

*Projet Arles**Architectures logicielles et systèmes
distribués**Rocquencourt*

THÈME 1B



*R*apport
d'Activité

2002

Table des matières

1. Composition de l'équipe	1
2. Présentation et objectifs généraux	1
3. Fondements scientifiques	2
3.1. Introduction	2
3.2. Développement orienté architecture de systèmes	2
3.3. Architectures middleware pour utilisateurs mobiles	4
4. Domaines d'application	6
5. Logiciels	6
5.1. Introduction	6
5.2. Environnement de développement orienté architecture	7
5.3. Infrastructure middleware pour la mobilité	7
5.4. Système de fichiers distribués ad hoc	8
6. Résultats nouveaux	8
6.1. Introduction	8
6.2. Développement orienté-architecture de systèmes distribués	8
6.2.1. Problématique	9
6.2.2. Environnement de développement	9
6.2.3. Composition dynamique de systèmes	10
6.2.3.1. Composition de services.	12
6.2.3.2. Composition de services en environnement mobile.	12
6.3. Architecture middleware pour l'intelligence ambiante	13
6.3.1. Problématique	13
6.3.2. Gestion de l'accès aux données	13
6.3.2.1. Système de fichiers distribués ad hoc.	14
6.3.2.2. Caches Web coopératifs ad hoc	15
6.3.3. Gestion de l'accès aux services	15
8. Actions régionales, nationales et internationales	16
8.1. Actions européennes	16
8.1.1. Projet IST DSoS	16
8.1.2. Projet IST OZONE	16
8.1.3. Projet ITEA Vivian	17
8.2. Relations bilatérales internationales	17
8.3. Réseaux et groupes de travail internationaux	17
8.3.1. CaberNet	17
8.3.2. iTrust	17
9. Diffusion des résultats	18
9.1. Animation de la communauté scientifique	18
9.1.1. Comités de programme	18
9.1.2. Autres responsabilités sur un plan international	19
9.1.3. Autres responsabilités sur un plan national	19
9.2. Enseignement	19
9.3. Accueil de stagiaires	19
9.4. Participation à des colloques, séminaires, invitations	20
10. Bibliographie	20

1. Composition de l'équipe

Responsable scientifique

Valérie Issarny [DR INRIA]

Assistante de projet

Sylvie Loubressac [TR INRIA]

Ingénieur associé

Viet Khoi Nguyen [jusqu'au 17/12/2002]

Collaborateur extérieur

Nicole Levy

Chercheur post-doctorant

Nikolaos Georgantas [bourse INRIA/région IdF, depuis le 15/10/2002]

Ingénieurs experts INRIA

Anis Ben Arbia [jusqu'au 31/08/2002]

Daniele Sacchetti [depuis le 25/03/2002]

Chercheurs doctorants

Malika Boulkenafed [bourse INRIA/région IdF, Université de Paris 6]

Teresa Higuera [bourse INRIA, Université de Rennes I, jusqu'au 31/01/2002]

Jinshan Liu [bourse INRIA, Université de Versailles Saint-Quentin en Yvelines, depuis le 01/10/2002]

Françoise Sailhan [bourse INRIA/région IdF, Université de Paris 6]

Ferda Tartanoglu [bourse INRIA, Université de Paris 6]

2. Présentation et objectifs généraux

Le principal objectif du projet ARLES, créé au début de l'année 2002, est *l'étude de langages, de méthodes, d'outils et d'infrastructures d'exécution pour l'aide au développement de systèmes distribués efficaces et sûrs de fonctionnement*. Par *système distribué*, nous entendons un système logiciel regroupant des fonctionnalités des niveaux applicatif et exécutif, et s'exécutant sur un ensemble de sites répartis, interconnectés par un réseau de communication. Dans ce cadre, nous sommes plus spécifiquement intéressés par la réalisation des fonctions du niveau exécutif des systèmes distribués, lesquelles ont trait à la gestion des ressources distribuées sous-jacentes, de manière à :

- Optimiser les performances des systèmes tant du point de vue de la consommation des ressources que de celui des temps de réponse.
- Accroître la sûreté de fonctionnement des systèmes, laquelle regroupe les critères de disponibilité, de fiabilité, de sécurité et de sûreté.

Le développement de systèmes distribués reste encore mal maîtrisé, non seulement de par la complexité inhérente à ces systèmes (p.ex., hétérogénéité, concurrence), mais également de par leur évolution continue (p.ex., prise en compte de nouvelles technologies). Aussi, nous pensons qu'il est essentiel de fournir des solutions aux deux points suivants :

- Développement rigoureux des systèmes distribués, ce qui suppose le moyen de raisonner sur leurs propriétés fonctionnelles et extra-fonctionnelles.
- Infrastructures d'exécution distribuée (c.-à-d., infrastructures *middleware* et systèmes sous-jacents) intégrant de manière efficace les nouvelles composantes technologiques.

Afin de répondre à l'objectif fixé, notre approche repose sur le développement de systèmes distribués à partir de la description de leur architecture, ce qui se ramène à un développement des systèmes par composition. Ce choix est motivé par :

- Notre expérience dans le domaine du développement de systèmes distribués à partir de la description de leur architecture logicielle, qui définit une structuration du système, nous a convaincus de son intérêt pour aboutir à des systèmes robustes et efficaces. La robustesse vient de la possibilité de pouvoir spécifier formellement, et par conséquent de raisonner sur, le comportement de systèmes logiciels complexes. L'efficacité vient de la possibilité de spécialiser la composition du système distribué au regard des exigences de l'application et de l'environnement d'exécution, et ainsi de limiter l'intégration de fonctions coûteuses et non nécessaires.
- Dans la pratique, l'émergence de standards d'architectures de systèmes distribués conduit à la définition de composants réutilisables (qualifiés de COTS pour *Component Off The Shelf*) des niveaux applicatif et exécutif. Par ailleurs, de nombreux systèmes distribués sont construits par intégration de composants existants (ou *systèmes legs* pour *legacy systems*). La problématique du développement de systèmes distribués devient donc principalement orientée vers la composition de systèmes.

Les activités de recherche du projet ARLES sont plus spécifiquement centrées sur le développement de systèmes distribués relevant de l'intelligence ambiante. Ces systèmes ont vocation à permettre l'accès aux services et données en tout lieu, à tout instant, ce qui requiert une composition dynamique des services applicatifs et exécutifs, en fonction de l'environnement d'exécution. L'objectif de nos travaux est ainsi d'offrir à moyen terme une *architecture de composition de systèmes*, laquelle recouvre :

- Un environnement de développement orienté architecture, comprenant : (i) un langage de description d'architecture (ou ADL), qui soit spécialisable pour supporter la mécanisation d'une partie du développement, (ii) des spécialisations de l'ADL avec des méthodes et outils associés, principalement orientées vers la composition dynamique de systèmes offrant des garanties de qualité.
- La définition de nouveaux styles d'architectures de systèmes, dédiés à la composition dynamique de systèmes intégrant des sites mobiles, et offrant des garanties de performance et de sûreté de fonctionnement.
- Une infrastructure d'exécution pour systèmes distribués intégrant des sites mobiles, éventuellement de relativement faible capacité, et interagissant au moyen d'un réseau sans fil local (p.ex., IEEE 802.11, Bluetooth) ou global (p.ex., GSM, GPRS, UMTS). Nous nous intéressons en particulier à l'exploitation de réseaux *ad hoc*, qui permettent de s'affranchir d'une infrastructure de communication pour l'accès aux services et données distants.

3. Fondements scientifiques

3.1. Introduction

Mots clés : *architectures logicielles, composition de systèmes, mécanisation, systèmes distribués, architectures middleware, qualité de service, sûreté de fonctionnement, mobilité, intelligence ambiante, ubiquité.*

Les activités de recherche du projet ARLES relèvent des domaines des architectures logicielles et des systèmes distribués. Nous étudions plus spécifiquement le développement orienté architecture de systèmes distribués et les architectures *middleware* pour utilisateurs mobiles, qui font respectivement l'objet des 2 sections suivantes.

3.2. Développement orienté architecture de systèmes

Depuis le début des années 90, une partie de la communauté du domaine de l'ingénierie du logiciel s'est intéressée à l'exploitation de la description d'architectures logicielles pour simplifier le développement de systèmes complexes. Ceci a donné naissance au domaine des architectures logicielles, dont les travaux ont porté sur la définition de langages pour la description rigoureuse d'architectures logicielles, et sur la proposition de méthodes et d'outils associés à ces langages afin de mécaniser l'analyse, la conception et la

construction de systèmes complexes à partir de leur description architecturale. L'architecture logicielle d'un système est centrée sur la définition de la structure de ce dernier et abstrait les détails d'implémentation. Une architecture logicielle de système précise ainsi la configuration du système en termes des composants le constituant (c.-à-d., les unités de calcul et de stockage), et des connecteurs (c.-à-d., les protocoles d'interaction) employés pour combiner les composants. Le développement orienté architecture d'un système distribué recouvre plus spécifiquement :

- La description de l'architecture du système, c'est-à-dire, la structuration du système en terme d'interconnexion de *composants* (représentant des services applicatifs et exécutifs) *via* des *connecteurs* (représentant des protocoles d'interaction et plus généralement des infrastructures d'exécution distribuée).
- L'implémentation des composants et connecteurs en privilégiant la réutilisation de logiciels.
- La construction du système par composition des implémentations des composants et connecteurs, ce qui se ramène à produire le code de liaison ou plus généralement d'intégration (*wrapper*).

Les travaux dans le domaine des architectures logicielles se sont par conséquent concentrés sur la définition de *langages de description d'architectures* (ou ADL pour *Architecture Description Language*) pour la modélisation de systèmes à partir de leur description architecturale, ainsi que sur la définition de méthodes et d'outils associés pour mécaniser certaines des tâches du processus de développement. A titre d'illustration, la figure 1 fournit une spécification, sous forme de diagramme, d'une architecture client-serveur suivant l'ADL Wright. Cet ADL s'appuie sur une modélisation du système en CSP et permet ainsi l'emploi de l'outil FDR pour la mécanisation de l'analyse de propriétés du système. Succinctement, l'architecture comprend les composants client et serveur, composés *via* le connecteur RPC implantant l'appel de procédure à distance. La description de l'architecture proposée permet de vérifier la compatibilité du comportement des composants avec celui du connecteur et de vérifier des propriétés globales du système (typiquement, l'absence d'interblocage avec l'ADL Wright).

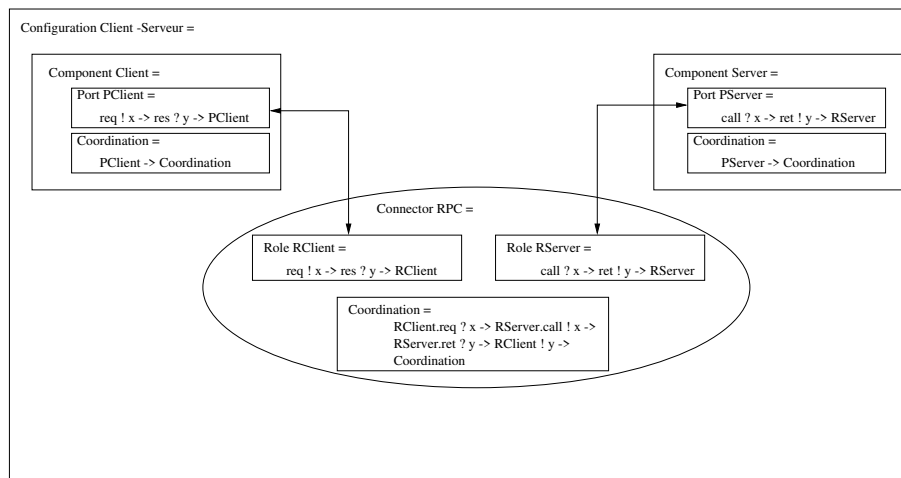


Figure 1. Une spécification d'architecture client-serveur

Notre problématique porte principalement sur la caractérisation des architectures de systèmes distribués du point de vue de la réalisation du niveau exécutif (également qualifié de réalisation des propriétés *non fonctionnelles* ou *extra-fonctionnelles*). Le développement de systèmes distribués reste en effet une tâche difficile pour le développeur d'applications qui doit non seulement maîtriser la réalisation des fonctionnalités spécifiques à l'application mais également celle de la gestion des ressources (p.ex., intégration de fonctions du niveau exécutif pour l'optimisation des performances ou l'accroissement de la fiabilité). Les nouvelles

architectures de systèmes distribués simplifient en partie cette dernière tâche *via* la définition d'architectures *middleware* qui offrent des services exécutifs (ou *services systèmes*) réutilisables. Toutefois, le développement de tout système distribué requiert la spécification, la sélection et la composition adéquates des services applicatifs et exécutifs, requis par l'application visée, ce qui n'est pas trivial.

L'intérêt de la mécanisation de la sélection et de la composition de services exécutifs est encore faiblement perçu dans un contexte statique, car il y a peu de services implémentés dans un souci de réutilisation, qui sont effectivement disponibles à ce jour. Par ailleurs, la spécification formelle du comportement des services, qui est requise pour cette mécanisation, est souvent considérée comme trop complexe par les développeurs. Une telle mécanisation devient toutefois impérative dans un contexte de construction dynamique des systèmes distribués, comme exigé par les nouveaux domaines d'application des systèmes distribués et notamment celui de l'intelligence ambiante. De par l'expérience des membres du projet ARLES dans le domaine de la mécanisation du développement de systèmes distribués dans un contexte statique, et de par le besoin croissant de fournir des solutions à la réalisation de l'intelligence ambiante, le projet ARLES se concentre sur l'étude de solutions à la composition dynamique de systèmes distribués pour ce dernier domaine d'application. Dans ce cadre, une architecture de système distribué décrit, de manière plus ou moins concrète suivant le niveau de son élaboration, une composition de services applicatifs et exécutifs, pour la réalisation d'un service applicatif particulier.

3.3. Architectures middleware pour utilisateurs mobiles

La définition d'architectures *middleware* est apparue au début des années 90 pour faciliter le développement de systèmes distribués. Une architecture *middleware* comprend : (i) un système logiciel, appelé *broker*, qui offre les fonctionnalités de base pour la gestion des interactions entre composants logiciels s'exécutant dans des espaces d'adressage distincts, et (ii) des services exécutifs additionnels mettant en œuvre des propriétés extra-fonctionnelles avancées (p.ex., services de persistance, de sécurité, de tolérance aux fautes). Les architectures *middleware* qui sont parmi les plus utilisées à l'heure actuelle sont celles qualifiées d'orientées objet ; celles-ci sont basées sur la notion d'appel de procédure à distance étendue avec un modèle objet. Des exemples connus sont les architectures CORBA de l'OMG, DCOM de Microsoft et *Enterprise Java Beans* (EJB) de Sun. Le (World Wide) Web peut également être qualifié d'architecture *middleware* facilitant l'échange de données au-dessus de l'Internet ; citons également l'architecture de services Web, basée sur XML, qui va vraisemblablement jouer une part importante dans la réalisation des futurs systèmes distribués. Les travaux de recherche dans le domaine des architectures *middleware* s'attachent à faire évoluer les architectures existantes, de manière à répondre aux exigences des nouvelles applications distribuées. Ces exigences peuvent être grossièrement classées en deux catégories et sont respectivement liées à : (i) l'évolution de l'infrastructure matérielle (p.ex., utilisation de réseaux sans fil, de terminaux de capacité variée) et (ii) l'intégration d'applications d'entreprise qui ne peuvent, en général, pas (ou du moins difficilement) être modifiées (c.-à-d., les systèmes legs). Les évolutions récentes dans le domaine des architectures *middleware* qui offrent un intérêt direct pour le développement de systèmes distribués du domaine de l'intelligence ambiante relèvent des fonctionnalités suivantes :

- **Intégration de services Web** : L'architecture de services Web vise à permettre l'intégration d'applications hétérogènes disponibles sur le Web, qualifiées de services Web, au travers de l'introduction de schémas XML pour la définition des interfaces des services et des données échangées. Un avantage clé de cette architecture, comparée aux autres architectures *middleware*, pour le développement de systèmes du domaine de l'intelligence ambiante est qu'elle tend à être adoptée par une majorité d'utilisateurs et de développeurs. Il devient ainsi possible de supposer un langage de description et un protocole de communication pour les services, qui soient majoritairement (voire universellement) acceptés, ce qui est primordial pour permettre la réalisation de la vision de l'intelligence ambiante. Par ailleurs, la volonté d'intégrer des applications à l'échelle du Web conduit à des solutions à la description d'interfaces qui sont plus élaborées que les solutions simplement basées sur les langages de définition d'interfaces,

en introduisant notamment des langages déclaratifs pour la définition de conversations et de composition de services. Toutefois, l'utilisation effective de l'architecture de services Web dans le contexte de l'intelligence ambiante requiert d'étudier son adéquation dans un environnement mobile, notamment pour ce qui concerne les critères de performance, ainsi que son évolution pour traiter les différentes propriétés extra-fonctionnelles inhérentes aux systèmes distribués.

- **Localisation et découverte de services :** Les évolutions en matière d'informatique mobile ont conduit à des exigences de (re-)configuration dynamique des systèmes distribués. Ceci a donné lieu à la définition de différents protocoles pour la localisation et la découverte de services (ou ressources). Les protocoles de localisation se présente sous la forme de services de courtage qui gèrent des bases de données de services et qui traitent des requêtes de services étant donné leur nom, adresse ou encore description abstraite. Les protocoles de découverte de services définissent des processus spontanés qui permettent de découvrir les services disponibles sur le réseau, en général local, à tout instant. Les solutions existantes à la localisation et découverte de services offrent une base solide à la réalisation de systèmes du domaine de l'intelligence ambiante. Toutefois, différents problèmes restent à être abordés afin d'intégrer au mieux la connectivités des sites mobiles, les exigences de propriétés extra-fonctionnelles des applications, ou encore les éventuelles contraintes de ressources des sites mobiles.
- **Gestion de données nomades :** La gestion de données nomades a été l'un des premiers sujets abordés dans le domaine des systèmes distribués en la présence de sites mobiles. Les premiers travaux ont plus spécifiquement porté sur le traitement de l'occurrence de déconnexions lors d'interactions avec un serveur de données. Dans ce contexte, les solutions apportées ont trait au préchargement des données par anticipation des accès et à la définition de protocoles de *réconciliation* de données, de manière à permettre la modification de données partagées depuis un poste mobile malgré l'occurrence de déconnexions. Avec l'apparition des ordinateurs de poche, leur utilisation croissante et les évolutions en matière de réseaux de communication sans fil, il devient aussi nécessaire de revoir la gestion des ressources locales de façon à prendre en compte la capacité limitée des terminaux et la diversité des réseaux. Ceci conduit à des études au niveau du système d'exploitation de ces terminaux ainsi qu'au niveau des systèmes distribués. Dans le premier cas, il s'agit de revoir les politiques de gestion de ressources locales afin de permettre l'exécution d'applications traditionnelles (p.ex., applications multimédias) malgré les contraintes de limitation de ressources et notamment d'énergie. Dans le second cas, il s'agit de proposer des architectures *middleware* permettant d'adapter les données et les calculs à la capacité des sites mobiles et des réseaux utilisés ; les solutions proposées s'appuient en général sur la définition de serveurs *proxy* (typiquement, ces serveurs adaptent les flux de données transmis aux sites mobiles). De manière complémentaires aux travaux susmentionnés, des études sont entreprises sur le stockage de données personnels sur des serveurs non sûrs à *proximité*, de manière à éviter la réplication des données sur les sites mobiles de faible capacité tout en maximisant la disponibilité des données depuis ces sites. Les résultats dans le domaine de la gestion de données nomades sont généralement applicables à celui de l'intelligence ambiante. Toutefois, les problèmes posés relèvent de l'intégration effective de machines de faible capacité ou encore de la sécurisation des données, qui font encore l'objet d'études. Indiquons également l'étude nécessaire de l'intégration des réseaux sans fil tant locaux (*ad hoc* ou basés sur une infrastructure) que globaux, incluant l'utilisation conjointe de ces réseaux.
- **Exploitation des divers réseaux sans fil :** Les évolutions récentes des réseaux sans fil offrent des caractéristiques primordiales pour la réalisation de systèmes du domaine de l'intelligence ambiante. En particulier, l'accroissement du débit de ces réseaux permet l'obtention de temps de réponse acceptables et leur diversité permet une plus grande connectivité pour les terminaux mobiles. Dans ce cadre, les réseaux *ad hoc* (intégrant des protocoles de routage *ad hoc*) sont particulièrement attractifs car ils ne requièrent pas la disponibilité d'une infrastructure dans la portée de communication du site mobile pour que celui-ci puisse interagir avec des sites distants.

Toutefois, l'exploitation effective et combinée des divers réseaux sans fil restent un problème ouvert, qui nécessite l'étude de solutions spécifiques au niveau des architectures *middleware* de manière à notamment intégrer la consommation énergétique associée aux communications ou encore traiter les problèmes de sécurité inhérents à ces réseaux.

Les architectures *middleware* et en particulier celles destinées à répondre aux nouvelles exigences des applications offrent des solutions de base à la définition d'une architecture *middleware* pour effectivement construire des systèmes distribués du domaine de l'intelligence ambiante. Toutefois, les principaux problèmes posés, qui ont trait à la composition dynamique et à l'exécution efficace des systèmes dans un environnement ouvert, ne trouvent pas encore de solutions satisfaisantes et font pour la plupart l'objet de différentes études.

4. Domaines d'application

Mots clés : *Systèmes pour l'intelligence ambiante, systèmes d'information, systèmes mobiles, services Web.*

Le domaine d'applications visé par le projet ARLES est celui de l'intelligence ambiante où nous nous concentrons plus spécifiquement sur les applications relevant des systèmes d'information grand public. Dans ce cadre, une partie de nos travaux est aussi d'intérêt pour le développement des systèmes d'information d'entreprise où la composition du système est statique et où la principale difficulté réside dans l'intégration de systèmes complexes existants (c.-à-d., les systèmes legs). Nos études s'appliquent à ce domaine dans ce sens où les solutions étudiées au problème de la spécification du comportement des systèmes et de leur composition peuvent être exploitées pour guider la spécification du comportement des systèmes legs et faciliter ainsi leur intégration. Une distinction de notre travail vient de l'infrastructure d'exécution requise, qui est moins problématique pour les systèmes d'information d'entreprise puisque leur réalisation peut s'appuyer sur des infrastructures existantes comme celles définies par l'OMG, Sun ou Microsoft. Notons que les services que nous visons peuvent être multimédias. De tels services requièrent des études spécifiques du fait des contraintes temporelles associées. Aussi, dans un premier temps, nous privilégierons des services manipulant des données discrètes.

Notons que le domaine d'applications que nous visons est volontairement vaste. Nous préférons en effet aborder notre problématique sous un angle général, même si la prise en compte d'applications particulières permet d'obtenir des solutions plus performantes, et est parfois nécessaire à l'apport de solutions effectives. Nous considérons toutefois des applications spécifiques pour montrer l'utilisation pratique de nos résultats et éventuellement guider nos choix, si cela s'avère nécessaire pour produire des résultats effectifs. Les applications que nous considérons dans ce cadre incluent celles qui seront développées dans le cadre du projet européen OZONE (voir § 8.1.2). Ces applications relèvent de l'*extended home environment* et nous sommes fortement impliqués dans leur définition. Dans ce cadre, nous contribuons en particulier à la réalisation d'un démonstrateur en collaboration avec les projets INRIA IMARA, Parole, Langue&Dialogue, MAIA et SARDES, qui permet l'accès à des services Web grand public, éventuellement composites, depuis divers terminaux mobiles, au moyen d'une interface multimodale combinant la parole et le geste [23].

5. Logiciels

5.1. Introduction

Afin de valider nos résultats, nous sommes amenés à développer un certain nombre de prototypes de systèmes. Les logiciels ainsi développés sont : (i) un environnement de développement orienté-architecture basé sur UML, (ii) une infrastructure *middleware* pour la mobilité basée sur l'architecture de services Web et (iii) un système de gestion de fichiers distribués qui supporte le partage de données entre sites mobiles à portée de communication *via* le réseau local sans fil sous-jacent.

5.2. Environnement de développement orienté architecture

Participants : Valérie Issarny [correspondante], Viet Khoi Nguyen.

L'environnement de développement orienté-architecture que nous développons vise à faciliter la conception et l'analyse de systèmes logiciels complexes sans requérir une expertise dans le domaine des méthodes formelles de la part des développeurs. L'environnement permet ainsi aux développeurs de décrire une architecture au moyen du langage UML et éventuellement d'intégrer des attributs de qualité. Ces attributs permettent d'effectuer une analyse du système tant d'un point de vue qualitatif par *model checking* que d'un point de vue quantitatif pour ce qui concerne les propriétés de performance et de fiabilité. Les descriptions architecturales enrichies d'attributs de qualité donnent en effet lieu à la génération automatique de modèles formels de systèmes qui sont traités par des outils d'analyse existants (c-à-d., SPIN pour le *model checking*, QNAP pour l'analyse de performance, et SURE/ASSIST pour l'analyse de fiabilité).

Le prototype d'environnement que nous avons développé jusqu'ici comprend :

- Un langage de description d'architectures spécialisable, basé sur UML
- Des spécialisations du langage de description d'architectures pour la définition d'attributs de qualité relevant de l'analyse qualitative du système par *model checking*, et de l'analyse quantitative du système en termes de performance et de fiabilité.
- Des outils de génération automatique de modèles traités par les outils SPIN, QNAP et SURE/ASSIST à partir des descriptions d'architectures enrichies des attributs de qualité.

L'ensemble de ces fonctionnalités a été intégré dans l'environnement graphique Rational Rose, *via* le développement d'*add-ins* spécifiques [21].

5.3. Infrastructure middleware pour la mobilité

Participant : Valérie Issarny [correspondante], Daniele Sacchetti, Françoise Sailhan.

Dans le cadre du développement d'une infrastructure *middleware* pour la mobilité, notre objectif est de permettre le développement de systèmes du domaine de l'intelligence ambiante, c'est-à-dire, permettre l'accès à des services en tout lieu, à tout instant, depuis différents terminaux et en particulier des sites mobiles de faible capacité. Nous avons débuté la conception d'une telle infrastructure à l'été 2002, dont les fonctions de base ont été définies [25]. Un premier prototype devrait être opérationnel dans le courant de l'année 2003.

L'exigence de disponibilité des services en tout lieu, à tout instant nous a conduits à nous appuyer sur une technologie largement acceptée pour ce qui concerne le noyau de l'infrastructure *middleware*. Nous avons ainsi naturellement retenu l'architecture de services Web comme base de définition de notre infrastructure. Les spécificités de notre infrastructure viennent de :

- La définition de services Web qui permettent un processus de découverte et de composition dynamique de services efficaces, tant en terme de consommation de ressources que de temps de réponse, considérant que les services peuvent être eux-mêmes hébergés par des sites mobiles.
- La définition d'un service *middleware* de découvertes de services Web, tant au niveau du réseau local que de l'Internet, en fonction de la connectivité des sites. Nous nous intéressons notamment à la prise en compte de réseaux *ad hoc* de manière accroître la connectivité des sites.
- La spécialisation dynamique des connecteurs de manière à offrir les propriétés de qualité exigées pour les services, en particulier en termes de sécurité et de performance.

Le prototype que nous développons s'appuie à ce jour sur Java et Apache Axis pour le développement de serveurs SOAP dédiés à des sites mobiles de faible capacité, le protocole SLP pour la découverte de services en environnement local et le réseau IEEE 802.11b. Le prototype de l'infrastructure *middleware* sera en outre progressivement enrichi de manière à intégrer nos différents résultats de recherche dans ce domaine.

5.4. Système de fichiers distribués ad hoc

Participants : Valérie Issarny [correspondante], Anis Ben Arbia, Malika Boulkenafed.

Les services de gestion de données nomades existants ont pour la plupart été conçus pour des réseaux sans fil basés sur une infrastructure. Dans le cadre de nos activités de recherche, nous avons développé un prototype de système de fichiers distribués, appelé ADHOCFS, qui exploite les spécificités des réseaux *ad hoc* (un saut) et permet ainsi à des sites mobiles à portée de communication les uns des autres et appartenant à un même domaine de sécurité, de partager directement leurs données. Plus précisément, les données liées à un domaine de sécurité sont stockées sur un serveur de données fixe et sont en outre répliquées (cachées) sur les sites mobiles suite à des requêtes au serveur, ou à des sites mobiles à portée de communication ayant déjà une copie de la donnée. Le prototype du système ADHOCFS a été développé au-dessus du système Linux et du réseau IEEE 802.11b. Les fonctionnalités du système incluent :

- La gestion de groupes dynamiques *ad hoc* qui regroupent les sites mobiles à portée de communication les uns des autres et appartenant au même domaine de sécurité. Un groupe dynamique *ad hoc* est ainsi la base de la réalisation d'un cache distribué coopératif, réparti sur l'ensemble des sites du groupe, lesquels interagissent suivant le modèle *peer-to-peer*.
- La gestion de la cohérence des données sur les sites mobiles où : (i) la cohérence des données répliquées sur les sites d'un même groupe obéit à un protocole de cohérence forte de manière à supporter des applications de travail collaboratif comme les réunions, et (ii) la cohérence des données répliquées sur des sites n'appartenant pas à un même groupe obéit à un protocole de cohérence optimiste permettant une mise à jour concurrente des données.
- Le gestion de la sécurité de manière à garantir que toute donnée n'est accédée que par les (utilisateurs des) sites ayant accès au domaine de sécurité dont relève la donnée.

Un prototype du système ADHOCFS regroupant les fonctionnalités mentionnées ci-avant est disponible depuis l'été 2002. Nous enrichissons actuellement ce prototype de manière à accroître la disponibilité des données au sein des groupes *ad hoc* via la création de répliques préventifs qui pallient la déconnexion de sites mobiles stockant des copies de données non disponibles par ailleurs.

6. Résultats nouveaux

6.1. Introduction

Le projet ARLES vise l'étude de solutions au développement orienté architecture de systèmes distribués, qui permettent une composition de systèmes offrant des garanties de qualité (ou propriétés extra-fonctionnelles) aux utilisateurs. A cette fin, nous étudions la définition de langages, de méthodes, d'outils et d'infrastructures d'exécution qui facilitent le développement de systèmes distribués, en offrant des solutions à la mécanisation de la conception, de l'analyse et de la construction des systèmes. Dans ce cadre, nos travaux sont plus spécifiquement centrés sur le développement de systèmes distribués relevant de l'intelligence ambiante, c'est-à-dire qui permettent l'accès aux services et données en tout lieu, à tout instant, ce qui requiert une composition dynamique des services applicatifs et exécutifs, en fonction de l'environnement d'exécution. Nos activités se décomposent en deux classes principales qui relèvent respectivement (i) de solutions à la composition dynamique de systèmes dans le cadre d'un développement orienté-architecture de systèmes distribués et (ii) de la conception et mise en œuvre d'une architecture *middleware* supportant l'accès aux services et données en tout lieu, à tout instant. Les sections suivantes présentent la problématique de nos travaux et les résultats obtenus au cours de l'année 2002 pour ces deux classes d'activités.

6.2. Développement orienté-architecture de systèmes distribués

Participants : Nikolaos Georgantas, Valérie Issarny, Viet Khoi Nguyen, Nicole Levy, Jinshan Liu, Ferda Tartanoglu.

6.2.1. Problématique

Les résultats du domaine des architectures logicielles sont pour la plupart complémentaires. Ils se traduisent par des langages de description d'architectures (ou ADLs pour *Architecture Description Languages*) qui reposent majoritairement sur la définition de composants, de connecteurs et de configurations. L'intérêt d'un ADL particulier vient de la méthode sur laquelle il repose pour supporter le processus de développement de systèmes. Les travaux dans ce domaine ont été centrés sur l'aide à l'analyse du comportement des systèmes (l'ADL repose alors sur une méthode formelle particulière comme par exemple CSP pour le langage Wright) et à la construction (l'ADL est alors défini en relation avec une infrastructure d'exécution). Il n'y a donc pas une unique définition d'ADL mais un ensemble d'ADLs contribuant à l'aide au développement de systèmes, chacun supportant une des (sous-)tâches du processus de développement. Il est donc avantageux de définir un ADL de base (qui ne comprend que les types primitifs associés aux éléments architecturaux à la base de la définition de la structure d'un système -composant, connecteur, configuration-) et des spécialisations de cet ADL de base, pour :

- Supporter une tâche particulière du processus de développement comme par exemple l'ajout d'attributs de performance dans la définition des éléments architecturaux pour une analyse de la performance.
- Supporter le développement d'une famille de systèmes particulière (plus ou moins précise) en fixant les types d'éléments architecturaux pouvant être exploités pour le développement du système. Une telle spécialisation peut être apparentée au sous-typage des langages à objets.

Il apparaît ainsi intéressant de fournir un environnement de développement s'appuyant, non pas sur un unique ADL, mais sur un ADL spécialisable, où chaque spécialisation introduite assiste une tâche du processus de développement. C'est dans ce contexte que s'inscrit notre activité de recherche. Nous nous intéressons plus spécifiquement à l'aide au développement de systèmes distribués du point de vue de la réalisation de propriétés extra-fonctionnelles, dans le cas d'une composition dynamique en environnement mobile. Ceci nous conduit à avoir des activités se rapportant à :

- La conception et la réalisation d'un environnement de développement orienté architecture, construit autour d'un ADL spécialisable.
- La définition de nouvelles architectures de systèmes distribués offrant des propriétés extra-fonctionnelles et pouvant être composées dynamiquement dans un contexte mobile.

6.2.2. Environnement de développement

La conception de notre environnement de développement a débuté avant la création du projet ARLES. Nos activités de cette année ont principalement porté sur la mise en œuvre d'un prototype de l'environnement de développement [21] dont nous rappelons les principales caractéristiques dans ce qui suit.

La définition d'un ADL spécialisable, visant à intégrer dans un même environnement les différentes solutions apportées dans le domaine des architectures logicielles, a déjà été abordée dans la littérature. Nous trouvons notamment l'ADL ACME du CMU et plus récemment l'ADL fondé sur XML de UCI. Pour la définition de l'ADL de base de notre environnement de développement, nous nous appuyons sur les exigences suivantes :

- La spécialisation de l'ADL, ce qui conduit à n'introduire que les notions élémentaires à la base de la définition d'une architecture, à savoir la définition de la structure du système en termes de composants, de connecteurs et de configurations, sans contraindre le style architectural.
- La définition d'un ADL fondé sur un langage standard, de manière à garantir une utilisation effective de notre environnement mais aussi la disponibilité d'une spécification adéquate des services disponibles, sujets à composition.

Au regard de ce qui précède, nous avons défini un ADL minimal reposant sur la définition des éléments structurels du système, et fondé sur le langage UML (*Unified Modeling Language*). Nous avons retenu UML

car, son utilisation se généralise, il permet d'exploiter des outils d'aide au développement existants, et une traduction en XML est disponible. L'inconvénient d'UML est l'imprécision de sa sémantique, ce qui devrait être en partie résolu à terme au regard des travaux en cours. Ceci n'a par ailleurs qu'une incidence mineure sur la définition précise de notre ADL puisque ce dernier fait l'objet d'une définition propre et n'utilise que peu d'éléments d'UML. Nous avons intégré le profil UML correspondant à notre ADL dans un environnement existant, l'environnement Rational Rose, que nous avons en outre enrichi d'un analyseur de contraintes OCL (*Object Constraint Language*), implémenté dans le langage OCAML. L'analyseur OCL permet de vérifier la correction syntaxique des descriptions d'architectures, au regard des contraintes OCL associées aux éléments architecturaux.

A titre d'illustration, la figure 2 donne une définition de composant et de connecteur au moyen de notre ADL, dans le cadre d'un système offrant les fonctionnalités d'une agence de voyages, à partir de la composition de services Web existants. Le composant *FlightReservation* défini, offre des fonctions de réservation de vols à partir de services Web de compagnies aériennes. Ce composant requiert ainsi les interfaces de ces derniers services (voir la définition associée à *Interfaces*), avec lesquels il interagit *via* le protocole HTTP, en tant que client (voir définition de *Port* donnée informellement par *HTTPClientSpec*). Le connecteur *HTTP* abstrait un connecteur multi-partie basé sur le protocole HTTP. Ce connecteur comprend notamment le rôle (défini ici de manière informelle par *HTTPServerSpec*) implantant la partie serveur du protocole pour ce qui est des interactions avec les services des compagnies aériennes. Le comportement du protocole est par ailleurs défini informellement par *HTTPProtocolSpec*. Cette définition de l'ADL s'apparente naturellement à celle de l'ADL Wright si l'on considère une substitution des définitions informelles par des processus CSP. Nous avons proposé une spécialisation dans ce sens de notre ADL, en nous appuyant sur l'outil SPIN. Remarquons que nous avons retenu SPIN/PROMELA pour permettre une analyse comportementale des systèmes au niveau architectural ainsi que pour supporter la composition systématique d'architectures *middleware* dans un cadre statique [9].

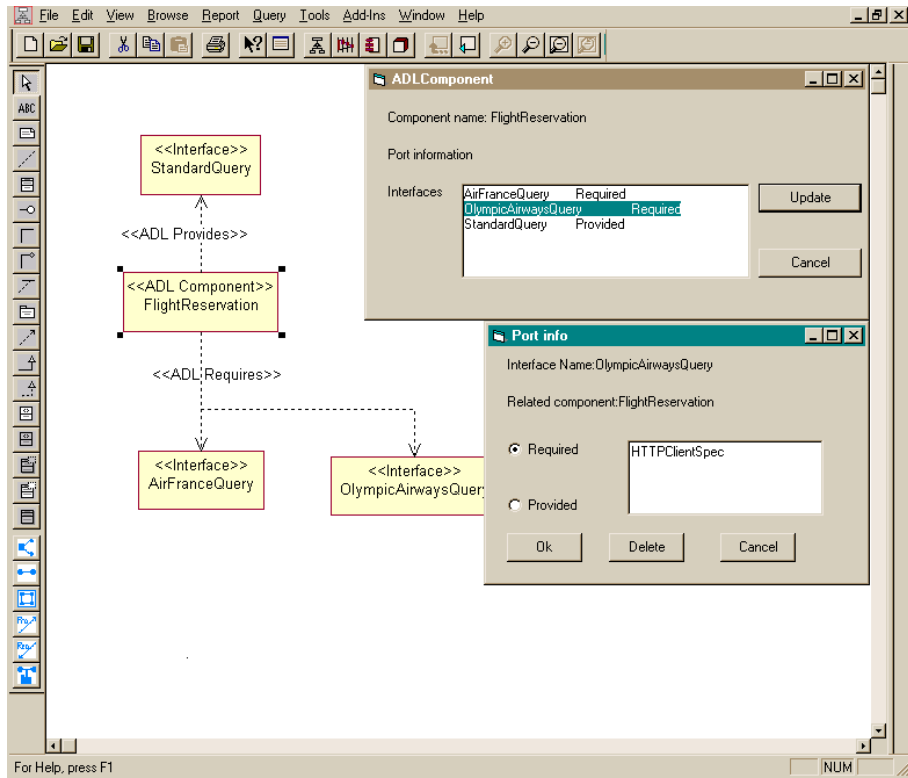
Etant donnée la définition de l'ADL de base de notre environnement, notre objectif est d'en offrir différentes spécialisations pour supporter la mécanisation de la conception, de l'analyse et de la construction des systèmes distribués, en nous concentrant sur la qualité des systèmes produits. Nous avons ainsi proposé des solutions à l'analyse qualitative (au moyen de la vérification de modèles - *model checking*) et quantitative (du point de vue de la fiabilité et de la performance). Celles-ci reposent sur la spécialisation de la définition des éléments architecturaux par l'intégration d'attributs de qualité ne nécessitant pas de connaissance de modèles formels par les développeurs. Les descriptions d'architectures obtenues à partir de ces spécialisations de l'ADL sont ensuite automatiquement traduites en des modèles formels, traités par des outils d'analyse existants [13].

L'environnement de développement que nous avons proposé supporte la conception de systèmes logiciels distribués complexes, en offrant notamment des fonctionnalités d'analyse de la qualité des systèmes développés. Nous avons en particulier étudié son exploitation pour l'aide à la conception d'architectures *middleware* spécialisés, c'est-à-dire, la composition de services *middleware* nécessaires à la garantie des exigences extra-fonctionnelles d'un applicatif donné [12].

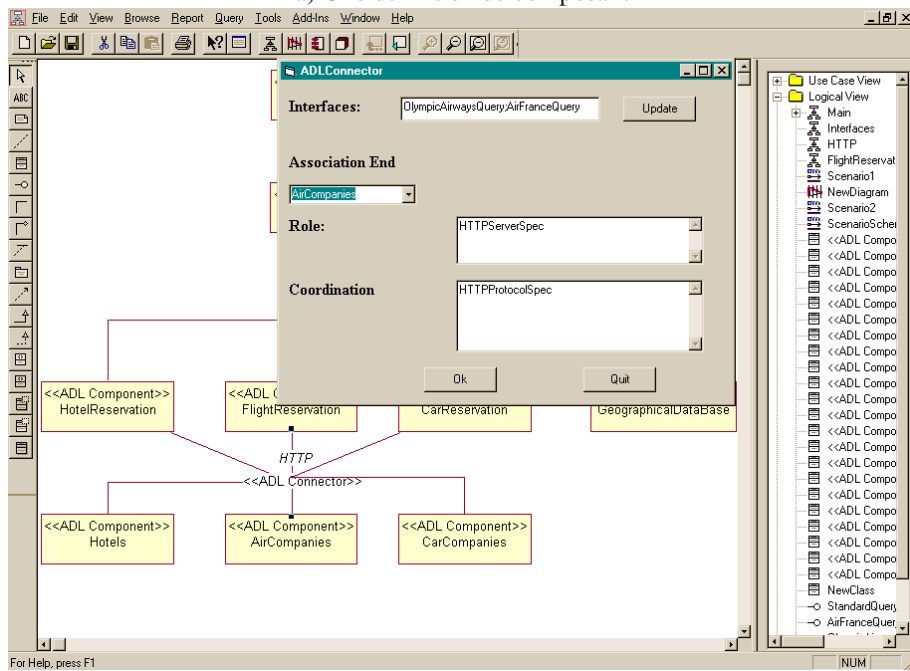
L'environnement que nous avons développé permet essentiellement un développement classique de systèmes distribués, en ce sens qu'il supporte une approche statique où les composants des systèmes sont *a priori* connus et modifiables par les développeurs. Le domaine d'application de l'intelligence ambiante et notre souci de promouvoir le développement de systèmes offrant des garanties de qualité (spécifiquement, performance et sûreté de fonctionnement) aux utilisateurs, nécessitent l'apport de nouvelles solutions au niveau de la définition des architectures de systèmes pour en supporter la composition dynamique.

6.2.3. Composition dynamique de systèmes

Les systèmes relevant de l'intelligence ambiante doivent permettre aux utilisateurs d'accéder aux services et contenus en tout lieu, à tout instant. Dans ce cadre, nous étudions la définition d'une architecture de système distribué qui réalise la composition dynamique de services satisfaisant la requête de l'utilisateur en fonction de son environnement. Une telle architecture regroupe ainsi (i) une solution à la description de services qui en permet une composition dynamique et (ii) une architecture *middleware* qui supporte la composition effective



a) Une définition de composant



b) Une définition de connecteur

Figure 2. Définition de composant et connecteur

de services à l'exécution. Une exigence clé pour une telle solution est qu'elle soit utilisée par la majorité afin de garantir une description uniforme des services mis à disposition ainsi que l'emploi d'un protocole de communication commun. Ceci nous a conduits à baser nos travaux sur l'architecture de services Web [16]. Nos travaux relevant de la définition d'une architecture *middleware* pour la composition dynamique de services à l'exécution sont présentés dans la section 6.2 suivante. Les exigences associées à la définition de langages déclaratifs pour décrire les services mis à disposition afin d'en permettre une sélection et une composition dynamique s'apparentent à celles associées à la définition d'ADL. Il s'agit de : (i) fournir une description des services qui abstraient les détails d'implémentation et (ii) préciser la structure des services composites, tout en supportant l'analyse de la qualité du système produit. Dans le cadre d'une composition dynamique, il faut en outre préciser le comportement du service composite et prendre en compte les spécificités de l'environnement d'exécution comme le déploiement des services sur des sites mobiles, d'éventuellement faible capacité. Comme indiqué précédemment, nous nous appuyons sur l'architecture de services Web et exploitons donc les langages de description d'interfaces associée à cette architecture, qui permettent de définir les opérations offertes par un service ainsi que les séquences d'appels à un service. Nos travaux de cette année ont porté sur : (i) la définition d'un langage de composition de services en ne considérant pas leur possible exécution sur des sites mobiles dans un premier temps, et (ii) la définition de services qui soient adaptés à une composition dynamique en environnement mobile.

6.2.3.1. Composition de services.

La composition de services en environnement ouvert où les services ne dépendent pas d'une même organisation fait l'objet de différents travaux depuis ces deux dernières années suite à l'émergence de l'architecture de services Web. Dans ce cadre, l'objectif est de permettre l'intégration de services autonomes fournis par différentes organisations en offrant un langage de composition adéquat. Si l'on ne considère que le cas de services implantant des *business processes* et s'exécutant sur des serveurs fixes, le problème majeur réside dans la réalisation de services composites qui soient sûrs de fonctionnement. La solution généralement retenue pour le développement de services composites sûrs de fonctionnement, consiste à s'appuyer sur la notion de transactions qui est souvent supportée au niveaux des services à composer, en partie du fait du couplage fort des services Web avec un système de gestion de base de données. Toutefois, la mise en œuvre de transactions distribuées n'est pas adaptée dans un environnement ouvert, où les sites sont autonomes et où la latence d'accès à ces services peut être élevée. Ceci a conduit à retenir le modèle de transactions ouvertes dont l'annulation dépend de la définition d'opérations de compensation. En général, les mécanismes de tolérance aux fautes de recouvrement arrière dont relèvent les transactions ne sont pas adaptés pour les systèmes distribués ouverts. C'est pourquoi nous avons proposé une solution qui est fondée sur un mécanisme de recouvrement avant, et plus spécifiquement un mécanisme de traitement d'exceptions coopératif [19]. Notre solution repose sur un langage de composition de services, basé sur la définition d'actions atomiques coopératives réparties sur les services composés, couplé avec un support d'exécution qui ne requiert pas de modification au niveau des services composés [20]. Nous examinons à présent une solution à la génération automatique de services composites, étant donnée leur définition ainsi qu'à la définition de mécanismes systèmes visant à accroître les performances des services composites en termes de temps de réponse.

6.2.3.2. Composition de services en environnement mobile.

La composition de services en environnement mobile soulève le problème d'offrir un processus de composition efficace en termes de consommation de ressources et de temps de réponse mais également en terme de qualité du service composite au regard de propriétés de sécurité, disponibilité et temps de réponse. Nous avons étudié une première solution à ce problème en nous concentrant, dans un premier temps, sur la gestion des interactions directes entre services, la gestion de l'ensemble de la configuration d'un service composite faisant partie de nos futurs travaux. Nous avons ainsi proposé une solution à la description de services qui supporte [25][16] :

- La sélection dynamique de services tout en minimisant le coût à l'exécution de la fonction d'appariement associée.
- La spécialisation systématique des connecteurs du point de vue de la mise en œuvre de propriétés de sécurité et de performance.

Nos futurs travaux dans ce cadre portent sur la définition d'un processus de sélection et composition dynamiques de services qui optimise la qualité des services composites en termes de propriétés de sécurité, disponibilité et de performance.

6.3. Architecture middleware pour l'intelligence ambiante

Participants : Anis Ben Arbia, Malika Boulkenafed, Nikolaos Georgantas, Valérie Issarny, Daniele Sacchetti, Françoise Sailhan.

6.3.1. Problématique

Les évolutions en matière de réseaux sans fil et d'ordinateurs de petite taille, permettent *a priori* à tout utilisateur d'accéder à des services numériques en tout lieu. La figure 3 donne un aperçu des interactions possibles, à terme, depuis un même site mobile où les ronds représentent des sites mobiles, et le cercle la portée du WLAN par rapport au site mobile qui se trouve en son centre. Le choix de l'une ou l'autre de ces interactions dépend de plusieurs facteurs, qui outre la fonctionnalité visée, relèvent de la consommation de ressources et en particulier énergétique, du temps de réponse, et de la sûreté de fonctionnement induits par telle ou telle interaction. L'objectif de nos travaux est de mieux comprendre ces critères par l'étude de services exécutifs supportant la réalisation de systèmes distribués dans un tel environnement. Nous avons subdivisé notre étude en deux points :

- Gestion de l'accès aux données.
- Gestion de l'accès aux services que l'on peut également qualifier de composition dynamique de services.

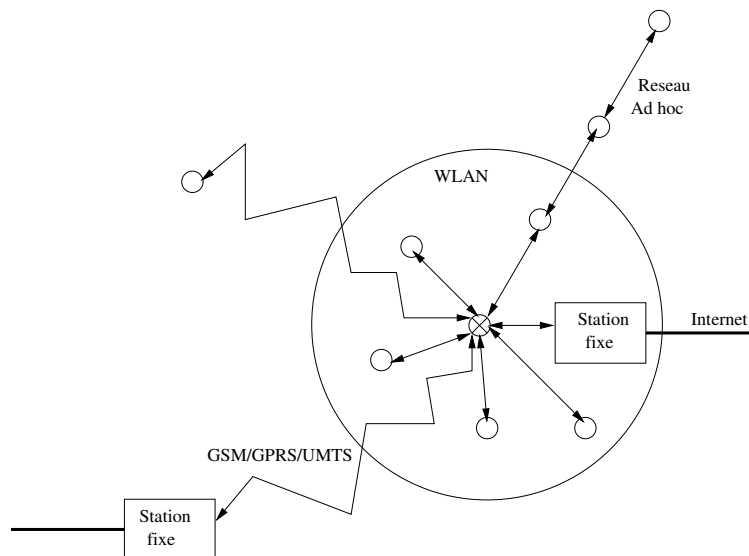


Figure 3. Interactions en environnement mobile

6.3.2. Gestion de l'accès aux données

L'objectif de nos travaux est ici de permettre à un utilisateur d'accéder à ses données depuis tout terminal mobile, en tout lieu, sans requérir une connexion systématique à un site distant spécifique. Notre approche pour offrir une garantie de sûreté de fonctionnement et de performance lors de l'accès aux données, repose sur l'emploi du *caching*. Les données deviennent accessibles, non seulement depuis les sites les hébergeant, mais également depuis les sites dont les utilisateurs ont des centres d'intérêt communs. Nous considérons ainsi que

les sites mobiles implantent des caches de données des serveurs. Nous distinguons deux cas suivant que les données sont privées ou publiques.

6.3.2.1. Système de fichiers distribués *ad hoc*.

Pour un accès simplifié aux données privées tout en offrant des garanties de sûreté de fonctionnement et de performance, nous examinons la conception et l'implantation d'un système de gestion de fichiers distribués, appelé ADHOCFS, où l'accès à un fichier se traduit de manière transparente par un accès au site *le plus proche* stockant le fichier. Le critère de proximité, qui reste à définir précisément, est notamment évalué en termes de temps de réponse et de consommation énergétique. La réalisation de ce système débouchera sur un ensemble de services exécutifs, réutilisables pour le développement de divers systèmes distribués intégrant des sites mobiles. Nous étudions plus spécifiquement les aspects suivants :

- *Caching et partage de données dans un WLAN* : Il s'agit ici de considérer le partage de fichiers entre des sites mobiles pouvant interagir directement *via* le WLAN en mode *ad hoc*. Nous sommes typiquement dans le cas de la coopération entre utilisateurs ayant accès aux mêmes fichiers. L'intérêt de cette solution est qu'elle conduit *a priori* à une optimisation du temps de réponse et, de manière générale, à une optimisation de la consommation des ressources. Toutefois, cette approche requiert des solutions relatives à la gestion de groupes de sites mobiles, de la cohérence des données et de la sûreté de fonctionnement, comme précisé ci-après, ainsi qu'un travail d'expérimentation conséquent pour sa validation.
- *Groupes de sites mobiles* : Différents protocoles de découverte de services ont été proposés pour permettre à des sites mobiles d'interagir directement lorsqu'ils se trouvent à portée de communication l'un de l'autre *via* le WLAN sous-jacent. Nous avons ainsi proposé une solution à la gestion de groupes dynamiques de sites mobiles, à partir de ces protocoles, les groupes étant définis en fonction de l'application. Dans le cas du système ADHOCFS, il s'agit de définir des groupes de sites au regard des fichiers qu'ils sont susceptibles de partager, identifiés par l'appartenance à des *domaines de sécurité* communs.
- *Cohérence des données* : La gestion de la cohérence des données réparties sur des sites mobiles a fait l'objet de différents travaux depuis le début des années 90. Ceux-ci sont principalement centrés sur le traitement des déconnexions, abordant le préchargement de données et leur réconciliation. Notre intérêt se porte plus spécifiquement sur la gestion de la cohérence de données au sein d'un groupe de sites mobiles, l'objectif étant d'examiner une solution optimale en terme de consommation de ressources induite. Nous avons ainsi proposé un protocole de cohérence hybride [14] qui implante : (i) une cohérence forte au sein d'un groupe afin de supporter des applications de travail collaboratif et (ii) une cohérence optimiste au niveaux des réplicas appartenant à des groupes distincts.
- *Sécurité* : La sécurité est un problème incontournable dans le contexte de la mobilité. Nous nous intéressons au problème du partage sécurisé des données, et plus spécifiquement à des solutions minimisant la consommation de ressources sur les sites mobiles. Dans ce cadre, nous avons entrepris une collaboration avec le projet CODES afin d'étudier une solution au calcul de clés privées au niveau des groupes.
- *Disponibilité* : Les sources de défaillance en environnement mobile sont nombreuses, du fait des déconnexions volontaires ou non, des sites. Nous examinons la définition de protocoles de réplication pour ce contexte particulier, de manière à accroître la disponibilité des données, tout en optimisant la consommation de ressources induite. Nous avons ainsi proposé une solution de base à la réplication pro-active des données au sein d'un groupe, qui prennent en compte les profils des différents terminaux du groupe.

Un premier prototype du système ADHOCFS est opérationnel. Ce prototype, implémenté en OCAML au-dessus du système Linux, gère le partage de données entre sites mobiles interagissant *via* IEEE 802.11 en

mode *ad hoc*, et la cohérence des données partagées par de tels sites. Ce prototype sera ensuite enrichi par l'intégration de nos solutions aux problèmes de la sécurité et de la disponibilité des données.

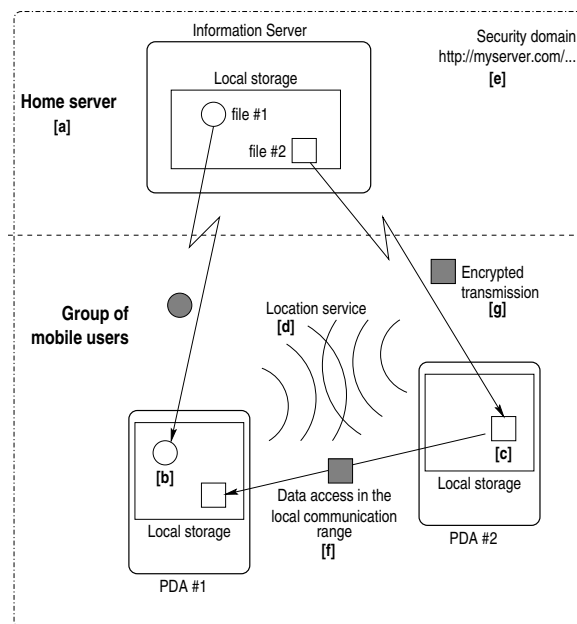


Figure 4. Partage de fichiers dans un WLAN

6.3.2.2. Caches Web coopératifs *ad hoc*

Pour la gestion de l'accès à des données publiques depuis un site mobile, nous nous concentrons sur un type de données particulier, à savoir les données du Web qui sont parmi les plus populaires. Le *caching* de données dans le Web a fait l'objet de nombreux travaux, du fait de son intérêt majeur pour diminuer les temps de réponse pour les utilisateurs. Le *caching* de données est généralement couplé à un préchargement qui, s'il conduit à un accroissement de la consommation de la bande passante, peut s'avérer bénéfique pour diminuer les temps de réponse. Nous nous intéressons ici à étudier le bénéfice potentiel du *caching* de données réparti sur des sites mobiles. Nous avons ainsi mené une étude sur le sujet en considérant la recherche de pages Web depuis des sites mobiles accessibles *via* un réseau *ad hoc*, plutôt que suivant une approche classique s'appuyant sur un accès depuis un serveur *proxy* fixe [17]. Il s'agit ici d'examiner le gain potentiel en termes de temps de réponse, de disponibilité et de consommation de ressources. La solution que nous avons proposée combine l'exploitation de protocoles de routage *ad hoc* et d'infrastructures de manière accroître la connectivité des sites. Cette solution repose sur un protocole coopératif de gestion de caches et a été évaluée en terme de consommation énergétique induite [18].

6.3.3. Gestion de l'accès aux services

Comme indiqué dans la section 6.2, nous étudions des solutions à la composition dynamique de services, suite à des requêtes de services des utilisateurs. Notre objectif est ici de comprendre l'impact de cette facilité sur la gestion des sites mobiles où nous comptons généraliser nos résultats dans le domaine de la gestion de l'accès aux données en environnement mobile, au cas de l'accès aux services. Il s'agit ainsi d'étudier :

- Le *caching* de services sur les sites, de manière à optimiser la composition dynamique de systèmes en termes de consommation de ressources et de temps de réponse.
- La coopération entre sites dans la perspective de gérer les centres d'intérêt communs et ainsi identifier les sites ayant potentiellement fait appel aux mêmes services.

- La gestion des services exécutés localement, et notamment leur adaptation en fonction des ressources disponibles.
- La composition de services ne faisant intervenir que des sites mobiles.
- L'utilisation de différents réseaux sans fil.
- L'accroissement de la qualité des services.

A ce jour, les seuls travaux existants sont ceux relatifs à la définition de protocoles pour la découverte de services (ou ressources). Notre objectif est de définir une solution complète à la découverte et intégration de services en environnement mobile, couvrant aussi bien les services de haut niveau (p. ex., services Web traditionnels) que ceux de bas niveau (p. ex., ressources) en prenant en compte les divers réseaux de communications exploitables et la qualité des services accessibles, mais aussi de leur utilisation. Au cours de cette année, nous avons défini l'architecture *middleware* de base qui repose sur un protocole de communication standard et un service de découverte prenant en compte la connectivité des sites [16]. Nos futurs travaux portent sur l'étude de solutions aux problèmes énoncés précédemment.

8. Actions régionales, nationales et internationales

8.1. Actions européennes

8.1.1. Projet IST DSoS

Participants : Nikolaos Georgantas, Valérie Issarny, Viet Khoi Nguyen, Ferda Tartanoglu.

- Intitulé : IST DSoS - *Dependable Systems of Systems*
- URL : <http://www.newcastle.research.ec.org/dsos/index.html>
- Activité concernée : §6.2
- Période : [Avril 2000 - Mars 2003]
- Partenaires : Université de Newcastle (RU) - coordinateur, CNRS-LAAS (Toulouse), QinetiQ (RU), INRIA (Rocquencourt), LRI (Orsay), Université d'Ulm (Allemagne), Université technique de Vienne (Autriche).

L'action DSoS vise à fournir des solutions facilitant la composition de systèmes de systèmes sûrs de fonctionnement, à partir de systèmes informatiques autonomes. Cette action se concentre notamment sur la conception, le placement et les propriétés des interfaces des systèmes à composer. Elle examine également la définition de méthodes et outils pour la construction effective des systèmes à partir de la spécification des interfaces des sous-systèmes à composer, ainsi que pour l'évaluation et la validation de la sûreté de fonctionnement des systèmes composés.

8.1.2. Projet IST OZONE

Participant : Valérie Issarny, Daniele Sacchetti, Françoise Sailhan.

- Intitulé : IST OZONE - *New technologies and services for emerging nomadic societies*
- URL : <http://www.extra.research.philips.com/euprojects/ozone/>
- Activité concernée : §6.2 et 6.3
- Période : [Novembre 2001 - Juin 2004]
- Partenaires : Philips Research Eindhoven (Pays Bas) - coordinateur, EPICOID (Pays Bas), INRIA (Lorraine, Rennes, Rhône-Alpes, Rocquencourt), IMEC (Belgique), PRF (Suresne), THOMSON (Rennes), Université technique d'Eindhoven (Pays Bas).

L'action OZONE vise à développer de nouveaux concepts, techniques et outils pour fournir un environnement informatique invisible, afin de permettre une utilisation nomade, grand public, des technologies de l'information. Une base importante du projet est l'utilisation des nouvelles technologies pour une recherche

et une utilisation de l'information, centrées sur l'utilisateur, comparée à l'approche actuelle qui est centrée sur l'infrastructure informatique. Ceci requiert des solutions adaptées au niveau des interfaces utilisateurs, de l'environnement logiciel et de l'infrastructure matérielle.

8.1.3. *Projet ITEA Vivian*

Participants : Anis Ben Arbia, Malika Boulkenafed, Valérie Issarny, Daniele Sacchetti.

- Intitulé : ITEA VIVIAN - *Opening mobile platforms for the development of component-based applications*
- URL : <http://www-nrc.nokia.com/Vivian/index.html>
- Activité concernée : §6.3
- Période : [Juin 2000 - Aout 2002]
- Partenaires : Nokia NRC (Finlande) - coordinateur, ADISOFT (Allemagne), CAS (Allemagne), Université technique d'Helsinki (Finlande), INRIA (Rocquencourt), INT (Evry), MEMODATA (Caen), PALMWARE (Toulouse), PARAVANT (Finlande), Philips Research Eindhoven (Pays Bas), Unicom (Finlande)

L'action Vivian porte sur la définition d'une plate-forme pour ordinateurs de poche personnels, sans fil (*e.g.* Psion serie 5, *Communicator* de Nokia), afin de faciliter la construction d'applications logicielles distribuées pour ces systèmes. La solution retenue s'appuie sur une plate-forme *middleware* à base de composants, de type CORBA. Dans ce cadre, les problèmes abordés portent sur la définition et répartition des services composant l'architecture *middleware* sachant que les services devant s'exécuter sur les ordinateurs personnels doivent s'accommoder des ressources limitées disponibles.

8.2. Relations bilatérales internationales

Valérie Issarny participe à une coopération bilatérale Franco-Portugaise avec l'INESC-Porto, soutenue par INRIA/ICCTI. Le thème de cette coopération est l'adaptation dynamique d'architectures logicielles de systèmes distribués.

8.3. Réseaux et groupes de travail internationaux

8.3.1. *CaberNet*

- Intitulé : IST NoE CABERNET - *Network of Excellence in Distributed and Dependable Systems*
- URL : <http://www.newcastle.research.ec.org/cabernet/>
- Période : [Janvier 2001 - Décembre 2003]
- Partenaires coordinateurs : Université de Newcastle (RU) - coordinateur principal, Université technique de Vienne (Autriche), INRIA (Rocquencourt), LAAS-CNRS (Toulouse), Université de Kaiserslautern (Allemagne), Université de Bologne (Italie), Université de Pise (Italie), Université de Twente (Pays Bas), Université de Lisbonne (Portugal), Université de Cambridge (RU), Université de Lancaster (RU).

Cabernet est un réseau d'excellence dans le domaine des systèmes distribués et des systèmes sûrs de fonctionnement. Sa mission est de coordonner la recherche européenne de haut niveau dans ces domaines, dans le but de la rendre visible aux gouvernements et aux acteurs industriels d'une part, et d'améliorer la qualité de l'enseignement d'autre part.

8.3.2. *iTrust*

- Intitulé : IST WG iTRUST - *Working Group on Trust Management in Dynamic Open Systems*
- <http://www.itrust.uoc.gr/>
- Période : [Septembre 2002 - Août 2005]

- Partenaires : Université de Crete - coordinateur, CCLRC (RU), CNR-ISTC (Italie), HP (RU), Imperial College (RU), INRIA (Rocquencourt) Intracom SA (Grèce), King's College (RU), Nine by Nive Co (RU), Pleafs Information Systems SA (Grèce), Queen Mary University College (RU), Sintef telecom and Informatics (Norvège), Trinity College Dublin (RU), Université autonome de Barcelone (Espagne), University of Dortmund (Allemagne), Université d'Oslo (Norvège), Université Strathclyde (RU), Virtual Trip Ltd (Grèce),

Le groupe de travail iTrust vise l'étude interdisciplinaire de la modélisation de la confiance des systèmes informatiques ouverts et dynamiques, au travers de l'organisation de workshops regroupant des experts européens des domaines de l'informatique, du droit, de la philosophie et des sciences sociales.

9. Diffusion des résultats

9.1. Animation de la communauté scientifique

9.1.1. Comités de programme

- V. Issarny est membre du comité de programme de SAINT'02 - *Symposium on Applications and the Internet*. Janvier-Février 2002, Nara, Japon.
- V. Issarny est membre du comité de programme d'AlgoTel'02. Avril 2002, Montpellier, France.
- V. Issarny est membre du comité de programme de WPDRTS'2002 - *10th workshop on parallel and distributed real-time systems*. Avril 2002, Fort Lauderdale, USA.
- V. Issarny est membre du comité de programme de ISORC'2002 - *5th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*. Avril 2002, Washington DC, USA.
- V. Issarny est membre du comité de programme de ICSE 2002 - *24th International Conference on Software Engineering*. Mai 2002, Orlando, USA.
- V. Issarny est membre du comité de programme de ICSE'2002 *Workshop on Architecting Dependable Systems*. Mai 2002, Orlando, USA.
- V. Issarny est membre du comité de programme de *Workshop on Software Infrastructures for Component-Based Applications on Consumer Devices*. Septembre 2002. Lausanne, Suisse.
- N. Levy est membre du comité de programme de *SoMeT'02, First International Workshop on Lyee Methodology*. Octobre 2002, Paris, France.
- V. Issarny est membre du comité de programme de SIGOPS-EW'02 - *10th ACM SIGOPS European Workshop on « Can we really depend on an OS ? »*. Septembre 2003, Bordeaux, France.
- V. Issarny est membre du comité de programme de MDM'03 - *4th International Conference on Mobile Data Management*. Janvier 2003, Melbourne, Australie.
- N. Levy est membre du comité de programme des journées AFADL'03, *Approches formelles dans l'assistance au développement de logiciels*. Janvier 2003, Rennes, France.
- V. Issarny et N. Levy sont membres du comité de programme de ICSE'2003 *Workshop on Architecting Dependable Systems*. Mai 2003, Portland, USA.
- V. Issarny est membre du comité de programme de MCM'03 - *1st International ICDCS Workshop on Mobile Computing Middleware*. Mai 2003, Providence, USA.
- V. Issarny est membre du comité de programme de WWW'2003 - *12th International World Wide Web Conference*. Mai 2003, Budapest, Hongrie.
- V. Issarny est membre du comité de programme de *1st International Conference on Trust Management*. Mai 2003, Heraklion, Grèce.
- V. Issarny est membre du comité de programme de ESEC/FSE'03 - *4th joint meeting of the European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering*. Septembre 2003, Helsinki, Finlande.

- V. Issarny est membre du comité de programme de LADC'03 - *1st Latin-American Symposium on Dependable Computing*. Octobre 2003, Sao Paulo, Brésil.
- N. Levy est membre du comité de programme de SoMeT'03, *2nd International Workshop on Lye Methodology*. Octobre 2003, Stockholm, Suède.
- V. Issarny est membre du comité de programme de ICSE 2004 - *26th International Conference on Software Engineering*. Mai 2004, Edinburgh, RU.
- V. Issarny est membre du comité de pilotage de WIESS - *Workshop on industrial experiences with system software*, co-sponsorisé par USENIX, IEETCOS et ACM-SIGOPS.

9.1.2. Autres responsabilités sur un plan international

- V. Issarny est vice-présidente de l'ACM SIGOPS.
- V. Issarny est responsable du comité exécutif du consortium AIR&D.
- V. Issarny est responsable du *Research Coordination and Training Committee* de CaberNet.
- V. Issarny est membre du *core expert group* du groupe de travail de l'ISTAG, sur *Software technologies, services and distributed systems* (WG9).
- N. Levy est membre du Réseau thématique Européen ECET.
- M. Boulkenafed assure la maintenance de la base de données des conférences relevant du domaine des systèmes d'exploitation et publie la liste des conférences à venir dans chaque numéro de la revue OSR de l'ACM SIGOPS.

9.1.3. Autres responsabilités sur un plan national

- N. Levy est membre du groupe AFADL du GDR ALP.
- N. Levy est membre du groupe B du GDR ALP.
- N. Levy est directrice de l'Institut des Sciences et Techniques des Yvelines (ISTY), depuis juin. Cet institut de l'Université de Versailles est habilité à délivrer le titre d'ingénieur.
- N. Levy est membre élu du conseil d'administration de l'IUT de Vélizy.
- N. Levy est membre élu du conseil de département Informatique de l'IUT de Vélizy.
- N. Levy est membre élu de la commission de spécialistes 27^e section (Informatique), de l'Université de Versailles.
- N. Levy est membre élu de la commission de spécialistes 27^e section (Informatique) de l'Université d'Evry.

9.2. Enseignement

- N. Levy est responsable de l'option AOD (Architecture à Objets Distribués) du DEA MISI de l'Université de Versailles saint-Quentin en Yvelines.
- V. Issarny et N. Levy assurent l'enseignement du module principal de l'option AOD (Architecture à Objets Distribués) du DEA MISI de l'Université de Versailles saint-Quentin en Yvelines.
- V. Issarny et Ferda Tartanoglu ont assuré un cours d'une journée sur Java Temps Réel à l'école IN2P3 d'informatique temps réel.
- M. Boulkenafed et V. Issarny ont assuré le cours *Architectures middleware et XML* de la formation de 3^e cycle IPA du Pôle Universitaire Léonard de Vinci.
- F. Tartanoglu a assuré le cours *Architectures middleware* de la Licence professionnelle ISDRN (Intégrateur de Systèmes Distribués et Réseaux Numériques) de l'IUT de Velizy.

9.3. Accueil de stagiaires

Pendant l'année 2002, les membres du projet ont encadré ou co-encadré des stagiaires venant de différents établissements :

- R. Bashkar, Calcul de clés de groupes dans les réseaux *ad hoc* mobiles, Stage en collaboration avec le projet CODES (Stage de 6 mois)
- A. Pathak, Implémentation de la gestion de la cohérence dans un système de fichiers distribués intégrant des groupes *ad hoc* (Stage de 3 mois).
- M. Akin, Architecture logicielle de systèmes pour la mobilité. Université de Paris-Sud et Supélec (Stage de DEA et de fin d'étude).

9.4. Participation à des colloques, séminaires, invitations

Des membres du projet ont participé à des conférences et « workshops » ayant donné lieu à des publications dans des actes ; on se reportera à la bibliographie pour en avoir la liste. Certains ont également donné des présentations lors de séminaires ou workshops :

- V. Issarny. *Software Architectures of Dependable Systems : From Closed to Open Systems*. ICSE'2002 Workshop on Architecting Dependable Systems - Exposé invité, Orlando, USA, Mai 2002.
- V. Issarny. *OZONE : Nouvelles technologies et services pour les sociétés nomades*, Séminaire Aristote - Collège Polytechnique sur la nouvelle interaction homme-environnement, Palaiseau, Juin 2002.
- V. Issarny. *Infrastructure middleware pour systèmes distribués ad hoc*. Rencontre INRIA-Ilitech sur les architectures de systèmes distribués pour l'intelligence ambiante, Rocquencourt, Juin 2002.
- V. Issarny. *Mobile Systems & Middleware for Consumer-oriented Ambient Intelligence*, ERCIM/NSF Strategic Workshop on Middleware for Mobile Systems, Vienne, Autriche, Juillet 2002.
- V. Issarny. *Software Architectures of Dependable Systems : From Closed to Open Systems*. IBM Watson, Yorktown Heights, USA, Septembre 2002.
- M. Boulkenafed. *Adaptive Replication for Enhanced Data Availability within Mobile Collaborative Ad Hoc Groups*. CaberNet Radicals Workshop, Bertinoro, Italie, Octobre 2002.
- V. Issarny. *Towards the Development of Ambient Intelligence Systems : Supporting Data Sharing in Mobile Ad hoc Systems*. Université de Lancaster, Lancaster, RU, Novembre 2002.
- N. Levy. *De l'analyse des besoins à la spécification formelle en B : un cours en première année d'IUT*. Journée B : L'enseignement de B et des méthodes formelles, Nantes, Novembre 2002.

10. Bibliographie

Bibliographie de référence

- [1] P. A. BERNSTEIN. *Middleware : A Model for Distributed System Services*. in « Communications of the ACM », numéro 2, volume 39, 1996, pages 86-98.
- [2] D. GARLAN, D. SIEWIOREK, A. SMALAGIC, P. STEENKISTE. *Project Aura : Toward Distraction-Free Pervasive Computing*. in « IEEE Pervasive Computing », numéro 2, volume 1, 2002.
- [3] éditeurs J. C. LAPRIE., *Dependability : Basic Concepts and Terminology*. série Dependable Computing and Fault-Tolerant Systems, volume 5, Springer-Verlag, 1992.
- [4] P. A. LEE, T. ANDERSON. *Fault Tolerance Principles and Practice*. édition 2nd, série Dependable Computing and Fault-Tolerant Systems, volume 3, Springer - Verlag, 1990.

- [5] N. MEDVIDOVIC, R. TAYLOR. *A Framework for Classifying and Comparing Architecture Description Languages*. in « Proceedings of the Joint European Software Engineering Conference - ACM SIGSOFT Symposium on Foundations of Software Engineering », pages 60-76, 1997.
- [6] M. SHAW, D. GARLAN. *Software Architecture : Perspectives on an Emerging Disciplines*. Prentice Hall, 1996.
- [7] D. B. TERRY, M. M. THEIMER, K. PETERSEN, A. J. DEMERS, M. J. SPREITZER, C. H. HAUSER. *Managing Update Conflicts in a Weakly Connected Replicated Storage System*. in « Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles », pages 172-183, 1995.

Thèses et habilitations à diriger des recherche

- [8] T. HIGUERA. *Solutions à la gestion mémoire pour systèmes Java temps réels*. Thèse de doctorat, Université de Rennes I, Mars, 2002.
- [9] C. KLOUKINAS. *Composition d'architectures logicielles*. Thèse de doctorat, Université de Rennes I, Février, 2002.
- [10] S. ROUVRAIS. *Utilisation d'agents mobiles pour la construction de services distribués*. Thèse de doctorat, Université de Rennes I, Juillet, 2002.

Articles et chapitres de livre

- [11] T. HIGUERA, V. ISSARNY, M. BANÂTRE, G. CABILLIC, F. PARAIN, J. LESOT. *Memory Management for Real-time Java : an Efficient Solution using Hardware Support*. in « Real-Time Systems journal (Kluwer Pub.) », 2002.
- [12] V. ISSARNY, C. KLOUKINAS, A. ZARRAS.. *Systematic Aid for Developing Middleware Architectures*. in « Communications of the ACM », numéro 6, volume 45, Juin, 2002.
- [13] A. ZARRAS, V. ISSARNY. *Concurrency in Dependable Systems*. Kluwer Academic Publishers, Juin, 2002, chapitre Quality Analysis of Enterprise Information Systems.

Communications à des congrès, colloques, etc.

- [14] M. BOULKENAFED, V. ISSARNY. *Coherency Management in Ad hoc Group Communication*. in « Proc. of the Workshop on Software Infrastructures for Component-Based Applications on Consumer Devices », Lausanne, Suisse, Septembre, 2002.
- [15] T. HIGUERA, V. ISSARNY. *Analyzing the Performance of Memory Management in RTSJ*. in « Proc. of ISORC 2002 - The 5th IEEE International Symposium on Object-oriented Real-time Distributed Computing », Washington DC, USA, Mai, 2002.
- [16] V. ISSARNY, D. SACCHETTI, F. TARTANOGLU, F. SAILHAN. *Développement de systèmes pour l'intelligence ambiante : une solution basée sur les services Web*. in « Proc. of the 16ème Congrès DNAC : De Nouvelles Architectures pour les Communications », Paris, France, Décembre, 2002.

- [17] F. SAILHAN, V. ISSARNY. *Energy-aware Web Caching for Mobile Terminals*. in « Proc. of the ICDCS Workshop on Web Caching Systems », Vienne, Autriche, Juillet, 2002.
- [18] F. SAILHAN, V. ISSARNY. *Cooperative Caching in Ad Hoc Networks*. in « Proc. of the 4th International Conference on Mobile Data Management », Melbourne, Australie, Janvier, 2003.
- [19] F. TARTANOGLU, V. ISSARNY, A. ROMANOVSKY, N. LEVY. *Dependability in the Web Service Architecture*. in « Proc. of the ICSE Workshop on Architecting Dependable Systems », Orlando, USA, Mai, 2002.

Rapports de recherche et publications internes

- [20] F. TARTANOGLU, V. ISSARNY, A. ROMANOVSKY, N. LEVY. *Coordinated Forward Error Recovery for Web Services*. rapport technique, numéro CS-TR-786, University of Newcastle upon Tyne, Décembre, 2002.

Divers

- [21] *Demonstration Support for Architectural Design for Dependable SoSs*. Rapport de contrat DSoS, Octobre, 2002.
- [22] *Dependable Composition of Web Services*. Rapport de contrat DSoS, Octobre, 2002.
- [23] *Detailed Description of the Ozone Away Demonstrator Scenarios and Implementation*. Rapport de contrat OZONE, Novembre, 2002.
- [24] *Software Environment Layer : Design and Implementation Plan*. Rapport de contrat OZONE, Novembre, 2002.
- [25] *Detailed Design of the Ozone Distributed Middleware Infrastructure*. Rapport de contrat OZONE, Novembre, 2002.
- [26] *Ozone Specification*. Rapport de contrat OZONE, Juin, 2002.
- [27] *VIVIAN Persistent Storage Service*. Rapport de contrat VIVIAN, Septembre, 2002.