INRIA International program - Associate Team Proposal 2012

Name of the project: **SCADA**: Safe Composition of Autonomic Distributed Applications

A. Detailed project description

A1. Description of the research project - 2 pages maximum

- *Describe the scientific objectives of the project and the method used.*

Besides a formal collaboration between NIC Labs and OASIS team, the aim of the project is to contribute to programming models and languages for programming, running and debugging parallel and distributed applications. For this we will contribute both from a theoretical and practical perspectives to the design of languages, and their implementation and formalisation. In this project we will focus on composition models allowing to put together individual sequential code into complex applications featuring parallelism and distribution. More precisely we focus on two such composition models:

Algorithmic skeletons: Algorithmic skeletons are a high-level programming model for parallel and distributed computing, introduced by Cole. Skeletons take advantage of recurrent parallel programming patterns to hide the complexity of parallel and distributed applications. Starting from a basic set of patterns (skeletons), more complex patterns can be built by nesting the basic ones. Enforcing a clean decoupling of the behaviour (business code) and the parallelism management is the original feature offered by Algorithmic Skeletons. In recent years, the participants of this proposal have designed a type system for algorithmic skeletons and overcome some of the limitations of the skeleton programming paradigm such as exception handling.

Software components: Software components were designed to compose in a much reusable way objects or pieces of application by making their interconnections explicit when building applications. They provide a very convenient granularity for dealing with distribution as a component can be deployed as an independent entity. OASIS promoted in the recent years a software component model, the Grid Component Model, that features dynamic reconfiguration, parallelism and distribution. The contributions of the partners on component models are numerous; the participants of this proposal contributed to the formalisation, design, and implementation of GCM and its runtime support, featuring distribution, asynchronous communications, large-scale deployment, and support for adaptation and autonomic behaviour. Also the participants have a strong background in behavioural specification and verification of component applications in order to guarantee their safe behaviour.

In the same way as algorithmic skeleton decouple business logic from parallelisation concerns, the business logic should be well-separated from non-functional concerns, like monitoring, adaptation, security, etc... in order to facilitate the design of correct large applications. This way, the application can be designed only from its functional point of view, and then non-functional features can be attached at deployment time or at runtime. Of course this requires that adequate hooks are placed in the functional code to allow the interaction between functional and non-functional aspects.

In this project we aim at using complementary competencies from NIC Labs and OASIS in order to extend the algorithmic skeleton library Skandium (developed at NIC Labs). The first research objective consists in extending Algorithmic Skeletons with event handling support to further improve on the separation of functional and non-functional concerns in the patterns, allowing, for example, the monitoring and debugging of the applications.

Then, building on the previous objective, we aim to demonstrate how the use of GCM can provide autonomic features, and structured management of the non-functional aspects to the skeleton library. For this, events and monitoring information will be used as hooks to interconnect the GCM autonomic management features with algorithmic skeletons.

On a more formal basis, extending a programming language should not be done without a strong formal background specifying the language, formalising it, and proving its properties. We also aim at providing such a formal background, and even further formalise the language in a theorem prover. This will provide guarantees on the design of our language and on its properties.

To focus on guarantees on the program behaviour itself, we intend to place ourselves in the context of distributed components and contribute to a behavioural verification platform for the safe execution of distributed components. This objective is partially unconnected from the previous ones, except that, as stated above, non-functional features will be provided by components managing the skeletons, and that the behavioural specification and verification will particularly focus on non-functional features and component management. We intend to reinforce the long collaboration and shared knowledge on behavioural specification of distributed component systems, and on the verification of the correct behaviour of those systems by model-checking techniques. We will build on code generators and model generators in order to build safe-by-construction distributed components by using a semi-formal specification language that has been defined under the thesis of Antonio Cansado from NIC Labs under the supervision of Eric Madelaine from OASIS.

As a longer term research, we also plan on benefiting from this associate team proposal to start investigating on the interaction between theorem proving techniques and model-checking of applications: we have always used generic properties of the component model in our specification and verification tools, and the objective here is to mechanise and generalise the combined use of generic properties and application-specific verification techniques. Another long term goal consists in applying specification and verification techniques to algorithmic skeletons; then the mechanised formalisation we will realise in the first years of this associate team will be particularly useful.

- *Describe how, what and to what extent you intend to advance this research.*

Our objective is to benefit from the long history of collaborations between OASIS and NIC Labs and from the complementary knowledge of the two teams to achieve the research goals described above and further detailed in section A5. All of this will provide a research background which we expect to mature into, or by itself, a CIRIC project within a 5 years span.

To support this collaboration, in addition to individual exchanges and visits between INRIA and NICLabs we plan on organising two workshops at the end of the first and of the second year. The first one should be organised in Chile, and the second in France. The long history of collaboration between the participants and the creation of the CIRIC project next year should guarantee the success of this associate team, and allow us to design languages and tools that will be transferable, in a next stage, to the CIRIC project.

A2. Expected project outcomes and scientific value - 2 pages maximum

- *Project's characteristics, advancement beyond the state-of-the-art.*

This project will advance Algorithmic Skeleton beyond the state-of-the-art by allowing the implementation of non-functional concerns without lowering the programming model's high-level abstraction. The features we intend to provide for algorithmic skeletons are: event management, autonomic management, progress estimation and monitoring, and debugging of the application. For example, by applying state-of-the-art omniscient debuggers which allow backward navigation in the execution history of debugged programs.

Theorem proving techniques have emerged and gained importance in the past years because they are getting easier to apply and to maintain, and there is also a strong need, among the multiple theoretical results, for guaranteed proofs. Unfortunately, nowadays computer science and mathematics theories are getting too complex to be easily checked by a human, as typically the number of cases to consider when reasoning on a large distributed system is huge. Even if it can be feasible to consider all those cases in the proofs, an external reader would not have the time to do so. Theorem provers increase the confidence an external reader has in the formalisation and the proofs one would do. When a language/theory is mechanically formalised, checking a proof is restricted to: (i) check the assumptions; (ii) check the property proved is the one claimed by the author; and (iii) let the theorem prover automatically check the intermediate steps. Mechanical proofs will, thus, increase the confidence in the theory we provide, but also increase the quality of the produced theory by reducing the number of mistakes in the proofs and theorems. We propose here to contribute to the mechanical formalisation of algorithmic skeletons, but also to use theorem proving techniques to prove generic properties on component systems and their adaptation, in order to make behavioural verification possible on larger, reconfigurable component systems.

State-of-the-art techniques on verification of distributed components are usually limited to message-passing techniques which abstract-away dataflow. However, in real-world distributed components dataflow has a strong impact in synchronisations, and therefore it must be included for a realistic behaviour analysis. Within this project we expect to extend the state-of-the-art by using parameterized models which allow dataflow to be encoded, and therefore analysed. Finally, we expect to push generation of code skeletons to specifications that include dataflow, in order to generate code that is safe-by-construction.

Autonomic management of systems has gained a strong interest in the past years. Indeed, distributed and mobile systems are getting bigger and more complex. Additionally, applications must now be able to adapt to an evolving running environment, or to runtime evolution of the service that has to be offered. A first answer to this problem was proposed by component-oriented programming: if programs are decomposed into well-defined, well-decoupled runtime entities, then modifying the composition of those entities at runtime is possible, and achieves adaptation of the application. But the complexity of real applications is such that it is impossible for a human to manage a running system manually, and even to create adaptation procedures that take into account all the possible evolutions a system can perform, and all the possible runtime changes the application might have to face. For this, applications have to provide autonomic evolution capabilities, in such a way that adaptation procedures are written as simple rules, and the enactment of the adaptation is mainly handled by the middleware itself. In the past we designed a component structure to support autonomic adaptation in component systems. Our objective in this project is to adapt this work to provide autonomic behaviour to algorithmic skeletons.

- *General orientation of the project (basic research - applied research with or without the participation of industry).*

The goal of this project is to continue both theoretical and applied research with a vision of industrial applications in Chile and France through CIRIC, aiming for the production of tools which will be of interest for the software industry in parallel programming (algorithmic skeletons offering some quality of service guarantees), advanced state-of-the-art debugging tools (with omniscient debugging), and formal verification of software properties and constraints (Vercors platform). Furthermore, OASIS has experience in producing both theoretical and applied research, demonstrated both by the publication of theoretical papers, and by the creation of a spin-off company named ActiveEon in 2008.

- *Project's contribution to fostering young researchers.*

This project is composed mainly of young researchers who have recently finished their studies at INRIA Sophia-Antipolis or University of Chile DCC. On NIC Labs side this project will include: Antonio Cansado (PhD. 2008), Mario Leyton (PhD. 2008), Guilliaume Pothier (PhD. 2011). Additionally on the Chilean side we plan to continue building on this relationship by exchanging current Master and PhD students in this project such as: Gustavo Pabon, Pablo Valenzuela and Mario Cornejo, all Master and other students. On the OASIS side, Cristian Ruz (PhD. 2011), one PhD student, and a master student will be particularly involved in the project. This project would allow those young researchers to develop their international collaborations and to enrich their research fields. The coordinator of the project, Ludovic Henrio, has been a CNRS researcher since 2005, though not strictly a young researcher, he is not yet a well-established senior researcher. This project would also allow him to develop its international visibility.

A3. Presentation of each organization and lead researcher credentials - 2 pages maximum

- *Brief introduction of the two partner institutions / teams*

DCC – NIC Labs Description: NIC Labs is a technology transfer and research laboratory in the Department of Computer Science (DCC) of the University of Chile. NIC Labs is composed of roughly 20 people having approx 10 permanent staff, 5 of which have PhD in Informatics. NIC Labs focuses mostly on transfer and as such has produced several products on the area of Internet Networks and Parallel/Distributed Computing. Some of the active projects at NIC Labs include "IPv6 para Chile" project lead by NIC Labs which consists of a testbed for the adoption of IPv6 in Chile with 6 Industrial Partners representing 90% of the Chilean Internet Service Providers; Adkintun, an Internet QoS monitoring project in collaboration with the Chilean Government; and Skandium, a Java Library for parallel programming with Algorithmic Skeletons.

The main participants from NIC Labs are: **Mario Leyton**, PhD thesis at INRIA Sophia-Antipolis in OASIS Team during 2005-2008, **Antonio Cansado**, PhD thesis at INRIA Sophia Antipolis in OASIS Team during 2005-2008, **Guillaume Pothier**, PhD thesis at DCC U.Chile during 2006-2011, and **Gustavo Pabon** Master student at DCC U.Chile.

OASIS Description: The project-team OASIS from the INRIA Sophia Antipolis Méditerranée center, will participate in the associated team. The OASIS project-team is a joint project between INRIA, CNRS (UMR I3S) and University of Nice Sophia-Antipolis, which builds the basic principles, techniques and tools for construction, analysis, validation, verification and maintenance of reliable and efficient distributed systems. The main areas of application are distributed applications, grid computing and Clouds, and large-scale services. OASIS' research covers the search for basic calculi and models for distributed computing, the development of programming methodologies and middleware, and specification and verification techniques and software platforms, with a strong interdependence between these three aspects.

The main participants in the OASIS team to this associate team are: **Ludovic Henrio** (CR CNRS), **Eric Madelaine** (CR INRIA), **Françoise Baude** (Prof UNSA), **Cristian Ruz** (Engineer INRIA), **Nuno Gaspar** (PhD student INRIA), **Alexandra Savu** (Apprentice INRIA).

- *Background of the Principal Investigators on both sides*

Mario Leyton is currently a Researcher at NIC Labs. He has authored more than a dozen international publications on scientific journals and conferences, winning best paper awards twice. He performed his PhD studies at INRIA Sophia Antipolis, France, with a scholarship co-financed by INRIA and CONICYT Chile; and received a Computer Science Engineer degree, with maximum distinction, from University of Chile in 2005. Among others, he developed the first spanish speaking ccTLD adoption of Internationalized Domain Names (IDN). He was also the main creator and developer of the ProActive Algorithmic Skeleton library: Calcium, and its successor Skandium (`http://skandium.niclabs.cl`). His main research interest is in parallel/distributed computing, in particular, structured parallel programming models.

Ludovic Henrio is a research scientist at CNRS in the OASIS team. His research activities mainly focus on formal models for programming and composing correct distributed systems. His research topics include distributed programming, object-oriented and component oriented programming, theorem proving, and languages for programming parallel and distributed systems. He has published over fifty international publications in journals, conferences, or book chapters. He formalised the active object model in the form of the ASP calculus during his PhD thesis, and also provided a formal specification for the GCM component model. Those results provided a solid basis for the implementation of the ProActive library and the GCM component model, and the behavioural verification of GCM applications.

- *Main publications of the participants related to the broad topic of the proposal (mark clearly the joint publications)*

OASIS/NICLABS: Related Joint Publications

- M. Leyton, L. Henrio, J. M. Piquer: *Exceptions for Algorithmic Skeletons.* Euro-Par (2) 2010: 14-25
- A. Cansado, L. Henrio, E. Madelaine: *Transparent First-class Futures and Distributed Components.* Electr. Notes Theor. Comput. Sci. 260: 155-171 (2010)
- A. Cansado, L. Henrio, E. Madelaine, P. Valenzuela: *Unifying Architectural and Behavioural Specifications of Distributed Components.* Electr. Notes Theor. Comput. Sci. 260: 25-45 (2010)

- T. Barros, R. Ameur-Boulifa, A. Cansado, L. Henrio, E. Madelaine: *Behavioural models for distributed Fractal components.* Annales des Télécommunications 64(1-2): 25-43 (2009)
- A. Cansado, E. Madelaine: *Specification and Verification for Grid Component-Based Applications: From Models to Tools.* FMCO 2008: 180-203
- D. Caromel, L. Henrio, M. Leyton: *Type Safe Algorithmic Skeletons.* PDP 2008: 45-53

NICLabs Related Publications

- H. Gonzalez-Velez, M. Leyton: *A survey of algorithmic skeleton frameworks: high-level structured parallel programming enablers.* Softw., Pract. Exper. 40(12): 1135- 1160 2010.
- M. Leyton, J. M. Piquer: *Skandium: Multi-core Programming with Algorithmic Skeletons.* PDP 2010: 289-296
- M. Leyton, J. M. Piquer: *A Skandium Based Parallelization of DNSSEC.* SCCC 2009: 87-94
- M. Leyton: *Algorithmic Skeleton* – Wikipedia 2009 [1]

OASIS Related Publications

- F. Baude, D. Caromel, C. Dalmasso, M. Danelutto, V. Getov, L. Henrio and C. Pérez: *GCM: A Grid Extension to Fractal for Autonomous Distributed Components.* Annals of Telecommunications - Special Issue on Software Components – The Fractal Initiative, Springer, 2008
- L. Henrio, F. Kammüller, and M. Uzair Khan: *A Framework for Reasoning on Component Composition* – FMCO 2009, Springer
- 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) pp 123–128, IEEE Computer Society
- C. Ruz, F. Baude, B. Sauvan: *Flexible adaptation loop for component-based SOA applications.* In 7th International Conference on Autonomic and Autonomous Systems(ICAS 2011). Note: Best paper award. May 2011. IEEE Computer Society

A4. History of the collaboration between the teams - 1 page maximum

The OASIS team and the DCC laboratory have a long history of bilateral collaboration. This started with the INRIA/CONICYT Proximos project (2002-2004) between the Polate team at U. Chile and the OASIS team at INRIA. The research subjects included meta-object protocols, concurrent and distributed programming methods, and distributed systems verification. It gave results in terms of one PhD thesis in co-tutelle (Javier Bustos), and a number of common publications [OOPSLA'03, GPCE'02, ECOOP'04, SCCC'04]. Then we launched the associated team Oscar (`http://www-sop.inria.fr/oasis/oscar/` – 2004-2006), including two teams from INRIA participated: OASIS in Sophia-Antipolis, and Obasco in Nantes, building a natural follow-up of Proximos on the same research keywords. This Associate Team, together with the CONICYT student exchange programs, allowed us to set-up internships for 5 master students and 4 PhD thesis, plus 1 co-tutelle PhD thesis (Eric Tanter). Today Eric Tanter is professor at U. Chile, and 3 of the 4 chilean students that got their PhD in Sophia-Antipolis are back in Santiago, with a research/teaching position at U. Chile. We held 3 workshops, and published 4 journal papers and 10 international conference papers co-authored by the partners.

After the end of Oscar, we went to a different collaborative project, the Stic-Amsud project ReSeCo, (OASIS and Everest teams at INRIA, U. Chile and U. Diego Portales in Chile, Uni-

[1] `http://en.wikipedia.org/wiki/Algorithmic_skeleton`

versities of Cordoba in Argentina, and Montevideo in Uruguay); the subject was safety and security of distributed software components.

- *Existing activities correlated with the main objective of the project,*

In parallel with the last collaborations, we continued a fruitful student exchange, and two other chilean students got a CONICYT funding for doing their PhD at INRIA Sophia-Antipolis (2008-2011). However, it seems that recent changes in the CONICYT procedure make now these exchanges much more difficult. In recent years OASIS has also collaborated with NIC Labs in the further advancement of Algorithmic Skeletons and their formalisms. We have worked on extending Algorithmic Skeletons to support high-level exceptions and provided formal semantics for this extension as well as a practical implementation in the Skandium Library.

- *Necessity and significance of this collaboration.*

Currently, INRIA is in the process of installing a research and transfer center in Chile with the collaboration of the Chilean Government and U. Chile (NIC Labs) among others. The CIRIC project aims to foster research and technological transfer into the industry and has a projected span of 10 years with a 3 year periodic review. Our previous collaboration and expertise between OASIS and NIC Labs provides a natural ground in which CIRIC can be installed and developed by building on the close relationship between OASIS (INRIA) and NIC Labs (U. of Chile). At the same time both OASIS and NIC Labs will be able to improve our previously successful relationship both in terms of scientific research and knowledge exchange in the form of Master and PhD students. Furthermore, the close relationship between OASIS and NIC Labs and the previous scientific results is ideal to hopefully produce research and transfer results within CIRIC's 3 year periodic review span.

While NIC Labs members have a strong expertise in algorithmic skeletons, code generation, networks, tools and platforms for parallel and distributed programming, OASIS members have a deep knowledge of verification and formalisation techniques, autonomic computing, model-checking, and theorem provers. NIC Labs research has an applied orientation with a transfer axis; this practical approach will be greatly improved when combined with OASIS's theoretical knowledge and tools on distributed systems, formalisms, and software verification.

- *Added value of this project for each partner.*

As stated above, the two teams are complementary, and thus this project will profit of this complementarity to bring new research directions. The research axis investigated by the associate team will also play the role of background research for the next years of the CIRIC project. Overall, the project will provide both research opportunities on each side, and opportunity for industrial transfer through CIRIC.

Additionally, the possibility of sending students from Chile to INRIA Sophia-Antipolis provides further development and opportunities for the advancement of Informatics and Computer Science in Chile by fostering new researchers with skills and fields of expertise currently scarce or completely inexistent in Chile.

A5. Scientific work program

Duration of the project : 3 years

- *Describe the scientific tasks planned for the duration of the project.*

Task 1: Non-functional concerns in Skeleton Programming with Events

This task aims to tackle the inversion of control problem present in algorithmic skeleton programming which yields a batch-like computation. The goal of this work is to provide events hooks customized for each skeleton pattern in the library to weave non-functional concerns into the skeleton program while maintaining the high-level abstractions and promoting separation of concerns. Those hooks will be the interaction point between algorithmic skeletons and the autonomic managers that can be attached to them: autonomic manager will take decisions to achieve a required quality of service, or any non-functional requirements, and will interact with the functional skeletons through events.

We want to formalise the skeleton language with event capabilities and prove basic properties of this model. We will investigate basic properties on the composition between the treatment of several events, and on the behaviour of the skeleton+event programming model. We will start by investigating properties on the equivalence between different forms of composition between skeletons and programs, and possibly provide a specification for well-formed event+skeletons programs.

This work would be a good opportunity to formalise the skeleton language in a theorem prover, in the same spirit as the current work of Frédéric Loulergue et al. but focusing on higher level representation, and higher level properties, like typing, overall behaviour, etc... The first step here would be the formalisation of our previous works: skeleton language, typing, and exception handling.

First year expected results:
i) Extension of patterns to provide event hooks and their implementation in the Skandium Library.
ii) Paper formalisation of the event hook skeleton pattern.

Second and third year expected results:
i) Mechanised formalisation of Skeletons, and on events for skeletons.
ii) Experiment with the new programming methodology (expressiveness and benchmarks).

Task 2: Tools for Programming, Monitoring & Debugging Skeleton Applications

This task aims to provide tools to program, monitor and debug applications, particularly focusing on Algorithmic Skeletons but not only limited to this programming model. In such lines a programming tool should facilitate the safe composition of skeleton patterns; a monitoring tool should provide feedback to the user on the progress of the execution of the skeleton application; and a debugging tool should provide useful information for the understanding of a problematic behaviour in the skeleton application.

For example, a possible monitoring feature is to estimate a skeleton application's progress dynamically, based on some dynamic cost-model.

The goal of this work is to provide a programming tool for programming models such as algorithmic skeletons which also verifies the safeness/correctness of the program (ex: skeleton follow the safe by construction principle), possibly displays monitoring status of the application, and also provides state-of-the-art debugging of the application.

First year expected results:
i) Design programing (safe type composition) and progress monitoring tools for algorithmic skeleton applications.

ii) Implement Eclipse Plugin for the support of a state-of-the-art omniscient debugger (or back-in-time debugger) engine prototype.

Second and third year expected results:
i) Integrate debbugger engine into algorithmic skeleton programming model;
ii) Implement programming and monitoring tool into the Eclipse Plugin tool and further improve omniscient debugger engine.

Task 3: Autonomic Skeletons

Allowing specific interactions between functional and non-functional layers is required and is one of the key goals of the common research workplan. To this goal, we will make explicit which are the interaction points (mainly event-handling). Also, skeletons are entities that can evolve during their lifetime, which means that both the behaviour and the composition pattern be dynamically modifiable. We thus need a programming technology that could allow the management layer to be both easy to develop, and allows dynamic evolution. A natural proposition is indeed to rely upon a software component-based approach. In particular, components should be dynamically composable, and as we target parallel and distributed skeletons, components must also feature distribution, and possibly parallelism so that the components supporting the management code can themselves run efficiently. So far, we succeeded to define the control part of GCM-based applications by using GCM components themselves. This allowed to build control parts of component-based distributed applications capable to steer some aspects of the functional level behaviour from the non-functional level one. The addressed aspects pertained mostly to the structuration of the functional code, like adding or removing dynamically some components. More recently, we demonstrated that the use of components in the control part of component-based service based (SOA/SCA) distributed applications can support the most elaborated management model (the autonomic MAPE control loop promoted by IBM). The way we design the loop is totally flexible and can so evolve in case one wants it to follow the dynamic evolution of the business part.

From this point we want to investigate on applying our methodology for designing autonomic components to the skeleton framework. A first step would be a simple integration, where a skeleton interpreter will be encapsulated in a GCM component that manages it. To achieve this, our adaptation procedures will have to be adapted to the skeleton framework by using events to trigger adaptation actions.

Then, further integration of the two frameworks can be envisioned. First, we could benefit from the distributed nature of GCM components to run a distributed set of skeleton runtimes and administrate them (note that the Skandium library does not feature distribution). Also we can investigate further integration of autonomic aspects inside skeletons, making the skeleton runtime itself autonomic, instead of managed by an external entity.

First year expected results:
Running a simple skeleton runtime wrapped inside a component.

Second and third year results:
i) Implement autonomic component able to encapsulate and fully manage a skeleton runtime.
ii) Investigate further integration between algorithmic skeletons and autonomic distributed components (*e.g.* provide a distributed autonomic runtime for executing algorithmic skeletons).

Task 4: Behavioural Verification and Code Generation for Safe Distributed Components

Another trend of the project relies on guarantees on the system behaviour. There are two approaches for that: (i) checking that the code is correct, for example through static analysis; the problem with this approach is that it is difficult to analyse code because of the expressive power of programming languages. And (ii), generating code that is safe-by-construction; this means that properties that hold within a system specification are hold in the system implementation. We plan to work on (ii) with two different paths: one is endowing the programmer with high-level patterns such as Skeletons in which she/he does not control the synchronisations and the communication among tasks and therefore one can predict the system behaviour; another way is by allowing the programmer to specify how its program will synchronise and communicate, to verify the specification against the properties we seek for, and then to generate code. For this, we will develop a Domain Specific Language (DSL) that allows the system to be specified, and is simple enough to be analysed. This way the programmer is able to specify the control and data flow of the system in a way that it can be verified using (semi)automatic tools. Finally, the last piece of the workflow is to provide the programmer with a code generator for the DSL that will create code usable in standard programming languages, for example Java – this code is safe-by-construction.

Also, we plan on investigating on the interaction of theorem provers and model checking for verifying program behaviour. Mechanically proving theorems comes at a high cost, but allows much generality and confidence in the theorem proved; our idea is to use theorem provers for proving generic properties on the composition model, on a class of applications or on generic adaptation procedures. Those properties can then be used to simplify the model-checking performed in the rest of this task. Applying generic proofs to assist model checking could drastically reduce the state-space to be explored to prove the correct behaviour of an application. This way we expect to be able to prove the correct behaviour of bigger or more dynamic applications.

First year expected results:
i) Investigate how to generate code that can be filled-in with implementation-specific details using as starting point the JDC language defined within Antonio Cansado's thesis.
ii) From previous works, provide a mechanised formalisation of the component model allowing to handle formal proofs.

Second and third year results:
i) Build a code generation prototype.
ii) Investigate on runtime verification and how to integrate the system specification with a debugger.
iii) Build tools for generating behavioural models and code generation, endowed with a GUI for easy development.
iv) Investigation on the impact of mechanically proved properties on model-checking, especially in the context of component reconfiguration.
v) Proofs of the first properties allowing us to verify the behaviour of reconfigurable component systems.

- *Explain how these tasks are distributed among the participating teams.*

Both of the teams will participate to each task. Formalisation and proofs, and autonomic computing is the domain of the OASIS team, whereas algorithmic skeleton, code generation, and debugging are among the research interests of NIC Labs, but such a research plan cannot be realized without the strong collaboration of both teams on all those aspects, as demonstrated by previous results of the participants. Behavioural specification and verification is a domain shared by the two teams.

*Outline how the partners will coordinate their joint activities and the specific part of each organization (*e.g. *workshops, website, videoconferencing, etc.)*

The joint work will be coordinated by the following activities and tools:

- Frequent video/audio conferences (approx. 1 per month)
- Individual visits of students or researcher (see next section), approximately 3 visits of two to four weeks each year
- A shared revision control repository for the remote collaboration on code and documents such as SVN or GIT
- Organisation of a workshop at the end of the first and of the second year.

Additionally to these crucial collaboration points, a website will be created for the external visibility of the project.

A6. Exchanges scheduled for Year 1

Exchanges scheduled from France to the partner country (researchers' name, including students, and expected duration of stays)

Three researchers should be doing long visits to Chile in the first year (Ludovic HENRIO, Eric Madelaine, and Cristian Ruz), for approximately 2-3 weeks each, these visits will be complemented by a workshop held in Chile. We estimate each stay will cost approximately 3000 euros.

Exchanges scheduled from the partner country to France (researchers' name, including students, and expected duration of stays

One student and one Chilean researcher should be visiting INRIA in the first year (Student: Gustavo Pabon, and researcher: Guillaume Pothier). Each of the stay should last between 2 and 4 weeks, and should cost approximately 3000 euros.

Other exchanges and travels

One joint workshop is planned in Chile for the end of first year, with approximately 5 researchers from Sophia Antipolis going to Santiago de Chile. The workshop organisation should cost approximately 5000 euros: this will cover organisation and travels of two French researchers; to reduce travel costs, we will plan long visits of the French researchers in the same period as the joint workshop, bringing 3 more researchers to the workshop.

We plan also to allocate part of our budget to the participation to conferences in order to present joint publications resulting from the work of the associate team. On the first year we expect to present 2 joint papers, which should cost us approximately 2500 euros. In the next years additional budget from NIC Labs or CIRIC could fund part of those expenses.

NIC Labs participation

In the frame of the CIRIC project, the international partner (NIC Labs) plans to provide funding for travels both for conferences and internships for people from Chile to Sophia-Antipolis. However, as CIRIC is in the process of starting and its budgets are still being drafted, the initial contribution from the International partner will be provided by NIC Labs; expected contribution should cover 1-2 travels for year 1 and up to 5000 euros in expenses. The international partner's contribution is expected to increase in years 2 and 3 once CIRIC is fully operational. In other words, since CIRIC will start during the 1st year of associate team, budget will initially come directly from NICLabs.

A7. Additional information

(This section has been left empty on purpose).

B. Project funding

B1. Expected expenditures for Year 1

Note: in the following table we count only 2 workshop participants as the 3 other visits will be scheduled at the same period as the workshop, and the visitors will also take part to the workshop; overall, OASIS team will have 5 participants to the workshop.

Also 2 missions are planned in the first year for presenting joint-papers in conferences.

		Senior researcher	Postdoc	PhD student	Intern Intern	Other Other	Total Total
Visits from	Nb of persons	2	1			2	5
	Event	long stay	long stay			workshop	
Inria to partner	Estimated costs	6000	3000			5000	14000
Visits from	Nb of persons		1		1	2	2
	Event		long stay		long stay	joint-paper	
partner to Inria	Estimated costs		3000		3000	5000	11000

B2. Budget proposal for Year 1

Financial support request to INRIA[a] under the Associate Team program [a] The maximum amount awarded by INRIA is 20000 Eur per year	20000 Eur
Expected financial and/or material contribution from the international partner to the project *Please detail...* The financial support is provided by the CIRIC project, longer stays and a much higher financial contribution is expected for the next years	5000 Eur
Total	25000 Eur