

APPEL A PROJETS DE RECHERCHE

ATTENTION : Cette partie (I) sera également à saisir directement sur le site de soumission

I - FICHE D'IDENTITE DU PROJET

Référence :

(reprendre la référence qui vous sera attribuée automatiquement par le logiciel de soumission)

Titre du projet (*maximum 120 caractères*)

Familles de Langages pour Familles de Systèmes (FLFS)

Acronyme ou titre court : FLFS

Secteur disciplinaire principal (*cf. liste en dernière page de ce dossier*) :

STIC

Mots-clés (deux max. choisis dans la liste des mots-clés proposés pour ce secteur sur le logiciel de soumission)

Qualité et vérification du logiciel - Réseaux systèmes architectures.

Mots-clés libres du secteur principal (max. 4)

Génie logiciel, programmation générative

Autre secteur disciplinaire secondaire¹ (*cf. note de bas de page*):

Mots-clés (deux max. choisis dans la liste des mots-clés proposés pour ce secteur sur le logiciel de soumission)

Mots-clés libres du secteur secondaire (max. 4)

¹ Un dossier ne sera effectivement évalué par un secteur secondaire que si un des participants au projet relève de ce secteur secondaire.

Résumé du projet (maximum 5000 caractères)

Traditionnellement le développement de logiciels ne s'appuie pas sur une connaissance approfondie du domaine ciblé, mais l'introduit de façon ad hoc et partielle avec une démarche peu formalisée et peu outillée, pour satisfaire des objectifs immédiats. Se faisant, les logiciels sont trop souvent appréhendés isolément, rendant difficile toute factorisation conceptuelle ou de mise en œuvre. Pourtant, il est fondamental de constater que les logiciels appartiennent toujours à une famille. Au sein d'une même famille, ils partagent un ensemble de points communs et exhibent des variations spécifiques. Nous désirons capturer cette réalité par l'utilisation de familles de langages.

Du point de vue du développement logiciel, une famille de logiciels constitue un *métier*, c'est-à-dire, un vocabulaire, des notations, des règles, des représentations et des protocoles qui sont propres à un domaine. Par exemple, le domaine de la téléphonie comprend un ensemble de concepts, de règles, de protocoles et d'interfaces qui encadrent très précisément le développement de services de téléphonie.

Objectif

Notre initiative de recherche vise à placer le métier au centre du processus de développement de logiciels. Elle a pour ambition de lever les verrous actuels du génie logiciel concernant la production à grande échelle, la robustesse, la fiabilité, la maintenance et l'évolutivité des composants logiciels. L'innovation consiste à concevoir un processus de développement logiciel en fonction d'un métier particulier (ou domaine applicatif). Ce processus couvre toutes les étapes du génie logiciel en combinant les trois approches émergentes suivantes :

- La modélisation métier aussi appelée " *ingénierie de modèles* " ;
- Les langages de programmation métiers ou langages dédiés aussi appelés " *Domain-Specific Languages* " qui s'opposent aux langages généralistes et monolithiques ;
- La programmation générative et en particulier les aspects comme vecteurs de transformation de modèles et de programmes.

Ces trois approches ont déjà démontré isolément des bénéfices génie logiciel concrets et reconnus ; leur combinaison permettra d'appréhender la totalité du processus de développement logiciel dédié à un métier donné.

L'idée que nous désirons soutenir dans ce projet est que l'évolution de la recherche dans les domaines de l'ingénierie des modèles, des langages dédiés (DSLs), et de la séparation des aspects mène actuellement à une forte convergence au niveau international. Par ce regroupement de trois équipes françaises, chacune reconnue dans son domaine, nous pensons pouvoir contribuer à cette convergence en faisant avancer l'état de l'art de façon significative. Pour ceci nous désirons démontrer expérimentalement que la notion de famille de systèmes (ou de ligne de produits) peut s'appuyer sur un ensemble de langages dédiés que nous nommons famille de langages. Ces langages vont permettre de capturer des besoins amont (coté utilisateur final) aussi bien que des caractéristiques de mise en œuvre (coté plate-forme d'implémentation). La notion de chaîne de langages permettant d'exprimer des "mappings" explicites entre ces différents formalismes nous apparaît aujourd'hui un concept très prometteur. A partir d'expérimentations préliminaires menées dans le domaine de la téléphonie, nous sommes convaincus qu'il est maintenant possible d'organiser ces chaînes de langages et de traducteurs des deux cotés d'un langage pivot nommé DSPL (Domain Specific Programming Language), en amont du coté des besoins utilisateurs et en aval du coté de la traduction vers une plate-forme. Nous désirons confirmer cette intuition sur un sujet principal (téléphonie) et sur quelques sujets annexes (ordonnancement dans les systèmes, gestion de règles métier, gestion de données, réseaux de "streaming", etc.).

Résultats attendus

- Des études de cas appuyant l'hypothèse avancée ci-dessus et portant sur les domaines mentionnés (téléphonie, ordonnancement dans les systèmes, adaptation de composants, etc., règles métier, gestion de données et de flots de données, etc.)
- Un atelier de génie logiciel basé sur Eclipse (<http://www.eclipse.org/>) incluant une palette d'outils destinés à la création et à l'utilisation des langages dédiés.
- Des publications montrant comment les études de cas outillés confortent l'hypothèse de la prise en charge des familles de systèmes par les familles de langages

Méthodologie

Il s'agit de réunir trois équipes françaises déjà fortement impliquées dans chacune des technologies des modèles, des langages dédiés et des aspects dans la perspective d'élaborer, d'intégrer puis de partager les outils du continuum modélisation, programmation et implémentation. Ces outils seront validés dans le domaine applicatif central de la téléphonie et des domaines annexes (ordonnancement dans les systèmes d'exploitation, gestion de règles métier, gestion de données, systèmes distribués basés sur les flots ("streaming"), etc.)

Abstract (Not exceed 5000 car.)

Traditionally, software development does not rely on an in-depth knowledge of the target domain. Instead, domain-specific knowledge is integrated in software development process in an ad hoc and partial fashion, without much formal basis or tools. In doing so, software systems are tackled in isolation, making conceptual or implementation factorization difficult. Yet, it is fundamental to observe that programs always belong to a family. In this family, they share commonalities and expose specific variations.

From a software development viewpoint, a program family represents a *domain of expertise*, that is, a vocabulary, notations, rules and protocols that are specific to a domain. For example, the telephony domain consists of a set of concepts, rules, protocols and interfaces that represent a precise framework to be used for the development of telephony services.

Objectives

The goal of our research proposal is to place *domain expertise* at the centre of the software development process. It is aimed to lift the current limitations of software engineering regarding large scale software production, robustness, reliability, maintenance and evolution of software components. Our key innovation is to introduce a software development process parameterized with respect to a specific domain of expertise. This process covers all the stages of software development and combines the following three emerging approaches:

- Domain-specific modelling, also known as model engineering;
- Domain-specific languages, in contrast with general-purpose languages;
- Generative programming and in particular aspect-oriented programming as a means to transform models and programs.

These three approaches have already demonstrated concrete and well-recognized software engineering benefits, in isolation; their combination will permit to cover the entire software development process dedicated to a specific domain of expertise.

Expected results

1. A domain-specific approach to software development,
2. An Eclipse-based programming environment including a toolbox for creating and using domain specific languages,
3. A set of experiments, described in research papers and showing how the common tools may comfort the idea that system families (e.g. product lines) may be supported by families of coordinated languages, all of these languages related to a specific domain, but some more on the end-user side and others on the implementation platform side.

Methodology

Our approach is to forming a tightly-coupled consortium of three research groups, already strongly involved in each of the technologies – model engineering, domain-specific languages, and aspects – towards developing, integrating, and sharing the tools that will target the continuum modeling, programming, and implementing. These tools will be validated in the context of the main application domain of telephony and secondary application domains of operating system scheduling, business rule management, data engineering and distributed networks of streaming systems.

Coordinateur du projet² (Partenaire 1)

Civilité	Nom	Prénom	Laboratoire (nom complet)	Type (établissement public, fondation, association, entreprise)
Mr	Cointe	Pierre	LINA/EMN. Laboratoire Informatique de Nantes Atlantique	Association ARMINES/EMN

Nom des responsables scientifiques des autres partenaires

Civilité	Nom	Prénom	Laboratoire (nom complet)	Type (établissement public, fondation, association, entreprise)
Mr	Consel	Charles	INRIA Futurs (Bordeaux)	EPST
Mr	Bézivin	Jean	LINA. Laboratoire Informatique de Nantes Atlantique	CNRS FRE 2729

Nombre de personnes impliquées dans ce projet (en équivalent temps plein : ETP)³:Chercheurs et enseignants-chercheurs permanents **5**

Post-doctorant(s) déjà recruté(s) _____ Doctorant(s) _____ Ingénieurs et techniciens _____

Personnes à recruter 3 doctorants

Durée du projet : 36 mois

² Rappel : le coordinateur du projet doit consacrer au moins 30% de son temps de recherche au projet

³ Quelque soit la catégorie de personnel, il s'agit ici, pour chaque personne impliquée dans le projet, de multiplier son temps de recherche par le pourcentage de temps qu'il consacrerà à ce projet.

Dimensionnement total du projet

Coût complet du projet : 823 878€
Aide financière demandée : 291 570 €
Effort en personnel demandé : 108 homme. mois

Je déclare exactes toutes les informations contenues dans ce document et m'engage à envoyer une copie de ce dossier à chacun des établissements ou organismes de rattachement de mon laboratoire.

Visa du directeur du
laboratoire

Cointe Pierre,
20/03/06
Date et signature du **coordinateur
du projet** précédé de la mention
" Lu et approuvé "

Cointe, Pierre
20/03/06

En cas de recouvrement thématique avec d'autres appels à projets (AAP) lancés par l'ANR, les coordinateurs de projet devront veiller à choisir l'AAP le mieux adapté à leur projet. Les personnes impliquées dans plusieurs AAP soumis à l'ANR devront le mentionner dans le tableau " demandes de contrats en cours d'évaluation " (Section D du document).

Programme blanc 2006

APPEL A PROJETS DE RECHERCHE

II - PRESENTATION DETAILLEE DU PROJET

A - Identification du coordinateur et des autres partenaires du projet

Acronyme ou titre court du projet :

A-1 – Partenaire 1 = Coordinateur du Projet

Un coordinateur, responsable scientifique du projet, doit être désigné par les partenaires.

Civilité ⁴	Nom ²	Prénom ²
Monsieur	Cointe	Pierre
Grade ²	Professeur 1 ^{ère} classe des écoles des mines	
Mail ²	cointe@emn.fr	
Tél ²	02 51 85 82 00	Fax ² 02 51 85 82 49

Laboratoire ² (nom complet)					
LINA, Laboratoire d'Informatique de Nantes Atlantique CNRS FRE 2729 Projet EMN-INRIA OBASCO					
N° Unité (s'il existe)		FRE 2729			
Adresse complète du laboratoire ²					
LINA/Ecole des Mines de Nantes/Centre Armines 4 rue Alfred Kastler 44 307 Nantes cedex					
Ville ²	Nantes	Code postal ²	44307	Région ²	Pays de Loire
Organismes de tutelle (indiquer le ou les établissements et organismes de rattachement, souligner l'établissement susceptible d'assurer la gestion du projet) :					
Centre Commun Ecole des Mines de Nantes-Armines Bruno Restif, Directeur du Développement 60 bd Saint Michel 75272 Paris cedex 6					

Principales publications :

Liste des principales publications ou brevets (max. 5) de l'équipe partenaire 1 (définie tableau ci-dessous) au cours des cinq dernières années, relevant du domaine de recherche couvert par la présente demande dans l'ordre suivant : Auteurs (en soulignant les auteurs faisant effectivement partie de la demande), Année, Titre, Revue, N°Vol, Pages. N'indiquez pas les publications soumises.

1. Éric Tanter, Jacques Noyé, Denis Caromel, and Pierre Cointe. *Partial Behavioral Reflection: Spatial and Temporal Selection of Reification*. In Ron Crocker and Guy L. Steele, Jr., editors, **proceedings of the 18th ACM SIGPLAN conference on Object-Oriented Programming, Systems, Languages, and Applications , OOPSLA 2003**, Anaheim, California, USA, pages 27-46, October 2003. ACM Press.
2. Pierre Cointe. *Towards Generative Programming*. **Unconventional Programming Paradigms, UPP 2004**, Mont St Michel, France, september 2004. Edited by J.-P. Banâtre and al. Pages 302-312, LNCS 3566, Springer Verlag 2005.
3. Pierre Cointe, Hervé Amiot Albin and Simon Denier. *From (meta) objects to aspects : from Java to AspectJ*. **Third International Symposium on Formal Methods for Components and**

⁴ Champ obligatoire

Objects, FCMO 2004, Leiden, The Netherlands, november 2004. Edited by M. Bonzangue and al, pages 70-94 LNCS 3657, Springer Verlag 2005.

4. Charles Consel, Fabien Latry, Laurent Réveillère and Pierre Cointe. *A Generative Programming Approach to Developing DSL Compilers*. **Generative Programming and Component Engineering conference, GPCE 2005**, ACM SIGPLAN, SIGSOFT, LNCS Springer Verlag, Tallinn, Estonia, Sept-29, Oct- 1 2005.
5. Pierre Cointe, Jacques. Noyé, Rémi Douence, Thomas Ledoux, Jean-Marc Menaud, Gilles Muller and Mario Südholt. Programmation post-objets : des langages d'aspects aux langages de composants. Revue RSTI série L'Objet. Des octets aux modèles. Vingt ans après : ou en sont les objets. Volume 10, numéro 4/2004, pages 119-144. Lavoisier Hermès 2004.

Ce projet fait-il partie des projets labellisés (ou en cours de labellisation) par un pôle de compétitivité (ou par plusieurs, en cas de projet interpôle) ? **NON**

Si oui, nom du pôle ou des pôles :

Partenaire 1 = Coordinateur du Projet

	Nom	Prénom	Emploi actuel	Discipline (à renseigner uniquement pour SHS)	% de temps de recherche consacré au projet	Rôle/Responsabilité dans le projet 4 lignes max
<i>exemple</i>	<i>MARTIN</i>	<i>Charles</i>	<i>Professeur</i>		<i>30%</i>	<i>Caractérisation des facteurs de transcription recombinants en système in vitro ...</i>
Coordinateur	Cointe	Pierre	Professeur		30%	Responsable scientifique. Encadrement du doctorant travaillant à la définition de langages d aspects et à la mise en œuvre de ceux-ci pour la compilation et la composition de langages dédiés.
Membres de l'équipe	Muller	Gilles	Professeur		5%	Expertise système d'exploitation.

Pour chacun des membres de l'équipe dont l'implication dans le projet est supérieure à 25%, fournir une biographie **d'une page maximum** qui comportera :

A/ Nom, prénom, âge, cursus, situation actuelle

B/ Autres expériences professionnelles

C/ Liste des 5 publications (ou brevets) les plus significatives des cinq dernières années

D/ Prix, distinctions

Pierre Cointe, Professeur, Ecole des Mines de Nantes

Pierre Cointe.

Né le 7 novembre 1954.

Doctorat de 3^{ième} cycle université en informatique, Université Paris-6, décembre 1981.

Doctorat d'Etat, Université Paris-6, décembre 1984.

Professeur de 1^{ère} classe des écoles des mines depuis juillet 1992.

Responsable du département informatique de l'école des mines de Nantes depuis sa création en 1992.

Responsable du projet INRIA-EMN OBASCO (Objets, ASpects et COmposants) depuis 2002.

Directeur adjoint du LINA, CNRS FRE 2729 depuis 2004.

Co-responsable du CPER Pays Loire, 2000-2006.

Autres expériences professionnelles

Depuis septembre 2004, responsable pour l'INRIA de la coordination des travaux des projets Triskell, Obsaco, Jacquard et PopArt dans le cadre du REX AOSD ;
Expert auprès de la MSTP DS 9 depuis 2003 ;

- De décembre 1999 à juillet 2003 membre externe nommé du CNU section 27 ;
- De mars 1999 à juillet 2002 membre externe de la commission d'évaluation de l'INRIA ;
- De janvier 1995 à décembre 1999 : responsable du laboratoire Jules Verne commun à l'EMN et OTI/IBM IBM travaillant sur la compilation des langages de la famille Java ;
- D'avril 1987 à juin 1992 : responsable du laboratoire commun Rank Xerox France, LITP, université Paris 6, travaillant sur les langages à objets (Smalltalk, CLOS, C++) ;
- De décembre 1985 à avril 1987 : maître de recherche au Centre Mondial de l'Informatique ;
- D'août 1981 à décembre 1985 : chargé de recherche à l'IRCAM ;
- De septembre 1979 à août 1981 : boursier DGRST au LITP, université Paris 6 ;
- De septembre 1977 à août 1979 : VSNA (E/C) au CERI d'Alger.

P. Cointe est l'un des pionniers européens du domaine des langages à objets. Il a contribué à la création et au développement de plusieurs conférences internationales dont ECOOP, OOPSLA et Reflection mais aussi nationales : LMO, JFLA et plus récemment Objet, OCM et JFLPA.

Publications significatives pour le projet

Éric Tanter, Jacques Noyé, Denis Caromel, and Pierre Cointe. *Partial Behavioral Reflection: Spatial and Temporal Selection of Reification*. In Ron Crocker and Guy L. Steele, Jr., editors, **proceedings of the 18th ACM SIGPLAN conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2003**, Anaheim, California, USA, pages 27-46, October 2003. ACM Press.

Pierre Cointe. *Towards Generative Programming. Unconventional Programming Paradigms, UPP 2004*, Mont St Michel, France, September 2004. Edited by J.-P. Banâtre et al. Pages 302-312, LNCS 3566, Springer Verlag 2005.

Pierre Cointe, Hervé Amiot Albin and Simon Denier. *From (meta) objects to aspects : from Java to AspectJ*. **Third International Symposium on Formal Methods for Components and Objects, FCMO 2004**, Leiden, The Netherlands, November 2004. Edited by M. Bonzangue et al, Pages 70-94, LNCS 3657, Springer 2005.

Charles Consel, Fabien Latry, Laurent Réveillère and Pierre Cointe. *A Generative Programming Approach to Developing DSL Compilers*. **Fourth International Conference on Generative Programming and Component Engineering, GPCE 2005**, ACM SIGLPAN, SIGSOFT, LNCS, Tallinn, Estonia, Sept-29, Oct- 1 2005.

Pierre Cointe, Jacques Noyé, Rémi Douence, Thomas Ledoux, Jean-Marc Menaud, Gilles Muller and Mario Südholt. **Programmation post-objets : des langages d'aspects aux langages de composants. Revue RSTI série L'Objet. Des octets aux modèles. Vingt ans après : ou en sont les objets**. Volume 10, numéro 4/2004, pages 119-144. Lavoisier Hermès 2004.

Gilles Muller, Professeur, Ecole des Mines de Nantes

Né le 4 août 1962

Docteur en informatique de l'Université de Rennes 1, juin 1988

Habilitation à diriger des recherches, Université de Rennes 1, octobre 1997

Expérience professionnelle

Octobre 1988 - février 2002 : Chargé de recherche INRIA

Octobre 1988 - Mars 1996 : projet Lsp, devenu ultérieurement Solidor.

Avril 1996 – février 2002 : projet Compose, créé avec Charles Consel

Depuis mars 2002 professeur 2^{ème} classe des écoles des mines, membre du projet INRIA/EMN OBASCO.

Membre du RTP 5 Systèmes Répartis du CNRS Département STIC. (Responsable JM Geib).

Responsable de l' Action Spécifique CNRS "méthodes et outils pour noyaux de systèmes d'exploitation"

Membre du conseil du LINA , Membre du conseil de l'Ecole doctorale ED-STIM

Le domaine d'expertise de Gilles Muller est la conception de systèmes d'exploitation. Il s'intéresse plus particulièrement au développement de nouvelles méthodologies reposant sur la programmation par aspects et les langages métier (*Domain Specific Languages*).

Gilles Muller est l'auteur de plus de 50 conférences et 10 articles de revues internationales dont TOCS, TOPLAS, ACM Computing Surveys, TCS, IEEE Software, IEEE Transactions on Reliability, OSDI, DSN, FTCS, OOPSLA, ECOOP et ICDCS. Il est co-auteur de 3 brevets dans les domaines de la tolérance aux fautes et de Java embarqué.

Gilles Muller a été le responsable de l'organisation du workshop ACM SIGOPS/EW2002 des conférences fédérées RENPAR/CFSE/SYMPA/JC 2005. Il a été member du comité de programme de conférences internationales telles que ACM EW, ACM Eurosys, IEEE DSN, EDCC, IEEE ICDCS, IEEE SRDS. Gilles Muller a été président du chapitre français de l'ACM/SIGOPS entre octobre 2002 et février 2005. Il est le vice président de l'ACM/SIGOPS depuis juillet 2003.

Publications significatives pour le projet

Julia L. Lawall, Hervé Duchesne, Gilles Muller, and Anne-Françoise Le Meur.

2005. Bossa Nova: Introducing Modularity into the Bossa Domain-Specific Language.

Fourth International Conference on Generative Programming and Component Engineering (GPCE'05).

LNCS, Springer Verlag.

Julia L. Lawall, Anne-Françoise Le Meur, and Gilles Muller.

2004. On Designing a Target-Independent DSL for Safe OS Process-Scheduling Components.

Third International Conference on Generative Programming and Component Engineering (GPCE'04),

Lecture Notes in Computer Science, volume 3286, pp. 436-455.

Gilles Muller, Julia L. Lawall, Scott Thibault, and Rasmus Erik Voel Jensen.

2003. A Domain-Specific Language Approach to Programmable Networks.

IEEE Transactions on Systems, Man, and Cybernetics – Part C, volume 33, number 3, pages 370-381.

D. McNamee, J. Walpole, C. Pu, C. Cowan, C. Krasic, A. Goel, C. Consel, G. Muller, R. Marlet.

2001. Specialization tools and techniques for systematic optimization of system software.

ACM Transactions on Computer Systems, vol 19, 2001, pages 217--251.

F. Mérillon, L. Réveillère, C. Consel, R. Marlet, G. Muller.

2000. Devil: An IDL for Hardware programming

Usenix/ACM Operating System Design and Implementation (OSDI), San Diego, October 2000.

Acronyme ou titre court du projet Familles de Langages pour Familles de Systèmes (FLFS)
Partenaire2

A-2 : Autres partenaires du projet (remplir une fiche par partenaire)

Un responsable scientifique de l'équipe partenaire doit être désigné

Partenaire 2		
Civilité ⁵	Nom ⁴	Prénom ⁴
M	Consel	Charles
Grade ⁴	Professeur des Universités	
Mail ⁴	Charles.Consel@inria.fr	
Tél ⁴	05 56 84 23 25	Fax ⁴ 05 56 84 21 80

Laboratoire ⁴ <i>(nom complet)</i>			
Laboratoire Bordelais de Recherche en Informatique (LaBRI)			
N° Unité <i>(s'il existe)</i>	UMR 5800		
Adresse complète du laboratoire ⁴			
Domaine Universitaire 351, cours de la Libération			
Ville ⁴	Talence	Code postal ⁴	33405
Région ⁴	Aquitaine		
Organismes de tutelle <i>(indiquer le ou les établissements et organismes de rattachement, souligner l'établissement susceptible d'assurer la gestion du projet) :</i>			
<u>INRIA</u> , ENSEIRB, CNRS, LaBRI			

Principales publications :

⁵ Champ obligatoire

1. C. Conseil, F. Latry, L. Réveillère, P. Cointe. *A Generative Programming Approach To Developing DSL Compilers*. In *Generative Programming and Component Engineering*, 2005
2. C. Conseil. Domain-Specific Program Generation; International Seminar, Dagstuhl Castle, chapter *From A Program Family To A Domain-Specific Language*, pages 19-29. Number 3016 in Lecture Notes in Computer Science, State-of-the-Art Survey. Springer-Verlag, 2004.
3. C. Conseil and L. Réveillère. Domain-Specific Program Generation; International Seminar, Dagstuhl Castle, chapter *A DSL Paradigm for Domains of Services: A Study of Communication Services*, pages 165 -- 179. Number 3016 in Lecture Notes in Computer Science, State-of-the-Art Survey. Springer-Verlag, 2004.
4. C. Conseil and L. Réveillère. *A Programmable Client-Server Model: Robust Extensibility via DSLs*. In Proceedings of the 18th IEEE International Conference on Automated Software Engineering (ASE 2003), Montréal, Canada, pages 70--79, November 2003. IEEE Computer Society Press.
5. F. Mérillon, L. Réveillère, C. Conseil, R. Marlet, and G. Muller. Devil: An IDL for Hardware Programming. In Proceedings of the Fourth Symposium on Operating Systems Design and Implementation, San Diego, California, pages 17--30, October 2000.

Partenaire 2

	Nom	Prénom	Emploi actuel	Discipline (à renseigner uniquement pour SHS)	% de temps de recherche consacré au projet	Rôle/Responsabilité dans le projet 4 lignes max
<i>exemple</i>	<i>MARTIN</i>	<i>Charles</i>	<i>Professeur</i>		<i>30%</i>	<i>Caractérisation des facteurs de transcription recombinants en système in vitro ...</i>
Responsable	Consel	Charles	Professeur		27%	Conduite du projet et co-encadrement du doctorant
Membres de l'équipe	Réveillère	Laurent	MCF		33%	Co-encadrement du doctorant

Pour chacun des membres de l'équipe dont l'implication dans le projet est supérieure à 25%, fournir une biographie **d'une page maximum** qui comportera :

A/ Nom, prénom, âge, cursus, situation actuelle

B/ Autres expériences professionnelles

C/ Liste des 5 publications (ou brevets) les plus significatives des cinq dernières années

D/ Prix, distinctions

Charles Consel, Professeur des Universités, ENSEIRB

Charles Consel

Né le 22 Août 1960

Doctorat d'informatique, Université de Paris 6, 1989.

Habilitation à Diriger des Recherches, Université de Rennes 1, 1991.

Professeur des Universités depuis 1993, 1^{ère} classe depuis 2000.

Directeur du département de Télécommunications de l'ENSEIRB depuis 2000.

Responsable de l'équipe de recherche INRIA Phoenix.

Membre du groupe de travail IFIP 2.11 sur la génération de programmes, depuis 2004.

Charles Consel est l'auteur de plus de 40 articles dans des conférences internationales et 25 articles de revues telles que : HOSC, TPLS, TCS, TSE, OSDI, ASE, GPCE, PLDI, ECOOP, ICDCS, PLILP, SRDS, POPL, ACM Computer Survey.

Charles Consel a été co-organisateur de l'école de Dagstuhl en 2003 sur les langages dédiés. Il a été co-président du comité de programme de la conférence GPCE 2002. Il a été éditeur invité de Theoretical Computer Science en 2000. Il a été membre de comité de programmes pour des conférences dans les domaines des langages de programmations et de la génération de programmes telles que : PLDI, ESOP, PEPM, PADO, ICFP, POPL, ECOOP, CC, GPCE.

Autres expériences professionnelles

- 1992 - 1993 Assistant Professor, Oregon Graduate Institute of Technology & Science
Beaverton, OR, USA
- 1989 - 1992 Associate Research Faculty, Yale University, New Haven, CT, USA.
- 1987 - 1989 Enseignant, Epita (a private institute of computer science), Paris, France
- 1986 - 1987 Assistant, University of Paris VIII, Paris, France
- 1985 - 1987 Chercheur assistant, Centre Mondial de l'Informatique, Paris, France
- 1984 - 1985 Ingénieur système, Société Axis Digital, Paris, France

Distinction

2004 Best paper Award pour l'article *Automatic Specialization of Protocol Stacks in OS kernels* paru dans Conference on Local Computer Networks.

Sélection de publications

C. Consel, F. Latry, L. Réveillère, P. Cointe. *A Generative Programming Approach to Developing DSL Compilers*. International Conference on Generative Programming and Component Engineering, GPCE 2005, ACM SIGLPAN, SIGSOFT, Tallinn, Estonia, Sept-29, Oct- 1 2005.

C. Consel. Domain-Specific Program Generation; International Dagstuhl Seminar, chapter From A Program Family To A Domain-Specific Language, pages 19-29. Number 3016 in LNCS, State-of-the-Art Survey. Springer-Verlag, 2004.

C. Consel, H. Hamdi, L. Réveillère, L. Singaravelu, H. Yu, and C. Pu. Spidle: A DSL Approach to Specifying Streaming Application. In International Conference on Generative Programming and Component Engineering, 2003.

C. Consel and L. Réveillère. *A Programmable Client-Server Model: Robust Extensibility via DSLs*. In Proceedings of the 18th IEEE International Conference on Automated Software Engineering, Montréal, Canada, pages 70--79, 2003.

F. Méry, L. Réveillère, C. Consel, R. Marlet, and G. Muller. *Devil: An IDL for Hardware Programming*. In Proceedings of the Fourth Symposium on Operating Systems Design and Implementation, San Diego, USA, pages 17--30, 2000.

Laurent Réveillère, Maître de Conférences, ENSEIRB

Laurent Réveillère,
Né le 22 Juillet 1974.
Doctorat en Informatique, Université de Rennes 1, décembre 2001.

Maître de Conférences à l'ENSEIRB depuis Septembre 2002.
Membre du conseil scientifique de l'ENSEIRB, depuis 2003.
Membre du groupe de travail IFIP 2.11 sur la génération de programmes, depuis 2004.

Laurent Réveillère est l'auteur d'un chapitre de livre et de sept articles dans des conférences internationales telles que OSDI, DSN, ASE, GPCE.

Laurent Réveillère a organisé le workshop du groupe IFIP 2.11 sur la génération de programmes en 2004.

Autres expériences professionnelles

Attaché Temporaire d'Enseignement et de Recherche à l'ENSEIRB, département Télécommunications, de 2001 à 2002.
Allocataire d'Enseignement Supérieur à l'Université de Rennes 1, de 1998 à 2001.

Prix

2003 Prix de la recherche en système décerné par le chapitre Français de l'ACM SIGOPS.

Sélection de publications

C. Consel, F. Latry, L. Réveillère, P. Cointe. *A Generative Programming Approach to Developing DSL Compilers*. International Conference on Generative Programming and Component Engineering, GPCE 2005, ACM SIGPLAN, SIGSOFT, Tallinn, Estonia, Sept-29, Oct- 1 2005.

C. Consel and L. Réveillère. Domain-Specific Program Generation; International Dagstuhl Seminar, chapter *A DSL Paradigm for Domains of Services: A Study of Communication Services*, pages 165 -- 179. Number 3016 in LNCS, State-of-the-Art Survey. Springer-Verlag, 2004.

C. Consel and L. Réveillère. *A Programmable Client-Server Model: Robust Extensibility via DSLs*. In Proceedings of the 18th IEEE International Conference on Automated Software Engineering, Montréal, Canada, pages 70--79, 2003.

L. Réveillère and G. Muller. *Improving Driver Robustness: an Evaluation of the Devil Approach*. In The International Conference on Dependable Systems and Networks, Göteborg, Sweden, pages 131--140, July 2001.

F. Mérillon, L. Réveillère, C. Consel, R. Marlet, and G. Muller. *Devil: An IDL for Hardware Programming*. In Proceedings of the Fourth Symposium on Operating Systems Design and Implementation, San Diego, USA, pages 17--30, 2000.

Acronyme ou titre court du projet FLFS Partenaire3
--

A-2 : Autres partenaires du projet ⁶

		Partenaire 3	
Civilité ⁷	Nom ⁴	Prénom ⁴	
M	Bézivin	Jean	
Grade⁴	Professeur des Universités		
Mail ⁴	Jean.Bezivin@univ-nantes.fr		
Tél ⁴	02 51 12 58 13	Fax ⁴	02 51 12 58 12

Laboratoire ⁴ (nom complet)			
Laboratoire Informatique de Nantes Atlantique LINA CNRS FRE 2729			
N° Unité (s'il existe)	FRE 2729		
Adresse complète du laboratoire ⁴			
Université de Nantes - 2 rue de la Houssinière 44322 Nantes cedex 3 – France			
Ville ⁴	Nantes	Code postal ⁴	44322
Région ⁴	Pays de Loire		
Organismes de tutelle (indiquer le ou les établissements et organismes de rattachement, souligner l'établissement susceptible d'assurer la gestion du projet) :			
<u>LINA</u> , Université de Nantes, CNRS FRE 2729.			

Principales publications :

⁶ Remplir une fiche par équipe partenaire

⁷ Champ obligatoire

1. J. Bézivin: *On the Unification Power of Models*. In Software and System Modeling (SoSym), Volume 4, Number 2, May 2005, pp 171-188, Springer-Verlag.
2. J. Bézivin: *Sur les principes de base de l'ingénierie des modèles*. In RSTI-L'Objet, Volume 10, Number 4, pp145-157, October 2004.
3. J. Bézivin, E. Breton: *Applying the basic principles of model engineering to the field of process engineering*. In CEPIS, UPGRADE, The European Journal for the Informatics Professional, Volume V, Number 5, 2004.
4. I. Kurtev, J. Bézivin, M. Aksit: *Technological Spaces: An Initial Appraisal*. CoopIS, DOA'2002 Federated Conferences, Industrial track, Irvine, 2002
5. J. Bézivin, O. Gerbé: *Towards a Precise Definition of the OMG/MDA Framework*, Automated software Engineering, ASE'2001, San Diego, Californie, USA, (2001).

Partenaire 3

	Nom	Prénom	Emploi actuel	Discipline (à renseigner uniquement pour SHS)	% de temps de recherche consacré au projet	Rôle/Responsabilité dans le projet 4 lignes max
<i>exemple</i>	<i>MARTIN</i>	<i>Charles</i>	<i>Professeur</i>		<i>30%</i>	<i>Caractérisation des facteurs de transcription recombinants en système in vitro ...</i>
Responsable	Bézivin	Jean	Professeur		40%	Conduite du projet et encadrement du doctorant
Membres de l'équipe						

Pour chacun des membres de l'équipe dont l'implication dans le projet est supérieure à 25%, fournir une biographie **d'une page maximum** qui comportera :

Jean Bézivin, Professeur des Universités, Université de Nantes

Jean Bézivin

Né le 22 Février 1946

DEA, Université de Grenoble 1, 1970.

Doctorat, Université de Rennes 1, 1975

Habilitation à Diriger des Recherches, Université de Rennes 1, 1987.

Professeur des Universités depuis 1988, 1^{ère} classe depuis 2004.

Responsable adjoint de l'équipe de recherche ATLAS.

Co-fondateur de la série de conférences Tools

Co-fondateur et membre du steering committee des conférences ECOOP et Models/UML

Jean Bézivin a été co-organisateur de l'école de Dagstuhl en 2004 sur l'ingénierie des langages pour le développement par modèles. Il a été éditeur invité de SoSym en 2004. Il a été membre de comité de programmes pour des conférences dans les domaines liés à l'ingénierie des modèles : Models/UML, EDOC, GPCE, etc. Il a assuré un cours à l'école CEA/INRIA/EDF de 2003. Il a conduit un groupe de travail de l'OFTA sur l'ingénierie des modèles de 2000 à 2003.

Autres expériences professionnelles

1970 - 1972 Assistant à l'Université de Rennes 1
1972-1973 Research Fellow à Queen's University, Belfast
1973-1974 Research Associate à Concordia University, Montreal
1974-1978 Assistant à l'Université de Rennes 1
1978-1988 Maître de conférences à l'Université de Brest
1988-2005 Professeur à l'Université de Nantes

Distinction

2004 IBM Eclipse Innovation Grant pour le développement d'un environnement de transformation de modèles.

Sélection de publications

J. Bézin: *On the Unification Power of Models*. In Software and System Modeling (SoSym), Volume 4, Number 2, May 2005, pp 171-188, Springer-Verlag.

J. Bézin: *Sur les principes de base de l'ingénierie des modèles*. In RSTI-L'Objet, Volume 10, Number 4, pp145-157, October 2004.

J. Bézin, E. Breton: *Applying the basic principles of model engineering to the field of process engineering*. In CEPIS, UPGRADE, The European Journal for the Informatics Professional, Volume V, Number 5, 2004.

I. Kurtev, J. Bézin, M. Aksit: *Technological Spaces: An Initial Appraisal*. CoopIS, DOA'2002 Federated Conferences, Industrial track, Irvine, 2002

J. Bézin, O. Gerbé: *Towards a Precise Definition of the OMG/MDA Framework*, Automated software Engineering, ASE'2001, San Diego, Californie, USA, (2001).

Programme blanc 2006

B - Description du projet

La partie (B) devra être rédigée en anglais à l'exception des projets de recherche pour lesquels le français s'impose.

Cette possibilité concerne en particulier les projets en SHS pour lesquels le français peut être utilisé dans le cadre d'une évaluation internationale.

Acronyme ou titre court du projet : Famille de Langages pour Familles de Systèmes
--

B-1 – Objectifs et contexte : (2 pages maximum en Arial 11, simple interligne)

Contexte

Traditionnellement le développement de logiciels ne s'appuie pas sur une connaissance approfondie du domaine ciblé, mais l'introduit de façon ad hoc et partielle avec une démarche peu formalisée et peu outillée, pour satisfaire des objectifs immédiats. Se faisant, les logiciels sont trop souvent appréhendés isolément, rendant difficile toute factorisation conceptuelle ou de mise en œuvre. En fait, il est fondamental de constater que les logiciels appartiennent toujours à une famille. Au sein d'une même famille, les dits logiciels partagent un ensemble de points communs et exhibent des variations spécifiques.

Du point de vue du développement logiciel, une famille de logiciels constitue un *métier*, c'est-à-dire, un vocabulaire, des notations, des règles, des représentations et des protocoles qui sont propres à un domaine. Par exemple, le domaine de la téléphonie comprend un ensemble de concepts, de règles, de protocoles et d'interfaces qui encadrent très précisément le développement de services de téléphonie.

Objectifs

Notre initiative de recherche place le métier au centre du processus de développement de logiciels. Elle a pour ambition de lever les verrous actuels du génie logiciel concernant la production à grande échelle, la robustesse, la fiabilité, la maintenance et l'évolutivité des composants logiciels. Pour ce faire, l'innovation au cœur de cette initiative, consiste à concevoir un processus de développement logiciel en fonction d'un métier particulier (ou domaine applicatif). Ce processus couvre toutes les étapes du génie logiciel et combine les trois approches émergentes suivantes :

- La modélisation métier (aussi appelée " *ingénierie de modèles* ") ;
- Les langages de programmation métiers (aussi appelés " *Domain-Specific Languages* ") par opposition aux langages généralistes et monolithiques à la C ou Java ;
- La programmation générative en particulier les aspects comme vecteurs de transformation de modèles et de programmes généralistes.

D'un point de vue génie logiciel, ces trois approches ont déjà démontré isolément des bénéfices génie logiciel concrets et reconnus ; leur combinaison permettra d'appréhender la totalité du processus de développement logiciel dédié à un métier donné.

La modélisation métier capture et manipule les concepts spécifiques à un domaine et met en œuvre un processus de conception qui respecte le cahier des charges de ce domaine. Elle transfère aux différentes corporations (gestionnaires, biologistes, ...) la responsabilité de l'expression de leur besoins propres permettant aux informaticiens de se (re) concentrer sur l'ingénierie des langages.

Le langage de programmation métier prolonge donc la modélisation métier au niveau du processus de programmation. Le langage métier est un langage de programmation haut niveau qui offre des abstractions et des notations spécifiques à un domaine. En outre, il définit un cadre de programmation strict dans lequel des propriétés critiques au métier peuvent être garanties et vérifiées.

La programmation générative traite de la génération automatique d'un logiciel à partir d'un programme écrit dans le langage métier. Le très haut niveau d'abstraction du langage métier requiert des approches génératives puissantes telles que les aspects, les annotations, la métaprogrammation et de manière plus générale diverses techniques de transformation de programmes pour traduire ces langages métiers

vers des langages généralistes. Ces techniques sont garantes de la portabilité, de fiabilité, de l'efficacité, de la réutilisation et de l'adaptabilité du " composant logiciel " généré.

Finalement, l'approche par aspects, pour les modèles de conception et les langages de programmation, permet d'abord, en séparant les préoccupations, d'adapter incrémentalement une application à un domaine, et ce, par tissage de propriétés transverses. Ce tissage doit aussi, à terme, permettre d'appréhender la programmation d'une application comme une coordination de plusieurs langages métiers capturant chacun une de des facettes spécifiques (à l'image de la coordination des différents corps de métier dans le bâtiment et de la superposition de leurs différents plans d'exécution).

L'étude de notre nouveau processus de développement logiciel sera concrétisé par un atelier de génie logiciel configurable par un métier et incluant une palette d'outils de génération de programmes. Cet atelier sera lui-même validé par une étude de cas, avec démonstrateurs, conduite dans le domaine de la téléphonie.

B-2 – Description du projet et résultats attendus : (8 pages maximum en Arial 11, simple interligne)

L'originalité et le caractère ambitieux du projet devront être explicités. L'interdisciplinarité et l'ouverture à diverses collaborations seront à justifier en accord avec l'orientation du projet. La capacité de ou des équipes "porteuse(s)" devra être attestée par la qualification et les productions scientifiques antérieures de leurs membres. Leurs rôles dans les différentes phases du projet devront être précisés et la valeur ajoutée des collaborations entre les différentes équipes sera argumentée. On décrira le déroulement prévisionnel et les diverses phases intermédiaires ainsi que les méthodologies employées. Les moyens demandés devront être en accord avec les objectifs scientifiques du projet.

Uniquement dans le domaine des sciences humaines et sociales, les projets de recherche peuvent impliquer la production de données statistiques. Dans ce cas l'accès au financement de l'ANR implique l'obligation de déposer ces données, documentées, dans un centre d'archivage et de diffusion auprès des chercheurs, et de les mettre à disposition de la communauté scientifique (éventuellement au terme d'un embargo de durée déterminée).

Résultats attendus

Les résultats de notre projet couvrent toutes les étapes du processus de génie logiciel proposé. De ce point de vue notre action doit être vue comme interdisciplinaire au sein du génie logiciel. De manière plus précise nous souhaitons établir un continuum entre la conception (l'ingénierie des modèles), la programmation (l'ingénierie des langages de programmation) et l'implémentation (l'ingénierie des intergiciels et des plates-formes).

Tout d'abord, nous élaborerons une nouvelle méthodologie de développement de logiciels prenant le métier cible comme point de départ. Cette méthodologie s'appuiera sur un atelier de génie logiciel configurable pour un métier donné et intégrant des outils de transformation et d'exécution de modèles et de langages. Notre projet sera validé par une application dans le domaine de la téléphonie, et des réalisations annexes dans les domaines de l'ordonnancement des systèmes d'exploitation, de la gestion de règles métier, de la gestion des données hétérogènes et des flots de données.

Méthodologie

La méthodologie proposée vise donc à couvrir le continuum modélisation-programmation-implémentation par l'utilisation de chaînes de langages et de traducteurs associés. Ces langages vont permettre de capturer des aspects différents, allant des besoins utilisateurs à la prise en compte des caractéristiques des plates-formes d'implémentation. Chaque étape de la méthodologie devra alimenter l'étape suivante.

La phase modélisation de notre projet comprend trois volets distincts : la modélisation métier, les contraintes et les transformations. Chaque volet nécessite la conception d'un environnement de métamodélisation et les outils associés.

- Modélisation métier. Elle vise à définir des modèles offrant des notations, graphiques et/ou textuelles, représentant des abstractions spécifiques à un métier donné. Ces notations peuvent s'appuyer totalement ou partiellement sur des notations déjà utilisées par les experts du domaine visé, plaçant ainsi ces acteurs au centre du processus. Dans notre approche, les modèles ont plutôt une nature déclarative et non exécutable, pour permettre d'élaborer des spécifications évolutives, composables et de haut niveau.
- Modélisation des contraintes. Ce volet permet de définir des règles de vérification de modèles permettant d'assurer la conformité d'un modèle donné à un jeu de contraintes particulières. Ces règles rendent explicites des contraintes métiers émanant de réglementations spécifiques, de connaissances informelles, etc.
- Modélisation des transformations. Un modèle de transformation définit des règles de transformation permettant de générer un programme en langage métier, à partir d'un modèle donné. En ciblant un langage métier, cette étape de transformation permet un meilleur étagement du processus de production de logiciels, réduisant ainsi le fossé sémantique entre le langage source et le langage cible. Cette approche diffère de celle, plus traditionnelle, consistant à introduire une chaîne, plus ou moins complète, de transformations de modèles menant à une implémentation dans un langage et un environnement généralistes.

La phase programmation s'appuie sur l'introduction d'un langage métier. Comme pour la modélisation, ce langage offre des notations et des abstractions spécifiques au métier visé et peut conserver une nature déclarative. Toutefois, les programmes écrits dans ce langage sont exécutables. En outre, le langage métier détaille des considérations opérationnelles, sciemment ignorées jusque la :

- Des contraintes opérationnelles (ressources, contraintes temps réel...);
- Une architecture logicielle (qui peut prendre la forme d'une machine virtuelle métier);

- Des protocoles métiers.

La phase implémentation a pour objectif de générer une implémentation à partir d'un programme écrit dans un langage métier. Parce qu'un langage métier est haut niveau, son processus de compilation est généralement plus compliqué que celui d'un langage généraliste. De fait, il nécessite des outils spécifiques également de haut niveau pour s'assurer de sa fiabilité. En fait :

- La vérification de programmes peut requérir des analyses statiques et dynamiques.
- La génération de code vise à établir un chemin vers un langage généraliste, nécessitant l'introduction d'un ensemble de détails d'implémentation (bibliothèques, composants, squelettes...).

Notre projet s'appuie sur l'utilisation d'approches et d'outils de programmation générative contribuant à la définition du processus de compilation d'un langage métier. Nous comptons ainsi utiliser l'approche par aspects pour une compilation progressive d'un programme métier en programme généraliste. Les aspects seront également utilisés pour effectuer certaines vérifications statiques et insérer (tisser) du code pour reporter d'autres vérifications à l'exécution du programme généré. L'utilisation d'autres techniques de programmation générative, telle que la spécialisation de programmes, permettra de générer du code générique sans perte de performance.

Atelier génie logiciel

Une tâche essentielle au projet consistera à développer un atelier de génie logiciel sous environnement Eclipse incluant une palette d'outils de programmation générative. Pour des raisons de commodité et d'efficacité nous partirons du projet GMT (Generative Model Transformer) déjà ouvert (<http://eclipse.org/gmt/>) et destiné à recevoir des prototypes de recherche. Ceci nous permet un démarrage beaucoup plus rapide du projet. En particulier, les outils d'ATLAS (le transformateur de modèles ATL ou le tisseur de modèle AMW) sont déjà disponibles sur le site et constituent une plateforme de production de langages métier AMMA (ATLAS Model management Architecture). L'objectif est donc d'ouvrir de nouveaux sous projets de GMT correspondant aux activités proposées ici. Notre contribution globale consistera à enrichir GMT non seulement par plusieurs prototypes correspondant à un ou plusieurs sous-projets complémentaires, mais aussi par des démonstrateurs et des validations sur les trois domaines applicatifs (téléphonie, informatique ambiante, systèmes d'exploitation).

En fait l'ambition du projet consiste à concevoir une véritable plate-forme expérimentale de production et d'évaluation de langages métiers en s'appuyant essentiellement sur le cas d'étude de la téléphonie, et en s'inspirant également de cas d'étude secondaires (ordonnancement dans les systèmes d'exploitation, satisfaction de contraintes et gestion de règles métier, etc.). On donnera une importance particulière à l'étude des chaînes de traducteurs et plus généralement à la gestion globale des langages et des programmes exprimés dans ces langages.

Application au domaine de la téléphonie (SPL)

Le domaine des services de téléphonie est actuellement en évolution rapide suite à la convergence des réseaux de télécommunications, des réseaux informatiques et du multimédia. Le protocole SIP (Session Initiation Protocol) est au centre de cette évolution en proposant une approche complète pour la téléphonie sur IP. Il a été standardisé par l'IETF, adopté par l'ITU et utilisé, par exemple, dans l'UMTS. Outre les fonctions de téléphonie classique, SIP permet de gérer la vidéo, la présence, les alertes, la messagerie instantanée, mais peut également être utilisé pour les jeux en réseau, les tableaux partagés, ainsi qu'une multitude de supports multimédia.

La programmation de services de téléphonie pour une plate-forme SIP est souvent présentée comme l'un de ses atouts majeurs. En effet, c'est le moyen de traduire la multitude de fonctionnalités offerte par SIP en nouveaux services pour l'utilisateur. L'offre de services est donc un paramètre important du succès de SIP.

Toutefois, la téléphonie est appréhendée par le grand public comme une ressource indispensable, au même titre que l'eau et l'électricité. Par conséquent, la "programmabilité" offerte par une plate-forme SIP crée un risque de compromettre la fiabilité de cette ressource. En effet, elle permet à tout programmeur de développer des services dont le manque de fiabilité peut perturber, voir interrompre, la disponibilité de cette ressource.

En outre, la programmation de services SIP demande un éventail très large de compétences, en plus de celles, peu répandues, du domaine de la téléphonie : réseaux, systèmes distribués, protocoles annexes, etc.

Pour faciliter le développement de services de téléphonie, l'équipe Phoenix conçoit un langage de programmation métier, nommé SPL (*Session Processing Language*). La sémantique de SPL a été formalisée, ce qui permet de vérifier des propriétés sur les services écrits dans ce langage et ainsi de garantir leur fiabilité. Au-delà de la représentation textuelle, il serait souhaitable d'introduire une version graphique de SPL pour rendre la programmation de services accessibles à des non-informaticiens.

Dans le cadre de ce projet, Phoenix propose de modéliser la spécification de services de téléphonie et de définir les méta-données nécessaires à la compilation de modèles en SPL. De plus, nous développerons un compilateur basé sur des approches et outils de programmation générative pour cible deux plates-formes SIP reconnues dans l'industrie : JAIN SIP et Microsoft Live Communications.

Autres domaines d'application

Le domaine métier central sur lequel se fera l'expérimentation est celui de la téléphonie. Cependant d'autres métiers seront pris en compte de façon secondaire, essentiellement par les équipes ATLAS et OBASCO. Ils permettront de démontrer que les résultats ne sont pas valables uniquement dans un domaine, mais peuvent s'appliquer génériquement sur différents domaines. OBASCO propose une étude de l'ordonnancement dans les systèmes d'exploitation et ATLAS propose deux études, l'une dans le domaine des règles métier et des moteurs de règles et l'autre dans la gestion de données et de flots de données.

L'ordonnanceur est le service d'un système d'exploitation qui choisit quel processus doit s'exécuter sur le processeur à un instant donné. De fait, l'ordonnanceur a un rôle clé dans le respect des échéances temporelles et plus généralement de la qualité de service (QoS) que doivent satisfaire les applications multimédia. Malheureusement, les ordonnanceurs présents dans les systèmes patrimoniaux sont souvent limités aux techniques de priorité ou de tourniquets, ce qui rend très difficile le respect de contraintes de QoS. Idéalement, on souhaiterait pouvoir associer à une classe d'application spécifique un ordonnanceur approprié et satisfaisant les contraintes de QoS. Cependant le développement d'un ordonnanceur au sein d'un noyau de système nécessite une expertise "métier" très pointue que peu de programmeurs possèdent. Un traitement linguistique de cette problématique devrait pouvoir permettre de réaliser des avancées importantes.

On a vu apparaître un nombre très important de formalismes de règles métier dans les dernières années, formalismes d'ailleurs souvent basés sur XML. La différence entre ces multiples langages correspond au contexte d'utilisation, au niveau d'expertise des utilisateurs ainsi qu'à d'autres paramètres comme le niveau de préoccupation normative. Maintenant ces différents langages doivent permettre de générer des programmes exécutables, éventuellement sur une plate-forme spécialisée. Une cible classique est un moteur de règles comme ceux fournis par la société ILOG ou par ses concurrents. Du côté de la plate-forme on trouve donc des langages comme IRL (ILOG Rule Language) tandis que du côté des utilisateurs on trouve des langages comme PRR (Production Rule Representation) récemment proposé à l'OMG (<http://www.w3.org/2004/12/rules-ws/paper/53/>). La prise en compte des aspects utilisation et/ou implémentation dans ces langages constitue un excellent domaine d'expérimentation pour les DSLs, d'autant plus que le nombre de proposition de formalismes est très important (RBML, Blaza SRL, RuleML, etc.). Une collaboration avec ILOG ou certains de ses concurrents comme Fair Isaac a déjà permis de valider l'intérêt de ce type d'approche.

L'un des défis industriels les plus importants aujourd'hui est de permettre la prise en compte de la multiplicité des formats de données par des langages métiers. Dans des corporations comme AT&T des avancées de recherche importantes ont été réalisées récemment par l'utilisation de DSL de description de données. L'un des exemples typiques est PADS (<http://www.padsproj.org/>). Même si les chercheurs américains ont une avance considérable dans ce domaine (voir par exemple Yitzhak Mandelbaum. *The Next 700 Data Description Languages. POPL*, January 2006), il est possible d'intégrer partiellement cette vision dans notre projet.

Allant au-delà de la description des formats de données, l'approche des familles de langages peut également permettre la prise en compte de réseaux complexes de traitement de flots de données. Les capteurs de données ne produisent plus actuellement de simples flots de bits ou d'octets, mais des données complexes pouvant se caractériser par des formalismes précis de définition de métadonnées. La surveillance de phénomènes sismiques dangereux ou la construction d'un système aéronautique ou

médical complexe s'apparentent à la gestion d'un nombre important de langages dédiés participant à un réseau de traitement (distributed data-centric stream management network). La prise en compte de ce troisième type d'application permet d'élargir le champ d'investigation sur les familles de langages.

Programme de travail

Notre programme de travail se décompose en cinq volets principaux : lancement, modélisation, programmation, implémentation et démonstrateurs. Par chacun de ces volets, nous listons les équipes coordinatrices et les tâches principales. Hormis le premier, chaque volet est conduit par au moins deux équipes pour couvrir tous les aspects en amont et en aval ; les autres équipes pourront toutefois effectuer des tâches spécifiques. Le volet démonstrateurs est quant à lui coordonné par l'équipe Atlas qui interviendra transversalement aux différents démonstrateurs développés par les deux autres équipes : Obasco et Phoenix. Au-delà de son soutien en matière de modélisation, l'équipe Atlas élaborera des procédures de méthodologie et de contrôle permettant de concrétiser le continuum prôné par notre projet.

Présentation des tâches

1. Lancement

Participants principaux : Atlas, Obasco et Phoenix.

Tâches :

- Définition des terminologies et concepts qui seront communs à toutes les équipes de ce projet de recherche. Ce travail est capital pour le bon déroulement de nos travaux ; il sera raffiné tout au long du projet. Il fera l'objet d'une contribution auprès de la communauté génie logiciel.
- Identification, évaluation et intégration des méthodologies, techniques et outils utilisés dans le cadre de ce projet de recherche. Cette identification constituera la synthèse des propositions émanant de chaque partenaire du projet. Ce processus sera itéré pendant toute la durée du projet.

2. Modélisation

Participants principaux : Atlas et Phoenix

Tâches :

- Etude et définition d'un environnement global de gestion de modèles. Cet environnement permettra de gérer la complexité additionnelle provenant de la prise en charge nécessaire d'un grand nombre de composants de modélisation de natures diverses (modèles, métamodèles, transformations, contraintes, injecteurs, extracteurs, etc.). Pratiquement il s'agit essentiellement de fournir aux niveaux de la gestion des modèles des fonctionnalités équivalentes à ce que fournissent des outils comme Make, ANT ou Maven au niveau de la gestion du code.
- Réalisation concrète sous Eclipse de l'environnement de gestion de modèles.
- Etude et définition d'un outil de spécification de règles de vérification de modèles. Cet outil devra s'intégrer dans l'environnement présenté ci-dessus. Il pourra s'appuyer sur le langage OCL mais devra être applicable quel que soit le métamodèle, c'est-à-dire pas seulement pour des langages de la famille UML.
- Réalisation concrète sous Eclipse de l'outil de spécification et de validation de règles.
- Etude et définition d'un langage étendu de transformation de modèles en programmes écrits dans un langage métier donné. Le point de départ sera le langage ATL qui devra être étendu notamment pour pouvoir générer des syntaxes abstraites et concrètes correspondant aux langages métier.
- Réalisation concrète sous Eclipse du langage étendu de transformation de modèles.

3. Programmation

Participant principaux : Phoenix et Obasco

Tâches :

- Définition d'une méthodologie de conception de langages métiers, prenant en compte l'ingénierie des domaines en amont, et la programmation générative en aval.
- Identification et évaluation d'outils de traitement allant de la représentation textuelle d'un programme à une représentation arborescente.

4. Implémentation

Participants principaux : Obasco et Phoenix

Définition d'un processus de compilation haut niveau pour des langages métiers, basé sur les techniques et outils de programmation générative.

Définition d'un langage d'aspects pour la compilation.

i. Définition d'un langage d'aspects pour la vérification de propriétés.

ii. Identification de techniques et d'outils de programmation générative pour l'optimisation de programmes issus de langages métiers.

Définition de l'architecture d'un atelier intégrant les techniques et outils utilisés.

5. Démonstrateurs

Coordinateur : Atlas

a) Services de téléphonie

Participants principaux : Phoenix

iii. Modélisation : utilisation d'un environnement pour la modélisation des services de téléphonie.

iv. Programmation : Interfaçage avec le langage métier SPL.

v. Implémentation : définition d'un processus de compilation pour SPL composé de techniques et d'outils de programmation générative et ciblant les plates-formes de téléphonie JAIN SIP et Microsoft Live Communications.

b) Autres exemples

Participants principaux : Obasco et ATLAS

Organisation

Planification

Le tableau ci-dessous présente l'implication de chaque partenaire dans chaque volet du projet. Chaque étape (S_i) du projet correspond à un semestre.

Sem.	Atlas	Obasco	Phoenix
S ₁	Volet 1	Volet 1	Volet 1
S ₂	Volet 2	Volet 3	Volet 3
S ₃	Volet 2 (vers 3)	Volet 3	Volet 3
S ₄	Volet 2 (vers 3 & 4)	Volet 4	Volet 4
S ₅	Volet 5	Volet 5	Volet 5
S ₆	Volet 5	Volet 5	Volet 5

Interaction entre les partenaires

Un atelier de travail réunissant tous les partenaires aura lieu chaque trimestre dans les différents sites. Ces ateliers de travail auront de multiples objectifs.

- Une présentation des résultats intermédiaires de chaque partenaire.
- Une évaluation et une validation des techniques et outils produits et/ou identifiés par chaque partenaire.
- Une évaluation comparative et une validation des différentes étapes de développement des démonstrateurs du projet.
- L'analyse critique des choix effectués (concepts, techniques, outils, etc.) et définition d'une tâche de raffinement et/ou d'itération sur ces choix.

Originalité et caractère ambitieux du projet

La modélisation d'un domaine applicatif donné est maintenant fréquemment effectuée préalablement au développement proprement dit. Toutefois, peu de contraintes fortes pèsent sur les résultats de l'étape de modélisation, puisque celle-ci n'est pas nécessairement reliée au processus génération de logiciels. Ainsi, c'est encore trop souvent la phase de programmation qui permet d'identifier des incomplétudes ou des erreurs dans la modélisation.

A l'autre extrême de la programmation, l'utilisation d'un langage généraliste ne permet pas d'introduire des éléments spécifiques au domaine. Ces éléments doivent être codés dans le langage généraliste ; cette approche les rend souvent inexploitable pour des outils de vérification ou d'optimisation.

Notre projet de recherche innove en proposant une approche globale au développement de logiciels. Le métier visé constitue la dimension transversale du processus de développement. Par ailleurs, nous proposons de coupler fortement le développement de notre approche à des applications concrètes. Ainsi, notre projet de recherche comprend trois études de cas variées et réalistes (téléphonie, systèmes d'exploitation et informatique ambiante), avec le développement de démonstrateurs. Ce processus de conception conjointe permettra d'aboutir à une approche validée.

Enfin au niveau académique, les travaux réalisés devront permettre de préciser les intuitions de la programmation générative au sens de Czarnecki & Eisenecker en explicitant les relations entre les concepts de modèles, les langages dédiés et les aspects.

Équipes porteuses et collaborations

Les trois équipes partenaires de ce projet ont déjà eu des collaborations dans le cadre de projets nationaux et internationaux.

Ainsi, le projet Obasco participe au "langage lab" du nouveau réseau d'excellence AOSD <http://www.aosd-europe.net/index.htm> (septembre 2004, août 2007) et travaille à la définition d'un métamodèle de langages d'aspects permettant de rendre compte des différents langages de coupes et d'actions existants.

L'équipe Atlas participe quant à elle à l'IST Modelware <http://www.modelware-ist.org/> et au nouveau projet ModelPlex. Elle bénéficie d'un financement émanant du consortium Eclipse pour développer son langage de transformation de modèles <http://www.eclipse.org/>.

Les membres du projet ATLAS ont participé à l'action spécifique CNRS MDA dont le thème était l'ingénierie dirigée par les modèles.

Les membres de OBASCO et de Phoenix ont collaboré sur le développement d'un langage métier pour les pilotes de périphériques. Ils ont récemment initié une étude sur l'utilisation des techniques de programmation générative pour la compilation des langages métiers.

B-3 – Justification scientifique des moyens demandés pour chaque équipe partenaire impliquée dans le projet.

On présentera ici une justification scientifique des moyens demandés pour chacun des partenaires impliqués dans le projet, en distinguant les demandes en équipement, fonctionnement, personnels. Pour les demandes d'équipement, préciser si les achats envisagés doivent être complétés par d'autres sources de crédits, le montant et l'origine des crédits complémentaires qui seront utilisés.

Les moyens demandés dans le cadre de ce projet sont de trois ordres : un doctorant pour chaque partenaire et pour trois ans, un budget mission pour les ateliers de travail trimestriels et un budget mission pour la participation à des conférences internationales dans les thématiques de notre projet. Par ailleurs, nous organiserons un événement de portée internationale à la fin du projet. Cet événement aura pour but de diffuser les résultats obtenus et d'associer le plus largement possible les chercheurs des trois communautés que couvre notre projet. Enfin l'organisation de la vingtième édition de la conférence ECOOP à Nantes en juillet 2006 (voir <http://2006.ecoop.org/>) sera déjà l'occasion de mettre en place l'approche prônée en la resituant dans la perspective post-objets, par le lancement du premier workshop international sur le sujet (voir <http://phoenix.labri.fr/DSPD/>).

Le doctorant présent dans chaque équipe aura pour mission d'intégrer et d'étendre les outils existants de chaque partenaire et d'en développer de nouveaux pour les besoins du projet. Par ailleurs, il contribuera au développement des démonstrateurs du projet en synergie avec le projet Eclipse GMT. Les outils et les démonstrateurs de notre projet sont des enjeux importants dont la réalisation permettra d'avoir un impact significatif dans le domaine du génie logiciel et des trois domaines applicatifs étudiés. Le sujet applicatif de chaque doctorant portera sur l'un des domaines mentionnés (téléphonie, ordonnancement de systèmes, règles métier, description de données, gestion de flots "streaming", etc.)

Partenaire 1-2-3 : chacune des partenaires demande le financement d'un doctorant pour 3 ans ainsi que le financement des frais de mission pour un montant approximatif de 100 000€ chacun. L'idée bien sur est de créer une synergie entre ces trois thésards en développant notre programme de recherche.

Propositions d'experts et confidentialité

Les membres du comité d'évaluation et du comité de pilotage sont astreints à la confidentialité.

- Possibilité de fournir une liste de 3 à 5 noms d'experts français ou étrangers (avec coordonnées complètes : adresse postale et adresse électronique) susceptibles d'évaluer le projet avec lesquels les équipes participant au projet n'ont ni conflit d'intérêt, ni collaborations en cours.
 - Possibilité éventuelle de fournir une liste de 5 noms max. d'experts auxquels les participants au projet ne souhaitent pas que le projet soit envoyé s'il y a risque de conflits d'intérêts.
- a. **Dave Thomas** dave@bedarra.com and www.davethomas.net., Ottawa Office, Suite 412, 1 Stafford Road, Ottawa Ontario, Canada K2H 1B9; Phone or fax: 1-888-299-144. D. Thomas is cofounder/chairman of Bedarra Research Labs (www.bedarra.com), www.Online-Learning.com and the Open Augment Consortium (www.openaugment.org) and a founding director of the Agile Alliance (www.agilealliance.com). He is an adjunct research professor at Carleton University, Canada and the University of Queensland, Australia. Dave is the founder of and past CEO of Object Technology International (www.oti.com) creators of the Eclipse IDE Platform, IBM VisualAge for Smalltalk, for Java, and MicroEdition for embedded systems.
 - b. **Gregor Kiczales** gregor@cs.ubc.ca , Professor Software Practices Lab, Department of Computer Science, University of British Columbia, 201-2366 Main Mall,, Vancouver, B.C., Canada V6T 1Z4
 - c. **Jean-Pierre Banâtre** jbanatre@irisa.fr IRISA Campus Universitaire de Beaulieu. 35042 Rennes/France
 - d. **Serge Salicki**, serge.salicki@thalesgroup.com THALES Research and Technology, Advance Software Department, Domaine de Corbeville, 91404 Orsay cedex – France, Tel: + 33 (0)1 69 33 09 14, Fax : +33 (0)1 69 33 08 65, Cell: 06 80 58 21 70
 - e. **Kathleen Milsted** kathleen.milsted@francetelecom.com Distributed Systems Architecture, France Telecom, RD / MAPS / GRE, 28 Chemin du Vieux-Chene - BP 98, 38243 Meylan Cedex, 38243 Maylan, France, tel: +33-4-76 76 45 94

Programme blanc 2006

C - Moyens financiers et humains demandés par chaque équipe partenaire du projet

Chaque équipe partenaire remplira une fiche de demande d'aide selon les modèles proposés ci-dessous (laboratoire public ou fondation ; entreprise ou association) en fonction de son appartenance.

Programme blanc 2006

Fiche de demande d'aide – Entreprise / Association

Acronyme ou titre court du projet Familles de Langages pour Familles de Systèmes (FLFS)

Partenaire n°1 LINA/EMN/Centre Armines Responsable scientifique : Cointe Pierre

Calcul de l'aide demandée à l'ANR et estimation du coût complet du projet pour le partenaire :

				Euros HT
	Nbre Homme. mois	Coût Homme. mois Salaire chargé	Nombre de personnes impliquées	
Dépenses de personnel ⁽¹⁾ catégorie 1 catégorie 2 etc.	12,6	8601	2	108471
Dépenses de personnel non permanent à recruter ⁽²⁾ catégorie 1 catégorie 2 etc.	36	3000		108000(Q)
Amortissements des équipements (>4000 €) Nature et justification de la dépense				(R)
Petits matériels, consommables, fonctionnement, etc.				(S)
Frais de missions si montant >5% de la somme demandée, justification de la dépense				9000(T)
Prestations de service externes ⁽³⁾ , sous-contractant				(U)
Prestation de service interne à l'entreprise ou à l'organisme				(V)
Total frais fonctionnement				9000 =S+T+U+V
Frais généraux (assistance, encadrement, coût de structure) ⁽⁴⁾				18037(Y)
Coût complet du projet				243 508
Assiette de l'aide ⁽⁵⁾				243 508C
Aide demandée ⁽⁶⁾ Voir le fichier "informations complémentaires"				100 000

- (1) Il s'agit du personnel qui serait affecté au projet mais qui est présent dans le laboratoire ou l'entreprise indépendamment de la réussite de l'appel de l'agence. Salaire mensuel chargé (charges salariales et patronales). Pour les enseignants-chercheurs ne compter que la part salariale correspondant à la part recherche (50% du salaire pour 100% de temps consacré à la recherche).
5 grandes catégories (CDD ou CDI) : Ingénieur, chercheur, enseignant chercheur, technicien, autres. Lorsque dans une même catégorie plusieurs personnes de salaire différent sont mentionnées indiquer la valeur moyenne. Pour les laboratoires publics ou fondation, ces données ne servent qu'à calculer le coût complet du projet.
- (2) Personnel non statutaire directement affecté au projet exprimé en hommes mois. Les dépenses éligibles se limitent aux salaires et aux charges sociales. Exemple : post-doc (catégorie 1), ingénieur d'études (catégorie 2), etc.
- (3) Propriété intellectuelle, location de matériel, service, etc.
- (4) Pour les associations et TPE, les frais généraux peuvent être au maximum = 4% de R + 8% de (P+Q+S+T+U). Pour les sociétés civiles, les entreprises hors TPE, les GIE, les centres techniques, les frais généraux peuvent être au maximum de = 7% de (R+S+T+U) + 68% de (P+Q)
- (5) Assiette de l'aide signifie dépenses éligibles à une aide de l'agence
- (6) En cas d'aide accordée par un autre financeur sur les mêmes dépenses que celles listées dans le tableau, il peut y avoir une diminution de l'aide accordée par l'ANR pour rester conforme à la réglementation.

Programme blanc 2006

Fiche de demande d'aide – Laboratoire public / Fondation

Acronyme ou titre court du projet

Familles de Langages pour Familles de Systèmes (FLSF)

Partenaire 2 - Coordinateur : Consel Charles

Calcul de l'aide demandée à l'ANR et estimation du coût complet du projet pour le laboratoire du partenaire

Avant de remplir ce tableau il vous faut décider quel sera votre établissement gestionnaire (cf notes 5 et 6 en bas de page)

				Euros HT	Taux spécifiques à chaque établissement	
	Nbre Homme .mois	Coût Homme.mois (salaire chargé)	Nombre de personnes impliquées			
Dépenses de personnel ⁽¹⁾						
Professeur	5	7 500	1	37 500	1,2	77 400
Maître de Conférences	6	4 500	1	27 000		
Dépenses de personnel non permanent à recruter ⁽²⁾						
CDD (Doctorant)	36	2 300	1	82 800	2,5	207 000
Equipements (>4000 €) détail § B-3						0
Petits matériels, consommables, fonctionnement, etc				3 000	16,27%	3 488
Frais de missions si montant > 5% de la somme demandée, justification § B-3				5 000	16,27 %	5 813
Prestations de service externes, sous-contractant ⁽³⁾						0
Total des dépenses de fonctionnement				8 000		9 301
Frais généraux (assistance, encadrement, coût de structure) (max 4 % du coût total des dépenses)						3 684
Assiette de l'aide ⁽⁴⁾						95 785
Aide demandée ≤ Z ⁽⁵⁾						95 785

Coût complet du projet ⁽⁶⁾

297 385

- (1) Il s'agit du personnel qui serait affecté au projet mais qui est présent dans le laboratoire ou l'entreprise indépendamment de la réussite de l'appel de l'agence. Salaire mensuel chargé (charges salariales et patronales). Pour les enseignants-chercheurs ne compter que la part salariale correspondant à la part recherche (50% du salaire pour 100% de temps consacré à la recherche).

5 grandes catégories (CDD ou CDI) : Ingénieur, chercheur, enseignant chercheur, technicien, autres. Lorsque dans une même catégorie, plusieurs personnes de salaire différent sont mentionnées indiquer la valeur moyenne. Pour les laboratoires publics ou fondations, ces données ne servent qu'à calculer le coût complet du projet.

- (2) Personnel non statutaire à recruter pour le projet exprimé en hommes mois. Les dépenses éligibles se limitent aux salaires et aux charges sociales. Exemple : post-doc (catégorie 1), ingénieur (catégorie 2), technicien, autre.
- (3) Propriété intellectuelle, location de matériel, service, etc.
- (4) Assiette de l'aide signifie dépenses éligibles à une aide de l'agence.
- (5) L'aide demandée doit correspondre au montant HT augmenté éventuellement de la TVA non récupérable. La TVA non récupérable est actuellement, par exemple, de 88% pour le CNRS et l'INRA, de 94% pour l'Inserm et de 100% pour les universités. En conséquence pour une demande qui sera gérée par l'INRA, le taux de TVA non récupérable est $0,88 \times 0,196 = 0,1725$, ce qui conduit à inscrire dans la colonne de droite pour une demande HT de 10 000 euros, $10000 \times (1 + 0,1725)$ soit 11 725 euros soit une demande d'aide de 11 725 euros si le partenaire veut disposer de 10 000 euros dans la réalisation de son projet.
En cas d'aide accordée par un autre financeur sur les mêmes dépenses que celles listées dans le tableau, il peut y avoir une diminution de l'aide accordée par l'ANR pour rester conforme à la réglementation.
- (6) Pour le calcul en coût complet, il faut augmenter le salaire chargé d'un taux d'environnement, qui tient compte des conditions d'environnement des personnels (infrastructure, par exemple). Par exemple, ce taux est à l'heure actuelle de 1,8 pour l'Inserm et le CNRS.

Programme blanc 2006

Fiche de demande d'aide – Laboratoire public / Fondation

Acronyme ou titre court du projet
Familles de Langages pour Familles de Systèmes (FLSF)

Partenaire 2 - Coordinateur (nom, prénom) : Bézivin Jean

Calcul de l'aide demandée à l'ANR et estimation du coût complet du projet pour le laboratoire du partenaire

Avant de remplir ce tableau il vous faut décider quel sera votre établissement gestionnaire (cf notes 5 et 6 en bas de page)

				Euros HT	Taux spécifiques à chaque établissement	
	Nbre Homme .mois	Coût Homme.mois (salaire chargé)	Nombre de personnes impliquées			
Dépenses de personnel ⁽¹⁾						
Professeur	7	7 500	1	52 500	1,2	63 000
Dépenses de personnel non permanent à recruter ⁽²⁾						
CDD (Doctorant)	36	2 300	1	82 800	2,5	207 000
Equipements (>4000 €) détail § B-3						0
Petits matériels, consommables, fonctionnement, etc				3 000	16,27%	3 488
Frais de missions si montant > 5% de la somme demandée, justification § B-3				5 000	16,27 %	5 813
Prestations de service externes, sous-contractant ⁽³⁾						0
Total des dépenses de fonctionnement				8 000		9 301
Frais généraux (assistance, encadrement, coût de structure) (max 4 % du coût total des dépenses)						3 684
Assiette de l'aide ⁽⁴⁾						95 785
Aide demandée ≤ Z ⁽⁵⁾						95 785

Coût complet du projet ⁽⁶⁾ **282 985**

- (1) Il s'agit du personnel qui serait affecté au projet mais qui est présent dans le laboratoire ou l'entreprise indépendamment de la réussite de l'appel de l'agence. Salaire mensuel chargé (charges salariales et patronales). Pour les enseignants-chercheurs ne compter que la part salariale correspondant à la part recherche (50% du salaire pour 100% de temps consacré à la recherche).

5 grandes catégories (CDD ou CDI) : Ingénieur, chercheur, enseignant chercheur, technicien, autres. Lorsque dans une même catégorie, plusieurs personnes de salaire différent sont mentionnées indiquer la valeur moyenne. Pour les laboratoires publics ou fondations, ces données ne servent qu'à calculer le coût complet du projet.

- (2) Personnel non statutaire à recruter pour le projet exprimé en hommes mois. Les dépenses éligibles se limitent aux salaires et aux charges sociales. Exemple : post-doc (catégorie 1), ingénieur (catégorie 2), technicien, autre.
- (3) Propriété intellectuelle, location de matériel, service, etc.
- (4) Assiette de l'aide signifie dépenses éligibles à une aide de l'agence.
- (5) L'aide demandée doit correspondre au montant HT augmenté éventuellement de la TVA non récupérable. La TVA non récupérable est actuellement, par exemple, de 88% pour le CNRS et l'INRA, de 94% pour l'Inserm et de 100% pour les universités. En conséquence pour une demande qui sera gérée par l'INRA, le taux de TVA non récupérable est $0,88 \times 0,196 = 0,1725$, ce qui conduit à inscrire dans la colonne de droite pour une demande HT de 10 000 euros, $10000 \times (1 + 0,1725)$ soit 11 725 euros soit une demande d'aide de 11 725 euros si le partenaire veut disposer de 10 000 euros dans la réalisation de son projet.
En cas d'aide accordée par un autre financeur sur les mêmes dépenses que celles listées dans le tableau, il peut y avoir une diminution de l'aide accordée par l'ANR pour rester conforme à la réglementation.
- (6) Pour le calcul en coût complet, il faut augmenter le salaire chargé d'un taux d'environnement, qui tient compte des conditions d'environnement des personnels (infrastructure, par exemple). Par exemple, ce taux est à l'heure actuelle de 1,8 pour l'Inserm et le CNRS.

Programme blanc 2006

D - Récapitulatif global de la demande financière pour le projet

Acronyme ou titre court du projet Famille de Langages Familles de Systèmes

a-Estimation du coût complet de cette demande

(reporter les valeurs (CC) des fiches des différents partenaires)

	Coût complet
Coordinateur (Partenaire 1)	243 508
Partenaire 2	297 385
Partenaire 3	297 985
...	
Total Le total obtenu doit être le même que celui calculé par le logiciel de soumission	823 878

b-Total de l'aide demandée

(reporter les valeurs (Aide demandée) des fiches des différents partenaires)

	Aide demandée
Coordinateur (Partenaire 1)	100 000
Partenaire 2	95 785
Partenaire n	95785
...	
Total Le total obtenu doit être le même que celui calculé par le logiciel de soumission	291 570

c- Effort en personnel demandé

(reporter les valeurs des fiches des différents partenaires)

	en homme. mois
Coordinateur (Partenaire 1)	36
Partenaire 2	36
Partenaire 3	36
...	
Total Le total obtenu doit être le même que celui calculé par le logiciel de soumission	108

Contrats publics et privés sur les trois dernières années (effectués et en cours)

Nom du membre participant à cette demande	% d'implication	Intitulé de l'appel à projets Source de financement Montant attribué	Titre du projet	Nom du coordinateur	Date début - Date fin
Pierre Cointe	15%	NoE	NoE AOSD	A. Rachid (U Lancaster)	Sept 2004/2007
Gilles Muller	10%	ACI Sécurité Informatique	COSS: Composition et raffinement de systèmes sûrs	Mamoun Fillali	2003-2006
Charles Consel Laurent Réveillère	10% 10%	ACI Sécurité Informatique	COSS: Composition et raffinement de systèmes sûrs	Mamoun Fillali	2003-2006
Gilles Muller Pierre Cointe	12% 8,5%	ANR Blanc	Coccinelle	Gilles Muller	2006-2008
Charles Consel Laurent Réveillère	30 % 15 %	Open development platforms for software and services. Europe IST 2.3.2.3 220 000 €	Amigo : Ambient intelligence for the networked home environment	Harmke de Groot Philips Research	Sept 2004 Mars 2008

Demandes de contrats en cours d'évaluation ⁸

Nom du membre participant à cette demande	% d'implication	Intitulé de l'appel à projets Source de financement Montant demandé	Titre du projet	Nom du coordinateur

⁸ Mentionner ici les projets en cours d'évaluation soit au sein de programmes du GIP-ANR, soit auprès d'organisme, de fondations, à l'union européenne, etc. que ce soit comme coordinateur ou comme partenaire. Pour chacun donnez le nom de l'appel à projets, le titre du projet et le nom du coordinateur.

--	--	--	--	--

Secteurs disciplinaires

- **Sciences et technologies de l'information et de la communication (STIC),**