Other

Laurent Pellegrino [Apprentice, Master IFI, since oct 09]
Marc Valdener [Apprentice, Master IFI, since oct 09]
Etienne Valette d'Osia [Apprentice, master ISI, to oct 09]
Laurent Vanni [Apprentice, master ISI, to oct 09]

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 (semantic 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:
 - 3 ETSI TC Grid standards are now officially approved and published by the project: “GCM Interoperability Deployment standard”, “GCM Application Interoperability Description” in 2008, and “GCM Fractal ADL” [41] in 2009.
 - and 1 new work items have been approved towards standardisation: ‘ “GCM Management API”

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 [51]. We have been involved in the design of the Grid Component Model (GCM) [53], 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; not all GCM features are implemented in ProActive yet. ProActive/GCM is intended to become the reference implementation of the GCM, as was 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 [3]. On the other hand, we generate “safe by construction” code skeletons, from high level specifications; this guarantees the behavioural properties of the components. We then use generic tools from the verification community to check properties of these models. We concentrate on behavioural properties, expressed in terms of temporal logics (safety, liveness), of adequacy of an implementation to its specification and of correct composition of software components.

4. Application Domains

4.1. Grid Computing

As distributed systems are becoming ubiquitous, Grid computing, and the more recently emerging close concept known as Cloud computing are facing a major challenge for computer science: seamless access and use of large-scale computing resources, world-wide. The word "Grid" was 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)

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 large-scale distributed 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 [55], and we have conducted 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 services are composed and orchestrated through an Enterprise Service Bus. But, to scale, the ESB must itself be distributed, and must incorporate from design time, the necessary mechanisms to handle large-scale distribution and possibly huge amount of end-user or technical (service discovery, registry, orchestration and monitoring engines, etc) services. Moreover the bus itself should be deployable seamlessly on any heterogeneous combination of host machines. In this wide context, we intend to use GCM components for building

the bus itself, giving it the required Internet scale capabilities (this subject is specifically addressed in the context of the SOA4ALL IP FP7 project the team is involved in, taking the OW2 PEtALS ESB from the PEtALS link SME as starting point)

- 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 conduct 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 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 and deployment problems. 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.-D. Doan, F. Huet [correspondant], E. Isnard, M. Khan, V. Legrand, E. Mathias, P. Naoumenko, F. Perrina, M. Rivera, G. Sigety, F. Viale, F. Bratu, F. Fontenoy, O. Smirnov.

ProActive is a Java library (Source code under GPLv3 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 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 (Apache Axis2 and CXF),
- HTTP transport,
- ssh, rsh, RMI/ssh tunnelling,
- Globus: GT2, GT3, GT4, gsi, Unicore, ARC (Nordug)
- LSF, PBS, Sun Grid Engine, OAR, 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 deployment framework, which has been standardised by the ETSI, allows the deployment of ProActive, native including MPI applications on almost all Grid/cluster protocol: Windows CCS, Sun Grid Engine, LSF, OAR, PBS, SSH, RSH etc. It also supports all the major virtualization products such as VMware, Xen, KVM and Virtualbox;
- The communication layer that can rely on RMI, HTTP, IBIS, RMI/ssl or RMI/ssh. In particular, this last protocol can cross firewalls in many cases;
- Two mechanisms: the ProActive message routing protocol (PAMR), and another based on the use of the ssh proxy command mechanism have been designed to allow ProActive Programming deployment in the context of limited ingoing and outgoing network connections. This is particularly useful in the following cases:
 - Network address translation devices;
 - Firewalls allowing only outgoing connections (this is the default setup of many personal firewall);
 - Virtual Machines with a virtualized network stack.
- GCM component support;
- The graphical user interface IC2D offers many views of an application such as:
 - the Monitoring view which allows better control and monitoring;
 - the TimIt view that draws graphics on time statistics;
 - the CharIt view providing charts any ProActive numerical values.
- The ability to exploit the migration capability of active objects, in network and system management;
- 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 job scheduler for scheduling many kinds of jobs like Java, ProActive, native, scripts executables, Matlab and Scilab programs, etc.

- A resource manager able to manage various hosting machines, gained through SSH, PBS, Amazon EC2, GCM or Virtual Machine.
- 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;
- Remote data access for accessing data stored in a remote data space;
- MPI code wrapping to deploy MPI application using the GCM deployment;
- Remote debugging support;
- Active object and component exposition as web services using Axis2 or CXF;

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 former ObjectWeb, now OW2 Consortium. OW2 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] [47] and to the web pages <http://www.objectweb.org> and <tp://proactive.inria.fr/> which list several white papers.

The following new features have been developed in 2009:

- Dataspaces for remote data access
- GCM Deployment now supports all the major virtualization products (VMware, Xen, KVM, Virtualbox)
- New legacy and MPI code wrapping
- New Web Service exportation for active objects and components (Apache Axis2 and CXF)
- Remote debugging support
- New PAMR (ProActive Message Routing) over SSH communication protocol
- New RMI over SSL communication protocol, and use of the SSH proxy command
- Distributed logging framework (log4j)
- Annotation based static code checking
- Enhanced HTTP support (Jetty based)

5.2. Vercors platform

Participants: A. Cansado, E. Madelaine [correspondant], R. Gascon, M. Adrover, A. Parisy, A. Bouchakhchoukha.

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 the Vercors tools, strengthening the component editor, defining a major revision of the data-type library, and going through a major refactoring of the model generation module (ADL2N). We also have started a work on translation of the pNets model to the Fiacre format, that should be one of the future “standardized” input format for the models-checkers we use [28], but this is not yet ready to be implemented. The middle-term goal, here, is to provide a full coherent chain of tools, from specification to model-checking, integrated in the Eclipse Vercors environment.

We have continued our work on model-checking of systems containing finite state machines (FSM) connected by unbounded Fifo queues. We have extended our prototype tool for reachability in this kind of systems, with the goal of demonstrating how such algorithms could be controlled in practice [30].

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 ζ -calculus, and its semantics were published by Abadi and Cardelli [46]. In collaboration with Florian Kammüller (Technische Universität Berlin) we modelled the ζ -calculus and a distributed functional calculus, based on ASP – ASP_{fun} – in the Isabelle/HOL theorem prover¹.

This year, we have published previous results on ASP_{fun}. Results include type safety, absence of deadlocks, and absence of cycle of dependencies between processes or futures. We think these results could be used in the design of service-oriented (distributed) architectures communicating with requests-futures.

This year we focused on the representation of variables by locally nameless techniques allowing more elegant proofs and simpler theorems, at the price of a slightly more complicated theory. ζ -calculus formalisation has been transformed into a locally nameless representation and should be submitted to the “archives of formal proofs”; we started the transformation of the model for ASP_{fun}.

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

6.1.2. Skeleton-based parallel programming

Participant: L. Henrio.

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 aspect 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.

This year, we focused on the management of exceptions in skeletons, in order to capture helpful exception traces for the skeleton programmer. We took the opportunity of this new development to improve the formalism for skeleton semantics we developed previously. This work is under submission.

This work has been realized in collaboration with Mario Leyton (NIC Labs, Universidad de Chile).

6.2. Component-oriented Grid Programming

6.2.1. Mechanised formalisation of a Distributed Component Model

Participants: L. Henrio, M. Khan, M. Rivera.

¹ formalisation and proofs available at <http://www-sop.inria.fr/oasis/Ludovic.Henrio/misc.html>

Component models provide a good structure for programs, and allow management of programs at runtime. The structure of component applications describes component dependencies, and can be used to reconfigure automatically the component system. Reconfiguration enables adaptation to new requirements. In this context we want to develop a mechanised model of a component model in order to prove properties on the implementation of the component model itself or on management algorithms designed for this model.

In this context, we start from a component model based on the GCM plus asynchronous invocations with futures between components. Our components communicate via asynchronous requests and replies where the requests are en-queued at the target component. The component making the invocation receives a future which represents the result of this invocation. Our model is precise enough to enable the specification of a formal semantics. Futures play a major role for such asynchronous components. ProActive/GCM is a variant of this model.

We provided a reduction semantics for the component model. This reduction semantics has been specified in the Isabelle theorem prover², and will be used to prove properties on the component model and its implementations [16].

The above mentioned work has been further extended and a paper is under submission. This paper presents a model for distributed components communicating asynchronously using futures. Our model supports first class futures: futures can be transmitted to other components as request parameters or return values. As a consequence, future references can be dispersed among components. When the result is available for a future, it needs to be transmitted to all interested components, as determined by the particular future update strategy being used. We present formal semantics of our component model incorporating formalization of one such future update strategy. Our model has been mechanically formalized in Isabelle/HOL, together with the proof of properties. This approach validates the actual implementation of the future update strategy itself.

6.2.2. 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 [49] (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 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 [56]. We published a more detailed implementation of this example [19], by establishing a mapping between GCM components and the architectural elements from Autonomic Computing. We show the benefits of using a component-oriented architectural approach for designing self-adaptive systems in general. We highlight the architecture of the membranes of GCM components, as a

²available at <http://www-sop.inria.fr/oasis/Ludovic.Henrio/misc.html>

solution to design and implement feedback loops, which are the heart of the dynamic self-adaptive behavior. The fact that the managers can be reconfigured gives the possibility to dynamically update self-adaptive strategies.

In 2009, efforts have been made to improve the ADL (Architecture Description Language) used to describe the component architecture, by introducing componentized membranes at the level of the architecture description. The idea of the improvement, apart from introducing components inside the membrane, is to reach a better separation between the membrane and the functional content, in order to allow specialists to design and develop separately the respective parts of a GCM component. Although we focus on separation of aspects, the communication between those two parts is crucial, because the membrane has to supervise the components inside the functional content. To allow this communication, we defined explicit interfaces, through which the membrane and the content can exchange messages. The detailed improvement of the model is explained in [39]. We also plan to use this new ADL to describe and implement new examples of autonomic systems with GCM components.

6.2.3. *Reconfiguring Distributed Components*

Participants: L. Henrio, M. Rivera.

We have finalised the definition of a framework to support distributed reconfiguration of component systems. This work extended 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. This work is presented in [17].

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.

6.2.4. *SPMD and Components*

Participants: F. Baude, 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.

This work is done in collaboration with applied mathematicians (namely the CAIMAN/NACHOS 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 by defining a complex collective interface called gather-multicast, which is, in fact, a concatenation of a gather and a multicast interface [57]. These improvements came 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.

The usage of GCM components 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 whenever a direct link exists between involved resources. Otherwise, the standard $M \times N$ communication is still available. These concepts and results over the Grid5000 platform were published in [21] and have shown the usefulness and the importance of introduced interfaces and their optimisations.

Recently, we have also included in the implementation of the hierarchical SPMD the support of SSH message tunnelling and forwarding to make possible the integration of resources behind firewalls and NAT addressing. Such an approach has made possible the integration of resources in multiple domains even without direct network connectivity, without much of overhead, be these domains parts of computing grids or computing clouds. The implementation and practical results have been presented in a paper that has been submitted to CCGrid'2010, and contributing material for chapter submitted to a book entitled *Cloud Computing: Principles, Systems and Applications* to be published by Springer in 2010.

The implementation of the hierarchical SPMD has proved to be general enough to be used in different contexts where applications running in different domains need to transparently communicate. For this reason, this solution is currently being adapted to the SOA4All project specificities, aiming to provide a communication layer capable of federating Distributed Enterprise Service Buses at internet scale. Section 6.4.4 and [18] describe this work in more details.

6.2.5. 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 triggered by an executing component-based 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.

Our current implementation takes profit of the componentised membrane of GCM components, to introduce a Monitoring Component which stores the relevant information for doing performance monitoring, in a distributed, scalable and low intrusive way [23]. We have performed tests over a small proof-of-concept application, while we are in the way to extend to a more real scenario.

The monitoring scheme must be general enough to be used for any GCM application, whatever its hierarchical architecture, and to allow for scalability considering that no single component must need to keep all the information of the system (see [23]); 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. This feature will be also promoted to GCM components applications seen as SCA applications. Indeed, membranes of GCM components will host controllers to give these components an SCA personality. Among the controllers in the membrane, we will be able to plug those dedicated to monitoring, thus, getting for free, an innovative way to monitor SCA based SOA applications.

6.2.6. 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 the Galaxy INRIA 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 explored different ways to model a service composition with GCM components and identified the best way to use GCM at the different steps of the workflow life-cycle: deployment, dynamic and distributed workflow execution, and monitoring of the execution.

We achieved a first proof-of-concept [43] enacting the dynamic and possibly distributed execution of a BPEL process. As BPEL processes can be enacted in any BPEL engine, we chose to evaluate the Active BPEL engine, however we plan to evaluate another engines like e.g. Apache ODE, Orchestra, or the BPEL engine of the Xservices platform.

Up to now, our prototype is able to intercept calls between the engine and the involved services in the process. This gives us a finer and better control over the bindings and allows us to change them at will (like e.g. when an error occurs, a machine goes down or a network failure or software issue in the impacted service). We thus progressed in the design and implementation of a reconfiguration tool for SCA-based applications. Our reconfiguration tool manages components that are described using any kind of workflow language (BPEL, XPDL or any other). Our tool is in charge of the deployment of the needed workflow engines, and also, it effectively manages and modifies at will during the workflow execution, the SCA bindings to the involved services. A publication has been submitted to a journal mid-september (IJARAS). Demonstration of the proof-of-concept was done at Java One in June 2009.

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.

The JDC language aims at the specification of distributed components, in the context of the Grid Component Model (GCM). The long-term goal is to generate code that will be guaranteed to satisfy the (behavioural) properties proved on the JDC code. The language is Java-based, and provides means to describe both architectural and behavioural aspects of GCM applications.

The PhD thesis of Antonio Cansado was defended at the very end of 2008 [52]. The JDC language defined in this work is now being reformulated, and submitted as a journal paper. JDC has a graphical subset that is the basis for the graphical editors of the Vercors platform (5.2).

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. The details of this work appear in [32].

We are currently working on extending our implementation to support mixed-strategies. We want to be able to specify an active object or component (and later on at future) level which strategy should be used for a active object or component. Another interested and non-trivial 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, F. Bongiovanni, L. Pellegrino.

Researches on P2P networks have focused not only on the network architecture but also on the semantics 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 ([33], [34], [18]). 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. We are also investigating the possibility to add the Publish/Subscribe (Pub/Sub) paradigm on top of the semantic space. The semantic space is created on top of multiple Structured Overlay Networks (SONs) (CAN and Chord), which differ in topology, routing schemes and maintenance. This Pub/Sub layer would have to be generic enough to work on any SONs. That is, most of the existing SONs share a DHT abstraction layer (get/put/remove), and we would like to take advantage of this commonality between SONs in order to build a fault-tolerant pub/sub layer abstraction on top of the DHT abstraction.

6.4.3. Grid Computing for Computational Finance

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

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. Instead, in this work we focus 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 [50], [58], [24]. 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 [14]. 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 previous years, we also investigated the parallelisation of the Classification-Monte Carlo algorithm of Picazo (CMC) for pricing very high dimensional BA options and performed experiments on the Grid'5000 multi-sites test-bed. The results were published in the Workshop on High Performance Computational Finance at the Supercomputing Conference [54]. 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, [29]. Based on the Master/Slave API of the ProActive library, we designed and implemented an API specially for parallel handling of Monte Carlo simulations and a financial benchmark suite for the Plugtest participants.

In 2009, continuing from the works of the fifth Grid Plugtest for finance, we studied the use of the benchmark suite as a tool for comparing indirectly the grid middlewares performance. Our publication was accepted for presentation, and an extended version of the paper is now under the final proceedings review procedure [22]. Furthermore we investigated on the numerical validation of the simulated results of the benchmark suite (i.e. the problem of the dimension reduction for basket option pricing). The full detail results were reported in the INRIA technical report [29]. For the problem of the use of different classification algorithms of the Picazo algorithm (CMC), we performed the CMC algorithm with several test-cases in order to figure out the trade-off of accuracy and computational time for each classification algorithm.

6.4.4. *Federating DSBs at Internet Scale Upon a Component-Based Approach*

Participants: F. Baude, V. Legrand, E. Mathias, C. Ruz.

The EU-funded NESSI³ *Service Oriented Architecture for All* (SOA4All) Project aims at realizing 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 that integrates complementary and evolutionary technical advances (i.e., SOA, context management, Web principles, Web 2.0 and semantic technologies) into a coherent and domain-independent service delivery platform. In other terms, one expected conceptual outcome of the project is a paradigm often acknowledged as *Service Web* or *Service Cloud*. A Service Cloud must be able to span the whole Internet to allow end-users to use and coordinate external services, potentially executed anywhere on the globe.

In practice, the idea is to be capable to host all needed technical services on a domain-independent service delivery platform. This complex platform may imply on the definition of a hybrid underlying infrastructure. To reach the Internet scale, we require to connect in a seamless manner resources gained from clusters, Grids, Clouds, assuming they are made available to the SOA4All platform for further transparent use by all web-connected end-users devices [18]. Such a requirement is however compliant with the current trend by which scale-out, outsourcing, Software as a Service has become popular terms reflecting a shift on the way enterprises support and organize their IT services.

The underlying solution we push forward to integrate resources in a seamless manner is based on GCM components and has already been successfully employed to couple independent MPI applications running in different domains (Section 6.2.4). The general architecture and concepts of the ESB federation will be published in [18].

This component-based communication layer will be responsible for the routing of messages among services distributed in a multidomain environment. Current work consists in adapting the existing solution as an independent communication layer for the PETS Enterprise Service Bus. We are also investigating a multi-level registration and lookup strategy integrated to the routing component infrastructure to solve the issue of service localization. The semantic space will be also connected to the federated ESB [34], and monitoring information going through the ESB will be gathered using specific GCM components; it will then be delivered to the external SLA and analysis tool [38].

³Networked European Software and Services Initiative

6.4.5. Standardisation of the Grid Component Model

Participants: D. Caromel, L. Henrio, E. Madelaine, 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): adopted, March 2009
- GCM Application Programming Interface: stable draft, June 2009 (final version planned january 2010)

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) and service oriented computing.

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. 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.

Modelica Interface Feasibility Study

The feasibility study focused on the options available for interfacing ProActive with the Modelica language (and some Modelica implementations). Modelica is a language describing simulations in the same way Simulink(Matlab) or Scicos(Scilab) do. It defines a standard for such simulation modelization that many different softwares (open-source or commercial) implement. The study chose the Open Modelica Compiler (an open-source implementation of Modelica) as a reference implementation. It analyses how the compilation and the execution of a modelica model can be automated using the ProActive Scheduler. The study realizes as well the transcription of a Simulink model given by Thales to Modelica. The study concludes first that even though adaptation of models from Simulink to Modelica are feasible, some specific Simulink blocks don't have equivalents in Modelica, and therefore, adaptation techniques need to be implemented. Secondly, the study warns about the numerical stability issues faced when using the Open Modelica Compiler. In regard of these issues, it recommends taking into consideration other commercial implementations, to achieve a better stability.

Installation of a ProActive Matlab toolkit

The ProActive Matlab toolkit developed during last year contract was installed at Thales offices in Chatou. It allows the run Simulink simulation in parallel on a couple remote machines. A week was spent working with the engineers, to understand their everyday computation problems. The ProActive Scheduler and ProActive Matlab toolkit were installed. The ProActive Agent (a daemon program) was installed too.

It allows using machines as computational engines when the engineer usually working on the machine is not there (i.e. at night, during the weekend, etc...). A formation about the usage of ProActive Scheduler was done to Thales IT department. A formation about the usage of ProActive Matlab toolkit and the ProActive Agent was done to the engineers.

This contract started in November 2008 for 14 months, and has a budget of 120 kEuros. The partners are Thales Avionics, Digiteo, and ActiveEon.

7.4. 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 technological drivers of this ADT. The different usage scenarios as well as the different tools developed at infrastructure, application, and business levels demonstrate that this platform is able to support the design, modelling, deployment, and execution of business processes. In the same time, the ADT targets 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 axes: 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 [48], [27], [37].

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 code applications, including MPI. Several publications showed 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, ending in June 2009.

8.1.3. ANR OMD2

The aim of this project is to define, develop and experiment a collaborative platform of multidisciplinary optimization - As "platform" we consider here a software environment hosting heterogeneous code and data, geographically distributed in equally heterogeneous machines. These codes can be sequential or parallel. These machines can be data servers, supercomputers, PC farms, etc. - As "collaborative" we consider this environment to be able to host, control and allow communication of these codes transparently for the user, according to their own work habits. - As "multidisciplinary optimization", we consider the collection of

methods and numerical tools, objects of the other tasks in OMD2 and previously realized during project OMD. It has been decided that ProActive will be used as middleware for communication between machines, and that the ProActive Scheduler will be used as a basis for the collaborative platform. Similarly, Scilab will be used as a common language to describe optimization strategies.

The project started January 1st 2009 and will last 3 years, with a total budget of 214 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. 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. Project will terminate in March 2010.

8.2.3. *Strep GridCOMP*

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

GridCOMP main goal is the design and implementation of a component-based framework suitable to support the development of efficient grid applications. The framework 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. It has been prolonged until Feb 2009.

8.3. International Collaborations

8.3.1. *Stic-Amsud ReSeCo*

ReSeCo (*Reliability and Security of Distributed Software Components*) is a collaboration of INRIA with partners of the south American CONESUD, namely Un. of Cordoba (Argentina), Un. of Montevideo (Uruguay), Un. Diego Portales and Un. de Chile (Chile). 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. The final workshop of ReSeCo will be held in December 2009 in Cordoba (Argentina).

8.3.2. *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 December 2010.

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

- **Eric Madelaine**
Program committee chair and steering committee member of FACS'09, FMCO'09,
Program committee member of conferences Euromicro'09 (SEAA'09), FESCA'09, PDMC'09, and
reviewer for the journals SCP (Science of Computer Programming) and IET-Software.
- **Françoise Baude**
Organizing committee member of *Semantic Extensions to Middleware: Enabling Large Scale
Knowledge Applications* workshop at the *On The Move to Meaningful Internet Systems and Ubiqui-
tous Computing Federated Conferences and Workshops*, Nov. 2009
Program committee member of ECOOP (European Conference on Object-Oriented Programming)
2009, *Workshop for Component-Based High-Performance Computing (CBHPC2009)* with Super-
computing 2009.
Member of the *Best STIC PhD Jury 2009* of ASTI, the *Association Française des Sciences et
Technologies de l'Information et de la Communication*.
Reviewer for *Concurrency and Computation: Practice and Experience*, Wiley.
Member of the laboratory council of the CNRS-I3S laboratory, from 2008. Member of the steering
committee of the *pôle Système* of the *Groupe de Recherche CNRS Architectures, Systèmes, Réseaux*,
from mid 2009.
- **Ludovic Henrio**
Program Committee member of FESCA'09, FOCLASA'09, FESCA'10. Internal Reviewer for the D
3.2.7 BIONETS deliverable.
- **Fabrice Huet**
Vice Chair of the "Parallel and distributed programming" topic at EuroPar2009.
Program committee member of HPDC09, CCGrid09, PCGrid09 PDMC09 and HiPerGrid09.

9.2. Short Visits and external collaborators

- Florian Kammüller, Technische Universität Berlin, 2 weeks visit. Florian works with us in the context of several common developments and joint works in the domain of mechanised formalisation of programming languages and framework.

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 Distributed Algorithmics (Master 2 IFI, with the involvement of Ludovic Henrio, Francesco Bongiovanni); Concurrent Programming (Licence 3), and numerous exercise labs involvement (in XML and Web technologies, multithreaded programming, ...)
- **Denis Caromel**
Course convenor of Distributed Programming and Multi-Tiers Architectures (Master 1) with the involvement of Brian Amedro.
Compute and data grids: large scale distributed systems, Master 2 Ubinet, Master of Science in Ubiquitous Networking and Computing.
- **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)
- **Eric Madelaine** Course Convenor of Semantics of Distributed and Embedded Systems (Master 1)
- **Ludovic Henrio** Course Convenor of Semantics of Distributed and Embedded Systems (Master 1), and involvement in a few Master 2 courses.
- **Paul Naoumenko**
Operating Systems and shell scripting (EPU MAM3, together with Hélène Renard),
Practical session for Distributed Systems (Master 1, with Fabrice Huet)
- **Virginie Legrand Contes**
Course and practical sessions of Service Oriented Architecture (Master 2 MIAGE NTDP)

9.3.2. Theses

The following theses are in preparation:

- Brian Amedro: "Optimizations for Efficient and High-Level Active Object Programming on Multi-Cores and Multi-Processes" (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), directors Françoise Baude and Ludovic Henrio.
- Viet Dung Doan: "Adequation of grid computing to computation intensive calculations in the financial domain" (Since Oct 2006), directors 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), directors 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.
- Francesco Bongiovanni: “Self organizing overlay networks and generic overlay computing systems” (Since Oct 2008, joined OASIS in Sept 2009), directors Françoise Baude and Fabrice Huet.

9.3.3. Internships

- Marc Adrover: “A Datatype Library for the pNets model”
- Axel Baudet: “Hierarchical Resource management and Scheduling”
- Adel Bouchakhchoukha: “Translation from pNets model to Fiacre language”
- Arnaud Gastinel : “Heterogeneous grid experimentations of DiscoGRID applications”
- Ibrahim Hayeck: “Automatic Generation of RCP Eclipse for IC2D, Scheduler and Vercors”
- Milosz Kmiecik: “Data Space for Grid and SOA in ProActive: Accessing and Managing Remote Files and Data”
- Antony Parisy: “Graphical Environment for the Specification of Distributed Components”
- Laurent Pellegrino: “Structured P2P Networks with ProActive”
- Mathieu Schnoor: “Hierarchical Resource management and Scheduling”
- Marc Valdener: “Optimization and Benchmarks of Distributed/Multi-Core programming within ProActive”
- Marek Zawirski: “Data Space for Grid and SOA in ProActive: Accessing and Managing Remote Files and Data”

9.3.4. Member of thesis Committees

- **Françoise Baude**
 - Reviewer and member of PhD thesis committee of Jean-Denis Lesage (26/11/2009), *Couplage dans les applications interactives de grande taille*. Directors Bruno Raffin and Denis Trystram, MOAIS EPI.
 - Reviewer and member of Habilitation à diriger des recherches committee of Fabienne Boyer (04/12/2009), *Gestion de l'adaptabilité dans les applications réparties*. SARDES EPI.
- **Denis Caromel**
 - External reviewer and member of PhD thesis committee of Guillermo López Taboada, *Design of Efficient Java Communications of High Performance Computing*, may 2009. Directors Juan Touriño Domínguez and Ramón Doallo Biempica.

9.4. Participation in conferences and workshops

9.4.1. Invited Talks

- **Denis Caromel**
 - “Challenges and Opportunities on Parallel & Distributed Programming for large-scale: from Multi-core to Clouds”, keynote at CCGrid 2009, May 18-21, 2009, Shanghai, China
 - “ProActive Parallel Suite: A Strong Programming Model Bridging Distributed and Multi-Core Computing”, keynote at 17th Euromicro. Feb. 19 2009, Weimar, Germany
 - “Architecture + Model for Multi-Core and Distribution”, Invited Keynote at CAL 2009, Mar. 24, Nancy, France
 - “Effective SOA + GRIDs with ProActive Parallel Suite”, at OW2 Annual Conference & Solution Linux Open Source, Mar 31 - Apr 2 Paris, France
 - “ProActive Parallel Suite: Multi-Cores to Clouds to Autonomicity”, Invited Keynote at IEEE ICCP, Cluj, Roumania, August 28, 2009
 - “Accelerate on Windows Desktops, HPCS and Clouds with ProActive Parallel”, SC09, November 17, Portland, Oregon, USA, Invited Talk at Microsoft Luncheon
 - “Bridging Multi-Core and Distributed Computing: all the way up to the Cloud”, invited Keynote at ICPADS, Dec. 8-11, Hong Kong-Shenzhen, China
- **Eric Madelaine**
Invited talk at Université de Nantes: “Specification and Verification of Distributed Component-based Applications”

9.4.2. Talks in Workshops and Conferences

- **Ludovic Henrio**
FMCO’09 Symposium, Nov 2009: “Formalism and platform for autonomous distributed components”
Reseco workshop, Dec 2009: “Formalism and platform for autonomous distributed component”.
- **Eric Madelaine**
ReSeCo workshop’09: “Verifying distributed systems with unbounded channels”
- **Viet Dung Doan**
4th Int. Conference on High Performance Scientific Computing, Modeling, Simulation and Optimization of Complex Processes: “SuperQuant Financial-Benchmark Suite for Performance Analysis of Grid Middlewares”
- **Marcela Rivera**
Sinter’09: “A reconfiguration framework for distributed components”
- **Elton Mathias**
EuroPAR’09: “Grid-enabling SPMD Applications through Hierarchical Partitioning and a Component-Based Runtime”
- **Muhammad Khan**
Doctoral Symposium, Ecoop 2009: “Update Strategies for first class future”
- **Paul Naoumenko**
ICAS’09 : “Structural reconfiguration: an autonomic strategy for GCM components”
- **Cristian Ruz**
SCCC’09 : “Enabling SLA monitoring for component-based SOA applications”

9.4.3. Tutorials

- Fabrice Huet. Tutorial at the ASCI Course A14: Advanced Grid Programming Models: “The ProActive grid programming environment”, May 2009, Amsterdam.

9.4.4. Miscellaneous

- **Virginie Legrand**
Journées CNRS du GdR Génie de la Programmation et du Logiciel, January 28, “Exécution distribuée et agile de compositions de services”
- **Paul Naoumenko**
SAC-FIRE workshop organized by EU FET projects on Situated and Autonomic Communications, 3rd March 2009. “Distributed pong talk and demo: dynamic and autonomic reconfiguration of composite services.”

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] L. BADUEL, F. BAUDE, D. CAROMEL, A. CONTES, F. HUET, M. MOREL, R. QUILICI. *Programming, Composing, Deploying for the Grid (chapter 9)*, in "Grid Computing: Software Environments and Tools", Springer, 2006, ISBN: 1-85233-998-5.
- [3] T. BARROS, R. BOULIFA, E. MADELAINE. *Parameterized Models for Distributed Java Objects*, in "Forte'04 conference, Madrid", LNCS 3235, Springer Verlag, 2004, <http://hal.inria.fr/inria-00087222>.
- [4] F. BAUDE, D. CAROMEL, F. HUET, L. MESTRE, J. VAYSSIÈRE. *Interactive and Descriptor-based Deployment of Object-Oriented Grid Applications*, in "11th IEEE International Symposium on High Performance Distributed Computing", 2002, p. 93–102.
- [5] 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

Articles in International Peer-Reviewed Journal

- [11] T. BARROS, R. BOULIFA, A. CANSADO, L. HENRIO, E. MADELAINE. *Behavioural Models for Distributed Fractal Components*, in "Annals of Telecommunications", vol. 64, n^o 1–2, jan 2009, <http://hal.inria.fr/inria-00268965>, also Research Report INRIA RR-6491CL.
- [12] 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", vol. 64, n^o 1, 2009, p. 5–24, http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1007/s12243-008-0068-8&sa_campaign=Email/ACE/PaginatedUKIT.
- [13] A. CANSADO, L. HENRIO, E. MADELAINE. *A Specification Language for Distributed Components*, in "Science of Computer Programming", 2010, To appear.
- [14] V. D. DOAN, M. BOSSY, F. BAUDE, I. STOKES-REES. *Grid computing of Monte Carlo based American option pricing: Analysis of two methods*, in "Mathematics and Computers in Simulation", 2010, To appear.

Invited Conferences

- [15] A. CANSADO, E. MADELAINE. *Specification and Verification for Grid Component-based Applications: from Models to Tools*, in "FMCO 2008, Berlin Heidelberg", F. S. DE BOER, M. M. BONSANGUE, E. MADELAINE (editors), LNCS, n^o 5751, Springer-Verlag, 2009, p. 180-203.
- [16] L. HENRIO, F. KAMMÜLLER, M. RIVERA. *An Asynchronous Distributed Component Model and Its Semantics*, in "FMCO 2008", F. S. DE BOER, M. M. BONSANGUE, E. MADELAINE (editors), Lecture Notes in Computer Science, vol. 5751, Springer, dec 2009, p. 159-179, <http://www.springer.com/978-3-642-04166-2> DE .

International Peer-Reviewed Conference/Proceedings

- [17] B. BANNOUR, L. HENRIO, M. RIVERA. *A Reconfiguration Framework for Distributed Components*, in "SINTER Workshop on Software Integration and Evolution @ Runtime", ACM, 2009.
- [18] F. BAUDE, I. FILALI, F. HUET, V. LEGRAND, E. MATHIAS, P. MERLE, C. RUZ, R. KRUMMENACHER, E. SIMPERL, C. HAMERLING, J.-P. LORRÉ. *ESB Federation for Large-Scale SOA*, in "25th ACM Int. Symposium on Applied Computing", March 2010, To appear AT .
- [19] F. BAUDE, L. HENRIO, P. NAOUMENKO. *Structural reconfiguration: an autonomic strategy for GCM components*, in "5th International Conference on Autonomic and Autonomous Systems(ICAS 2009)", IEEE Xplore, 2009, p. 123–128.
- [20] L. HENRIO, F. KAMMÜLLER. *Functional Active Objects: Typing and Formalisation*, in "Foundations of Coordination Languages and Software Architectures (FOCLASA'09)", 2009 DE .
- [21] E. MATHIAS, V. CAVÉ, S. LANTERI, F. BAUDE. *Grid-enabling SPMD Applications through Hierarchical Partitioning and a Component-Based Runtime.*, in "15th Int. European Conference on Parallel and Distributed Computing (Euro-Par 2009)", LNCS, vol. 5704, 2009, p. 691–703.

Workshops without Proceedings

- [22] A. GAIKWAD, V. D. DOAN, M. BOSSY, F. ABERGEL, F. BAUDE. *SuperQuant Financial-Benchmark Suite for Performance Analysis of Grid Middlewares*, in "4th Int. Conference on High Performance Scientific Computing, Modeling, Simulation and Optimization of Complex Processes, March 2-6 2009, Hanoi, Vietnam", Institute of Mathematics, Vietnamese Academy of Science and Technology, March 2009, 70, Abstract.
- [23] C. RUZ, F. BAUDE. "Enabling SLA monitoring for component-based SOA applications", in ""XXVIII International Conference of the Chilean Computer Society (SCCC'09)"" , IEEE, Nov. 2009, Accepted for presentation.

Scientific Books (or Scientific Book chapters)

- [24] I. STOKES-REES, F. BAUDE, V. D. DOAN, M. BOSSY. *Managing parallel and distributed Monte Carlo simulations for computational finance in a grid environment*, Springer, 2009, p. 183-204US.

Books or Proceedings Editing

- [25] F. S. DE BOER, M. M. BONSANGUE, E. MADELAINE (editors). *Formal Methods for Components and Objects, 7th International Symposium, FMCO 2008, Sophia Antipolis, France, October 21-23, 2008, Revised Lectures*, Lecture Notes in Computer Science, vol. 5751, Springer, dec 2009, 299, <http://www.springer.com/978-3-642-04166-2> NL .

Research Reports

- [26] B. BANNOUR, L. HENRIO, M. RIVERA. *A Reconfiguration Framework for Distributed Components*, n^o RR-6911, INRIA, 2009, <http://hal.inria.fr/inria-00379268/en/>, Research Report.
- [27] F. BAUDE, E. MATHIAS, C. PÉREZ. *Resource selection algorithm and tools for deploying DiscoGrid applications on grids and clouds*, ANR DiscoGrid project, Deliverable from the WP4: Application Deployment, Jun 2009, Technical report.
- [28] A. BOUCHAKHCHOUKHA. *Translation from pNets model to Fiacre language, for the verification of parallel, distributed and concurrent applications* , Master 2 MDFI, Univ. Aix-Marseille II, 2009, Technical report.
- [29] V. D. DOAN, A. GAIKWAD, M. BOSSY, F. BAUDE, F. ABERGEL. *A financial engineering benchmark for performance analysis of grid middlewares* , n^o RT-365, INRIA, 2009, <http://hal.inria.fr/inria-00387324/fr/>, Technical Report.
- [30] R. GASCON, E. MADELAINE. *Verifying distributed systems with unbounded channels*, INRIA, 2009, <http://www-sop.inria.fr/oasis/SAFA/abstracts09/safa09-GM.pdf>, Technical Report.
- [31] C. HAMERLING, V. LEGRAND, F. BAUDE, E. MATHIAS, C. RUZ, M. FRIED, R. KRUMMENACHER, P. MERLE, N. DOLET. *SOA4All Runtime*, n^o D1.4.1B, SOA4All IP Project Deliverable from the SOA4All Runtime WP (WP1), August 2009, Technical report AT .
- [32] M. U. KHAN, L. HENRIO. *First Class Futures: a Study of Update Strategies*, INRIA a CCSD electronic archive server based on P.A.O.L [<http://hal.inria.fr/oai/oai.php>] (France), 2009, <http://hal.inria.fr/inria-00435573/en/>, RR-7113, Technical report.

- [33] R. KRUMMENACHER, I. FILALI, F. HUET, F. BAUDE. *Distributed Semantic Spaces: A Scalable Approach To Coordination*, n^o D1.3.2A, SOA4All IP Project Deliverable from the SOA4All Runtime WP (WP1), February 2009, Technical report AT .
- [34] R. KRUMMENACHER, M. FRIED, F. HUET, I. FILALI, L. PELLEGRINO, C. HAMERLING. *Distributed Semantic Spaces: A first Implementation*, n^o D1.3.2B, SOA4All IP Project Deliverable from the SOA4All Runtime WP (WP1), August 2009, Technical report AT .
- [35] R. KRUMMENACHER, C. HAMERLING, J.-P. LORRÉ, V. LEGRAND, C. RUZ, F. BAUDE, C. PEDRINACI, M. RICHARDSON, T.-P. LOBO. *SOA4All reference architecture specification*, n^o D.1.4.1A, SOA4All IP Project Deliverable from the SOA4All Runtime WP (WP1), February 2009, Technical report AT UK ES .
- [36] R. KRUMMENACHER, I. TOMA, C. HAMERLING, J.-P. LORRÉ, F. BAUDE, V. LEGRAND, P. MERLE, C. RUZ, C. PEDRINACI, D. LIU, T.-P. LOBO. *SOA4All Reference Architecture Specification*, n^o D1.4.1A, SOA4All IP Project Deliverable from the SOA4All Runtime WP (WP1), February 2009, Technical report AT ES UK .
- [37] E. MATHIAS, A. GASTINEL, C. PÉREZ, S. CHAUMETTE, J. OUOBA, F. BAUDE. *Resource selection algorithm and tools for deploying DiscoGrid applications on grids and clouds*, ANR DiscoGrid project, Deliverable from the WP6: Validation and performance assessment, Sept 2009, Technical report.
- [38] A. MOS, C. PEDRINACI, J. M. GOMEZ-PEREZ, G. A. REY, C. HAMERLING, F. BAUDE, C. RUZ. *Service Monitoring and Management Tool Suite Design*, n^o D2.3.1, SOA4All IP Project Deliverable from the SOA4All Studio WP (WP2), February 2009, Resubmission, Technical report UK ES .
- [39] P. NAOUMENKO. *An extension for ADL definition of GCM components, taking into account the full specification of the membrane*, n^o RR-7001, INRIA, 2009, <http://hal.inria.fr/inria-00407788/en/>, Technical Report.
- [40] H. PFEFFER, L. BASSBOUSS, P. NAOUMENKO, D. MIORANDI, D. LOWE, M. ION, L. JANNE, L. HENRIO, F. BAUDE. *Bio inspired service creation and evolution*, n^o D.3.2.6, BIONETS IP Project Deliverable from the Autonomic Service Life-Cycle And Service Ecosystems WP (3.2), Sept 2009, Technical report DE FI IT .

Patents and standards

- [41] *ETSI TS 102 829: GRID;Grid Component Model; Part 3: GCM Fractal Architecture Description Language (ADL)*, European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2009.

Other Publications

- [42] F. BAUDE, L. HENRIO, P. NAOUMENKO. *Distributed pong demo: dynamic and autonomic reconfiguration of composite services*, March 2009, SAC-FIRE workshop organized by EU FET projects on Situated and Autonomic Communications.
- [43] F. BAUDE, V. LEGRAND. *Exécution distribuée et agile de compositions de services*, Janvier 2009, Journées du GDR CNRS Génie de la Programmation et du Logiciel, Groupe de Travail COSMAL (Composants Objets Services : Modèles, Architectures et Langages).
- [44] M. U. KHAN. *Update strategies for first class futures*, July 2009, 19th Doctoral Symposium, Ecoop 2009, Genova, Italy.

- [45] OASIS TEAM. *Grid Component Model - GCM*, April 2009, OW2 Annual Conference, Poster session Fractal Ecosystem.

References in notes

- [46] M. ABADI, L. CARDELLI. *A Theory of Objects*, Springer-Verlag, New York, 1996.
- [47] L. BADUEL, F. BAUDE, D. CAROMEL, A. CONTES, F. HUET, M. MOREL, R. QUILICI. *Programming, Composing, Deploying for the Grid (chapter 9)*, in "Grid Computing: Software Environments and Tools", O. F. CUNHA (editor), Springer, 2006, ISBN: 1-85233-998-5.
- [48] F. BAUDE, D. CAROMEL, N. MAILLARD, E. MATHIAS. *Hierarchical MPI-like Programming using GCM Components as Implementation Support*, Dec. 2006, Selected and presented at the CoreGRID workshop: Grid Systems, Tools and Environments, Grids@work II conference.
- [49] 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)", ACM Digital Library, Oct 2007, Invited Paper.
- [50] 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.
- [51] 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.
- [52] A. CANSADO. *Formal Specification and Verification of Distributed Component Systems*, Université de Nice - Sophia Antipolis – UFR Sciences, December 2008, Ph. D. Thesis.
- [53] 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>.
- [54] 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.
- [55] 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", Springer, 2006, <http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0003.pdf>, ISBN: 978-0-387-47656-8. Also as the CoreGrid TR003 report.
- [56] D. LINNER, H. PFEFFER, S. STEGLICH, F. BAUDE, L. HENRIO, P. NAOUMENKO. *Service Probes Implementation and Evaluation*, n^o D.3.4.1, BIONETS IP Project Deliverable from the Service Probes WP (3.4), June 2008, http://www.bionets.eu/docs/BIONETS_D3_4_1.pdf, Technical report.

- [57] 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.

- [58] 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.