

**DOSSIER DE CANDIDATURE
À LA DÉLÉGATION AU SEIN DE L'INRIA
POUR L'ANNÉE 2007
(RENOUVELLEMENT)**

Candidat : Pascal POIZAT

Maître de conférences, UFR SFA, Université d'Évry Val d'Essonne

**Programme de recherche :
Coordination et adaptation de services
dans le cadre des architectures logicielles
dédiées à l'informatique diffuse et mobile**

**Équipe-projet d'accueil souhaitée :
ARLES, UR de Rocquencourt**

CONTENU

Ce dossier de candidature comprend les éléments suivants :

– une lettre de motivation	4
– une fiche individuelle de renseignements	5
– un <i>curriculum vitæ</i>	6
synthèse de la candidature avec entre autres :	
– un résumé de mes activités de recherche	6
– un résumé de mon programme de recherche en délégation	7
– un programme de recherche détaillé	13
– le détail des travaux effectués à l'INRIA depuis mon arrivée (09/2006–02/2007)	17
– une liste complète de publications	18
– une lettre du chef de projet d'accueil souhaité	23
– une lettre d'autorisation de mon établissement (Université d'Évry)	24
– des lettres de recommandation	25

DÉCLARATION DE CANDIDATURE

Je soussigné POIZAT Pascal
déclare présenter ma candidature à la délégation (renouvellement) à l'INRIA pour l'année 2007.

Mon programme de recherche s'intitule

Coordination et adaptation de services
dans le cadre des architectures logicielles dédiées à l'informatique diffuse et mobile

En cas d'accord, je demande à effectuer cette délégation au sein du projet ARLES de l'Unité de Recherche de Rocquencourt.

à Rocquencourt, le 15 février 2007

LETTRE DE MOTIVATION

Mes thématiques de recherche portent sur l'application des méthodes formelles au génie logiciel, et plus particulièrement sur deux points : d'un côté les techniques permettant la structuration des spécifications de systèmes et de l'autre les liens entre ces techniques formelles et le produit final, à savoir le logiciel. Cette dualité permet selon moi d'améliorer l'utilisation de ces techniques hors du milieu académique. La structuration, promeut le *diviser pour régner*, elle permet une meilleure séparation des préoccupations et donc un meilleur passage à l'échelle des techniques formelles. Le développement de processus formels complets, automatiques ou semi-automatiques, allant de l'analyse des besoins au développement et à la réutilisation de code permet quant à lui l'utilisation de ces techniques dans un cadre plus industriel.

D'abord axée sur les problèmes liés à l'intégration de types de données dans les spécifications comportementales, mon activité de recherche concerne depuis plusieurs années la définition de techniques formelles dédiées aux systèmes communicants et aux architectures logicielles à base de composants. Je m'intéresse plus particulièrement à de nouveaux moyens permettant d'augmenter la réutilisabilité et la compatibilité des entités constituant les systèmes logiciels : la coordination et l'adaptation.

Dans ce cadre, j'ai mis en place en 2002 un groupe de travail au sein de mon laboratoire sur les objets, les composants et les architectures logicielles. J'organise aussi depuis 2004 un atelier international sur la coordination et l'adaptation d'entités logicielles, WCAT, avec des collègues européens. Je fais partie de plusieurs comités de programme de conférences et ateliers autour de ces thèmes.

Mon objectif est de développer ces thématiques de coordination et d'adaptation qui permettent de mettre en œuvre la complémentarité entre méthodes formelles et génie logiciel et ce, en les confrontant à un domaine d'application concret et qui constitue un domaine de recherche stratégique, celui des systèmes ubiquitaires et de l'informatique diffuse.

Cette complémentarité et ce domaine d'application privilégié se retrouvent dans les problématiques développées par le projet ARLES qui s'intéresse, dans le cadre des systèmes distribués, au développement d'architectures à base de services et de techniques formelles dédiées à leur conception, ainsi qu'à leur validation par la réalisation de solutions logicielles (intergiciels) permettant la mise en œuvre effective et efficace de moyens d'inter-opérabilité et de construction par composition.

Depuis septembre 2006 je suis en délégation au sein du projet ARLES. Si l'utilité des techniques de coordination et d'adaptation pour l'amélioration des mécanismes d'inter-opérabilité et de composition est claire, nous avons pu détecter plusieurs verrous à leur application dans le cadre de l'informatique diffuse. Nous avons déjà proposé plusieurs solutions qui ont permis de lever certains de ces verrous. Nous sommes actuellement en train de rédiger un projet d'ANR Jeunes-chercheurs (ARLES, LAMSADE/Univ. Paris Dauphine, IBISC/Univ. Evry) sur les thèmes de la composition dynamique automatique de services.

Je suis en poste dans une université assez jeune, ce qui impose des contraintes en termes de responsabilités administratives et d'heures complémentaires, et limite les possibilités de développement de mes thèmes de recherche. Le renouvellement de ma délégation au sein du projet ARLES me permettra de poursuivre nos travaux sur l'application des techniques de coordination et d'adaptation, en proposant à moyen terme des solutions formelles mais aussi leur prise en compte au sein d'outils de conception de systèmes distribués et d'intergiciels dédiés aux systèmes ubiquitaires et à l'informatique diffuse.

Je compte par ailleurs, à terme, tirer profit de ce renouvellement pour rédiger une habilitation à diriger des recherches sur ces problématiques de coordination et d'adaptation de composants dans le cadre des architectures logicielles.

FICHE INDIVIDUELLE DE RENSEIGNEMENTS

Nom : POIZAT Prénom : Pascal
 Date et lieu de naissance : 11/08/1972, Amboise (France)
 Nationalité : Française Sexe : M
 Adresse postale : INRIA Rocquencourt
 Projet ARLES
 Domaine de Voluceau - Rocquencourt
 78153 Le Chesnay Cedex, France
 N° de téléphone : (+33/0) 1 39 63 59 69
 N° de télécopie : (+33/0) 1 39 63 54 00
 Adresse électronique : pascal.poizat@inria.fr
 Page Web : <http://www.ibisc.univ-evry.fr/~poizat/>

DIPLÔMES

Doctorat :

- Doctorat de l'Université de Nantes, spécialité Informatique (mention très honorable) soutenu le 20 décembre 2000
 jury : F. Benhamou (Pr, Univ. Nantes, Président), M. Heisel (Pr, Univ. Magdebourg, Rapportrice), G. Reggio (Pr, Univ. Gênes, Rapportrice), C. Choppy (Pr, Univ. Paris XIII, Directrice), M. Lemoine (IR, ONERA Toulouse, Examineur), J.-C. Royer (MCF, Univ. Nantes, Co-encadrant).

Autres diplômes (à partir du niveau maîtrise) :

- DEA Informatique, Université de Nantes (mention bien)
 obtenu en juin 1995
- Maîtrise Informatique, Université de Nantes (mention assez bien)
 obtenue en juin 1994

SITUATION PROFESSIONNELLE ACTUELLE

Statut et fonction : Maître de conférences, classe normale, 4ème échelon
 Établissement : UFR SFA, Université d'Évry Val d'Essonne (Évry - France)
 Laboratoire : IBISC FRE CNRS 2873 (fusion du LaMI et du LSC en janvier 2006)
 Date d'entrée en fonction : 1er septembre 2001

FORMATION ET PARCOURS DEPUIS LE DEA

ÉTABLISSEMENTS	FONCTIONS ET STATUTS	DATES		OBSERVATIONS
		d'entrée en fonction	de cessation de fonction	
INRIA Rocquencourt	Délégation	01/09/2006	31/08/2007	Projet ARLES 4ème échelon ATER plein
Université d'Évry	Maître de conférences, classe normale	01/09/2001	(en cours)	
IUT de Nantes	ATER	01/10/2000	31/08/2001	
Université de Nantes	Doctorant allocataire moniteur	01/10/1997	30/09/2000	
CCAS de Couëron	Service National	15/01/1996	14/09/1997	

CV – SYNTHÈSE DE LA CANDIDATURE

Nom: POIZAT Prénom: Pascal
 Projet d'affectation souhaité : ARLES, UR de Rocquencourt

1. Résumé de l'activité de recherche

Je m'intéresse aux problématiques liées à la structuration des spécifications de systèmes en génie logiciel. Structurer c'est combiner le *diviser pour régner* et le *recomposer pour réutiliser*. L'objectif premier est un gain en productivité et en efficacité au niveau de la production de logiciel. La *structuration* est un terme générique qui recouvre aussi bien l'intégration de plusieurs caractéristiques (par exemple les données, les comportements, les contraintes temporelles) au sein d'une spécification globale que l'utilisation d'architectures logicielles et de composants. Ces notions ont comme beaucoup en génie logiciel d'abord vu le jour sous l'angle de la programmation : programmation à objets, à base de composants ou d'aspects – Aspect Oriented Programming (AOP). Elles sont actuellement étudiées à un niveau plus abstrait, dès la conception des systèmes : méta modèles et approches guidées par les modèles (Model Driven Architecture, MDA), langages de définition d'architectures (Architectural Description Languages, ADL) ou encore conception par aspects (Aspect Oriented Software Development, AOSD). Dans ce cadre, mes recherches ont porté sur :

- l'intégration de spécifications et les spécifications mixtes entre 1998 et 2003 ;
- les modèles formels d'architectures et de composants logiciels depuis 2001.

Intégration de spécifications et spécifications mixtes. Ce thème est celui développé dans ma thèse, soutenue en décembre 2000 [1]¹. Les systèmes informatiques peuvent en général être décomposés en plusieurs aspects pour tenter de répondre au problème de complexité croissante qui les caractérise. Les *spécifications mixtes* ont pour objectif de prendre en compte plusieurs de ces aspects comme la dynamique (les comportements), la statique (les données) et la structuration (la communication, la distribution). La notion de *spécification hétérogène* apparaît lorsque plusieurs langages sont utilisés pour décrire les différents aspects, ce qui est rendu indispensable par l'absence de langage universel. Dans les deux cas, le problème posé est alors de définir une sémantique globale au *recollement* de ces aspects [29].

Dans ce contexte, je me suis plus particulièrement intéressé au cours de ces dernières années à l'intégration de types de données dans les systèmes de transitions et plus particulièrement ceux d'UML (Unified Modeling Language). Plusieurs sémantiques [29,7] et des prototypes d'outils [12,7] ont été définis. Je me suis aussi intéressé à la vérification des spécifications par traduction vers des langages dédiés, comme LOTOS ou SDL. Un point important de cette thématique est pour moi que les méthodes utilisées puissent l'être par le plus grand nombre, et servir de support à la définition d'outils ou de prototypes d'outils et à l'apprentissage de ces méthodes dans le cadre de l'enseignement de la conception et du développement de systèmes complexes comme par exemple les systèmes distribués communicants. C'est pourquoi je me suis focalisé sur l'utilisation de *sémantiques opérationnelles*, plus facilement outillables. Par ailleurs, je me suis intéressé aux aspects en amont (analyse des besoins, liens avec UML) [32,31,27] et en aval (génération de code) [31,18] de cette conception, cherchant en cela à proposer une intégration de mes recherches dans un processus plus global de génie logiciel. Les modèles développés dans le cadre de cette thématique (Systèmes de Transitions Symboliques –STS, "glu" de recollement) sont au cœur de la thématique suivante qui a été dès le départ un champ privilégié d'application.

Modèles formels d'architectures et de composants logiciels. Cette thématique est actuellement mon thème de recherche principal. Son objectif est la définition de modèles et langages de composants formellement bien définis. J'ai d'abord abordé la vérification de systèmes de composants communiquant de façon asynchrone [21], les techniques de vérification par abstraction des STS [17] (dans l'objectif d'éviter les problèmes d'explosion du nombre d'états) ainsi que l'exécutabilité et l'implantation de modèles à base de STS [18]. Une proposition d'ADL formel intégrant les concepts de "vues" (aspects), de recollement externe par glu et de vérification par abstraction a récemment aussi été faite [6].

Je me suis aussi intéressé aux mécanismes de *coordination* de composants équipés d'interfaces comportementales (Behavioural Interface Description Languages, BIDL). L'intérêt de la coordination est de

¹Les références bibliographiques de ce document correspondent à ma liste de publications, disponible page 18.

s'abstraire des détails concernant les entités du système et de s'intéresser de façon séparée aux aspects de communication et d'interaction. D'un point de vue plus concret, cela signifie étudier les *connecteurs* indépendamment des composants au sein des architectures logicielles. La coordination peut aussi être vue comme l'étude, au niveau conceptuel, des mécanismes de fonctionnement des intergiciels. Mon travail se place à ce niveau dans l'optique de comparer différents moyens de description de la coordination. Cette étude comparative pourrait permettre, à moyen terme, d'envisager la définition de mécanismes d'interopérabilité pour intergiciels fonctionnant selon différentes techniques (ex : espaces de tuples, approches événementielles). Cela passe par la définition de "ponts" entre ces techniques, c'est-à-dire de mécanismes coordinateurs suffisamment expressifs. Un travail autour de l'expressivité de vecteurs de synchronisation étendus est présenté dans [19], où nous utilisons ces vecteurs pour coordonner les systèmes de diagrammes d'états étendus définis dans [23].

Je m'intéresse enfin, depuis plusieurs années, au problème de l'*adaptation logicielle* [11,2-5]. L'adaptation logicielle vise à résoudre les problèmes d'incompatibilité au sein des architectures de composants par la construction automatique d'entités (adaptateurs) permettant de compenser ces problèmes. Cette technique est au cœur de mon projet de délégation et est donc plus détaillée dans la suite [page 13].

Nous avons proposé dans [16] une technique d'adaptation à base de correspondances d'interfaces comportementales qui améliorerait les principales approches du domaine. Cette technique s'applique à un nombre quelconque de composants mais procède par calcul d'un adaptateur global sur un système clos (les composants sont fixés). Des premiers résultats permettant l'adaptation incrémentale, dans le but d'éviter le calcul d'un adaptateur global, ont été présentés dans [15]. Depuis mon arrivée au sein du projet ARLES, je me suis intéressé à la réduction de la taille des adaptateurs [53], à la définition de mécanismes d'adaptation qui fonctionnent sur des systèmes ouverts [52], et à la prise en compte d'informations sémantiques dans le processus d'adaptation [51]. Tous ces éléments sont rendus indispensables par le domaine d'application privilégié, celui de l'informatique diffuse. Ces résultats sont détaillés dans la présentation de mes travaux au sein du projet ARLES [page 17].

Conjointement à ces recherches, je co-organise depuis 2004 un atelier international sur la coordination et l'adaptation d'entités logicielles à la conférence ECOOP [2,4,5].

2. Résumé du programme de recherche

Mon programme de recherche s'intitule "*Coordination et adaptation de services dans le cadre des architectures logicielles dédiées à l'informatique diffuse et mobile*". Il s'inscrit dans deux des défis scientifiques prioritaires de l'INRIA, à savoir, "concevoir et maîtriser les futures infrastructures des réseaux et des services de communication" et "garantir la fiabilité et la sécurité des systèmes à logiciel prépondérant", ainsi que dans le premier objectif prioritaire de l'UR de Rocquencourt, "concevoir et maîtriser les réseaux et les systèmes, traiter l'information distribuée".

Le développement de *informatique ubiquitaire* et des périphériques mobiles type PDA (Personal Digital Assistant, assistant numérique personnel), ainsi que celui des *services logiciels* à la base de nouvelles applications, rend nécessaire le développement de nouvelles architectures matérielles mais surtout logicielles permettant l'échange et la coordination de services. Le génie logiciel à base de composants a su de son côté démontrer son intérêt dans le cadre d'une informatique plus axée systèmes d'information et doit maintenant s'attacher à développer les moyens de conception et de sécurité de l'informatique mobile.

Le caractère spécifique de cette dernière, et en particulier son aspect *dynamique et distribué*, rend indispensable le développement de moyens d'analyse de composants ou services distribués qui devront pouvoir être corrigés de façon dynamique et automatique lorsque des problèmes d'interopérabilité sont détectés. C'est tout l'objet de *l'adaptation logicielle* [11], qui se différencie en cela de la maintenance ou évolution et de la personnalisation ou paramétrage. Étant donné un "*mapping*" (*i.e.*, une description abstraite de propriétés de correspondance entre composants ou entre un composant et son contexte d'utilisation) l'adaptation consiste (i) à détecter si l'assemblage des composants est exempt d'incompatibilités et respecte les contraintes spécifiées au niveau du mapping, et (ii) le cas échéant à obtenir de façon automatique une pièce logicielle assurant la compatibilité d'usage entre les composants. Par ailleurs, en amont, l'informatique distribuée rend indispensable la définition de nouveaux moyens de conception d'applications basées sur des services distribués. La problématique principale dans ce cadre est actuellement celui de la coordination de services de façon dynamique.

Dans le cadre de ma délégation au sein du projet ARLES, je désire étudier l'apport des techniques de coordination et d'adaptation automatiques aux architectures à base de services en environnement mobile. Je me base pour cela, entre autres, sur une combinaison des approches à base de *composition sémantique* développées dans ARLES et mes résultats en termes d'adaptation.

3. Sélection commentée de publications

La liste complète des mes publications pourra être trouvée page 18.

– Thématique modèles formels d’architectures et de composants logiciels :

- [16] C. Canal, **P. Poizat** and G. Salaün. Synchronizing Behavioural Mismatch in Software Composition. Proc. of the International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS’2006), LNCS 4037, pages 63–77, 2006.
Cet article présente des techniques d’adaptation logicielle basée modèle qui permettent de construire automatiquement à partir de contrats abstraits d’adaptation des modèles d’adaptateurs assurant la correction comportementale (non blocage, respect du contrat) d’assemblages de composants. Un outil, *Adaptor*, implante ces techniques.
- [17] **P. Poizat**, J.-C. Royer and G. Salaün. Bounded Analysis and Decomposition for Behavioural Descriptions of Components. Proc. of the International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS’2006), LNCS 4037, pages 33–47, 2006.
Cet article présente des techniques permettant d’éviter le problème de l’explosion du nombre d’états lors de la vérification de systèmes de composants décrits par des modèles à base de STS. Ces techniques permettent d’abstraire des parties du système et rendent alors possible la vérification de propriétés.
- [18] S. Pavel, J. Noyé, **P. Poizat** and J.-C. Royer. A Java Implementation of a Component Model with Explicit Symbolic Protocols. Proc. of the International Workshop on Software Composition (SC’2005), LNCS 3628, pages 115–124, 2005.
Cet article présente un modèle hiérarchique de composants logiciels basé sur les STS et son implémentation en Java. Son objectif est de remplir le fossé entre les modèles conceptuels de composants et les langages de programmation basés composants. Il s’agit d’une approche de type *container* qui se base sur l’utilisation d’un contrôleur de communications paramétré par un STS.
- [21] O. Maréchal, **P. Poizat** and J.-C. Royer. Checking Asynchronously Communicating Components using Symbolic Transition Systems. Proc. of the International Symposium on Distributed Objects and Applications (DOA’2004), LNCS 3291, pages 1502–1519, 2004.
Cet article s’intéresse au problème de la disponibilité des composants sous l’angle du test du caractère borné ou non des files d’attente en mode de communication asynchrone (un dépassement de la capacité de ces files pouvant conduire à un déni de service). Il propose une technique d’abstraction des files d’attente et montre un certain nombre de bonnes propriétés de la simulation fournie par cette abstraction.

– Thématique intégration de spécifications et spécifications mixtes :

- [7] C. Attiogbé, **P. Poizat** and G. Salaün. A Formal and Tool-Equipped Approach for the Integration of State Diagrams and Formal Datatypes. IEEE Transactions on Software Engineering, 33(3), pages 157–170, 2007.
Cet article formalise une démarche générique d’intégration de données formelles (Z, B ou spécifications algébriques) dans les diagrammes d’états (ex : UML). Un prototype, *xCLAP*, a servi à valider l’approche opérationnelle choisie.
- [12] C. Choppy, **P. Poizat** and J.-C. Royer. The Korrigan Environment. J.UCS 7(1), pages 19–36, 2001.
Cet article présente *Korrigan*, un environnement logiciel dédié aux spécifications mixtes (dont : méta modèles, principes de traduction et de génération de code) développé pendant ma thèse.
- [29] C. Choppy, **P. Poizat** and J.-C. Royer. A Global Semantics for Views. Proc. of the International Conference on Algebraic Methodology and Software Technology (AMAST’2000), LNCS 1816, pages 165–180, 2000.
Cet article présente l’idée de *glu* logique comme moyen expressif de description du recollement et de la communication dans les spécifications mixtes. Une sémantique opérationnelle y est présentée.
- [31] **P. Poizat**, C. Choppy and J.-C. Royer. From Informal Requirements to COOP : a Concurrent Automata Approach. Proc. of the World Congress on Formal Methods in the Development of Computing Systems (FM’99), LNCS 1709, pages 939–962, 1999.
Cet article présente une démarche de conception de spécifications mixtes données/comportements et les mécanismes de traduction de ces spécifications, soit à des fins de vérification (LOTOS, SDL) soit à des fins de prototypage (code Active Java).

4. Réalisation et diffusion de logiciels

- Korrigan [12] (1998–2000)
Cet environnement, écrit en python, permet la traduction du langage développé dans le cadre de ma thèse vers divers formalismes cibles à des fins de vérification (comme LOTOS pour pouvoir utiliser la boîte à outils CADP développée par le projet VASY de l'INRIA Rhône-Alpes). L'environnement permet aussi la génération de prototypes dans un langage concurrent orienté objet, Active Java. Le noyau de cet environnement est fait d'une librairie de classes (réification) de spécifications et d'un module de spécification dynamique, CLAP.
- CLAP [12] (1999) et xCLAP [7] (2004)
Cette librairie de classes, écrite en python (\approx 15000 lignes) permet de définir tout type de système basé états-transitions (ex : LTS, STS). Différents algorithmes génériques (ex : produit/recollement, atteignabilité) ont été implantés. Des sorties en Xfig (réification du format de cet outil) ou dot/graphviz sont possibles. Cet outil a servi de base à une extension permettant l'animation de diagrammes d'états d'UML étendus avec des données formelles (Z, Spéc. algébriques), xCLAP [7]. CLAP a aussi servi récemment de base aux travaux menés en collaboration avec l'École des Mines de Nantes sur la vérification de STS (outil SyCLAP). CLAP et xCLAP sont disponibles librement sur le Web : <http://www.ibisc.univ-evry.fr/~poizat/outils-fr.php>.
- Eclipse Transition Systems, ETS [55] (2005)
Ce plugin Eclipse écrit en java est un portage et une refonte de CLAP dans le cadre du projet RNRT STACS [page 9] qui permet la description et le produit de systèmes de transitions (LTS ou STS). Ses points-clés sont un langage de composition plus expressif que celui de CLAP, en relation avec mes travaux sur la coordination d'entités logicielles [19], et la conservation de l'information de composition au sein des STS, utile pour développer des techniques compositionnelles de test orienté modèle (projection des objectifs de test sur les composants du système).
- Adaptor [50] (2006–2007)
Cet outil, écrit en python (\approx 9000 lignes) implante les mécanismes d'adaptation logicielle développés dans le cadre de mes recherches sur ce thème [16]. Adaptor dispose d'une interface graphique qui rend transparentes les interactions avec les outils TINA (LAAS, Toulouse) et CADP (projet VASY, INRIA Rhône-Alpes). Sa première version fonctionnait sur les systèmes clos. La dernière distribution d'Adaptor (janvier 2007) prend en compte les résultats de mes travaux depuis mon arrivée dans le projet ARLES et supporte l'adaptation incrémentale de systèmes ouverts [52]. La distribution d'Adaptor inclut une base de validation d'approximativement 70 exemples (25000 lignes de XML) et est disponible sous licence LGPL depuis : <http://www.ibisc.univ-evry.fr/Members/Poizat/Adaptor>.
La première version d'Adaptor a fait l'objet d'une démonstration à la conférence FMOODS'06 en marge de la présentation de [16].
- Préalablement, pendant mon DEA j'ai réalisé un interpréteur de langage orienté objet fonctionnant à base de réécriture conditionnelle positive (un moteur de réécriture a été entièrement programmé pour cela). Cet interpréteur est implanté au dessus du langage CLOS.

5. Valorisation et transfert technologique

J'ai été co-rédacteur et participant d'un projet RNRT labellisé lors de l'appel 2002 [12/2003-12/2006] portant sur le test orienté modèle et la spécification, abstraits et compositionnels, de systèmes (STACS). Les principaux verrous étaient le caractère hétérogène des spécifications et surtout leur taille rendant nécessaire l'utilisation de systèmes symboliques (STS). Ce projet a donné lieu à des collaborations entre IBISC, le CEA Saclay, Thalès Communications et Ligeron S.A. Dans ce cadre, je me suis intéressé principalement à la définition de STS avec conservation de la structure de composition [55]. Je suis co-rédacteur et porteur d'un projet ANR Jeunes-chercheurs prévu pour l'appel 2007 [page 17].

6. Encadrement d'activités de recherche

- **encadrement de thèses :**
 - (2006-) encadrement (30%) avec P. Le Gall (Pr) et Marc Aiguier (MCF) de la thèse de M. Mabrouki (Univ. Évry) portant sur la coordination d'entités dans le cadre de systèmes biologiques complexes.
- **encadrements de DEA :**
 - (Février 2006 -) encadrement (50%) avec S. Ben Mokhtar (Doctorante projet ARLES) du M2 Recherche de S. Beauche (Paris 6). Il s'agit d'étudier l'application de l'adaptation logicielle aux architectures à base de services en étudiant le problème de l'adaptation entre services décrits selon

différents niveaux d'abstraction (OWL-S/BPEL).

- (2005) encadrement (100%) du DEA d'O. El Gares (Univ. Évry) sur l'étude d'architectures de composants hétérogènes. Il s'agissait d'étudier l'un des verrous du projet RNRT STACS sur le caractère hétérogène (SDL, UML, Esterel) des spécifications du partenaire industriel. L'approche a consisté à étudier les possibilités d'un encodage vers un langage unique, LOTOS.
- (2004) encadrement (60%) avec P. Le Gall (Pr) du DEA de R. Layadi (Univ. Évry) sur l'applicabilité des algèbres de processus à la définition d'ADL permettant la prise en compte de communications synchrones et asynchrones. Une première version d'algèbre à été proposée (syntaxe et sémantique opérationnelle).
- (2004) encadrement (90%) avec J.-C. Royer (Pr, École des Mines de Nantes) du DEA d'A. Sedkaoui (Univ. Évry). Ce stage a permis de proposer une implantation en Java d'un modèle d'ADL et de composants dirigés par des protocoles explicites décrits sous forme de STS.
- **encadrements de niveau maîtrise (sélection) :**
 - (1999) encadrement (50%) d'un projet (Travail d'Étude et de Recherche, TER) de 4 étudiants, CLAP, une librairie de classes pour systèmes de transitions très générique. Elle permet différentes opérations (ex : produits, atteignabilité) sur tout type de systèmes (LTS, STS). Ce projet a depuis fait l'objet de plusieurs extensions dans le cadre de prototypes de recherche. L'un des étudiants, G. Salaiïn, a fait une thèse et est actuellement Assistant de Recherche à l'Université de Málaga ;
 - (2005) encadrement d'un projet (TER) puis d'un stage d'été (2 étudiants, financement RNRT) portant sur le portage et la refonte de CLAP en tant que *plugin* pour la plate-forme Eclipse (ETS).
 - (2006) encadrement d'un projet (TER) portant sur l'implantation et le développement d'une interface graphique d'un outil pour la conception d'adaptateurs de systèmes à base de composants (Adaptor). L'un des étudiants de ce TER, S. Beauche est actuellement en M2 Recherche à Paris 6 et commence un stage sur l'adaptation d'architectures à base de services dans le projet ARLES.

Pour une description exhaustive (–03/2005), voir le dossier type maître de conférences disponible ici :

<http://www.ibisc.univ-evry.fr/~poizat/recherche-fr.php>.

7. Enseignement

J'ai enseigné, hormis les interventions entrant dans le cadre d'un contrat de formation avec la Poste et celles entrant dans celui de mon service national, successivement dans le cadre du Département Informatique de l'UFR des Sciences et des Techniques de l'Université de Nantes (moniteur, 64 h/an), dans celui du Département Informatique de l'IUT de Nantes (ATER, 192 h/an) et depuis 2001 dans celui de l'UFR Sciences Fondamentales et Appliquées de l'Université d'Évry Val d'Essonne (maître de conférences, entre 195 et 245 h/an pour une moyenne de 210 h/an). J'ai enseigné en premier, en second et en troisième cycle, tant dans des filières recherche que professionnelles. Je donne ici les éléments clés de mes enseignements. Une description plus détaillée (contenus des cours, volumes, etc.) pourra être trouvée dans un dossier type maître de conférences (–03/2005) disponible ici :

<http://www.ibisc.univ-evry.fr/~poizat/recherche-fr.php>.

Des supports pour les cours sont disponibles depuis mes pages Web enseignements :

<http://www.ibisc.univ-evry.fr/~poizat/enseignement-fr.php>.

Mes enseignements (cours, TD ou TP) ont concerné quatre grands thèmes : (i) réseaux, architecture et systèmes, (ii) génie logiciel et spécifications, (iii) bases de données et publication de l'information et (iv) bases de l'informatique.

1. Réseaux, Architecture et Systèmes : internet et applications Web, client-serveur et aspects avancés de Java, administration système, protocoles et services, architecture des ordinateurs, systèmes d'exploitation ;
2. Génie logiciel et spécifications : combinaison et intégration de langages de spécification, spécifications dynamiques, UML ;
3. Bases de données et publication de l'information : bases de données, communication (sites Web), bureautique ;
4. Bases de l'informatique : algorithmique et langages impératifs (C, Pascal, python), langages fonctionnels, langages orientés objet (java), graphes et recherche opérationnelle.

Au cours de ces années, j'ai eu la responsabilité des cours magistraux suivants :

- 2003-2006, co-responsable de l'option "Combinaison et Intégration de Langages de Spécification" en DEA/M2 Informatique : de 7 à 9h de cours selon les années sur la prise en compte de données dans la spécification de comportements dynamiques et sur l'algèbre de processus LOTOS ;

- 2002-2006, co-responsable (responsable sur 2004-2006) du cours "Génie Logiciel" en Maîtrise/M1 Informatique : 18h de cours et 18h de TD/TP sur les algèbres de processus et la logique temporelle. Ce cours a aussi été proposé dans le cadre de la formation doctorale en 2005-2006 ;
- 2002-2006, responsable du cours "Bases de données" en DESS/M2 Compétences Complémentaires en Informatique (niveau L3) : 12h de cours et 18h de TD/TP sur les systèmes de gestion de bases de données, SQL, la conception de schémas relationnels, ... ;
- 2002-2004, responsable du cours "SDL" en DESS/M2 Génie Logiciel, Économie, Droit et Normes (enseignant extérieur, Université de Nantes) : 3h de cours et 3h de TD sur le langage de spécification SDL ;
- 2001-2006, responsable du cours "Internet et applications" en DESS/M2 Compétences Complémentaires en Informatique (niveau L3) : 15h de cours et 18h de TP sur apache, HTML, javascript, PHP, JSP, ... ;
- 2001-2005, responsable du cours "Unix et client-serveur" en DESS/M2 Compétences Complémentaires en Informatique (niveau L3) : 15h de cours et 18h de TP sur Unix, les scripts shell et la programmation client-serveur avec java (threads et sockets) ;
- 2001-2002, responsable du cours "Réseaux" en DESS/M2 Compétences Complémentaires en Informatique (niveau L3) : 15h de cours et 21h de TD sur les couches et protocoles réseaux ;
- 2001-2003, responsable du cours "Mise à niveau Unix" en Licence MIAGE : 12h de cours et 18h de TP sur Unix, les scripts shell et les bases de l'administration d'une machine sous Unix ;
- 2000-2001, responsable du cours "Spécifications Formelles 2" en Année Post Premier Cycle (niveau L2, IUT de Nantes) : 8h de cours et 24h de TD sur les réseaux de Petri, algèbres de processus, automates et Statecharts ;
- 2000-2001, responsable du cours "Réseaux - Administration - Services" en Année Post Premier Cycle (niveau L2, IUT de Nantes) : 10h de cours et 20h de TP sur l'installation, la configuration et l'administration de services réseau.

8. Diffusion de l'information scientifique

- co-rédaction en 2006 d'un chapitre sur les langages de description d'architectures dans un livre sur les méthodes formelles pour les systèmes répartis et coopératifs (public : ingénieurs et étudiants en Master) ;
- rédaction en 2006 d'un chapitre de livre sur le langage formel SDL (public : industriels et étudiants en Master) ;
- co-rédaction en 2006 d'un glossaire sur les méthodes formelles (46 pages) dans un livre portant sur les méthodes formelles (public : industriels et étudiants en Master) ;
- présentation de l'apport des langages de script (support : python) pour la programmation scientifique lors d'un colloque de physique (International Workshop on Computing for Heavy Ion Physics) organisé en 2005 ;
- animation d'un atelier "De la recherche au jeu" lors des journées nationales "La semaine de la science" en 1998 (aspects présentés : programmation concurrente et intelligence artificielle).

9. Mobilité

- mobilité thématique entre mon sujet de DEA (programmation objet à base de réécriture) et celui de thèse (intégration de spécifications formelles) et depuis quelques années vers les composants formels et l'adaptation ;
- mobilité géographique entre ma thèse (effectuée à l'IRIN, Nantes) et mon poste à l'Université d'Évry ;
- mon statut et le fait que l'Université d'Évry soit jeune impliquent un certain nombre de contraintes au niveau des heures supplémentaires et des charges administratives, ce qui rend plus difficile la mobilité en cours d'année (hors délégation ou CRCT). Conscient toutefois de son importance, j'ai pu réaliser de courts séjours (2 semaines) au sein des Universités de Málaga et d'Extremadure en juin 2005. Un séjour à l'Université de Málaga est prévu en 2007.

10. Responsabilités collectives

- **recherche/laboratoire :**
 - membre extérieur titulaire de la Commission de Spécialistes d'Établissement (CSE, commission chargée du recrutement des enseignants-chercheurs) 27ème section (informatique) du CNAM Paris et membre suppléant de la CSE 27ème section de l'Université d'Évry ;

- membre élu du conseil du laboratoire IBISC ;
- membre des commissions Web et locaux d'IBISC (2001-2006), webmaster d'IBISC (2001-2006), administrateur du système de gestion de bases de données d'IBISC (2001-2006) ;
- co-organisation des ateliers WCAT (International Workshop on Coordination and Adaptation Techniques for Software Entities) à ECOOP depuis 2004 et de l'atelier FOCLASA (International Workshop on the Foundations of Coordination Languages and Software Architectures) à CONCUR en 2007 ;
- membre du comité de programme de WCAT depuis 2004, d'OCM-SI (atelier sur les Objets, Composants et Modèles dans l'ingénierie des Systèmes d'Information) depuis 2005, de CAL (Conférence francophone sur les Architectures Logicielles) depuis 2006 et de FOCLASA (International Workshop on the Foundations of Coordination Languages and Software Architectures) depuis 2006 ;
- création et responsabilité d'un groupe "Objets, Composants et Modèles" (OCM) au laboratoire IBISC, participant aux pôles de même nom au sein des GDR I3 et ALP ;
- correspondant au laboratoire IBISC du groupe AFADL du GDR ALP ;
- relecteur régulier pour le journal Technique et Science Informatiques (TSI) depuis 2001, de numéros spéciaux des journaux SCP et Fundamentae Informatica en 2006, et de conférences tant dans le domaine du génie logiciel (FM'99, PSI'01, FASE'02 et FASE'07, Z'02 et Z'03, RENPAR'02, IC-FEM'02 et ICFEM'05, IFM'03, TFM'04, TACAS'05, ICTAC'05 et ICTAC'06, AFADL'06, FORTE'06, BDA'06, JISBD'06) que des composants et objets (TOOLS'00, TOOLS Europe'01, WCAT depuis 2004, SAC'05 et SAC'06, OCM-SI depuis 2005, CAL depuis 2006, FOCLASA depuis 2006).
- **pédagogie/enseignement :**
 - directeur adjoint (2001-2004) puis directeur (2004-2006) du DESS/M2 Compétences Complémentaires en Informatique ;
 - responsable du tutorat en informatique entre 2001 et 2003 (8 tuteurs/an, dotation : \approx 5300-6100 EUR/an (35-40 kF)) ;
 - représentant élu du collège "autres catégories de personnel enseignant" au conseil d'institut de l'IUP MIAGE de Nantes (1998-2000) ;
- **autres :**
 - membre fondateur, secrétaire puis trésorier de Login, l'association des jeunes chercheurs en informatique de Nantes (1998-2001) ;
 - co-responsable d'un contrat de formation Université de Nantes - La Poste en 2000, \approx 4300 EUR (28 kF).

Pour une description exhaustive (-03/2005), voir le dossier type maître de conférences disponible ici : <http://www.ibisc.univ-evry.fr/~poizat/recherche-fr.php>

11. Autres éléments

J'ai pu présenter mes travaux, en plus des présentations liées aux publications, lors de différents séminaires, en particulier :

- mes travaux sur l'adaptation ont été présentés à un séminaire du projet ARLES en janvier 2006, lors d'un séminaire invité de l'ACI FIACRE (Fiabilité des Assemblages de Composants Répartis : modèles et outils pour l'analyse de propriétés de sécurité et de sûreté) en février 2006, lors d'une réunion du groupe de travail MeFoSyLoMa (Paris 6, Paris 9, Paris 13, CNAM, ENST) en mai 2006, d'un séminaire à l'INRIA Rhône-Alpes (invitation par le projet VASY) en mai 2006 et dans plusieurs séminaires d'équipes du laboratoire IBISC ;
- mes travaux sur la coordination et sur l'intégration de spécifications et le lien avec UML ont été présentés lors de deux séminaires invités aux Universités de Málaga et d'Extremadure (Espagne) en 2005 et lors d'un séminaire invité dans le cadre du groupe de travail QSL du LORIA (Nancy) lors d'une journée "autour d'UML" en 2003.

J'entretiens des collaborations de recherche régulières avec C. Canal (TEU, Université de Málaga, Espagne), J.-C. Royer (Pr, projet OBASCO, École des Mines de Nantes/INRIA), G. Salaün (Posdoctorant, projet VASY, INRIA Rhône-Alpes puis Assistant de Recherche, Université de Málaga, Espagne) et M. Tivoli (postdoctorant projet POP ART, INRIA Rhône-Alpes puis Université de l'Aquila, Italie) sur la problématique des composants et de leur formalisation.

PROGRAMME DE RECHERCHE DÉTAILLÉ

Intitulé du programme de recherche :

**Coordination et adaptation de services
dans le cadre des architectures logicielles dédiées à l'informatique diffuse et mobile**

Présentation du contexte

Je présente tout d'abord le contexte d'application de mon programme de recherche. Il se situe dans le cadre de *l'informatique ubiquitaire et des systèmes ouverts*, défis majeurs pour l'industrie du logiciel qui, après la problématique du développement de gros systèmes ou logiciels se heurte maintenant au développement de nouvelles applications basées sur les technologies émergentes des composants, de l'Internet et, à l'intersection des deux, de services et de combinaisons de services Web. J'explique ensuite ce que peuvent apporter les approches basées sur l'utilisation de *langages de description d'architectures* et celles basées sur les *services* à ce développement. Je termine cette introduction en rappelant l'apport des techniques de génie logiciel, leurs limites dans le contexte présent, et en positionnant clairement mon programme de recherche dans une optique de *génie logiciel pour le développement de services en informatique diffuse et mobile*.

Dans un second temps, j'aborde plus en détails les thématiques que je désire développer dans mon programme de recherche : *la coordination et l'adaptation de services*. Complémentaires, elles peuvent selon moi apporter des solutions de génie logiciel aux nouvelles applications de l'informatique diffuse ou mobile, et permettent de répondre à deux grandes problématiques :

- le problème de passage à l'échelle, c'est-à-dire la difficulté d'application des méthodes formelles de conception et d'analyse sur des systèmes complexes, soit en raison du nombre de composants en jeu, soit en raison de l'utilisation conjointe de différents niveaux de description de ces derniers : comportement, données, contraintes temporelles. La coordination et l'adaptation favorisent la conception indépendante de petits systèmes qui sont ensuite coordonnés de façon exogène (hors du code des systèmes coordonnés) et si besoin adaptés pour en construire de plus grands ;
- la sécurité et l'inter-opérabilité des plate-formes, l'objectif de l'adaptation étant de corriger automatiquement les incompatibilités entre composants.

Les travaux que j'ai effectués au sein du projet ARLES depuis mon arrivée en septembre 2006 sont présentés page 17.

Informatique ubiquitaire et systèmes ouverts. La demande pour des systèmes logiciels de plus en plus complexes, supportant de nouveaux services, et pour des domaines d'application plus larges change la façon dont le logiciel est spécifié, conçu et construit. Les systèmes *plug-and-play*, le développement progressif du Web comme moyen global de communication et d'échange de services et d'informations, le développement des systèmes collaboratifs permettant à leurs utilisateurs distribués sur un réseau de collaborer à la réalisation de certaines tâches, et même le développement de l'utilisation des périphériques mobiles et des réseaux sans fil, conduisent à un scénario d'*informatique ubiquitaire* ou diffuse (*pervasive computing*). Ces exemples partagent des caractéristiques communes, mis à part leur complexité : ils consistent en un ensemble d'entités logicielles concurrentes et *interagissantes*, habituellement *distribuées* sur un réseau, pour lequel peu d'hypothèses sur l'homogénéité des nœuds, leur disponibilité à un moment donné, ou même sur les évolutions futures de l'architecture peuvent être faites. Tous ces systèmes sont des exemples de *systèmes ouverts*, par opposition aux systèmes clos dans lesquels l'ensemble des problématiques citées devaient être pensées, contrôlées et connues à l'avance.

Architectures logicielles et services. Différents *intergiciels* ont vu le jour dans le but de permettre le développement de systèmes ouverts au dessus d'un ensemble de fonctionnalités de base liées par exemple au déploiement des entités constituant le système ou encore à la gestion de la communication entre elles. Cela a conduit à une hétérogénéité de plate-formes rendant difficile leur intégration et leur inter-opérabilité, or cela s'avère crucial dans le cadre de systèmes comme ceux liés à *l'intelligence ambiante* où différentes entités sur différents types de réseaux sont amenées à collaborer pour rendre des services complexes à l'utilisateur. Une piste prometteuse quant à la résolution de ces problèmes est l'utilisation d'approches à base de langages de description d'architectures logicielles (ADL) et d'*approches orientées services*, et plus particulièrement celles basées sur leur instanciation la plus aboutie, les *services Web* (*Web Services*). Les ADL [8], fournissent un haut niveau d'abstraction des systèmes, en s'intéressant à leurs aspects principaux : *composants* représentant les entités constitutives du système, *connecteurs*

représentant une abstraction des mécanismes d'interaction entre les composants et donc des intergiciels, et *configurations* représentant les assemblages composants-connecteurs. Les ADL ont montré leur efficacité comme mécanisme de structuration et de vérification d'architectures (ADL de première génération : Wright, Rapide). Les services Web fournissent quant à eux un moyen universel de développer les applications distribuées. Ils sont donc particulièrement adéquats dans le cadre de l'informatique diffuse, de l'informatique mobile et des petits équipements à ressources limitées.

Génie logiciel. Le génie logiciel a fait ses preuves dans le cadre des systèmes d'information, même si dans le cas de l'utilisation de méthodes formelles le problème du *passage à l'échelle* se pose encore lors de la prise en compte d'informations riches comme l'utilisation conjointe de descriptions des comportements et des données (spécifications mixtes) ou bien encore de celle d'informations temporelles. La *structuration* des spécifications est un moyen de répondre à la complexité des systèmes tant au niveau de leur conception que de leur vérification lorsque leur sécurité est importante. Le concept de structuration a fait l'objet de nombreuses instanciations au cours du temps : modules, objets, composants et architectures logicielles, aspects. Le génie logiciel doit maintenant réagir aux nouveaux défis liés à l'informatique ubiquitaire et aux systèmes ouverts par la prise en compte de leur aspect dynamique et distribué. L'utilisation conjointe de langages architecturaux pour systèmes distribués et d'intergiciels présentant un haut niveau de généricité et d'inter-opérabilité est une approche prometteuse développée dans le projet ARLES. Cependant, si cette approche donne maintenant le support technique sur lequel s'appuieront de nouvelles applications (informatique mobile, intelligence ambiante), des problèmes restent à résoudre en amont, au niveau de la combinaison ou coordination des services, à la vérification de leur inter-opérabilité et, le cas échéant, à leur correction. Il va sans dire qu'en raison du domaine d'application, ces techniques se doivent d'être automatiques et applicables dans un contexte dynamique.

C'est dans le cadre de cette application de techniques de génie logiciel aux architectures logicielles dédiées à l'informatique diffuse et mobile, que je désire travailler en délégation au sein du projet ARLES de l'UR de Rocquencourt. Un point important est lié aux possibilités de validation des approches proposées par l'interaction constante entre les deux axes de ce projet : le développement d'ADL et de techniques pour la modélisation de systèmes distribués d'un côté et l'implantation de ces idées dans le cadre d'infrastructures logicielles (intergiciels) de l'autre.

Dans la suite de ce programme, je présente tout d'abord la partie concernant la coordination et dans un second temps celle concernant l'adaptation.

Problématiques liées à la coordination

L'approche à base de composants a permis de rendre explicites non seulement les services fournis par ces derniers mais aussi ceux requis, rendant ainsi complète la liste des interdépendances entre composants. Plusieurs problèmes restent cependant posés par la composition de composants. Le premier est celui des moyens techniques à mettre en œuvre pour la composition. La plupart des approches se basent sur l'existence d'un intergiciel offrant des mécanismes de communication de base aux composants. Ce modèle impose cependant sa prise en compte lors du codage du composant, directement (appels aux fonctions de communication dans le corps du composant) ou indirectement (processus de création automatique, à partir des interfaces des composants, de squelettes prenant en compte ces appels). Ces approches se caractérisent donc par la non séparation des préoccupations au sein des composants (rôle métier ou fonctionnel des composants vis à vis des primitives de communication et d'interaction entre composants) ce qui nuit à leur réutilisabilité. Un second problème est celui d'assurer la correspondance entre services requis d'un composant et fournis d'un autre lorsque les correspondances ne sont pas exactes.

L'un des domaines permettant la résolution de ces problèmes de composition est celui des *modèles et langages de coordination*. Leur objectif est de fournir des mécanismes et primitives pour spécifier les interactions entre entités logicielles indépendamment de leurs aspects calculatoires (rôle métier). Étant donné un ensemble d'entités, l'objectif de la coordination est de les faire interagir correctement. On retrouve cette problématique sous les concepts d'*orchestration* et de *choreographie* dans le domaine des services Web. Les modèles de coordination font abstraction de toute information sur les entités coordonnées, hormis la donnée de leur interface. Il est donc important, dans l'objectif d'utiliser ces techniques, que les composants soient équipés d'*interfaces comportementales* (Behavioural Interface Description Languages, BIDL). En effet, les interfaces actuellement utilisées dans les modèles de composants (EJB, CCM par exemple) ne permettent pas d'assurer la compatibilité entre eux hormis au niveau, limité, des noms et

arguments de leurs services. Le problème de l'inter-opérabilité entre composants peut en fait se poser à quatre niveaux. Le premier est celui des signatures des services. Comme nous l'avons signalé il s'agit du niveau actuel des interfaces de composants (au niveau industriel). Le second est celui des BIDL. De plus en plus de travaux académiques s'intéressent à leur expressivité et le problème de la détection des incompatibilités à ce niveau est maintenant bien établi. La prise en compte d'informations comportementales dans les services Web (via par exemple OWL-S, BPEL ou des extensions) est actuellement possible. Les deux niveaux d'inter-opérabilité supérieurs concernent la qualité de service (temps d'exécution, ressources nécessaires) et la sémantique (fonctionnalité).

Dans le cadre de l'informatique mobile, ces trois derniers niveaux sont les plus intéressants. Le niveau comportemental permet d'attaquer le problème de coordonner des services simples dans l'objectif de rendre des services plus complexes. Son objectif premier est d'assurer l'absence de blocages (compatibilité comportementale). À ce niveau, la coordination ne doit cependant pas se faire sans prendre en compte la sémantique des services, seule capable de faire correspondre de façon sensée services requis et fournis (la correspondance de nom n'étant pas une information suffisante). **Un premier axe de recherche (P-C1)** consiste à étudier la combinaison d'approches de composition/coordination au niveau comportemental avec les approches au niveau sémantique développées récemment dans le projet ARLES. Ceci pourrait passer par la définition d'un ADL équipé de protocoles de niveaux comportemental et sémantique. Formellement, les outils de base adéquats semblent être les algèbres de processus (expressives, abstraites) et plus précisément le π -calcul et ses dialectes, qui ont été appliqués avec succès dans le cadre de l'analyse des propriétés de mobilité et de disponibilité de ressources. Un critère important sera le caractère dynamique mais surtout automatique des techniques développées, seul garant de leur implémentabilité dans le cadre d'un intergiciel.

Les modèles de coordination peuvent se ranger en deux catégories : orientés données (espaces de tuples de Linda ou des Javaspaces) ou orientés contrôle (approches événementielles, comme Manifold ou Reo). On retrouve cette dichotomie au sein des intergiciels mis en œuvre en programmation distribuée. **Un second axe de travail (P-C2)** intéressant est donc d'étudier les relations entre modèles de coordination au niveau conceptuel de façon à pouvoir envisager la définition de "ponts" permettant l'inter-opérabilité entre intergiciels hétérogènes, ce qui est un verrou actuel du domaine de l'intelligence ambiante.

Problématiques liées à l'adaptation

Le génie logiciel basé composants insiste sur la réutilisation de composants, dans l'objectif de proposer un vrai marché de composants et services (les composants sur étagère, Commercial-Off-The-Shelf – COTS), similaire en essence au marché des composants matériels et/ou électroniques. Le développement d'un tel marché a toujours été l'un des mythes du génie logiciel mais n'est jamais devenu une réalité. L'une des raisons en est que, contrairement à ce qui se passe avec les composants matériels, le logiciel n'est jamais réutilisé *tel quel*, mais un certain degré d'adaptation est toujours nécessaire.

La problématique de la modification d'entités logicielles n'est pas nouvelle en génie logiciel. Elle peut être retrouvée dans les travaux autour de l'évolution (ou maintenance) et de la paramétrisation (ou personnalisation). Il faut cependant noter de grandes différences avec l'adaptation. L'évolution suppose une compréhension complète du composant à modifier et une possibilité de modification libre de son code. Son objectif est de fournir un nouveau composant disponible pour l'achat et la réutilisation. La paramétrisation suppose un ensemble d'options ou paramètres prévus dès la conception du composant et sur lesquels l'utilisateur (humain ou composant) a la main. Dans les deux cas, il s'agit d'une approche *intrusive* et très peu automatisable. À l'inverse, l'*adaptation* est un processus dans lequel seule une vision partielle du système ou du composant est supposée. Elle est *automatique et non intrusive*. Son objectif est souvent contextuel et non lié à la réutilisation. L'adaptation peut être *statique* (à la conception) ou *dynamique* (à l'exécution). De toute évidence, dans le cadre de systèmes ouverts, l'adaptation dynamique est nécessaire. Lorsque qu'une incompatibilité est détectée, l'adaptation doit être effectuée sans avoir à stopper l'ensemble du système.

L'adaptation peut avoir lieu aux mêmes niveaux d'interface que la coordination, où différentes incompatibilités peuvent survenir et devoir être corrigées. Un état de l'art des techniques d'adaptation basées sur des méthodes formelles [11] nous apprend que son automatisation passe par l'utilisation de formalismes dynamiques (automates ou algèbres de processus) et procède toujours en trois étapes. Premièrement, une technique de détection d'incompatibilités est développée. De nombreux travaux du domaine des architectures logicielles peuvent ici être réutilisés. Dans un second temps, une description abstraite des propriétés

du système adapté, du médiateur entre composants (l'adaptateur) ou de simples correspondances entre interfaces est fournie. Il s'agit du contrat ou *mapping d'adaptation*. Le processus d'adaptation produit ensuite, automatiquement, à l'aide de ce *mapping* et des interfaces des composants à adapter, l'adaptateur permettant, lorsqu'il est placé au sein du contexte d'exécution, d'assurer la correction du système. Ceci peut se faire selon différentes techniques, non exclusives, mais jamais intrusives : distribution de l'adaptateur sur les composants, utilisation de l'héritage, d'interfaces actives, de *wrappers* ou bien encore de composants-adaptateurs. De façon abstraite, un adaptateur est un connecteur répondant à un objectif particulier (l'adaptation).

Au sein du projet ARLES, je désire étudier l'apport des techniques d'adaptation dans le cadre des systèmes mobiles. Cet apport présenterait un **intérêt à au moins deux niveaux** : l'hétérogénéité des protocoles mis en œuvre dans les intergiciels et l'adaptation de services non compatibles ou ne répondant que partiellement à la spécification d'un service plus complexe. **Dans le premier cas (P-A1)**, des techniques adaptatives, incorporées au sein d'un intergiciel générique pourraient permettre l'interaction entre services basés sur des plate-formes différentes. **Dans le second cas (P-A2)**, l'adaptation est une technique complémentaire à celles développées par le projet ARLES dans le cadre de l'intelligence ambiante et portant sur la composition sémantique de services. En effet, d'un premier côté l'adaptation a besoin d'un *mapping* qui pourrait être fourni en tant que sortie du processus de composition sémantique. À l'inverse, l'adaptation complète ce processus en attaquant des incompatibilités non considérées par la composition comme, dans le cas de services non atomiques constitués d'un protocole complexe d'échange de messages, le besoin de réordonnement des messages constituant ce service, ou bien encore dans le cas de correspondances non bijectives entre les services requis et fournis.

On l'aura compris, **coordination et adaptation sont complémentaires** en termes de détection des incompatibilités, de leur résolution, et de l'obtention automatique d'une ou plusieurs entités (composants/services ou connecteurs/intergiciels) destinés à composer de façon adéquate les services. **Mon programme de recherche de recherche s'intéressera donc aussi à la combinaison de ces deux approches (P-3)**.

Enfin, dans les deux cas, il sera important dans le déroulement de ce programme de recherche de valider l'applicabilité des méthodes et algorithmes de coordination ou d'adaptation développés. L'un des critères les plus importants concerne le **passage à l'échelle (P-4)**, c'est-à-dire l'applicabilité sur des architectures et des services de taille réaliste. La double compétence du projet ARLES (architectures logicielles/intergiciels) devrait permettre le prototypage de ces techniques au sein d'intergiciels et le projet Européen AMIGO (IST FP6 IP), axé autour de l'intelligence ambiante, fournir un cadre de test adéquat. Les contraintes fortes liées au domaine d'application, tant temporelles (durée des processus de coordination ou d'adaptation) que matérielles (mémoire limitée des équipements concernés), rendent par ailleurs indispensable la prise en compte d'informations non fonctionnelles de qualité de service lors de tout processus de composition de services. Coordination et adaptation ont un coût (en mémoire, énergie et temps) qu'il sera important de pouvoir anticiper, par exemple en tenant compte lors de la coordination ou bien en refusant un schéma d'adaptation trop coûteux. Les différents moyens d'implantation des adaptateurs cités plus haut pourront être expérimentés dans ce cadre pour choisir la meilleure approche dans le contexte de l'informatique mobile. Dans un objectif d'efficacité, des pistes autour d'approches compositionnelles de la coordination ou de l'adaptation (absence de vision globale du système), ainsi que l'utilisation d'approches réflexives ou de tissage dynamique d'aspect pourront être étudiées, par exemple en collaboration avec d'autres projets INRIA (OBASCO, POPART, VASY). Notons qu'il est possible que les contraintes du domaine d'application puissent par ailleurs conduire à devoir prendre en compte des modèles moins riches que ceux utilisés dans [16,36] (algèbres de processus et systèmes de transitions), comme par exemple les *types comportementaux*.

Positionnement de mon programme de recherche dans la politique scientifique de l'INRIA

Mon programme de recherche s'inscrit dans deux des défis scientifiques prioritaires de l'INRIA, à savoir, "concevoir et maîtriser les futures infrastructures des réseaux et des services de communication" et "garantir la fiabilité et la sécurité des systèmes à logiciel prépondérant", ainsi que dans le premier objectif prioritaire de l'UR de Rocquencourt, "concevoir et maîtriser les réseaux et les systèmes, traiter l'information distribuée".

TRAVAUX ET RÉSULTATS DEPUIS MON ARRIVÉE DANS LE PROJET ARLES (09/2006–02/2007)

Les références indiquées entre parenthèses (ex : **P-A2**) font référence aux éléments du projet présentés précédemment.

Réduction transactionnelle des modèles de composition [53] (P-4). Les processus de composition et d'adaptation automatique sont basés sur des opérations à base d'automates ou de systèmes de transitions étiquetés qui peuvent conduire à des modèles de compositions ou d'adaptateurs ayant de grands espaces d'états, ou à des modèles contenant des parties redondantes ou inutiles. Ceci limite l'applicabilité de ces techniques dans le cadre de l'informatique diffuse où, en l'absence bien souvent d'infrastructure, les équipements (mobiles) sont équipés de peu de ressources.

En nous basant sur l'observation que, dans les architectures à base de services comme celles développées dans le projet ARLES, les services ont une nature transactionnelle (pour rendre un service donné, les composants se basent sur plusieurs échanges d'événements atomiques et/ou appels de services), nous avons développé une technique de réduction comportementale qui prend en compte les transactions et est donc plus adéquate que les réductions disponibles dans le domaine des algèbres de processus (taux-réductions, bissimulations). Une perspective est de rendre le calcul des réductions applicables "*à la volée*", *i.e.*, pendant le calcul de la composition ou de l'adaptateur.

Adaptation incrémentale de systèmes ouverts [52] (P-A2,P-4). Dans mes travaux sur l'adaptation j'ai développé une approche à base de modèles pour le développement d'adaptateurs permettant d'assurer la correction de compositions de composants logiciels et d'augmenter leur niveau de réutilisabilité. Cette approche [16], comme l'ensemble des approches d'adaptation globale qu'elle étendait, suppose (i) la connaissance de l'ensemble des composants à adapter et (ii) la construction d'un adaptateur global pour les assembler (adaptateur global qui peut ensuite potentiellement être distribué sur l'architecture du système). Dans le cadre de l'application de l'adaptation aux architectures pour l'informatique diffuse, ces hypothèses sont trop restrictives. En effet, dans ce cadre les systèmes sont amenés à évoluer, avec des composants ou services qui peuvent entrer ou sortir du système à tout moment, par exemple en raison de la mobilité de l'utilisateur. Il n'est donc pas possible de supposer a priori la connaissance de l'ensemble des composants à adapter. Pour des raisons d'efficacité, il n'est pas non plus souhaitable d'avoir à reconstruire un adaptateur global dès qu'un changement se produit au sein du système.

Pour remédier à ces verrous, nous avons proposé une extension de l'approche globale pour supporter (i) l'adaptation de systèmes ouverts et (ii) la construction incrémentale de tels systèmes, dans laquelle les adaptateurs sont distribués sur l'architecture et où seulement certains adaptateurs doivent être recalculés en cas de modification dans le système. L'outil *Adaptor* a été étendu (distribution de janvier 2007) pour permettre aussi l'adaptation de systèmes ouverts.

Adaptation sémantique de services [51] (P-C1,P-A2,P-3). Des travaux ont abordé l'adaptation à différents niveaux des interfaces des composants, signature/opérations et comportements, et ont abordé théoriquement l'impact de l'adaptation sur la qualité de service. La prise en compte de la sémantique des services n'est par contre actuellement prise en compte qu'au sein des processus de composition (comme ceux proposés dans le projet ARLES) mais pas au niveau de ceux de l'adaptation. La prise en compte de la sémantique est cependant indispensable à la correction complète des assemblages de composants ou de services. Pour résoudre ce problème, nous avons proposé une approche qui se base sur l'extension de langages de description dynamique de services Web (BPEL) avec des informations sémantiques pour permettre la construction automatique d'adaptateurs assurant aussi la compatibilité sémantique. Cette approche devrait permettre de compléter les approches de composition automatique du projet ARLES qui sont actuellement limitées à des correspondances 1-1 (bijectives) entre services, ce qui n'est pas le cas de l'adaptation.

Adaptation dynamique de services. Je suis porteur d'un projet d'ANR Jeunes-chercheurs (collaboration ARLES, LAMSADE/Univ. Paris Dauphine, IBISC/Univ. Evry) en cours de rédaction. Il concerne la composition dynamique et automatique de services Web et ce de façon à supporter (i) les quatre niveaux d'interfaces des services (signatures, comportements, qualité de service, sémantique), (ii) leur intégration, (iii) la réalisation d'un prototype et (iv) l'évaluation concrète de l'impact de l'adaptation à l'exécution.

LISTE COMPLÈTE DES PUBLICATIONS

Une grande partie de mes publications est disponible depuis :

<http://www.ibisc.univ-evry.fr/~poizat/publications-fr.php>.

Sauf exception, l'ordre des auteurs est l'ordre alphabétique.

Les travaux indiqués par une étoile (*) ont été finalisés ou réalisés depuis mon arrivée à l'INRIA.

THÈSE

1. **Pascal Poizat**. *Korrigan : un formalisme et une méthode pour la spécification formelle et structurée de systèmes mixtes*. Thèse de doctorat, Université de Nantes. Décembre 2000.

ÉDITION D'ACTES ET DE NUMÉROS SPÉCIAUX DE JOURNAUX

2. *Coordination and Adaptation Techniques : Bridging the Gap Between Design and Implementation*. Steffen Becker, Carlos Canal, Nikolay Diakov, Juan Manuel Murillo, Pascal Poizat and Massimo Tivoli (éditeurs). Proceedings of the Third International Workshop on Coordination and Adaptation Techniques for Software Entities (WCAT'06) at ECOOP (European Conference on Object-Oriented Programming), juillet 2006, Nantes, France.
3. *Special Issue on Coordination and Adaptation Techniques*. Carlos Canal, Juan Manuel Murillo and **Pascal Poizat** (éditeurs). L'objet, volume 12, numéro 1. 2006.
4. *New Issues on Coordination and Adaptation Techniques*. Steffen Becker, Carlos Canal, Juan Manuel Murillo, **Pascal Poizat** and Massimo Tivoli (éditeurs). Proceedings of the Second International Workshop on Coordination and Adaptation Techniques for Software Entities (WCAT'05) at ECOOP (European Conference on Object-Oriented Programming), juillet 2005, Glasgow, Ecosse. ISBN 84-689-2747-3.
5. *Issues on Coordination and Adaptation Techniques*. Carlos Canal, Juan Manuel Murillo and **Pascal Poizat** (éditeurs). Proceedings of the First International Workshop on Coordination and Adaptation Techniques for Software Entities (WCAT'04) at ECOOP (European Conference on Object-Oriented Programming), juin 2004, Oslo, Norvège. ISBN 84-688-6782-9.

ARTICLES DE JOURNAUX AVEC COMITÉ DE LECTURE ET CHAPITRES DE LIVRES

6. ***Pascal Poizat** and Jean-Claude Royer. A Formal Architectural Description Language based on Symbolic Transition Systems and Modal Logic. Journal of Universal Computer Science (J.UCS). 42 pages. Springer Verlag. 2007. **accepté le 07/02/2007, à paraître**
7. *Christian Attiogbé, **Pascal Poizat** and Gwen Salaün. A Formal and Tool-Equipped Approach for the Integration of State Diagrams and Formal Datatypes. IEEE Transactions on Software Engineering, volume 33, numéro 3, pages 157–170, 2007.
8. **Pascal Poizat** and Thomas Vergnaud. *Langages de description d'architecture*. chapitre 5 de "Méthodes formelles pour les systèmes répartis et coopératifs". Hermès Lavoisier. 2006.
9. Henri Habrias, **Pascal Poizat** and Marc Frappier. *Glossary*. chapitre 20 de "Software Specification Methods". ISTE, Hermès Science. 2006.
10. **Pascal Poizat**. *SDL*. chapitre 12 de "Software Specification Methods". ISTE, Hermès Science. 2006. – ce chapitre est une réédition de [13].
11. Carlos Canal, Juan Manuel Murillo and **Pascal Poizat**. Software Adaptation. In numéro spécial "Coordination and Adaptation Techniques". L'Objet, volume 12, numéro 1, pages 9–31. 2006.
12. Christine Choppy, **Pascal Poizat** and Jean-Claude Royer. The Korrigan Environment. Journal of

Universal Computer Science (J.UCS), volume 7, numéro 1, pages 19–36. Springer-Verlag, 2001.

13. **Pascal Poizat**. *SDL : a Language based on Extended Finite State Machines with Abstract Data Types*. chapitre 9 de "Software Specification Methods. An Overview Using a Case Study". Formal Approaches to Computing and Information Technology (FACIT). Springer-Verlag, 2000.

CONFÉRENCES ET ATELIERS (WORKSHOPS) INTERNATIONAUX

14. *Steffen Becker, Carlos Canal, Nikolay Diakov, Juan Manuel Murillo, **Pascal Poizat** and Massimo Tivoli. *Coordination and Adaptation Techniques : Bridging the Gap Between Design and Implementation*. In M. Südholt and C. Consel, editors, *ECOOP 2006 Workshop Reader*. volume 4379 of *Lecture Notes in Computer Science*, pages 72–86. Springer Verlag. **à paraître**
15. ***Pascal Poizat**, Gwen Salaün and Massimo Tivoli. *An Adaptation-based Approach to Incrementally Build Component Systems*. In *International Workshop on Formal Aspects of Component Software, FACS'2006*. Electronic Notes in Theoretical Computer Science. **à paraître**
16. Carlos Canal, **Pascal Poizat** and Gwen Salaün. *Synchronizing Behavioural Mismatch in Software Composition*. In *International Conference on Formal Methods for Open Object-Based Distributed Systems, FMOODS'2006*, volume 4037 of *Lecture Notes in Computer Science*, pages 63–77. 2006.
17. **Pascal Poizat**, Jean-Claude Royer and Gwen Salaün. *Bounded Analysis and Decomposition for Behavioural Description of Components*. In *International Conference on Formal Methods for Open Object-Based Distributed Systems, FMOODS'2006*, volume 4037 of *Lecture Notes in Computer Science*, pages 33–47. 2006.
18. Sebastian Pavel, Jacques Noyé, **Pascal Poizat** and Jean-Claude Royer. *A Java Implementation of a Component Model with Explicit Symbolic Protocols*. In *International Workshop on Software Composition, SC'2005*, volume 3628 of *Lecture Notes in Computer Science*, pages 115–124, Springer Verlag, 2005.
19. Gwen Salaün and **Pascal Poizat**. *Interacting Extended State Diagrams*. In *Workshop on the Semantic Foundations of Engineering Design Languages, SFEDL'2004*. Electronic Notes in Theoretical Computer Science, volume 115, pages 49–57. 2005.
20. Carlos Canal, Juan Manuel Murillo and **Pascal Poizat**. *Coordination and Adaptation Techniques for Software Entities*. In J. Malenfant and B. M. Ostvold, editors, *ECOOP 2004 Workshop Reader*. volume 3344 of *Lecture Notes in Computer Science*, pages 131–145. Springer-Verlag, 2004.
21. Olivier Maréchal, **Pascal Poizat** and Jean-Claude Royer. *Checking Asynchronously Communicating Components using Symbolic Transition Systems*. In *International Symposium on Distributed Objects and Applications, DOA'2004*, volume 3291 of *Lecture Notes in Computer Science*, pages 1502–1519. Springer-Verlag, 2004.
22. Marc Aiguier, Fabrice Barbier and **Pascal Poizat**. *A Logic with Temporal Glue for Mixed Specifications*. In *International Workshop on the Foundations of Coordination Languages and Software Architectures, FOCLASA'2003*. Electronic Notes in Theoretical Computer Science, volume 97, pages 155–174. 2004
23. Christian Attiogbé, **Pascal Poizat** and Gwen Salaün. *Integration of Formal Datatypes within State Diagrams*. In *International Conference on Fundamental Approaches to Software Engineering, FASE'2003*, volume 2621 of *Lecture Notes in Computer Science*, pages 341–355. Springer-Verlag, 2003.
24. Christian Attiogbé, **Pascal Poizat** and Gwen Salaün. *Specification of a Gas Station Using a Formalism Integrating Formal Datatypes Within State Diagrams*. In *International Parallel and Distributed Processing Symposium, Formal Methods for Parallel Programming : Theory and Applications, FMPP-TA'2003*, IEEE Computer Society Press, 2003
25. Michel Allemand, Christian Attiogbé, **Pascal Poizat**, Gwen Salaün and Jean-Claude Royer. *SHE'S Project : a Report of Joint Works on the Integration of Formal Specification Techniques*. In *Workshop on Integration of Specification Techniques with Applications in Engineering, INT'2002*, 2002.

26. Christine Choppy, **Pascal Poizat** and Jean-Claude Royer. Formal Specification of Mixed Components with Korrigan . In H. Jifeng, editor, *Asia-Pacific Software Engineering Conference, APSEC'2001*, pages 169–176. IEEE Computer Society Press. 2001.
27. Christine Choppy, **Pascal Poizat** and Jean-Claude Royer. Specification of Mixed Systems in Korrigan with the Support of an UML-Inspired Graphical Notation. In H. Hussmann, editor, *International Conference on Fundamental Approaches to Software Engineering, FASE'2001*, volume 2029 of *Lecture Notes in Computer Science*, pages 124–139. Springer-Verlag. 2001.
28. Christine Choppy, **Pascal Poizat** and Jean-Claude Royer. The Korrigan Environment. In *International Workshop on Tools for System Design and Verification, FMTOOLS'2000*. 2000.
29. Christine Choppy, **Pascal Poizat** and Jean-Claude Royer. A Global Semantics for Views. In T. Rus, editor, *International Conference on Algebraic Methodology and Software Technology, AMAST'2000*, volume 1816 of *Lecture Notes in Computer Science*, pages 165–180. Springer-Verlag. 2000.
30. Christine Choppy, **Pascal Poizat** and Jean-Claude Royer. Integration and Composition of Static and Dynamic Views : Unifying Approach to Complex System Specification. In H. Ehrig, M. Grosse-Rhode and F. Orejas, editors, *Workshop on Integration of Specification Techniques with Applications in Engineering, INT'2000*, Technische Universitat Berlin, Bericht-Nr. 2000/04, ISSN 1436-9915, pages 12–20, 2000.
31. **Pascal Poizat**, Christine Choppy and Jean-Claude Royer. From Informal Requirements to COOP : a Concurrent Automata Approach. In J.M. Wing, J. Woodcock and J. Davies, editors, *Formal Methods, World Congress on Formal Methods in the Development of Computing Systems, FM'99*, volume 1709 of *Lecture Notes in Computer Science*, pages 939–962. Springer-Verlag. 1999.
32. **Pascal Poizat**, Christine Choppy and Jean-Claude Royer. Concurrency and Data Types : A Specification Method. An Example with LOTOS. In J. Fiadeiro, editor, *Recent Trends in Algebraic Development Techniques, Selected Papers of the 13th International Workshop on Algebraic Development Techniques WADT'98*, volume 1589 of *Lecture Notes in Computer Science*, pages 276–291. Springer-Verlag. 1999.

CONFÉRENCES ET ATELIERS (WORKSHOPS) NATIONAUX

33. Carlos Canal, **Pascal Poizat** and Gwen Salaün. Adaptation de composants logiciels. Une approche automatisée basée sur des expressions régulières de vecteurs de synchronisation. In *Conférence Francophone sur les Architectures Logicielles, CAL'2006*, pages 31–39. 2006.
34. Christian Attiogbé, **Pascal Poizat** and Gwen Salaün. Intégration de données formelles dans les diagrammes d'états d'UML. In J.-M. Jézéquel, editor, *Conférence sur les Approches Formelles dans l'Assistance au Développement de Logiciels, AFADL'2003*, ISBN 2-7261-1236-6, pages 3–17, 2003.
35. **Pascal Poizat** and Jean-Claude Royer. Une proposition de composants formels. In *Conférence sur les Langages et Modèles à Objets, LMO'2002*, pages 231–245. L'Objet, volume 8, numéros 1–2. 2002.

RAPPORTS DE RECHERCHE

36. Carlos Canal, **Pascal Poizat** and Gwen Salaün. Adaptation of Component Protocols using Synchronous Vectors. Rapport de Recherche ITI-05-10, Department of Computer Science, University of Málaga. 2005.
37. **Pascal Poizat**, Jean-Claude Royer and Gwen Salaün. Symbolic Bounded Analysis for Component Behavioural Protocols. Rapport de Recherche EMN 05//INFO, École des Mines de Nantes, 2005.
38. **Pascal Poizat** and Gwen Salaün. Formal Coordination of Communicating Entities described with Behavioural Interfaces. Rapport de Recherche 120-2005, LaMI, 2005.
39. Sebastian Pavel, Jacques Noyé, **Pascal Poizat** and Jean-Claude Royer. A Formal Component Model with Explicit Symbolic Protocols and its Implementation in Java. Rapport de Recherche EMN 05//INFO, École des Mines de Nantes, 2005.

40. Olivier Maréchal, **Pascal Poizat** and Jean-Claude Royer. Checking Asynchronously Communicating Components using Symbolic Transition Systems. Rapport de Recherche EMN 04/5/INFO, École des Mines de Nantes, 2004.
41. **Pascal Poizat** and Jean-Claude Royer. Korrigan : a Formal ADL with Full Data Types and a Temporal Glue. Rapport de Recherche 88-2003, LaMI, 2003.
42. Christian Attiogbé, **Pascal Poizat** and Gwen Salaün. Integration of Formal Datatypes within State Diagrams. Rapport de Recherche 83-2002, LaMI, 2002.
43. Christian Attiogbé, **Pascal Poizat** and Gwen Salaün. Intégration de données formelles dans les diagrammes d'états d'UML. Rapport de Recherche 02.03, IRIN, 2002.
44. Marc Aiguier, Fabrice Barbier and **Pascal Poizat**. A Logic for Mixed Specifications. Rapport de Recherche 73-2002, LaMI, 2002
45. Christine Choppy, **Pascal Poizat** and Jean-Claude Royer. The Korrigan Environment. Rapport de Recherche 00.10, IRIN, 2000.
46. Christine Choppy, **Pascal Poizat** and Jean-Claude Royer. A Global Semantics for Views. Rapport de Recherche 189, IRIN, 1999.
47. Christine Choppy, **Pascal Poizat** and Jean-Claude Royer. Control and Datatypes using the View Formalism. Rapport de Recherche 188, IRIN, 1999.
48. **Pascal Poizat**, Christine Choppy and Jean-Claude Royer. Un support méthodologique pour la spécification de systèmes mixtes. Rapport de Recherche 180, IRIN, 1998.
49. **Pascal Poizat**, Christine Choppy and Jean-Claude Royer. Une nouvelle méthode pour la spécification en LOTOS. Rapport de Recherche 170, IRIN, 1998.

EN COURS DE SOUMISSION

50. *Carlos Canal, **Pascal Poizat** and Gwen Salaün. Software Adaptation of Behavioural Mismatching Components. Soumission au Journal of Systems and Software le 18 février. 20 pages.
51. *Tarek Melliti, **Pascal Poizat** and Sonia Ben Mokhtar. Automatic Generation of Semantic Service Compositions using Behavioural Adaptation. Soumis à la conférence FMOODS'2007 (International Conference on Formal Methods for Open Object-Based Distributed Systems). 15 pages.
52. ***Pascal Poizat** and Gwen Salaün. Adaptation of Open Component-based Systems. Soumis à la conférence FMOODS'2007 (International Conference on Formal Methods for Open Object-Based Distributed Systems). 15 pages.
53. *Serge Haddad and **Pascal Poizat**. Transactional Reduction of Component Compositions. Soumission à la conférence FORTE'2007 (International Conference on Formal Methods for Networked and Distributed Systems) le 18 février. 15 pages.

DIVERS

54. **Pascal Poizat**, Gwen Salaün and Massimo Tivoli. *On Dynamic Reconfiguration of Behavioural Adaptation*. In *Int. Workshop on Coordination and Adaptation Techniques for Software Entities, WCAT'2006*. pages 61–69. 2006.
55. **Pascal Poizat**. Eclipse Transition Systems. Délivrable RNRT STACS. 2005.
56. Steffen Becker, Carlos Canal, Juan Manuel Murillo, **Pascal Poizat** and Massimo Tivoli. *Run Time and Design Time Issues on Implementing Software Adaptors*. WCAT'05 Workshop Report.
57. **Pascal Poizat**, Jean-Claude Royer and Gwen Salaün. *Formal Methods for Component Description, Coordination and Adaptation*. In *Int. Workshop on Coordination and Adaptation Techniques for*

Software Entities, WCAT'2004. pages 89–100. ISBN : 84-688-6782-9. 2004.

58. Marc Aiguier, Fabrice Barbier and **Pascal Poizat**. A Logic for Mixed Specifications. 16th International Workshop on Algebraic Development Techniques WADT'2002. 2002.
59. Christine Choppy, **Pascal Poizat** and Jean-Claude Royer. Control and Datatypes using the View Formalism. 14th International Workshop on Algebraic Development Techniques WADT'99. 1999.
60. Henri Habrias, **Pascal Poizat** and Jean-Yves Lafaye. A Study of Collaborative Work : Answers to a Test on Formal Specification in B. In J.M. Wing, J. Woodcock and J. Davies, editors, *FM'99 - Formal Methods, World Congress on Formal Methods in the Development of Computing Systems*, volume 1709 of *Lecture Notes in Computer Science*, pages 1856–1857. Springer-Verlag. 1999.
61. **Pascal Poizat**, Christine Choppy and Jean-Claude Royer. Concurrency and Data Types : A Specification Method. An Example with LOTOS. 13th International Workshop on Algebraic Development Techniques WADT'98. 1998.
62. Henri Habrias and **Pascal Poizat**. Collaborative work to answer a test on formal specification in B. In Henri Habrias and Steve E. Dunne, editors, *B'98 Education Session*, pages 33–43. Association de Pilotage des Conférences B. 1998.
63. **Pascal Poizat**. Résumé du livre : *Java, de l'esprit à la méthode*, L'Objet - Logiciel, Bases de données, Réseaux, 4(1), pages 106–107, 1998.
64. **Pascal Poizat**. Applications de la réécriture de termes aux modèles à objet. Mémoire de DEA Informatique, Université de Nantes, juin 1995.

LETTRE DU CHEF DE PROJET D'ACCUEIL SOUHAITÉ

V. Issarny, DR, projet ARLES, UR de Rocquencourt

(non public)

LETTRE D'AUTORISATION DE L'UNIVERSITÉ D'ÉVRY

(non public)

LETTRES DE RECOMMANDATION

(non public)

