UMR 6241

Laboratoire d'Informatique de Nantes Atlantique

Bilan scientifique 2006–2009

Projet scientifique 2012-2015

Version 1.0 du 15 septembre 2010

UNIVERSITÉ DE NANTES

ECOLE DES MINES DE NANTES

LINA, Université de Nantes – 2, rue de la Houssinière – BP 92208 – 44322 NANTES CEDEX 3
Tél. : 02 51 12 58 17 – Fax. : 02 51 12 58 97 – http://www.lina.univ-nantes.fr/

Table des matières

1 AeLoS **5**
Chapeau-résumé 5
1.1 Composition de l'équipe au 30/06/2010 6
1.2 Projet scientifique 2012-2015 7
1.2.1 Description du projet scientifique et objectifs 7
1.2.2 Stratégie scientifique : fil conducteur 10
1.2.3 Facteurs de réussite et de développement 10
1.3 Bibliographie 10
1.3.1 Bibliographie externe 11

Équipe
AeLoS

Chapeau-résumé

Dans le cadre du quadriennal 2012-2015, les équipes COLOSS et MODAL ont formé un nouveau projet d'équipe. Plusieurs facteurs sont à l'origine de ce rapprochement. Les activités des deux équipes font partie de la thématique plus large de Génie Logiciel, et certaines des thématiques (services, composants) sont partagées. Un rapprochement fondé sur ces thématiques est suggéré par les experts qui nous avaient évalué lors du précédent quadriennal. Nous avons alors retenu la sûreté (des logiciels et de leurs architectures) pour l'intégration et l'interaction concrète entre nos activités. Enfin, l'idée d'un pôle de génie logiciel fédérant les activités des différentes équipes du laboratoire fait son chemin depuis 2007, on a ici un premier aboutissement avec des enseignants-chercheurs de plusieurs composantes de l'Université.

Les rapports d'activités des deux équipes COLOSS et MODAL précèdent ce projet scientifique.

Le projet décrit ci-après émane du bilan des activités des deux équipes et de leur projection dans le futur. Nous rappelons ci-après la composition de la nouvelle équipe avant de présenter le projet scientifique, suivi de quelques références bibliographiques servant de balises de lecture.

1.1 Composition de l'équipe au 30/06/2010

L'équipe AeLoS (Architectures et Logiciels Sûrs) est créée en Juin 2010 sous la responsabilité de Christian ATTIOGBÉ. Elle va entamer son premier quadriennal, 2012-2015.

Permanents				
Nom	Prénom	Position	Institution	Arrivée
ANDRE	Pascal	MC	UN	01/09/2003
ARDOUREL	Gilles	MC	UN	01/09/2003
ATTIOGBE	Christian	PR2	UN	01/09/1994
LANOIX	Arnaud	MC	UN	01/09/2008
MOTTU	Jean-Marie	MC	UN	01/09/2009
OUSSALAH	Mourad	PR	UN	01/09/2001
TAMZALIT	Dalila	MC	UN	01/09/2002

Membres associés				
Nom	Prénom	Position	Institution	Remarques
HABRIAS	Henri	PR	UN	Emerite
VAILLY	Alain	MC	UN	Mission Univ. de Rabat

Doctorants				
Nom	Prénom	Position	Institution	Arrivée
MESSABIHI	Mohamed	A	UN	01/09/2007
HANOUSSE	Abdelhakim	PdL	UN	01/11/2008
AMIRAT	Abdelkrim	Bourse Algérien	UN	01/03/2007
AOUSSAT	Fadila	Bourse Tassili	France/Algérie	11/2009
BASTIDE	Gautier	Bourse E. M. Douai	E.M. Douai/UN	10/04-12/07
BOUKHADDOUM/	Souad	Bourse Tassili	France/Algérie	12/2009
CHARDIGNY	Sylvain	Bourse E. M. Douai	E.M. Douai/UN	01/06 - 10/09
HOCK-KOON	Anthony	A	UN	09/2008
GHADDAR	Ali	Cifre	BitaSoft/LINA	10/2009
LE GOAER	Olivier	A	UN	11/05-10/09
SADOU	Nassima		UN	de/10/03 à 12/07

Constitution de l'équipe

Tous les membres des équipes COLOSS et MODAL deviennent membres de la nouvelle équipe AeLoS.

1.2 Projet scientifique 2012-2015

Les équipes COLOSS et MODAL ont conjugué leurs efforts pour former une nouvelle équipe (**A**rchitecture **e**t **Lo**giciels **S**ûrs : AeLoS) dont le projet scientifique s'appuie sur trois thématiques précises où les compétences des membres sont manifestes. L'accent est mis sur une meilleure articulation de ces thématiques par rapport aux proximités thématiques relevées par les experts lors de la dernière évaluation (services, composants), aux complémentarités des travaux (approches ascendante et descendante), sur la conjugaison des moyens pour relever le défi des *architectures sûres* et du *logiciel sûr* à différents niveaux : celui des objets, des services, des composants et des architectures ; l'architecture et le logiciel sont vus en terme de composition des entités précédentes. L'approche formelle est transversale et permet d'attaquer le défi de la sûreté aussi bien pour les services, les composants que pour les architectures. A travers la thématique *architecture* [19, 14, 25, 9, 5] nous considérons une approche descendante du logiciel ; la thématique *composants logiciels* [15, 17, 10, 1] couvre elle l'approche montante. Enfin la thématique *multiformalisme et analyse multifacette* s'attaque au défi de l'interopérabilité et de l'analyse globale [22, 7, 16, 24, 10] du logiciel. Les domaines d'application sont ceux déjà considérés dans les équipes actuelles : systèmes communicants, fiables ou critiques, systèmes d'information à grande échelle, logiciels corrects pour l'Internet du futur, systèmes d'intelligence ambiante. Le projet s'intègre dans son ensemble dans les défis internationaux en cours [18, 17, 24, 13].

1.2.1 Description du projet scientifique et objectifs

La description est déclinée selon les trois thématiques où nous précisons à chaque fois les justifications des choix et nos objectifs précis.

La contribution de notre projet aux *Grand challenges* (”Verified Softwares” Hoare & Misra ; ”Towards Engineered Architecture Evolution”, Garlan) peut être résumée comme suit : la composition de composants corrects via des langages d'architecture efficaces, contribue à construire des applications logicielles correctes. Pour ce faire, différents langages, techniques et outils sont nécessaires à condition qu'ils soient ouverts et interopérables avec d'autres approches. L'intégration des activités (services, composants, architectures, sûreté) des deux équipes précédentes permet de faire face aux défis.

Styles de conception et d'évolution centrés architectures

Nos travaux visent la conception de nouveaux langages de description et l'évolution d'architectures logicielles distribuées à base de composants et de services [21, 8, 5]. Les communautés et conférences scientifiques concernées sont par exemple : ECSA [1], WICSA [2].

Motivations pour ces choix Ces travaux s'appuient sur :

- La nécessité d'étendre les concepts de base des ADLs (Architecture Description Languages) pour prendre en compte explicitement les styles architecturaux ;
- La nécessité de promouvoir un véritable support de réutilisation au niveau de la conception et de l'évolution ;
- La possibilité de normaliser une famille d'architectures améliorant ainsi la compréhension de l'organisation d'un système ;
- La possibilité d'offrir une meilleure description et comparaison des styles à travers la formalisation de leurs concepts et leurs mécanismes ;

1. European Conference on Software Architecture
2. Working IEEE/IFIP Conference on Software Architecture

– La prise en compte d'analyses spécifiques plus ciblées au style concerné.

Objectifs à moyen terme Il s'agit d'étudier, concevoir et développer des systèmes logiciels dynamiques et évolutifs. Nos recherches s'appuient sur les formalismes d'ADL avec un cadre méthodologique permettant la conception de nouvelles abstractions pour la définition de langages adaptés aux domaines d'application. Les styles architecturaux, en tant qu'abstraction de structure, de comportement et d'évolution jouent un rôle central et primordial ; ce sont des outils d'un très haut niveau d'abstraction. Les premiers styles ou styles de base ont émergé naturellement de l'expérience du développement logiciel et en particulier de la conception architecturale. Ils sont utilisés très tôt dans le processus de développement d'un système logiciel, au début de la conception architecturale.

Plus précisément, notre *challenge* consiste à offrir un formalisme support pour la modélisation de styles de conception et d'évolution architecturaux pour les systèmes dynamiques. L'idée est de fournir une base de styles de fondation comme par exemple, le style client-serveur et le style *pipe-filter* définis comme des spécialisations du style composant-connecteur (C&C). Ce formalisme se démarque des langages orientés architectures existants par sa capacité à décrire la dynamique d'une architecture (création de nouveaux éléments architecturaux à la volée, changement de la structure, mobilité, etc).

Nos travaux s'attaquent donc à la fois au verrou scientifique de l'élaboration de langages de styles architecturaux (leur définition, leur extension, leur raffinement, leur composition, leur évolution...) et, au verrou technologique du développement de ce type de langages à travers les paradigmes de composants et de services. Cette approche holistique et conceptuelle contribue à faciliter et à améliorer la spécification, la conception et l'évolution des architectures logicielles distribuées.

Spécification et vérification de composants logiciels

Forts de nos résultats précédents et des voies déjà ouvertes, notre ambition pour le nouveau quadriennal est de proposer des outils expérimentaux, transférables dans le monde industriel, pour mettre en œuvre la construction par assemblage [12, 15, 6] et raffinements successifs [3, 4, 2] de composants corrects. En dehors des incontournables conférences internationales FM, FME, ICSE, ESEC/FSE, les communautés/conférences concernées par cette thématique sont par exemple ICFEM, FACS, SEFM, ETAPS.

Motivations pour ces choix Plusieurs défis sont à relever : allier l'expressivité des modèles et l'aisance dans leur analyse, la constitution de bibliothèques de composants génériques prouvés, la compositionnalité pour les propriétés globales, l'hétérogénité sémantique, la généricité des modèles et de leur développement.

Objectifs à moyen terme Nous poursuivons la recherche de méthodes et de techniques de modélisation et de construction qui garantissent la correction des composants et des logiciels ; pour cela nous visons des concepts et techniques élégants dans leur définition formelle et simples d'emploi. La plateforme expérimentale COSTO servira pour la preuve des concepts ; elle sera étendue et continuera à être ouverte sur d'autres plateformes logicielles pour la vérification de propriétés et la génération de codes mais aussi pour servir de passerelle avec d'autres formalismes à composants et services. Nous avons pour ambition de diffuser COSTO dans le domaine public sous licence LGPL par exemple. Les composants seront implantables dans des environnements d'exécution ciblés ; par exemple des plateformes Java où il existe des outils de preuve et de test (Esc/Java, ...) qui nous permettront de maintenir la correction des codes obtenus et adaptés en bout de chaîne de raffinement.

Le défi est entier dans cette voie, complémentaire à celles qui privilégient la vérification des composants logiciels à postériori par les techniques de test ou l'évaluation de modèles (*model checking*) et qui valident une certaine construction –par rapport au *bugs* trouvés– mais ne garantissent pas de construction

correcte convenant aux besoins initiaux.

Nous recherchons donc des moyens pour :

- concevoir, développer ou restructurer des architectures et des systèmes à partir de composants et services prédéfinis et validés,
- faciliter leur adaptabilité et leur évolution pour offrir de nouvelles fonctionnalités ou de nouvelles architectures,
- réutiliser par adaptation ou instanciation, des composants génériques dans diverses applications ; les systèmes embarqués pour le contrôle en domotique en sont un exemple ; nous avons des collaborations sur ces aspects avec des partenaires académiques et industriels (Somfy, ClearSy, Smartesting).

Multiformalisme et analyse multifacette

Nous attaquons ici l'analyse ou la correction par construction de systèmes à composants hétérogènes [16, 11, 12], en allant des phases abstraites où les propriétés globales sont définies, jusque parfois à la phase d'implantation par des raffinements successifs de certains composants. Les nouvelles compétences en tests et Ingénierie Dirigée par les Modèles (IDM) vont contribuer à la multimodélisation et la vérification des propriétés lors des analyses formelles. Cette thématique s'exprime également dans les communautés/conférences à coloration "méthodes formelles" précédentes : FM, FME, ICSE, IFM, ICFEM, ESEC/FSE, FACS, SEFM, ETAPS, ICST.

Motivations pour ces choix Dans ce champ de recherche de grande envergure, mêlant les questions d'hétérogénité sémantique, de compositionalité, d'évolution, nous nous attaquons à un périmètre spécifique qui est celui de la correction de systèmes globalement asynchrones avec des composants logiciels corrects par construction. Il y a un besoin réel et crucial pour des grands logiciels à structure adhoc.

Objectifs à moyen terme Les objectifs à moyen terme sont de proposer, dans la continuité de nos résultats actuels, sur la base des fondements théoriques établis et d'outils existants, des méthodes outillées pour la construction ou l'analyse formelle globale de logiciels avec un fort degré d'interaction entre des composants variés. La vérification par tests de systèmes pair à pair (comme domaine d'application) sera étudiée de concert avec des collègues de l'équipe GDD.

Nous nous consacrons ici à : (i) l'élaboration des concepts, des mécanismes et des outils *multiparadigmes* (données, dynamique, interaction, temps) pour maîtriser l'interopérabilité au niveau sémantique, la voie de la dérivation systématique croisée (à la manière des connexions de Galois) entre modèles sera poursuivie ; (ii) l'adaptation des techniques de *Rely/Garanty* [20] pour l'interaction entre modèles issus de formalismes différents ; (iii) la définition de piles de modèles sémantiques avec des interfaces normalisées (à la manière des modèles ouverts qui ont fait leur preuve dans le domaine des réseaux) en adéquation avec des catégories de logiques ou de modèles.

Les passerelles entre modèles, langages, outils pourront être rigoureusement définis à partir de telles piles. Nous avions déjà montré dans nos résultats, la faisabilité d'une telle approche (entre algèbres de processus, B, réseaux de Petri, PVS), il s'agit ici de généraliser l'approche et développer les expérimentations dans notre plateforme ATACORA. Le domaine de l'ingénierie des modèles peut largement bénéficier de ces approches pour renforcer les aspects sémantiques lors des transformations effectuées sur les modèles, qui restent souvent syntaxiques.

Les défis à relever au niveau de la modélisation ou des spécifications formelles concernent : des problèmes d'hétérogénéité sémantique relatifs aux méthodes et modèles intégrés ; l'élaboration des environnements d'expérimentation et d'analyse formelle associés (vérification de propriétés globales). Le dernier recrutement dans l'équipe nous apporte des compétences en tests, qui seront exploitées dans le

volet ”techniques et outils” d’analyse.

1.2.2 Stratégie scientifique : fil conducteur

Le fil conducteur est la recherche de solutions mariant les fondements théoriques et des techniques pour construire des logiciells sûrs avec des éléments architecturaux, des composants et services prouvés corrects.

Cette recherche est balisée par les aspects architecturaux et les préoccupations de correction prouvée telles que exposées dans le contexte mondial des défis informatiques (*Verified Softwares : Theories, Tools and Experiment*, Hoare et Misra).

Nous nous efforçons de positionner nos explorations et nos résultats par rapport à ce contexte international pour assoir leur visibilité. Le recours aux fondements et résultats établis et l’ouverture de nos expérimentations sur des plateformes éprouvées participent à cette stratégie.

Nous cherchons des solutions pour l’ingénierie des architectures logicielles évolutives ; la formalisation des styles architecturaux pour les systèmes dynamiques et leurs utilisations dans des processus de développement centrés architectures. Il s’agit de proposer des concepts et des outils favorisant le développement orienté style de conception et d’évolution architecturale en vue de capturer l’expertise de conception et d’évolution pour un domaine spécifique. Notre proposition se base sur les différents outils formels utilisés pour la conception et l’évolution architecturales, au niveau langage de description d’architectures mais aussi, au niveau langages de méta-modélisation centrés architectures.

Enfin, un paramètre important dans le contexte actuel de la recherche scientifique est la coopération entre différents partenaires dans des consortiums montés autour des projets souvent à dimension multithématique. Dans cette optique nous veillons à la préservation de nos préoccupations : (1) construire, structurer, composer et analyser formellement des entités logicielles diverses ; (2) expérimenter et évaluer à différentes échelles à l’aide de prototypes, des cas d’étude génériques, puis proposer des solutions et des outils génériques, adaptables à grande échelle à différentes plate-formes académiques ou industrielles.

1.2.3 Facteurs de réussite et de développement

La diversité et la complémentarité des compétences des membres de l’équipe est indéniablement un facteur important de réussite dans la mesure où nous pouvons ainsi aborder les difficultés techniques sous différents angles. Le bon équilibre entre les aspects théoriques et pratiques (prototypages) est un de nos leviers.

Nous sommes impliqués dans différents groupes du GDR GPL et entretenons des collaborations avec plusieurs équipes au niveau national et quelques unes au niveau international. Ces aspects seront maintenus et renforcés.

Nous considérons comme acquis le soutien du laboratoire pour accompagner nos projets et nos demandes de financements et de ressources à tous les niveaux (industriels, académiques, institutionnels,...).

Le développement de la nouvelle équipe va se faire en misant aussi sur les partenariats avec d’autres équipes nationales et internationales, notament dans le cadre de montages et dépôts de projets répondant aux appels d’offres des agences de moyens.

1.3 Bibliographie

1.3.1 Bibliographie externe

[1] http ://www.mrtc.mdh.se/index.php ?choice=publications&id=2139
I. CRNKOVIC, S. SENTILLES, A. VULGARAKIS, et M. CHAUDRON. A classification framework for software component models. *IEEE Transaction of Software Engineering*, 2010, Submitted for publishing : 1–25. IEEE

[2] T. S. HOANG, H. KURUMA, D. A. BASIN, et J.-R. ABRIAL. Developing Topology Discovery in Event-B. *Sci. Comput. Program.*, 2009, 74(11-12) : 879–899

[3] T. S. HOANG, A. FURST, et J.-R. ABRIAL. Event-B Patterns and Their Tool Support. In D. V. HUNG et P. KRISHNAN, réds., *SEFM*, pages 210–219. IEEE Computer Society, 2009. ISBN : 978-0-7695-3870-9

[4] D. CANSELL, D. MÉRY, et C. PROCH. System-on-chip design by proof-based refinement. *STTT*, 2009, 11(3) : 217–238

[5] S. CHAKI, A. DIAZ-PACE, D. GARLAN, A. GARFUNKEL, et I. OZKAYA. Towards Engineered Architecture Evolution. In *Workshop on Modeling in Software Engineering 2009*, 2009

[6] http ://hal.archives-ouvertes.fr/hal-00423639/en/
P. ANDRE, G. ARDOUREL, et C. ATTIOGBÉ. Composing Components with Shared Services in the Kmelia Model. In Cesare PAUTASSO et Eric TANTER, réds., *7th International Symposium on Software Composition (SC'2008)*, Budapest Hongrie, volume 4954 of *Lecture Notes in Computer Science*, pages 125–140. Springer, 2008

[7] L. CRUZ-FILIPE, A. SERNADAS, et C. SERNADAS. Heterogeneous fibring of deductive systems via abstract proof systems. *Logic Journal of the IGPL*, 2008, 16(2) : 121–153

[8] D. GARLAN. Software Architecture. In B. W. WAH, réd., *Wiley Encyclopedia of Computer Science and Engineering*. John Wiley & Sons, Inc., 2008

[9] I. GORTON. Software architecture challenges for data intensive computing. *Software Architecture, Working IEEE/IFIP Conference on*, 2008, 0 : 4–6. IEEE Computer Society. ISBN : 978-0-7695-3092-5

[10] P. CHALIN. A sound assertion semantics for the dependable systems evolution verifying compiler. In *ICSE '07 : Proceedings of the 29th international conference on Software Engineering*, Washington, DC, USA, pages 23–33. IEEE Computer Society, 2007. ISBN : 0-7695-2828-7

[11] T. A. HENZINGER et J. SIFAKIS. The embedded systems design challenge. In J. MISRA, T. NIPKOW, et E. SEKERINSKI, réds., *FM*, volume 4085 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2006. ISBN : 3-540-37215-6

[12] papers/Basu-Bozga-Sifakis-06.pdf
A. BASU, M. BOZGA, et J. SIFAKIS. Modeling Heterogeneous Real-time Components in BIP. In *SEFM '06 : Proceedings of the Fourth IEEE International Conference on Software Engineering and Formal Methods*, Washington, DC, USA, pages 3–12. IEEE Computer Society, 2006

[13] G. T. LEAVENS, J.-R. ABRIAL, D. BATORY, M. BUTLER, A. COGLIO, K. FISLER, E. HEHNER, C. JONES, D. MILLER, S. PEYTON-JONES, M. SITARAMAN, D. R. SMITH, et A. STUMP. "roadmap for enhanced languages and methods to aid verification". In *GPCE '06 : Proceedings of the 5th international conference on Generative programming and component engineering*, New York, NY, USA, pages 221–236. ACM, 2006. ISBN : 1-59593-237-2

[14] P. KRUCHTEN, H. OBBINK, et J. STAFFORD. The Past, Present, and Future of Software Architecture. *IEEE SOFTWARE*, 2006, 0 : 22–30. IEEE Computer Society

[15] GREGOR GÖSSLER AND JOSEPH SIFAKIS. Composition for Component-based Modeling. *Sci. Comput. Program.*, 2005, 55(1-3) : 161–183

[16] B. K. AICHERNIG, H. JIFENG, Z. LIU, et M. REED. Integrating theories and techniques for program modelling, design and verification. In Meyer et Woodcock [23], 2005

[17] L. de MOURA, S. OWRE, H. RUESS, J. RUSHBY, et N. SHANKAR. Integrating verification components. In Meyer et Woodcock [23], 2005

[18] J.-R. ABRIAL. On Constructing Large Computerized Systems (a position paper). In Meyer et Woodcock [23], 2005

[19] M. SHAW. The coming-of-age of software architecture research. In *ICSE '01 : Proceedings of the 23rd International Conference on Software Engineering*, Washington, DC, USA, page 656. IEEE Computer Society, 2001. ISBN : 0-7695-1050-7

[20] T. A. HENZINGER, S. QADEER, et S. K. RAJAMANI. Decomposing Refinement Proofs using Assume-guarantee Reasoning. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, pages 245–252. IEEE Computer Society Press, 2000

[21] D. GARLAN. Software Architecture : a Roadmap. In *ICSE - Future of SE Track*, pages 91–101, 2000

[22] Y. KALFOGLOU, W. M. SCHORLEMMER, A. P. SHETH, S. STAAB, et M. USCHOLD, réds. *Semantic Interoperability and Integration*, volume 04391 of *Dagstuhl Seminar Proceedings*. IBFI, Schloss Dagstuhl, Germany

[23] B. MEYER et J. WOODCOCK, réds. volume 4171 of *Lecture Notes in Computer Science*. Springer. ISBN : 978-3-540-69147-1

[24] N. SHANKAR et J. WOODCOCK, réds. volume 5295 of *Lecture Notes in Computer Science*. Springer. ISBN : 978-3-540-87872-8

[25] R. N. TAYLOR, N. MEDVIDOVIC, et E. DASHOFY. *Software Architecture : Foundations, Theory, and Practice*. Wiley, John & Sons, Incorporated, 2009

LINA, Université de Nantes – 2, rue de la Houssinière – BP 92208 – 44322 NANTES CEDEX 3
Tél. : 02 51 12 58 17 – Fax. : 02 51 12 58 97 – http://www.lina.univ-nantes.fr/