Submission to EUROCONTROL Programme CARE INO III
Towards Fault-Tolerant Cooperative Air Traffic Management
Detailed Description of the project proposal

Coordinator: Jacques Malenfant, LIP6, Paris

Call of September 2006

1 Summary of the project

1.1 Context, background and motivation

1.1.1 Air traffic control challenges

The design of future air traffic management (ATM) systems appears as a grand challenge. Because of its very large scale, complexity, and dynamicity, we believe that the future of air traffic management lies in hybrid distributed cooperative control systems, including human experts (air traffic controllers) and also intelligent computer support through artificial agents. Such artificial agents could implement various roles and functions: e.g., assistants of human experts, training support, local decision support systems, information servers, monitoring and analysis of human activities (local and aggregated), etc. The concepts of multi-agent systems [1], where various agents interact and coordinate to achieve complex goals appears as a very promising conceptual framework for such challenges.

1.1.2 Fault-tolerance

As a distributed application, air traffic management control includes possibility of partial failures, as this is a fundamental characteristic of distributed applications. The fault tolerance research community has developped solutions (algorithms and architectures), notably based on the concept of replication, applied for instance to data bases. But, these techniques are almost always applied explicitely and statically. This means that it is the responsability of the designer of the application to identify explicitly which critical servers should be made robust and also to decide which strategies (active or passive replication) and their configurations (how many replicas, their placement).

Meanwhile, regarding new cooperative applications, which are very dynamic, such as air traffic management (other examples are: crisis management systems, intelligent sensors networks, etc.), it is very difficult, or even impossible, to identify in advance the most critical agents of the application. This is because the roles and relative importances of the agents can greatly vary during the course of computation, interaction and cooperation, the agents being able to change roles, strategies, plans, and new agents may also join or leave the application (open system). For air traffic control, the focus of activity and criticality may depend on many dynamic properties, related to the air traffic domain (e.g., locations of planes, flight plans, congestion in airports, dependences between controllers, etc.), as well as to the computational domain (dependences between assistant agents, relative importances of information servers, roles of agents, etc.).

Our approach is in consequence to give the capacity to the multi-agent system itself to dynamically identify the most critical agents and to decide which fiabilisation strategies to apply to them. This is analog to load balancing but for reliability. We may use some high-level knowledge at the level of agents and agents organizations (e.g., plans, roles, intention of communication) to better guide the adaptive replication and thus increase the robustness of the application.

1.1.3 Towards fault-tolerant cooperative air traffic control

We may identify various approaches of introducing agents in air traffic management systems.

One direction is in studying the impact of new air traffic control procedures. The idea is to run distributed interactive simulations, involving human participants augmented with assistant agents. The objective is to be able to test and refine strategies in an interactive and participatory way, through the use of assistant agents analysing and advising human participants. This work has actually been recently conducted as a collaboration between EUROCONTROL and our team, through a jointly supervised PhD (thesis of Minh Duc Nguyen, defended at Paris 6/LIP6 in February 2005 [2]). See also, e.g., [3].

In this direction, making more robust the assistant agents by replicating them seems as a promising approach to make the scenario and the assistance by assistant agents more robust. The dynamicity control looks useful to adjust resources to the controllers work load (e.g., depending on the planes changing locations).

Another direction, even more ambitious, is in using agents for cooperative decision support [4]. Agents may be used for local decision support at the level of one human controller. They can also help at the coordination of decisions between controllers, through the use of multi-agent coordination protocols. Some agents may also encapsulate information sources, monitoring and analysis tasks, locally and aggregate and coordinate them, etc. The ultimate target would be an hybrid - human and artificial - distributed cooperative system for air traffic control.

Our objective is to select some scenarios with EUROCONTROL in the field of air traffic control to better evaluate and assess our approach.

1.2 Scientific and technical results expected

This project includes several complementary aspects:

- the design of a prototype platform for adaptive replication, with innovative dynamicity characteristics (dynamic application of replication, dynamic change of strategy of replication.. . .). A first initial version (DarX platform) has been developed in our teams ;
- to study replication, coherence management and recovering policies, adapted to the specificities of agents ;
- to study how to combine agent replication techniques with other reliability and adaptation techniques : replanification, tasks reallocation between agents, etc., and in the first place with exception handling and cooperative recovery techniques ;
- to study the central question of the automatic control of the replication strategies (which agent, which strategy, which parameters. . .). Various types of information, system/network (communication, CPU. . .), agent level (nature of the communications, roles, plans, commitments. . .), models (of faults, of replication costs. . .), and explicit domain information (e.g., flight relative priorities, flight routes dependences, etc.), could be used, compared and combined (multi-criteria decision, learning. . .) ;
- to experiment on various scenarios in the field of air traffic management. Our objective is to design several scenarios of cooperative distributed air traffic control with EUROCONTROL, in order to test and validate our approach.

2 Detailed presentation of the project

2.1 General project objective

The omnipresence of handheld and portable computerized devices as well as the explosion of interconnection opportunities in terms of short- and long-range wireless networks currently push the rapid emergence of *ambiant intelligence* and *large scale cooperative computing*, a new paradigm shift in information processing. Thousands of computerized devices will interact as a large-scale distributed system in order to deliver individual users a wide spectrum of services, ranging from convenience applications, such as providing timely information about cultural events, to more critical ones, such as cooperative work and e-commerce, and finally to highly critical ones, such as coordination of rescue teams in emergency interventions or *decision support systems* in areas like *air traffic control*. Indeed, we believe that the future of air traffic management lies in hybrid distributed cooperative control systems, including human experts (air traffic controllers) and also intelligent computer support through artificial agents. Such artificial agents could implement various roles and functions: e.g., assistants of human experts, training support, local decision support systems, information servers, monitoring and analysis of human activities (local and aggregated), etc.

Building such applications needs to melt together distributed articial intelligence capabilities with high reliability in presence of limited resources: latency, space, processing power, etc. Hence, they are now increasingly designed as a set of autonomous and interactive entities, named agents, which interact and coordinate (multi-agents system [1]). Multi-agent systems provide a promising approach yet, in order to stand the stress of such critical applications, they need a breakthrough in the area of reliability and fault-tolerance [5].

The fault-tolerance research community has developed solutions (algorithms and architectures), some more *curative* e.g., based on exception handling and cooperative recovery [6], and some more *preventive*, notably based on the concept of replication, applied e.g. to data bases.

As discussed by [7], software replication in distributed environments has some advantages over other fault-tolerance solutions. First and foremost, it provides the groundwork for the shortest recovery delays. Also, generally it is less intrusive with respect to execution time. Finally, it scales much better. Another important advantage, on the design perspective, is that the use of software replication is relatively generic and transparent to the application domain. The designer does not have to explicitly specify the nature of the possible abnormal behaviors and the way to handle them. As we will see, our approach is furthermore transparent, as the task of deciding what entities to replicate and how to parameterize replication is handled automatically. That said, software replication focuses mostly on processor or network faults, and does not address the whole spectrum of possible faults (design, timing. . .). Thus, a general issue, and still subject of open research is how to combine various approaches for fault-tolerance in a single articulated methodology.

Software replication is generally applied explicitly and statically, at design time. Thus, it is the responsibility of the designer of the application to identify explicitly what critical components should be made robust and also to decide what strategies (e.g., active or passive replication) and their configurations (how many replicas, their placement, etc.).

Meanwhile, as pointed out above, new cooperative applications, such as air traffic management, and also crisis management systems, ambient intelligence, e-commerce, increasingly designed as multi-agent systems (MAS), are much more dynamic. In such applications, the roles and relative importance of the agents can greatly vary during the course of computation, of interaction and of cooperation, because the agents may change roles, plans and strategies. Also, new agents may join or leave the application (as an open system). It is thus very difficult, or even impossible, to identify in advance the most critical software components of the application. Because such applications are typically large scale, they put additionnal stress on the underlying distributed system which large scale makes it unstable by nature, at least in currently deployed technologies. That increases the needs for mechanism for adaptive fiabilization of the application.

Such new challenges reach the limits of traditional static approaches of replication, and motivate the study of adaptive replication mechanisms. One key issue is then the identification of the most critical components (agents) of the application at a certain time. (Here we may informally define the *criticality* of an agent as follows: the criticality of an agent, relative to an organization of agents it belongs to, is the measure of the potential impact of the failure of that individual agent on the failure of the organization.) Therefore, we consider using various levels of information to estimate criticality.

More precisely, examples of informations related to the focus of activity and criticality, could be: domain-dependent (e.g., locations of planes, flight plans, congestion in airports, dependences between controllers, etc.), agent level (e.g., dependences between assistant agents, relative importances of information servers, roles of agents, plans of agents), and basic computational level (e.g., communication load, etc.).

In a more general computation perspective, our project aims at studying, proposing and implementing a novel approach to reliability in distributed cooperative systems adapted to the dynamicity and the large-scale characteristics of multi-agents systems, as well as resource limitations observed in any field-deployed systems. The angle we take is to adopt standard reliability techniques, like replication, and consider the replicas as resources to be dynamically allocated to maximize reliability under the time, processing power, space and bandwith contraints.

Hence, maximizing the reliability while minimizing the cost in resources becomes a problem of continuously identifying the most critical agents and then reallocating replication resources to them. The key point in our approach is to establish a collaboration between the low-level replication service and the application in order to perform the best decision (optimal or near-optimal when possible) by using the knowledge of the application to guide the reallocation decisions.

More precisely, our approach is to give the capacity to the multi-agents level to dynamically identify the most critical agents and to decide which fiabilisation strategies to apply to them, and then to pass along this information to the underlying middleware to make it decide the best fiabilisation technique and to optimize its different parameters. This is analog, yet much more challenging, to load balancing but for fiabilisation. We want to *automatically* and *dynamically* apply fiabilization (mostly through replication mechanisms) *where* (to which agents) and *when* they are most needed.

To guide the adaptive fiabilization, we intend to use various levels of information, system level, like communication load, and application/agent level, like the roles agents are currently playing in the application, the plans they are currently executing and the criticality of tasks they have to do. Contrary to standard distributed systems, roles and plans are typical information available in multi-agents systems. We conjecture that both can deliver crucial inputs to the fault-tolerance layer to get a precise evaluation of the current criticality of agents but also to predict their future criticality given the tasks they will have to perform in the near future.

To achieve its goal, our project brings together researchers from the areas of distributed systems, multi-agents systems, error and exception handling, as well as decision and control. Our expected collaboration with EUROCONTROL brings in applications in air traffic management as primary test fields for our approach. An example is to design and implement an air traffic assistant system where agents working on behalf of each air traffic controller cooperatively monitor the air traffic and suggest to human controllers decisions about traffic control. This type of application has the core characteristics mentioned above: criticality, large-scale and distributed intelligence.

2.2 Context and state of the art

Several approaches address the multi-faced problem of fault tolerance in multi-agents systems. These approaches can be classified in two main categories. A first category focuses especially on the reliability of an agent within a multi-agents system. This approach handles the serious problems of communication, interaction and coordination of agents with the other agents of the system. The second category addresses the difficulties of making reliable mobile agents which are more exposed to security problems [8]. We do not discuss that category here as in most of mobile agent systems and applications, agents are considered independently (without cooperation between them).

Within the family of reactive multi-agents systems, some applications offer high redundancy. A good example is a system based on the metaphor of ant nests. Unfortunately, we cannot design all applications as reactive multi-agents systems, and moreover we cannot apply such simple redundancy scheme onto more cognitive multi-agents systems as this would cause inconsistencies between copies of a single agent.

Work by [9] offers dynamic cloning of specific agents in multi-agents systems. But their motivation is different, the objective being to improve the availability of an agent if it is too congested. They do not consider the recovery of tasks state upon failure. The agents considered implements only stateless sessions, i.e. functional tasks without state, where fault-tolerance could be ensured by simply redoing the aborted tasks..

S. Hagg introduces sentinels to protect the agents from some undesirable states [10]. Sentinels represent the control structure of their multi-agents system. They need to build models of each agent and monitor communications in order to react to faults. Each sentinel is associated by the designer to one functionality of the multi-agents system. This sentinel handles the different agents which interact to achieve the functionality. The analysis of his beliefs on the other agents enables the sentinel to detect a fault when it occurs. Adding sentinels to multi-agents systems seems to be a good approach, however the sentinels themselves represent failure points for the multi-agents system. Moreover, the problem solving agents themselves participate in the fault-tolerance process.

A. Fedoruk and R. Deters [11] propose to use proxies to make transparent the use of agent replication, i.e. enabling the replicas of an agent to collectively act as a unique entity regarding the other agents.

The proxy manages the state of the replicas. All the external and internal communications of the group are redirected to the proxy. However this increases the workload of the proxy which becomes a quasi central entity. To make it reliable, they propose to build a hierarchy of proxies for each group of replicas. They point out the specific problems of read/write consistency, resource locking also discussed in [12]. This approach lacks flexibility and reusability in particular concerning the replication control. The experiments have been done with FIPA-OS which does not provides any replication mechanism. The replication is therefore realized by the designer before run time.

In distributed computing, many toolkits include replication facilities to build reliable application. However, many of products are not enough flexible to implement an adaptive replication. MetaXa [13] implements in Java active and passive replication in a flexible way. Authors extended Java with a reactive metalevel architecture. However, MetaXa relies on a modified Java interpreter. GARF [14] realizes fault-tolerant Smalltalk machines using active replication. Similar to MetaXa, GARF uses a reflexive architecture to provide transparent replication strategies to the application. But, it is up to the programmer to decide where to apply these strategies. More recently, [15] brings reflection a step further than GARF by introducing a meta-model expressed in terms of object method invocations and data containers defined for objects' states.

The work by Kraus et al [16] proposes a solution for deciding allocation of extra resources (replicas) for agents. They proceed by reformulating the problem in two successive operational research problems (knapsack and then bin packing). Their approach and results are very interesting but it is based on too many restrictive hypothesis to be made adaptive.

2.3 Related projects

At Free University of Amsterdam, the Intelligent Interactive Distributed Systems Group (headed by Pr. Frances Brazier), runs a research project related to our general objectives. They designed an agent architecture intended for replication, named AgentScape. They separate the public part of an agent, which is immutable and may be freely replicated, from its private part which should be kept unique. They are studying how they may combine their agent architecture and our current adaptive replication platform, named DarX [17]. Note that they do not address yet the issue of adaptive replication control.

A related and much more general project is the Autonomic Computing Program of IBM. This program intends at the design of adaptive computing systems, being able to self adapt (self configure, self optimize, self repair...). They propose a general blueprint architecture (monitor, analyze, plan, execute). A prototype architecture/framework, named ABLE (Agent Building and Learning Environment), partially implements it and provides a toolbox of components (implemented as JavaBeans) for manipulating and using monitored information (rules, neural networks, statistics...). Autonomic computing has very wide spectrum and is a long term goal. Although fault-tolerance is one of its crucial part, the ABLE architecture by itself does not solve the problem, but the blueprint guidelines are an interesting input.

Chameleon [18] is an adaptive fault tolerance system using reliable mobile agents. The methods and techniques are embodied in a set of specialized agents supported by a fault tolerance manager (FTM) and by host daemons for handshaking with the FTM via the agents. Adaptive fault tolerance refers to the ability to dynamically adapt to the evolving fault tolerance requirements of an application. This is achieved by making the Chameleon infrastructure reconfigurable. Static reconfiguration guarantees that the components can be reused for assembling different fault tolerance strategies. Dynamic reconfiguration allows component functionalities to be extended or modified at runtime by changing component composition, and components to be added to or removed from the system without taking down other active components. Unfortunately, through its centralized FTM, this architecture is not scalable and the FTM represents a bottle-neck as well as a failure point for the system.

AQuA [19] is a middleware built above Ensemble [20]; it offers an adaptive fault tolerance scheme for CORBA-compliant software. AQuA provides its own built-in replication strategies and uses the monitoring features of Ensemble in order to determine and configure the right strategy for each component at runtime. Such decision making is left to AQuA's most prominent element: its Proteus dependability manager. Proteus operates in a way that is more user-independent than the pre-defined plans used in Chameleon: it samples the QoS requirements of every component it manages, and determines the right configuration with respect to information provided by monitoring entities, called *observers*. AQuA suffers from the same flaw as Chameleon: although the Proteus dependability manager is set as a replicated component within AQuA, thus decreasing the risk of its becoming a failure point, it still represents a bottle-neck for entire chunks of the software it supports. Also, AQuA limits its own potential by preventing more than one strategy to be applied in the same replication group.

3 Organization of the project

3.1 Summary

Our proposed project wants to give the capacity to the multi-agents system itself to dynamically identify the most critical agents, through system-level information and application/agent-level information, and to decide which fiabilisation strategies to apply to them. This is analog, but far more challenging, to "load balancing" but for fiabilisation.

The project builds on previous work done between some of the existing partners (multi-agents team and distributed systems team at LIP6 and INRIA). A prototype platform for adaptive replication has been designed, with innovative dynamicity characteristics (dynamic application of replication, dynamic change of strategy of replication...). A first initial version (DarX platform [21]) has already been implemented. This platform has been connected to a multi-agents platform, called DIMA [22], also developed in our team. The project aims at leveraging our initial experience with this platform and develop new work in several complementary aspects:

- to design of a new version of DarX which will integrate a new policy based mechanism for specifying the reliability needs of agents which will then be able to translate the policy into a dynamic choice of fiabilisation technique and of optimal parameters (number and placement of replicas, ...) ;
- to study replication, consistency management and recovering policies, adapted to the specificities of agents ;
- to study how to combine agent replication techniques with other fiabilisation and adaptation techniques: replanification, tasks reallocation between agents, etc., and in the first place with exception handling and cooperative recovery techniques ;
- to study the central question of the automatic control of the replication strategies (which agent, which strategy, which parameters...). Various types of information, system/network (communication, CPU...), agent level (nature of the communications, roles, plans, tasks, commitments, norms...) [23], as well as a cost model for the different operations, must be used and combined (multi-criteria decision, learning...) to dynamically compute optimal of at least achievable fiabilization policies ;
- to experiment with EUROCONTROL on realistic scenarios in the field of air traffic management.

3.2 Added value of the cooperation

Our project, multi-thematic, includes three partners/laboratories: LIP6-Paris, INRIA-Rocquencourt, LIRMM-Montpellier, plus our expected collaboration with EUROCONTROL:

Laboratoire d'Informatique de Paris 6 (LIP6) - This first - and coordinator - partner has three different teams involved in the project. The SMA (Systèmes Multi-Agents) team, within the DESIR (DEcision, Systèmes Intelligents et Recherche opérationnelle) research department, works on multi-agent systems design and implementation and the issue of their self-adaptation. The MoVE (Modélisation et VErification) team, within the RSR (Réseaux et Systèmes Répartis) scientific department, works on modeling and verification of distributed applications. The Décision team, within the DESIR (DEcision, Systèmes Intelligents et Recherche opérationnelle) research department, works on multi-criteria decision and ranking. (Note that LIP6 recently finalized a significant scientific restructuration.)

Institut National de Recherche en Informatique et en Automatique (INRIA)-Rocquencourt - The REGAL (Répartition et Gestion d'Applications à Large Échelle) INRIA-LIP6 project, works on large scale distributed architectures, notably including fault-tolerance issues.

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) - The Exception handling team of the D'OC project, works on exception handling and fault-tolerance for objects, components and agents.

It is important to note that the respective specialities of our respective research teams - multi-agent systems, adaptive agents, modeling of distributed applications, decision, fault-tolerant large scale distributed systems, exception handling, are different but very complementary. This is indeed a true multi-disciplinary project, distributed on three research labs: LIP6, LIRMM, INRIA-Rocquencourt. This project is a rare opportunity to combine researchers from various disciplines and backgrounds in a single project and to address a very important issue for future computer applications: self adaptation of its distribution and fiabilization policies. This issue is getting growing attention because of the increased distribution (large scale and peer to peer) and dynamicity (nomadism, open systems) of future distributed cooperative applications.

Note that the multi-agent systems team at LIP6 and the REGAL project at INRIA-LIP6 have already started a small-scale collaboration on the topic of fault-tolerant multi-agents systems. This project proposal is the opportunity to expand this preliminary work, by adding experts in the field of decision systems (LIP6) and in exception handling (LIRMM), and air traffic management experts at EUROCONTROL in order to evaluate the merits of our approach for air traffic management applications.

3.3 Members of the project

The permanent members (researchers with permanent position) are:

Name	*position*	*team*	*participation (%)*
Samir Aknine	Assistant Professor	SMA/DESIR/LIP6	50
Isabelle Alvarez	Researcher	Décision/DESIR/LIP6	20
Jean-Pierre Briot	Research Director	SMA/DESIR/LIP6	50
Christophe Dony	Professor	D'OC/LIRMM	40
Zahia Guessoum	Assistant Professor (HdR)	SMA/DESIR/LIP6	60
Jacques Malenfant	Professor	MoVE/RSR/LIP6	50
Olivier Marin	Assistant Professor	REGAL/INRIA-LIP6	60
Pierre Sens	Professor	REGAL/INRIA-LIP6	40
Christelle Urtado	Assistant Professor	D'OC/LIRMM	40
Sylvain Vauttier	Assistant Professor	D'OC/LIRMM	40

See mini-CVs of permanent members in the Annex.

3.4 PhD students

There are currently 2 PhD students working directly on the project: Nora Faci, (currently finishing), working on dependence-based decision strategies for replication [24] and Alessandro Luna-Almeida, working on plan-based decision strategies for replication [25].

We plan to have - ideally - up to 4 new PhD students working on:

- exception handling and fault-tolerance (advised by Christophe Dony) ;
- decision and control (advised by Samir Aknine and Isabelle Alvarez) ;
- replication algorithms and support (advised by Olivier Marin and Pierre Sens) ;
- air traffic control scenarios (advised by Jacques Malenfant and a researcher at EUROCONTROL).

4 Technical description of the project

In the project, we will study various approaches for fiabilisation of large scale distributed multi-agents applications. We will have a special focus on replication techniques, although we will also explore how we could further combine agent replication techniques with other fiabilisation and adaptation techniques, such as replanification, tasks reallocation between agents, and exception handling and cooperative recovery.

4.1 Replication for fiabilization

Replication of data and/or computation is indeed an effective way to achieve fault tolerance in distributed systems. This is thus our first target as a mechanism for fiabilization. A replicated software component is defined as a software component that possesses a representation on two or more hosts [14]. There are two main types of replication protocols:

- active replication, in which all replicas process concurrently all input messages,
- passive replication, in which only one of the replicas processes all input messages and periodically transmits its current state to the other replicas in order to maintain consistency.

Each type of replication strategy has its advantages and disadvantages. The active replication provides a fast recovery delay and enables to recover from byzantine failures, but leads to a high overhead. This kind of technique is dedicated to critical applications, as well as other applications with real-time constraints which require short recovery delays. Passive replication has a low overhead under failure free execution, but does not provide short recovery delays. If the active replica is found to be faulty, a new replica is elected among the set of passive ones and the execution is restarted from the last saved state. This technique requires less CPU resources than the active approach but it needs a checkpoint management which remains expensive in processing time and space. (See some of our previous work on optimizing checkpointing [26]).

The choice of the most suitable strategy is directly dependent of the environment context, especially the failure rate, the kind of failure that must be tolerated, and the application requirements in terms of recovery delay and overhead. Active approaches should be chosen either if the failure rate becomes too high or if the application design specifies hard time constraints. In all other cases, passive approaches are preferable. In particular, active approaches must be avoided when the computational elements run on a non-deterministic basis, where a single input can lead to several different outputs, as the consistency between replicas cannot be guaranteed in this type of situation.

4.2 A Prototype framework for supporting adaptive replication

Many toolkits (e.g., [14] and [27]) include replication facilities to build reliable applications. However, most of them are not quite suitable for implementing large-scale, adaptive replication mechanisms. For example, although the strategy can be modified in the course of the computation, no indication is given as to which new strategy ought to be applied; moreover, such a change must have been devised by the application developer before runtime. Besides, as each group structure is left to be designed by the user, the task of conceiving a large-scale software appears tremendously complex.

Therefore we have designed a specific and novel framework for replication, named DarX [28], which allows dynamic replication and dynamic adaptation of the replication policy (e.g., passive to active, changing the number of replicas). Moreover, DarX has been designed to easily integrate various agent architectures, and the mechanisms that ensure dependability are kept as transparent as possible to the application.

DarX is a framework to design reliable distributed agent applications. Each agent can be replicated an unlimited number of times and with different replication strategies (passive and active). Note that we are working on the integration of other replication strategies in DarX, including quorum-based strategies [29]. The number of replicas may be adapted dynamically. Also the replication strategy is reified such as one may dynamically change the replication strategy. DarX includes group membership management to dynamically add or remove replicas. It also provides atomic and ordered multi-cast for the replication groups' internal communication. Messages between agents, that is communications external to the group are also logged by each replica, and sequences of messages can be re-emitted for recovery purposes.

4.3 Error propagation issues

If a replica within a group does not answer, current policy is to consider it as off the group. This simple policy is minimal for keeping consistency but is not very flexible. We are planning to improve it with a more flexible approach that mix replication with alternative solution allowing to somehow control propagation and handling of runtime errors.

Indeed, the management of software failure also includes the issue of error notification and propagation within and across the agents. Because errors do propagate and can cause system to crash in an arbitrary way, this submission (as compared to previous submission last year) now includes a team, whose main participant is a renown expert in exception handling who has worked and collaborates on various aspects of exception handling [30, 31, 32, 32, 33, 34, 6]. Team's recent works concern exception handling for multi-agents systems [35, 36, 37]. With this added expertise it will be interesting to study how signaling algorithms including concertation policies could be applied in the framework of adaptive replication to handle situations in which agent do not answer, or would like to report exceptional answers.

4.4 Control of adaptive replication

The central issue we address is the automatic control of the replication strategies. The system must be able to identify at any time which agents are the most critical (that is, whose failure will have the greater impact on the whole application) and must be protected through replication.

Various types of information:

- system/network-level (communication, CPU…),
- agent-level, for instance : nature of the communications - for instance by using types of communication intention made explicit in FIPA Agent Communication Language standard - [38], roles [39], plans [25], commitments, norms [40]…
- domain-level, for instance : flight relative priorities, flight routes dependences… (to be defined with EUROCONTROL)
- and also models (of faults, of replication costs…)

could be used, compared [23] and combined [41].

The issue of combination is a multi-criteria decision problem. Regarding criticality of agents, one issue is important: the set of critical agents must be relatively robust to small changes of the information taken into account, and also to small changes of the parameters of the models (for instance, of replication costs). When the ressources are limited, the identification of the most critical agents can also be seen as a ranking problem [42].

Note that, as a decision problem, the system must not only identify the most critical agents but also decide which replication strategy (active, passive) and its parameterization (how many replicas, where are they placed/created, what consistency management protocol is used…). We want to investigate the relevance of various kinds of information and abstractions and the way to collect them (monitoring, observation).

4.5 Informations for decision

As stated above, various types of information may be used for decision representing various strategies to estimate relative criticality of agents. For agent-level information, the notion of role of an agent appears interesting because it captures some "importance" of the agent in the organization (e.g., producer, client, mediator, primary participant, default participant…). We already did a preliminary study based on the notion of roles which gave promising results [43, 39].

We also studied the relevance of alternative concepts such as: dependence networks [44, 45]. Intuitively, the more an agent has other agents depending on it, the more it is critical in the organization.

Another strategy uses the plans of an agent, i.e., the actions that the agent has planned to execute in the near future [46, 25]. This strategy has the advantage of estimating future criticality and not just instantaneous one.

Each type of strategy has its pros and cons: static or dynamic, cost, and nature of assumptions on the abstractions available (e.g., messages, roles, plans). Indeed, some type of information like references or messages is basic and general, while some higher-level abstractions (performatives, roles, plans, norms) may or not be supported by a given multi-agent architecture. We also started experiments to compare the relevance and cost of the various strategies [23] and a further issue is their combination seen as a multi-criteria decision problem.

Note that the strategies we explored are mostly bottom-up, as they use or infer information from the program elements or/and from execution, to estimate criticality of agents. We would also like to study a dual direction, top-down, based on first analysis and specifications of general dependability requirements, and then in using that information to guide replication control. Some directions are in using a dependability risk-driven approach or dependability cases [47].

Last, complementary to the collection of decision informations, we also intend to address the dynamic adaptation problem as a full-fledged sequential decision control problem. To achieve this goal, we will develop a complete cost model for all operations used in our fiabilization strategies, and then use this cost model to build an efficient and realistic closed-loop control model. After a careful and comprehensive identification of the decision parameters, as well as the implementation constraints, we will do a complete survey of possible techniques among which we will choose the most relevant to implement our control model. An important point we also want to address using autonomic computing ideas is the design of a good software architecture to integrate control issues into the combination of the multi-agents and the DarX middleware systems. Given our goal to allow the multi-agents system to provide inputs to the fiabilization strategies, such an architecture is crucial to the viability of the whole approach. Our FACOMA project therefore insists on this point and looks forward to provide important contributions along these lines.

4.6 Performance issues

We already conducted preliminary measurements (see details in [21] and in [28]) of the DarX middleware, notably the cost for: dynamic addition of replicas, message sending to a group with active replication, message sending to a group with passive replication, replication strategy switching. The performances are quite encouraging; most notably, [28] shows a case study wherein the overhead added by strategy switching is insignificant with respect to the supported application in terms of resource usage as well as computing time. We also recently conducted preliminary measurements of the cost of the adaptive control, to compare the overhead of various control strategies (based on roles, on dependence networks), with a planned publication. During the project, we plan to develop further these performance evaluation and comparisons. Meanwhile, note that our main goal for the project is robustness and adaptability and not in optimization for performance.

However, the main issue regarding performance is as follows: one of the fundamental reasons for adptivity in DarX is to restrict the overhead inherent to replication. Indeed, since one of the goals in DarX is to identify the subset of critical agents, and since this subset is then ordered in terms of criticity (for each agent: the impact of its loss on the overall application), fault tolerance is restrained to those agents which really require it, thus cutting down on the performance overhead traditional dependability solutions impose on their supported software.

4.7 Validation issues

We already conducted some preliminary experiments with small scenarios, of distributed meeting planification with assistant agents (intelligent PDAs), and also of crisis management systems. Our objective is to work together with EUROCONTROL in the field of air traffic management to better evaluate and assess our approach.

We may identify various approaches of introducing agents in air traffic control systems.

One direction is in studying the impact of new air traffic control procedures. The idea is to run distributed interactive simulations, involving human participants augmented with assistant agents. The objective is to be able to test and refine strategies in an interactive and participatory way, through the use of assistant agents analysing and advising human participants. This work has actually been recently conducted as a collaboration between EUROCONTROL and our team, through a jointly supervised PhD [2, 3].

In this approach, making more robust the assistant agents by replicating them seems as a promising approach to make the scenario and the assistance by assistant agents more robust. The dynamicity control looks useful to adjust resources to the controllers work load (e.g., depending on the planes changing locations).

Another direction, even more ambitious, is in using agents for cooperative decision support [4]. Agents may be used for local decision support at the level of one human controller. They can also help at the coordination of decisions between controllers, through the use of multi-agent coordination protocols. Some agents may also encapsulate information sources, monitoring and analysis tasks, etc. The ultimate target would be an hybrid - human and articifial - distributed cooperative control for air traffic control.

Last, note that the application field of air traffic control is not limitative. Our approach for reliability seems also promising in that kind of applications. Other application fields (e.g., distributed decision support, crisis management systems,ambient intelligence, distributed monitoring, electronic commerce...) could be also possible targeted applications.

5 Planned schedule and deliverables

5.1 Planned schedule

Our general planned schedule is as following:

- during the first year, in cooperation with EUROCONTROL, we will identify air traffic control scenarios. We will also study the issues and possible approaches, notably: 1) how to improve exception handling meschanisms for our DarX middleware ; 2) studying the use of tasks and plans as a pertinent information to guide the control of replication ; 3) studying how to apply ranking techniques to select what agents are to be replicated.
- during the second year, in coordination with EUROCONTROL, we will design and then implement and integrate solutions. Implementation will be Java-based. In complement to DarX, we will use the DIMA prototype multi-agents platform, with has been used for current experiments. We will also study the coupling between DarX and another multi-agents platform, such as JADE.
- during the third year, we will evaluate our approach and software prototypes on EUROCONTROL air traffic control scenarios.

5.2 Deliverables

From the kickoff of the project, the following schedule gives the expected intermediate and final results:

- End of year 1:
  1. Bibliography/state of the art (Teams: all).
  2. Initial specification of applicative scenarios (Teams: MoVE/LIP6,and SMA/LIP6), in coordination with EUROCONTROL.
  3. Stabilized implementation of the DarX platform (Teams: REGAL/INRIA).
  4. Finalization of the plan-based strategy (Teams: SMA/LIP6 and MoVE/LIP6).
  5. Comparative study of the various decision/information strategies and models to be applied at both the DIMA (multi-agents) level and the DarX (fault-tolerance layer) level (Teams: MoVE/LIP6, SMA/LIP6, and Décision/LIP6).
  6. Specification of a policy-based mechanism to communicate between DIMA and DarX about the criticality of agents (Teams: SMA/LIP6, REGAL/INRIA, MoVE/LIP6 and D'OC/LIRMM).
  7. Specification of a two-layer exception handling system between DarX and DIMA (Teams: D'OC/LIRMM, REGAL/INRIA, MoVE/LIP6, and SMA/LIP6).
  8. Specification of a multi-criteria and ranking decision architecture for combining multiple information/strategies and for decision (Décision/LIP6, SMA/LIP6).
- End of year 2:
  1. Refinement of the specifications of applicative scenarios (Teams: MoVE/LIP6 and SMA/LIP6) in coordination with EUROCONTROL.
  2. Specification of test cases to conduct the experimentations (Teams: all).

3. Implementation of the fault-torerance policy manager into DarX, including the integration of the decision about the optimal allocation of replication resources based on low-level information from the system (Teams: REGAL/INRIA, Décision/LIP6, and MoVE/LIP6).
4. Implementation of the fault-tolerance policy generator into DIMA, including the integration of the decision about the best policies using high-level information from the multi-agents application (Teams: SMA/LIP6 and Décision/LIP6).
5. Implementation of the exception handling system (Teams: D'OC/LIRMM, REGAL/INRIA, and MoVE/LIP6).
6. Port of the application to the DIMA/DarX platform (Teams: MoVE/LIP6 and SMA/LIP6).

- End of year 3:

1. Experiments and performance studies (Teams: all).
2. Report on experiments with the applicative scenarios (Teams: MoVE/LIP6 and SMA/LIP6) in coordination with EUROCONTROL.
3. Final implementation and delivery of the DarX/DIMA platform (Teams: all).

5.3 Table of main deliverables

Along the project, we plan to hold general meetings between the participants every six months. We will also deliver reports about the project every 6 months (semiannual reports). Last, we will create a Web site dedicated to the project, as required by the program. This Web site will be regularly updated.

The tentative schedule of deliverables is as following:

Label	*Type*	*Pilot team*	*Participants teams*	*Date*
Web site of the project	Site	LIP6	all	T0+1
Bibliography/state of the art	Report	LIRMM	all	T0+6
Definition of initial applicative scenarios	Report	LIP6	LIP6	T0+6
Semiannual report	Report	LIP6	all	T0+6
Refinement of the tasks and schedule	Report	LIRMM	all	T0+9
Stabilized implementation of the DarX prototype	Software	INRIA	LIP6, INRIA	T0+12
Semiannual report	Report	INRIA	all	T0+12
Comparative evaluation of various information/strategies	Report	LIP6	LIP6	T0+12
Refinement of applicative scenarios	Report	LIP6	LIP6	T0+15
Semiannual report	Report	LIRMM	all	T0+18
Prototype implementation	Software	INRIA	all	T0+24
Semiannual report	Report	INRIA	all	T0+24
Report on experiments	Report	LIP6	all	T0+30
Semiannual report	Report	LIRMM	all	T0+30
Final implementation	Software	INRIA	all	T0+36
Final report	Report	LIP6	all	T0+36

5.4 Expected results

We expect the following results:

- a better knowledge on the issues of fiabilization of cooperative applications, in the field of air traffic management,
- a software platform for dynamic replication, improving our current first prototype,
- a decision architecture and a corresponding prototype architecture, for automatically fiabilizing cooperative multi-agents applications.

- validation through several scenarios in the field of air traffic management (see Validation section).

6 Financial demand

We already have two PhD students working on the project (one finishing). We would like to have up to (ideally) 4 new PhD students, working: on exception handling and fault-tolerance (advised by Christophe Dony) ; on decision and control (advised by Samir Aknine and Isabelle Alvarez) ; on replication algorithms and support (advised by Pierre Sens) ; and on application to air traffic control (advised by Jacques Malenfant and a researcher of EUROCONTROL).

We include in the demand to EUROCONTROL program support for one PhD student. He will be the PhD student working on scenarios and prototype applications at Eurontrol. For the other ones, we expect to find other sources (fellowships). Current PhD students are funded (PhD fellowships from french Ministry of research or from other governments, like Brazil).

Thus our needs are limited to:

- *travel/missions* - Mostly, for presenting our results to conferences. Plus a little travel support for meetings with LIRMM/Montpellier partner.
- *equipment* - We conducted our first experiments on workstations connected through a local network. Through the REGAL INRIA-LIP6 project, we will be able to use a PC-cluster. We need to upgrade work stations for our researchers, plus stations for students (PhD students and masters).
- *PhD support* - One PhD student support.
- *other students support* - We would also like to provide some support to students (Master 2nd year projects) to work on the project.

The proposed 3 years budget is as following:

- travel/missions - to present results in scientific conferences.
  On average, 7 missions (2kEuros) every year (for 10 permanents + students).
  Budget: 18kEuros (LIP6) + 12kEuros (INRIA) + 12kEuros (LIRMM) = 42kEuros (total)
- equipment - work stations and laptops.
  New stations for students and upgrade for permanents. On average, 4 stations (3 kEuros) every year. We will also use this set of stations for first distributed experiments.
  Budget: 12kEuros (LIP6) + 12kEuros (INRIA) + 12kEuros (LIRMM) = 36kEuros (total)
- PhD support.
  PhD salary: 3kEuros (including all taxes) each month.
  Budget: 110kEuros (LIP6) = 110kEuros (total)
- support for masters students.
  Budget: 12kEuros (LIP6) + 12kEuros (INRIA) + 12kEuros (LIRMM) = 36kEuros (total)
- Budget grand total = 42 + 36 + 110 + 36 = 224kEuros + 15% institutional management overhead = 258kEuros

The proposed budget for the first year is one third of the total budget,
that is 258kEuros / 3 = 86kEuros

7 References

See the short bibliography (References) generated at the end of this document.

8 Annex - Mini CVs of permanents members of the project

8.1 Samir Aknine

Degrees:

- PhD in Computer Science, Université Paris Dauphine, Paris, December 2000.

Current position: Assistant Professor (*Maître de conférences*), Université Paris 6, since 2001.

3 selected publications:

- Alessandro Luna-Almeida, Samir Aknine, Jean-Pierre Briot, and Jacques Malenfant. Plan-based replication for fault-tolerant multi-agent systems. In *11th IEEE Workshop on Dependable Parallel, Distributed and Network-Centric Systems (DPDNS'06) 20th International Parallel and Distributed Processing Symposium (IPDPS'2006)*, Rhodes Island, Greece, April 2006.
- Jean-Pierre Briot, Zahia Guessoum, Samir Aknine, Alessandro L. Almeida, Nora Faci, Maíra Gatti, Carlos Lucena, Jacques Malenfant, Olivier Marin, and Pierre Sens. Experience and Prospects for Various Control Strategies for Self-Replicating Multi-Agent Systems, International Workshop on Software Adaptive and Self-Managing Systems (SEAMS'2006), co-located with the International Conference on Software Engineering (ICSE'2006), Shanghai, China, ACM - Sheridan Publications, May, 2006.
- Samir Aknine and Olivier Marin. Role of Replication Planning for Fault Tolerance Multiagent Systems, Fifth symposium on Adaptive Agents and Multi-Agent Systems (AAMAS-5), AISB'05 Convention, Paris, France, March 2005.

8.2 Isabelle Alvarez

Degrees:

- PhD in Computer Science, Université Paris 6, Paris, November 1992.

Current position: Researcher (*IGREF*), CEMAGREF, Paris.

3 selected publications:

- Isabelle Alvarez and S. Bernard. Ranking Cases with Decision Trees: a Geometric Method that Preserves Intelligibility. In *19th International Joint Conference on Artificial Intelligence (IJCAI'2005)*, Edinburgh, August 2005.
- Isabelle Alvarez. Sensitivity Analysis of the Result in Binary Decision Trees. In Proceedings of the 15th European Conference on Machine Learning, LNAI, No 3201, pp. 51–62, Springer-Verlag.
- Isabelle Alvarez. Explication géométrique du résultat dans les arbres de décision. Revue d'Intelligence Artificielle, 18(3) 411–441.

8.3 Jean-Pierre Briot

Degrees:

- PhD in Computer Science, Université Paris 6, Paris, December 1984.
- "Tenureship" (*Habilitation à diriger des recherches*) in Computer Science, Université Paris 6, Paris, July 1989.

Current position: Research Director (*Directeur de Recherche*) CNRS (DR2), since 1999.

Main responsabilities:

- Advising: 6 HdRs and 13 PhDs already defended.
- Program committee member of more than 50 conferences and workshops (including ECOOP, and PC chair of LMO'03 and JFSMA'03).

Publications:

- 4 books edited
- 16 chapters in books
- 11 articles in journals, plus editor of 2 special issues
- more than 70 communications in conferences (including ECOOP, OOPSLA, IJCAI, ECAI) and workshops

3 selected publications:

- Jean-Pierre Briot, Zahia Guessoum, Samir Aknine, Alessandro L. Almeida, Nora Faci, Jacques Malenfant, Olivier Marin, and Pierre Sens. Engineering control strategies for replication-based fault-tolerant multi-agent systems. In *International Workshop on Engineering of Fault-Tolerant Systems (EFTS'2006)*, Luxembourg, Luxembourg, June 2006.
- Zahia Guessoum, Nora Faci, and Jean-Pierre Briot. Adaptive replication of large-scale multi-agent systems - towards a fault-tolerant multi-agent platform. In Alessandro Garcia, Ricardo Choren, Carlos Lucena, Paolo Giorgini, Tom Holvoet, and Alexander Romanovsky, editors, *Software Engineering for Large-Scale Multi-Agent Systems IV*, number 3914 in Lecture Notes in Computer Science, pages 238–253. Springer Verlag, 2006.
- Jean-Pierre Briot and Khaled Ghédira (editors). Déploiement des systèmes multi-agents - Vers un passage à l'échelle - JFSMA'03. *RSTI, Hermes*, novembre 2003. Hors série (*Special issue*).

8.4 Christophe Dony

Degrees:

- PhD in Computer Science, Université Paris 6, Paris, February 1989.
- "Tenureship" (*Habilitation à diriger des recherches*) in Computer Science, Université Montpellier-II, Montpellier, January 1998.

Current Position: Professor, Université Montpellier-II, Montpellier, since 2002.

Main responsibilities:

- Advising : 5 PhDs already defended and 1 to be.
- Program committee member of various conferences and workshops including ECOOP, ESUG, MODELS, JFLA, LMO. Program Chair of LMO'2000.

Publications: Around 60 publications all included, (including ECOOP, OOPSLA, MODELS, two books edited, two french journals edited, two books chapters invited).

3 selected publications:

- Dony, C., Knudsen, J.L., Romanovsky, A.B., Tripathi, A. (editors). Advances Topics in Exception Handling Techniques. Volume 4119 of Lecture Notes in Computer Science., Springer (2006), To appear.
- Shui, A., Mustafiz, S., Kienzle, J., and Dony, C., Exceptional use cases. In: Proceedings of Models/UML, ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems, Montego Bay, Jamaica, (2005), pages 568–583.
- Dony, C., Urtado, C., Vauttier, S., Exception handling and asynchronous active objects : Issues and proposal. in Advances Topics in Exception Handling Techniques, Volume 4119 of Lecture Notes in Computer Science, Springer (2006), Chapter 5, To appear.

8.5 Zahia Guessoum

Degrees:

- PhD in Computer Science, Université Paris 6, Paris, May 1996.
- "Tenureship" (*Habilitation à diriger des recherches*) in Computer Science, Université Paris 6, Paris, December 2003.

Current position: Assistant Professor (*Maître de conférences*), Université de Reims, since 1998.

Main responsibilities:

- Advising: 3 PhDs already defended.
- Program committee member of more than 30 conferences and workshops (including AAMAS, JFSMA, SELMAS).

Publications:

- 4 books and journals edited
- 9 chapters in books
- 8 articles in journals
- more than 60 communications in conferences and workshops

3 selected publications:

- Zahia Guessoum and Nora Faci. Adaptive replication of large-scale multi-agent systems - towards a fault-tolerant multi-agent platform. ICSE 2006 Workshop on Software Engineering for Large-Scale Multi-Agent Systems V (SELMAS'06). Alessandro Garcia, Ricardo Choren, Carlos Lucena, Alexander Romanovsky (eds.). ACM, China, May 2006.
- Zahia Guessoum, Nora Faci, and Jean-Pierre Briot. Adaptive replication of large-scale multi-agent systems - towards a fault-tolerant multi-agent platform. In Alessandro Garcia, Ricardo Choren, Carlos Lucena, Paolo Giorgini, Tom Holvoet, and Alexander Romanovsky, editors, *Software Engineering for Large-Scale Multi-Agent Systems IV*, number 3914 in Lecture Notes in Computer Science, pages 238–253. Springer Verlag, 2006.
- Zahia Guessoum. Adaptive Agents and Multi-Agent Systems. In Distributed Systems Online Journal, IEEE Computer Society, 5(7):1–4, July, 2004.

8.6 Jacques Malenfant

Degrees:

- PhD in Computer Science, Université de Montréal, Montréal, Québec, Canada, March 1990.
- "Tenureship" (*Habilitation à diriger des recherches*) in Computer Science, Université de Nantes, Nantes, April 1997.

Current position: Professor, Université Paris 6, since 2003. (Professor since 1997).

Main responsabilities:

- Program committee member of 6 international conferences (including ECOOP, Reflection and ACM PEPM) as well as 12 national conferences (including LMO, JFPDLA, JFLA and JFPL, and PC chair of LMO'99).

Publications:

- 4 books edited (2 conference proceedings, 2 ECOOP Workshop reader)
- 2 chapters in books
- 5 articles in journals
- 10 papers in international conferences (OOPSLA, ECOOP, Reflection)
- 10 papers in national conferences (LMO, JFPL, JFLA, Journées de l'optimisation)
- 10 communications in international workshops

3 selected publications:

- Jean-Pierre Briot, Zahia Guessoum, Samir Aknine, Alessandro L. Almeida, Nora Faci, Jacques Malenfant, Olivier Marin, and Pierre Sens. Engineering control strategies for replication-based fault-tolerant multi-agent systems. In *International Workshop on Engineering of Fault-Tolerant Systems (EFTS'2006)*, Luxembourg, Luxembourg, June 2006.
- Jacques Malenfant, Marie-Thérèse Segarra et Françoise André. Dynamic Adaptability: the MolèNE Experiment. *Proceedings of the Third International Conference Reflection'2001*, No2192 in LNCS, Springer-Verlag, pp. 110–117, September 2001.
- Pierre Lécuyer et Jacques Malenfant. Computing Optimal Checkpointing Strategies for Rollback and Recovery Systems. *IEEE Transactions on Computers*, 37(4):491–496, April 1988.

8.7 Olivier Marin

Degrees:

- PhD in Computer Science, Université du Havre, France, December 2003.

Current position: Assistant Professor (*Maître de Conférences*), Université Paris 6, Paris, since 2004.

Main responsabilities:

- Co-advising: 4 PhDs, amongst which 2 have already been defended.
- Proceedings coordinator of EuroSys 2006
- Editorial board member of Distributed Systems Online

Publications:

- 14 communications in conferences and workshops

3 selected publications:

- Olivier Marin, Marin Bertier and Pierre Sens. DARX - A Framework for the Fault-Tolerant Support of Agent Software, In Proceedings of the *14th IEEE International Symposium on Software Reliability Engineering (ISSRE'2003)*, Denver, Colorado, USA, November 17-20, 2003.
- Olivier Marin, Marin Bertier, Pierre Sens, Zahia Guessoum, and Jean-Pierre Briot. DARX - a Self-Healing Framework For Agents to appear in the collected notes of the Monterey Workshop 2005, *Networked Systems: realization of reliable systems on top of unreliable networked platforms*, Springer Verlag.
- Jean-Pierre Briot, Zahia Guessoum, Samir Aknine, Alessandro L. Almeida, Nora Faci, Jacques Malenfant, Olivier Marin, and Pierre Sens. Engineering control strategies for replication-based fault- tolerant multi-agent systems. In *International Workshop on Engineering of Fault- Tolerant Systems (EFTS'2006)*, Luxembourg, Luxembourg, June 2006.

8.8 Pierre Sens

Degrees:

- PhD in Computer Science, Université Paris 6, Paris, December 1994.
- "Tenureship" (*Habilitation à diriger des recherches*) in Computer Science, Université Paris 6, Paris, December 2000.

Current position: Professor, University Paris 6, since 2003.

Main responsabilities:

- Co-advising: 8 PhDs (5 already defended)
- Program committee member of 10 conferences : (including NCA, SSS, AC, ISORC, CFSE, CDUR, DRUIDE, ASTI), chair of JCS 99, co-chair of CFSE'2005
- co-director of Laboratory of Computer Science (LIP6)
- Leader of the REGAL INRIA Project (joint research team/project between INRIA and LIP6)

Publications:

- 1 book edited
- 6 chapters in books
- 7 articles in journals (including SPE, JPDC, PPL, Journal Cluster Computing)
- 37 communications in international conferences and workshops (including DSN, ISSRE, ICCP, HOTOS, CCGRID, SPDP), more than 10 communication in french conferences.

3 selected publications:

- M. Bertier, L. Arantes, and P. Sens. Distributed mutual exclusion algorithms for grid applications: a hierarchical approach. *Journal of Parallel and Distributed Computing (JPDC)*, 66:128–144, 2006.
- Olivier Marin, Marin Bertier, and Pierre Sens. DARX - A Framework For The Fault-Tolerant Support Of Agent Software. Proc. of the 14th IEEE International Symposium on Software Reliability Engineering (ISSRE '03), Denver, CO, USA, November 2003.
- M. Bertier, O Marin, and P. Sens. Performance Analysis of Hierarchical Failure Detector. In Proceedings of the International Conference on Dependable Systems and Networks (DSN'03), San Francisco, CA, USA, June 2003 (IEEE Computer Press).

8.9 Christelle Urtado

Degrees:

- *Engineering diploma* in Computer Science, EERIE (Institute for studies and research in Computer Science and Electronics), Nîmes, July 1994.
- PhD in Computer Science, Université Montpellier-II, Montpellier, October 1998.

Current position: Assistant Professor, École des Mines d'Alès, since 1999.

Main responsibilities:

- Advising : 1 PhD already defended and 1 to be, 2 Masters already defended and 2 to be.
- Program committee member of several french-speaking conferences and workshops such as LMO, CAL, JMAC. Organizing Chair of LMO'2006.

Publications: Around 20 publications including journal articles (in both french and english), one edited book, three invited book chapters.

3 selected publications:

- F. Souchon, C. Dony, C. Urtado, and S. Vauttier. Improving exception handling in multi-agent systems. In C. Lucena, A. Garcia, A. Romanovsky, J. Castro, and P. Alencar, editors, *Advances in Software Engineering for Multi-Agent System*, number 2940 in LNCS. Springer-Verlag, 2004.
- Aurélien Campéas, Christophe Dony, Christelle Urtado, and Sylvain Vauttier. Distributed exception handling : Ideas, lessons and issues with recent exception handling systems. In *Proceedings of RISE'04 : International Workshop on Rapid Integration of Software Engineering techniques*, pages 75–85, Luxemburg, November 2004.
- Dony, C., Urtado, C., Vauttier, S., Exception handling and asynchronous active objects : Issues and proposal. in Advances Topics in Exception Handling Techniques, Volume 4119 of Lecture Notes in Computer Science., Springer (2006), Chapter 5, To appear.

8.10 Sylvain Vauttier

Degrees:

- *Engineering diploma* in Computer Science, EERIE (Institute for studies and research in Computer Science and Electronics), Nîmes, July 1994.
- PhD in Computer Science, Université Montpellier-II, Montpellier, December 1999.

Current position: Assistant Professor, École des Mines d'Alès, Nîmes, since 2000.

Main responsibilities:

- Advising : 1 PhD already defended and 1 to be, 2 Masters already defended and 2 to be.
- Program committee member of several french-speaking conferences and workshops such as LMO, OCM-SI, JMAC. Organizing Chair of LMO'2006.

Publications: Around 20 publications including journal articles (in both french and english), an edited book, two invited book chapters.

3 selected publications:

- F. Souchon, C. Dony, C. Urtado, and S. Vauttier. Improving exception handling in multi-agent systems. In C. Lucena, A. Garcia, A. Romanovsky, J. Castro, and P. Alencar, editors, *Advances in Software Engineering for Multi-Agent System*, number 2940 in LNCS. Springer-Verlag, 2004.
- Aurélien Campéas, Christophe Dony, Christelle Urtado, and Sylvain Vauttier. Distributed exception handling : Ideas, lessons and issues with recent exception handling systems. In *Proceedings of RISE'04 : International Workshop on Rapid Integration of Software Engineering techniques*, pages 75–85, Luxemburg, November 2004.
- Dony, C., Urtado, C., Vauttier, S., Exception handling and asynchronous active objects : Issues and proposal. in Advances Topics in Exception Handling Techniques, Volume 4119 of Lecture Notes in Computer Science., Springer (2006), Chapter 5, To appear.

References

[1] Jean-Pierre Briot and Yves Demazeau, editors. *Principes et architecture des systèmes multi-agents.* Collection IC2. Hermes, 2001.

[2] Minh Nguyen-Duc. *Vers la conception participative de simulations sociales : Application à la gestion du trafic aérien.* PhD thesis, University Paris-VI, Paris, France, February 2005. Thèse d'Université.

[3] Minh Nguyen-Duc, Vu Duong, and Alexis Drogoul. Agent-based modeling and experimentation for real-time collaborative decision making in air traffic management. In *Proceedings of the 24th Congress of the International Council of the Aeronautical Sciences (ICAS'04)*, Yokohama, Japan, 2004.

[4] Minh Nguyen-Duc, Vu Duong, Jean-Pierre Briot, and Alexis Drogoul. An application of multi-agent coordination techniques in air traffic management. In *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03)*, pages 622–625, Halifax NS, Canada, October 2003. (short paper).

[5] Jean-Pierre Briot and Khaled Ghédira (editors). Déploiement des systèmes multi-agents - Vers un passage à l'échelle - JFSMA'03. *Revue des Sciences et Technologies de l'Information (RSTI)*, November 2003. Actes des 11èmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA'03). Hors série.

[6] Christophe Dony, Jørgen Lindskov Knudsen, Alexander B. Romanovsky, and Anand Tripathi, editors. *Advances Topics in Exception Handling Techniques*, volume 4119 of *Lecture Notes in Computer Science*. Springer, 2006.

[7] R. Guerraoui and A. Schiper. Software-based replication for fault tolerance. *IEEE Computer*, 30(4):68–74, 1997.

[8] F. De Assis Silva and R. Popescu-Zeletin. An approach for providing mobile agent fault tolerance. In S. N. Maheshwari, editor, *Second International Workshop on Mobile Agents*, number 1477 in LNCS, pages 14–25. Springer Verlag, 1998.

[9] K. Decker, K. Sycara, and M. Williamson. Cloning for intelligent adaptive information agents. In *ATAL'97*, LNAI, pages 63–75. Springer Verlag, 1997.

[10] S. Hagg. A sentinel approach to fault handling in multi-agent systems. In C. Zhang and D. Lukose, editors, *Multi-Agent Systems, Methodologies and Applications*, number 1286 in LNCS, pages 190–195. Springer Verlag, 1997.

[11] A. Fedoruk and R. Deters. Improving fault-tolerance by replicating agents. In *AAMAS2002*, Boulogna, Italy, 2002.

[12] L. Silva, V. Batista, and J. Silva. Fault-tolerant execution of mobile agents. In *International Conference on Dependable Systems and Networks*, pages 135–143, 2000.

[13] M. Golm. Metaxa and the future of reflection. In *OOPSLA -Workshop on Reflective Programming in C++ and Java*, pages 238–256. Springer Verlag, 1998.

[14] R. Guerraoui, B. Garbinato, and K. Mazouni. Lessons from designing and implementing GARF. In *Proceedings Objects Oriented Parallel and Distributed Computatio*, volume LNCS 791, pages 238–256, Nottingham, 1989.

[15] J.-C. Fabre, M.-O. Killijian, J.-C. Ruiz-Garcia, and P. Thevenod-Fosse. The design and validation of reflective fault-tolerant Corba-based systems. *IEEE Distributed Systems Online*, 3(3), March 2002.

[16] Sarit Kraus, V.S. Subrahmanian, and N. Cihan Tacs. Probabilistically survivable MASs. In *IJCAI'03*, 2003.

[17] B.J. Overeinder, F.M.T. Brazier, and O. Marin. Fault-tolerance in scalable agent support systems: Integrating DarX in the AgentScape framework. In *Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid2003)*, 2003.

[18] Z. Kalbarczyk, R. K. Iyer, S. Bagchi, and K. Whisnant. Chameleon: A software infrastructure for adaptive fault tolerance. *IEEE Transactions on Parallel and Distributed Systems*, 10(6):560–579, 1999.

[19] Yansong Ren. *AQuA: A Framework for Providing Adaptive Fault Tolerance to Distributed Applications.* PhD thesis, 2001.

[20] O. Rodeh, K. Birman, and D. Dolev. The architecture protocols in the Ensemble group communication system. *ACM Transactions on Information Systems and Security (TISSEC)*, 2001.

[21] Olivier Marin, Pierre Sens, Jean-Pierre Briot, and Zahia Guessoum. Towards adaptive fault-tolerance for distributed multi-agent systems. In *Proceedings of the 4th European Research Seminar on Advances in Distributed Systems (ERSADS'2001)*, pages 195–201, Bertinoro, Italy, May 2001.

[22] Z. Guessoum and J.-P. Briot. From active objects to autonomous agents. *IEEE Concurrency*, 7(3):68–76, 1999.

[23] Jean-Pierre Briot, Zahia Guessoum, Samir Aknine, Alessandro L. Almeida, Nora Faci, Maíra Gatti, Carlos Lucena, Jacques Malenfant, Olivier Marin, and Pierre Sens. Experience and prospects for various control strategies for self-replicating multi-agent systems. In *International Workshop on Software Adaptive and Self-Managing Systems (SEAMS'2006), co-located with the International Conference on Software Engineering (ICSE'2006)*, Shanghai, China, May 2006. ACM - Sheridan Publications.

[24] Zahia Guessoum, Nora Faci, and Jean-Pierre Briot. Adaptive replication of large-scale multi-agent systems - towards a fault-tolerant multi-agent platform. In Alessandro Garcia, Ricardo Choren, Carlos Lucena, Paolo Giorgini, Tom Holvoet, and Alexander Romanovsky, editors, *Software Engineering for Large-Scale Multi-Agent Systems IV*, number 3914 in LNCS, pages 238–253. Springer-Verlag, 2006. Revision of [38].

[25] Alessandro Luna-Almeida, Samir Aknine, Jean-Pierre Briot, and Jacques Malenfant. Plan-based replication for fault-tolerant multi-agent systems. In *11th IEEE Workshop on Dependable Parallel, Distributed and Network-Centric Systems (DPDNS'06) 20th International Parallel and Distributed Processing Symposium (IPDPS'2006)*, Rhodes Island, Greece, April 2006.

[26] Pierre L'Écuyer and Jacques Malenfant. Computing optimal checkpointing strategies for rollback and recovery systems. *IEEE Transactions on Computers*, 37(4):491–496, April 1988.

[27] R. van Renesse, K. Birman, and S. Maffeis. Horus: A flexible group communication system. *CACM*, 39(4):76–83, 1996.

[28] O. Marin, M. Bertier, and P. Sens. Darx - a framework for the fault-tolerant support of agent software. In *14th IEEE International Symposium on Software Reliability Engineering (ISSRE'03)*, pages 406–417, Denver, Colorado, USA, November 2003.

[29] F. Belkouch, M. Bui, and L. Chen. Self-stabilizing quorum systems for reliable document access in fully distributed information systems. *Studies in Informatics and Control*, 7(4):311–326, 1998.

[30] Christophe Dony. Exception handling and object-oriented programming: Towards a synthesis. In *ACM SIGPLAN Notices - Proceedings of the Joint Conference ECOOP-OOPSLA'90, 25(10):322–330*, October 1990.

[31] Aaton Shui, Sadaf Mustafiz, Jorg Kienzle, and Christophe Dony. Exceptional use cases. In *Proceedings of Models/UML, ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems*, pages 568–583, Montego Bay, Jamaica, October 2005.

[32] Alexander Romanovsky, Christophe Dony, Jørgen L. Knudsen, and Anand Tripathi, editors. *Proceedings of the ECOOP 2005 Workshop on : Developing Systems that Handle Exceptions*. Technical Report, LIRMM, Montpellier, 2005.

[33] Alexander B. Romanovsky, Christophe Dony, Jørgen Lindskov Knudsen, and Anand Tripathi, editors. *Advances in Exception Handling Techniques*, volume 2022 of *LNCS*. Springer-Verlag, 2001.

[34] Christophe Dony, Christelle Urtado, and Sylvain Vauttier. Exception handling and asynchronous active objects : Issues and proposal. In Dony et al. [6], chapter 5.

[35] Frédéric Souchon, Sylvain Vauttier, Christelle Urtado, and Christophe Dony. Fiabilité des applications multi-agents : le système de gestion d'exceptions SaGE. In *Systèmes multi-agents - Défis scientifiques et nouveaux usages - Actes des Journées Francophones sur les Systèmes Multi-Agents 2004*, pages 121–134, Saint-Rémy-lès-Chevreuse, France, November 2004.

[36] Aurélien Campéas, Christophe Dony, Christelle Urtado, and Sylvain Vauttier. Distributed exception handling : Ideas, lessons and issues with recent exception handling systems. In *Proceedings of RISE'04 : International Workshop on Rapid Integration of Software Engineering techniques*, pages 75–85, Luxemburg, ovember 2004.

[37] F. Souchon, C. Dony, C. Urtado, and S. Vauttier. Improving exception handling in multi-agent systems. In C. Lucena, A. Garcia, A. Romanovsky, J. Castro, and P. Alencar, editors, *Advances in Software Engineering for Multi-Agent System*, number 2940 in LNCS. Springer-Verlag, 2004.

[38] Zahia Guessoum, Nora Faci, and Jean-Pierre Briot. Adaptive replication of large-scale multi-agent systems - Towards a fault-tolerant multi-agent platform. *ACM Software Engineering Notes*, 30(4):1–6, July 2005. Proceedings of the ICSE'05 4th International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'05), St Louis, MI, USA, May 2005.

[39] Zahia Guessoum, Jean-Pierre Briot, Olivier Marin, Athmane Hamel, and Pierre Sens. Dynamic and adaptive replication for large-scale reliable multi-agent systems. In *Software Engineering for Large-Scale Multi-Agent Systems*, number 2603 in LNCS, pages 182–198. Springer-Verlag, 2003.

[40] Maíra Gatti, Carlos Lucena, and Jean-Pierre Briot. On fault tolerance in law-governed multi-agent systems. In *5th International Workshop on Software Engineering for Large Scale Multi-Agent Systems (SELMAS'2006), co-located with the International Conference on Software Engineering (ICSE'2006)*, Shanghai, China, May 2006.

[41] Jean-Pierre Briot, Zahia Guessoum, Samir Aknine, Alessandro L. Almeida, Nora Faci, Jacques Malenfant, Olivier Marin, and Pierre Sens. Engineering control strategies for replication-based fault-tolerant multi-agent systems. In *International Workshop on Engineering of Fault-Tolerant Systems (EFTS'2006)*, Luxembourg, Luxembourg, June 2006.

[42] Isabelle Alvarez and S. Bernard. Ranking cases with decision trees: a geometric method that preserves intelligibility. In *Proceedings of IJCAI'2005*, August 2005.

[43] Jean-Pierre Briot, Zahia Guessoum, Sébastien Charpentier, Samir Aknine, Olivier Marin, and Pierre Sens. Dynamic adaptation of replication strategies for reliable agents. In *Proceedings of the Second Symposium on Adaptive Agents and Multi-Agent Systems (AAMAS-2)*, pages 13–21, AISB'02 Convention, London, U.K., April 2002. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour (AISB).

[44] Zahia Guessoum, Jean-Pierre Briot, Nora Faci, and Olivier Marin. Un mécanisme de réplication adaptative pour des SMA tolérants aux pannes. In *Systèmes multi-agents - Défis scientifiques et nouveaux usages - Actes des JFSMA'04. Revue des Sciences et Technologies de l'Information (RSTI)*, pages 125–139, November 2004. Hors série.

[45] Zahia Guessoum, Jean-Pierre Briot, and Nora Faci. Towards fault-tolerant massively multiagent systems. In Toru Ishida, Les Gasser, and Hideyuki Nakashima, editors, *Massively Multi-Agent Systems I*, number 3446 in LNCS, pages 55–69. Springer-Verlag, July 2005.

[46] Alessandro de Luna Almeida, Samir Aknine, Jean-Pierre Briot, and Jacques Malenfant. Méthode de réplication basée sur les plans pour la tolérance aux pannes des systèmes multi-agents. In *Actes des JFSMA'05. Revue des Sciences et Technologies de l'Information (RSTI)*, November 2005. Actes des 13èmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA'05). Hors série. Short paper.

[47] C.B. Weinstock, J.B. Goodenough, and J.J. Hudak. Dependability cases. Technical Report CMU/SEI-2004-TN-016, SEI, CMU, Pittsburgh, PA, USA, 2004.