

1 Curriculum Vitae

Nicolas STOULS, né le 23 mars 1980 à Suresnes (France), Marié

9 avenue Alsace Lorraine
38300 Bourgoin Jallieu

INRIA Saclay – Île-de-France
Parc Orsay Université, 4 rue Jacques Monod
91893 ORSAY Cedex, France

Téléphone portable : 06.86.62.89.05
Nicolas.Stouls@gmail.com

Téléphone bureau : 01.74.85.42.67
<http://www.lri.fr/~stoulsn/>

1.1 Formations

2007 : **Doctorat** d'informatique (Institut Polytechniques de Grenoble). Mention : *sans objet à l'IPG*.
Titre : « *Systèmes de transitions symboliques et hiérarchiques pour la conception et la validation de modèles B raffinés* ». Supervisé par *M-L. Potet* et *S. Boulmé* du Laboratoire d'Informatique de Grenoble (*LIG*) et co-financé par le *CNRS* et *ST-Microelectronics* via une bourse BDI.

2003 : **DEA** d'informatique systèmes et communications (Grenoble 1). Mention *Assez Bien*.
Titre : « *Génération de systèmes de transitions étiquetées à partir de spécifications B événementiel* ». Supervisé par *Marie-Laure Potet* et *Didier Bert* au *LIG*.
Financements : Bourse du mérite et enseignements (vacataire et tuteur d'informatique).

2002 : **Maîtrise** d'informatique (Grenoble 1). Mention *Bien*.
Financements : Tuteur d'informatique en première année de DEUG.

2001 : **Licence** d'informatique (Grenoble 1). Mention *Assez Bien*.

1.2 Formation complémentaire : Magistère (de 2000 à 2003)

Cette formation, réalisée à l'UFRIMA, a pour but de préparer les étudiants au monde de la recherche. Les étudiants doivent donc effectuer des stages de recherche, dans des organismes publics ou privés, durant deux ou trois ans, en parallèle de leur cursus normal (de la licence au DEA).

2002–2003 (Mention *Assez Bien*) : stage supervisé par *Daniel Dollé* (*Siemens Transportation Systems*) et portant sur l'« *étude et le développement de logiciels pour l'utilisation de B dans des projets de taille industrielle* ».

2001–2002 (Mention *Très Bien*) : stage supervisé par *Didier Bert* (*LIG*) et portant sur l'« *intégration du calcul de plus faible précondition à la Boîte à Outils B (BoB)* ».

2000–2001 (Mention *Bien*) : stage supervisé par *Marie-Laure Potet* et *Didier Bert* (*LIG*) et portant sur la « *modélisation en B d'un interpréteur d'objets JavaCard* ».

1.3 Fonctions assurées

09/2007–08/2008 : Post-doctorat au sein du projet ProVal de l'INRIA Saclay – Île-de-France (qui est également l'équipe Démons du LRI), dans le cadre du pôle de compétitivité System@tic. Ce travail est décrit en section 3.2.

01/2006–06/2007 : ATER dans l'équipe Démons du LRI. Les enseignements associés sont décrits en section 2, tandis que la partie recherche a été dédiée à la finalisation de ma thèse.

1.4 Outils professionnels

Langages et Méthodes de spécification : méthode B, Caduceus, Coq, JML, UML/OCL, ...

Langages de programmation : Java, C, Caml, ADA, Prolog, assembleurs, Lustre, JavaCard, ...

Base de données/internet : MySQL, HTML, PHP, CSS, ...

Systèmes d'exploitation : LINUX, DOS, Windows, MacOS

Langues : Français (*Langue maternelle*) et anglais (*Lu, écrit, parlé*)

2 Activités d'enseignement (Expérience et projet)

Ayant débuté mes enseignements en maîtrise en tant que tuteur, avant de pouvoir être vacataire, j'ai eu la possibilité d'enseigner dans différents UFRs, face à un public varié. J'ai ainsi enseigné tour à tour au DSU¹, à l'UFRIMA², à Polytech Grenoble³, à l'ENSIMAG⁴, à l'IE2⁵ ainsi qu'à l'IFIPS⁶, où j'ai pu expérimenter les différents types d'enseignement que sont les tutorats, les TPs, les TDs, les projets et les cours.

Dans le tableau 1, je résume le nombre d'heures d'enseignement que j'ai réalisé dans les différents niveaux d'étude. Par la suite, je détaille chacun de ces enseignements. En fin de cette section, je conclus en décrivant mon projet d'enseignement.

Statut	Cursus	Année	Licence				Master	
			Tutorat	TP	TD	Cours	TD	Cours
<i>ATER</i>	<i>Thèse 4A</i>	<i>2006-2007</i>	-	<i>45h⁶</i>	<i>52h^{6,7}</i>	<i>9h⁷</i>	-	-
<i>Vacataire</i>	<i>Thèse 3A</i>	<i>2005-2006</i>	-	-	-	-	<i>70h^{3,5}</i>	<i>15h³</i>
<i>Vacataire</i>	<i>Thèse 2A</i>	<i>2004-2005</i>	-	-	<i>10h⁴</i>	-	<i>70h^{3,5}</i>	<i>15h³</i>
<i>Vacataire et tuteur</i>	<i>Thèse 1A</i>	<i>2003-2004</i>	<i>20h²</i>	-	<i>36h³</i>	-	-	-
<i>Vacataire et tuteur</i>	<i>DEA</i>	<i>2002-2003</i>	<i>18h²</i>	<i>15h²</i>	<i>33h^{2,3}</i>	-	-	-
<i>Tuteur</i>	<i>Maîtrise</i>	<i>2001-2002</i>	<i>28h²</i>	-	-	-	-	-
	<i>Total</i>		<i>66h</i>	<i>60h</i>	<i>131h</i>	<i>9h</i>	<i>140h</i>	<i>30h</i>
	<i>Total (Équ. TD)</i>		44h	40h	131h	13,5h	140h	45h

TAB. 1 – Résumé des enseignements réalisés, selon le niveau d'étude et la filière

2.1 Enseignements

Dans cette section, j'ai organisé mes enseignements en quatre thématiques :

- Génie logiciel
- Architecture logicielle et matérielle

¹Département Scientifique Universitaire - UFR premier cycle scientifique de l'UJF.

²UFR d'Informatique et Mathématiques de Grenoble, rattaché à l'UJF.

³École d'ingénieur dépendant de l'UJF.

⁴École d'ingénieur en Informatique et Mathématiques Appliquées rattachée à l'IP Grenoble.

⁵UFR d'Informatique et Enseignement de second cycle, rattaché à l'université d'Orsay, Paris XI.

⁶Institut de Formation d'Ingénieurs de l'Université Paris Sud XI.

- Algorithmique et programmation
 - Soutien pédagogique
- Pour finir, j'effectue un bilan des aspects didactiques que j'ai acquis lors de ces enseignements.

2.1.1 Génie logiciel

■ *Projet de Génie logiciel*

Niveau (*Effectif* / *UFR*) : 2^{ème} année d'ingénieur (*20 étudiants* / *ENSIMAG*)

Années, heures effectives et nature : 04–05 et 05–06 : 1 semaine de cours et 3 mois de suivi (55h équ. TD)

Enseignant responsable : Catherine Oriat

Statut : Vacataire

Ce projet d'un mois à temps plein a pour objectif de permettre aux étudiants de réaliser un logiciel en respectant une méthodologie génie-logiciel où les phases d'analyse et de test sont particulièrement mises en avant. L'enseignement commence avec 5 cours de 3h permettant d'introduire le sujet et de présenter l'approche génie logiciel attendue. Des séances de suivi permettent ensuite aux enseignants de jouer le rôle du client en vérifiant la compréhension des besoins et la qualité du rendu.

Le programme réalisé est un compilateur fada (Fempto-ADA) devant générer un assembleur exécutable sur une machine virtuelle. Le logiciel rendu est partiellement évalué à l'aide d'outils automatiques, à savoir : une base de tests à dépiotage semi-automatique et des scripts de détection de suspicion de tricherie. Enfin, l'entretien final dure 45 minutes, ce qui permet de faire réaliser une petite maintenance au groupe en observant leur méthodologie de travail et leur connaissance du code.

■ *Outils pour le génie logiciel - introduction aux méthodes formelles*

Niveau (*Effectif* / *UFR*) : Master 1 (*25 étudiants* / *UFRIMA*)

Années, heures effectives et nature : 04–05 et 05–06 : 15h Cours + 15h TD

Enseignants responsables : Marie-Laure Potet et moi

Statut : Vacataire

Dans cette option, j'ai été responsable de l'intégralité de la formation du second semestre, qui est une introduction aux méthodes formelles. J'ai donc dû définir les objectifs attendus en termes des connaissances qu'ils devront maîtriser à l'issue de l'option, le programme pour y arriver ainsi que les méthodes finales d'évaluation. La méthode formelle utilisée pour illustrer ce cours a été la méthode B. Ce choix historique a deux avantages : proposer une syntaxe intuitive pour les étudiants et me faciliter l'agrément des cours et TDs par des exemples pratiques sortis de mes travaux de thèse.

Les objectifs fixés étaient de permettre aux étudiants de comprendre ce qui doit être prouvé pour établir la consistance d'un programme, ainsi que la correction d'un raffinement. Pour ce qui est du cheminement, je me suis fortement inspiré du cours déjà réalisé à l'ENSIMAG par Marie-Laure Potet. Nous avons d'ailleurs avancé de concert pour donner une homogénéité à ces deux formations similaires. Les modalités et dates d'examens n'étant pas les mêmes, nous avons cependant dû réaliser des sujets différents.

■ *Génie logiciel*

Niveau (*Effectif* / *UFR*) : Licence 3 (*21 étudiants* / *IE2*)

Années, heures effectives et nature : 06–07 : 33h TD

Enseignant responsable : Frédéric Voisin

Statut : ATER

Ce cours est une introduction aux méthodes de génie logiciel. Il s'organise autour d'une introduction aux méthodes d'analyse et de conception d'UML et fait le lien avec les techniques de test puis le principe de la preuve de programmes.

Dans ce TD, chacune des séances consiste à concevoir un système différent dont on donne un cahier des charges. Les étudiants sont donc amenés à extraire les besoins du client à partir d'un document écrit en français et à les reformuler sous la forme de diagrammes UML. Dans la seconde partie du cours, le lien est fait entre les notions UML et les méthodes de tests, ainsi que les différents critères de qualité. Enfin, les deux derniers TD permettent une rapide introduction aux méthodes formelles en abordant la preuve de programmes en se basant sur la logique de Hoare.

■ *Génie logiciel pour ingénieurs en alternance*

Niveau (*Effectif* / *UFR*) : 2^{ème} année d'ingénieur, spécialité informatique (*14 étudiants* / *IFIPS*)

Années, heures effectives et nature : 06–07 : 9h cours + 9h TD

Enseignants responsables : Frédéric Baucher et moi

Statut : ATER

Ce cours est similaire à celui décrit en section 2.1.1, mais s'adresse à des élèves ingénieurs en formation en alternance. Avec Frédéric Baucher, nous avons divisé le module en deux parties (l'expression des besoins et la vérification / validation) que nous nous sommes réparties. Je me suis occupé de la partie vérification / validation et suis donc intervenu sur les 6 derniers cours/TD. Dans une première moitié, j'ai commencé par décrire différentes techniques de test basées sur des études de flots de contrôle et de données en mettant en évidence des critères de couvertures. J'ai ensuite introduit les concepts généraux de la preuve de programme avec les approches à la Hoare et par raffinement. J'ai ensuite développé la présentation de la méthode B et son approche de développement par raffinement. Ce cours s'est conclu avec un TP sur machine dont le sujet⁷ est disponible en ligne.

2.1.2 Architecture logicielle et matérielle

■ *Architecture logicielle et matérielle*

Niveau (*Effectif* / *UFR*) : Licence 3 (*25 étudiants* / *UFRIMA*)

Années, heures effectives et nature : 02–03 : 18h TD / 03–04 : 2×18h TD

Enseignant responsable : Paul Amblard

Statut : Vacataire

L'enseignement d'ALM est une première introduction à l'architecture matérielle et à la manipulation de valeurs binaires pour des étudiants qui ont déjà quelques notions de programmation. Il se décompose en un cours magistral et 3 types de TD : Général, Soft et Hard. Le TD Général porte sur la logique booléenne et le fonctionnement général d'un ordinateur et d'un processeur (UAL, pipeline, périphériques, etc.), tandis que les TDs Soft et Hard s'intéressent respectivement à la programmation assembleur et à l'architecture fine des composants physiques.

⁷http://www.lri.fr/~stoulsn/Enseignements/2007_IFIPS_TP.tgz

Sur les 3 groupes que j'ai été amené à encadrer, je me suis occupé d'un groupe de TD Soft ainsi que de deux groupes de TD Général. Dans ces enseignements, j'ai pu compléter le programme avec des illustrations tirées de mes recherches sur les cartes à puce. Le côté un peu mystérieux de ces « *micro-ordinateurs* » a suscité un intérêt accru de la part des étudiants.

Enfin, le contrôle continu était réalisé sous la forme de sujets de TP à faire pour la semaine suivante. En accord avec les étudiants, j'ai ainsi pu profiter de ce mécanisme pour faire accomplir sur machine les questions non traitées des sujets de TD. Ce faisant, la contre-partie était que je corrige tous leurs rendus pour la semaine suivante (selon moi, il ne sert à rien de donner des exercices à faire si les étudiants n'ont pas de retour dans un laps de temps suffisamment court pour se souvenir de leurs problèmes).

2.1.3 Algorithmique et programmation

■ *Programmation fonctionnelle*

Niveau (*Effectif* / *UFR*) : Licence 2 (*30 étudiants* / *DSU*)

Années, heures effectives et nature : 02–03 : 15h TD + 15h TP

Enseignant responsable : Michel Lévy

Statut : Vacataire

Dans cet enseignement, j'ai été chargé d'encadrer un groupe d'étudiants pour les TDs et les TPs. Ce cours se basait sur le langage Caml. L'enseignement s'est principalement orienté sur une approche purement fonctionnelle et le point de vue algorithmique associé.

Leur dernier TP a consisté en la réalisation d'un jeu de morpion, dont la partie graphique a été encapsulée dans une interface fonctionnelle à effet de bord, permettant ainsi de leur cacher l'existence de la structure de contrôle séquentielle.

■ *Projet de compilation*

Niveau (*Effectif* / *UFR*) : Licence 1 (*26 étudiants* / *RICM*)

Années, heures effectives et nature : 04–05 : 1 semaine de suivi à temps partiel (10h équ. TD)

Enseignant responsable : Laurent Mounier

Statut : Vacataire

Ce projet consiste, pour les étudiants, à réaliser un compilateur depuis un langage d'assemblage simple. Le développement est effectué en ADA et dure une semaine à temps plein. Les trois enseignants se répartissent donc les 80 étudiants de la promotion en circulant à tour de rôle dans les salles et en suivant plus particulièrement les avancées d'un tiers des groupes. La soutenance finale est réalisée sur machine avec un court exposé et une démonstration du produit en 30 minutes pour chaque groupe.

■ *Projet de Travail d'Etude et de Recherche*

Niveau (*Effectif* / *UFR*) : Licence 3 (*21 étudiants* / *IE2*)

Années, heures effectives et nature : 06–07 : 10h TD + 45h TP

Enseignant responsable : Fatiha Zaïdi

Statut : ATER

Ce projet se déroule sur l'intégralité du second semestre et est une utilisation pratique du cours de

génie logiciel décrit en section 2.1.1. Les étudiants doivent choisir un sujet qu'ils réalisent en groupe de 3 personnes sur un semestre entier avec une séance de suivi chaque semaine.

Dans ce projet, on met particulièrement l'accent sur la partie analyse et conception. Les étudiants sont donc amenés à faire des études très approfondies de leur sujet. Ce faisant, ils isolent des ambiguïtés ou des sous-spécifications qu'ils doivent rapporter à leur enseignant qui joue alors le rôle de client. Un compte rendu est également attendu à chaque fin d'entrevue afin de notifier les choix qui ont été effectués.

Dans le cadre de ce projet, j'ai proposé l'un des 4 sujets utilisés en choisissant un programme ludique et simple⁸ afin de les motiver dans leur projet.

2.1.4 Soutien pédagogique

■ *Tutorat d'informatique*

Niveau (*Effectif* / *UFR*) : Licence 1 (*8 étudiants* / *DSU*)

Années, heures effectives et nature : 01–02 : 28h / 02–03 : 18h

Enseignant responsable : Christian Dupuis

Statut : Tuteur

Cet enseignement consiste à prendre en charge de petits groupes d'étudiants de première année de DEUG MIAS et SM qui sont en difficulté. Leur présence n'est pas obligatoire et le programme suit principalement les avancées du cours. Cependant, cela reste un soutien qui s'adapte à la demande. Il est donc arrivé à plusieurs reprises qu'il soit une préparation au cours à venir.

Chaque année, l'ouverture d'un cours de tutorat a été décidé suivant les besoins des étudiants. Il a ainsi pu couvrir un semestre (02–03) ou deux semestres (01–02), tandis qu'un suivi supplémentaire a systématiquement été mis en place entre les examens de Mai et les rattrapages.

■ *Soutien individualisé d'un sportif de haut niveau*

Niveau (*Effectif* / *UFR*) : Licence 1 (*1 étudiant* / *DSU*)

Années, heures effectives et nature : 03–04 : 20h TD

Enseignant responsable : Christophe Massit

Statut : Vacataire

Dans le cadre de ses études, un étudiant de l'UJF avait un aménagement de ses horaires pour lui permettre de continuer ses entraînements et ses compétitions sans qu'il soit pénalisé dans ses cours. J'ai alors été chargé de lui faire rattraper ses cours, TDs et TP de mathématiques et informatique de sa première année de DEUG. Durant cette année, il m'a donc fallu travailler en interaction avec les enseignants des différentes filières pour lui permettre d'atteindre un niveau suffisant dans un minimum de temps. De plus, nos horaires de rencontre ont parfois dû être très décalés avec les heures ou jours d'enseignement traditionnel.

⁸<http://www.lri.fr/~stoulsn/Enseignements/2007-TER-Sujet-RasendeRoboter/TER-RasendeRoboter.html>

2.1.5 Bilan

J'ai eu la possibilité de dispenser des enseignements dans un certain nombre de structures universitaires et dans différentes villes. Cela m'a permis de comparer les approches pédagogiques et de faire mes propres expériences.

En particulier, j'ai acquis la conviction qu'il est important de rattacher explicitement les enseignements à quelque chose de concret, tel qu'un autre cours ou une application pratique. Dans le cas contraire, le cours peut être mal perçu par les étudiants qui « *n'en voient pas l'utilité* ».

De la même manière, je pense que tout travail rendu par un étudiant doit être accepté par l'enseignant et corrigé dans un laps de temps court (que ce travail ait été noté ou pas et également s'il est spontané).

De plus, je suis convaincu qu'une équipe pédagogique participant à un même enseignement doit avoir une dynamique de groupe avec une interaction de tous. Ainsi, dans le cadre des enseignements dont je serai amené à avoir la charge, je compte proposer des réunions régulières et, dans la mesure du possible, des compte rendus de cours et TD. Ainsi, au delà du programme du cours, chacun pourra être tenu au courant des avancées et difficultés du groupe entier des étudiants. Notons que cette approche à base de compte-rendus permet également d'aider les enseignants les plus novices. Je l'ai découverte au sein de l'équipe pédagogique de génie logiciel et cela m'a particulièrement bien aidé dans mon intégration et ma découverte du programme.

Enfin, dans le cas particulier des projets, j'ai noté que les soutenances sont souvent courtes et ne laissent que peu de temps pour se faire une idée réelle de la qualité du résultat et de la méthodologie de développement appliquée par les étudiants. C'est pourquoi, je pense que, comme cela est fait à l'ENSIMAG, des soutenances de 30 à 45 minutes seraient un réel plus. Sans changer le temps de présentation des étudiants (10 à 15 minutes), cela laisse du temps pour poser quelques questions (5 à 10 minutes) et surtout les faire intervenir sur leur code, par exemple en leur faisant ajouter une fonctionnalité simple ou corriger un bug. Ainsi, il est possible d'assister à leur méthode d'analyse du problème, de répartition des tâches et leur politique de test. Il est alors cohérent de monter des groupes de travail de 3, 4, voire 6 étudiants.

3 Activités de recherche (Expérience et projet)

Dans cette section, je décris d'abord mes travaux de thèse, puis ceux actuels de mon post-doctorat. Je présente ensuite les stages que j'ai encadrés, avant de conclure en présentant mon projet de recherche.

3.1 Thèse

Titre	Systèmes de transitions symboliques et hiérarchiques pour la conception et la validation de modèles B raffinés.
Institution	Institut Polytechnique de Grenoble (<i>IPG</i>)
Financements	co-financée par le <i>CNRS</i> et <i>ST Microelectronics</i> via une bourse BDI.
Président	Jean-Claude Fernandez (<i>Professeur à l'université de Grenoble 1</i>)
Directeurs	Marie-Laure Potet (<i>Professeur à l'Institut Polytechnique de Grenoble</i>) Sylvain Boulmé (<i>Maître de conférence à l'Institut Polytechnique de Grenoble</i>)
Rapporteurs	Jacques Julliand (<i>Professeur à l'université de Franche Comté</i>) Michael Leuschel (<i>Professeur à l'université de Düsseldorf</i>)
Examineurs	Claude Marché (<i>Directeur de recherche à l'INRIA Saclay – Île-de-France</i>) Christophe Planat (<i>Chercheur chez ST Microelectronics</i>)
Soutenance	14 décembre 2007
Manuscrit	http://tel.archives-ouvertes.fr/tel-00216026/en/
Mots clefs	Méthode B - Systèmes de transitions - Raffinement - Vues - Aide au développement - Validation

Ma thèse a porté sur l'aide à la compréhension et à la validation de modèles B [Abr96]. Les spécifications formelles sont principalement utilisées pour développer des logiciels ayant une composante sécurité très forte. Afin de vérifier la conformité de tels systèmes par rapport au cahier des charges, il est souvent utile d'avoir différents points de vue du modèle. De plus, les modèles B sont décrits en termes de traitement des données. Ainsi, un second point de vue exhibant les comportements possibles du système peut apporter une aide précieuse à sa compréhension. Notre approche s'inspire de [BC00] et se base sur la génération de systèmes de transitions étiquetées représentant les comportements d'un modèle B. Les principales originalités de ce travail sont la génération d'un automate par résolution d'obligations de preuve, la définition de conditions de franchissement plus fines que la garde classique et la prise en compte du processus de raffinement, qui permet de développer un modèle en introduisant progressivement les exigences.

Afin d'obtenir une représentation aussi précise que possible, nous divisons les gardes des transitions en deux parties : une condition de déclenchabilité et une condition d'atteignabilité. La première est la condition sous laquelle un événement peut se déclencher depuis l'état de départ et la seconde définit la condition sous laquelle un événement peut atteindre l'état d'arrivée. Cette représentation permet d'exhiber l'origine de la condition tout en coupant les obligations de preuve [CF.5].

Le raffinement est l'un des points majeurs du processus de développement selon la méthode B. Nous prenons donc en compte ce mécanisme pour introduire, par étapes, la complexité du modèle, en utilisant des automates hiérarchiques [CF.5,CI.3]. Afin de conserver la structure générale de la représentation du système abstrait, nous avons défini la notion de projection d'états selon le lien de raffinement. Les états projetés sont alors représentés comme des sous-états d'états abstraits. Certaines propriétés du raffinement sont alors exploitées pour simplifier la construction du système de transitions associé à un raffinement. Enfin, nous avons également proposé certaines constructions telles que l'utilisation de sous-états initiaux et finaux, ainsi que la factorisation de transitions. L'objectif est alors de mieux

structurer la représentation tout en préservant une sémantique observationnelle équivalente.

Les résultats de ces travaux ont été exploités dans le contexte du projet GECCOO où nous avons expérimenté leur utilisation pour la vérification de propriétés comportementales non exprimables directement en **B**. En nous inspirant du langage JTPL [TH02], nous avons introduit [CI.3] un langage simple permettant d'exprimer des propriétés de trace en même temps en terme d'événements et d'états. Ce langage est basé sur quatre primitives : Enabled, Crossable, AlwaysEnabled et AlwaysCrossable. Dans certains cas particuliers, ces primitives peuvent être syntaxiquement évaluées sur le système de transitions étiquetées extrait du modèle **B** ce qui permet de rendre la formule décidable. Les problèmes d'indécidabilité ont alors été déportés dans la construction du système de transitions.

Enfin, la méthode proposée de construction d'un système de transitions a été implantée dans le logiciel *GénéSyst*⁹. Celui-ci a été utilisé par D. Bert dans le cadre du projet EDEMO1, s'intéressant à la modélisation de la sécurité d'un aéroport. Les figures générées ont servi à documenter le modèle **B** réalisé ainsi que comme support de communication avec les experts de l'aviation civile qui s'intéressaient à ce projet. L'outil a également été étudié comme piste d'avancement par A. Haddad dans le cadre du projet POSÉ, pour vérifier la conformité d'un modèle par rapport à sa politique de sécurité.

3.2 Post-doctorat

Titre	Modèles et preuves pour les plate-formes de confiance
Institution	Équipe-projet ProVal de l'INRIA Saclay – Île-de-France
Responsable	Christine Paulin (<i>Professeur à l'université de Paris Sud-XI</i>)
dates	09/2007 à 08/2008
Mots clefs	Propriétés de sécurité - Preuve - LTL - Chaîne de confiance - TPM

Ce post-doctorat s'intègre dans le projet *plates-formes de confiance* (PFC) du pôle de compétitivité parisien System@tic. Il porte sur la spécification et la vérification de propriétés de sécurité. L'enjeu général est ici de fournir des outils permettant la validation, avec un haut niveau de confiance, de systèmes d'informations vis-à-vis de politiques de sécurité.

L'une des problématiques consiste à définir un langage de description de propriétés de sécurité suffisamment expressif et permettant une validation des systèmes par un processus automatisable de telle sorte qu'il soit exploitable dans l'industrie. De plus, les propositions doivent prendre en considération les contraintes spécifiques associées aux différents types de systèmes cible (Carte à puce, serveur partagé, etc).

La piste d'étude retenue est de s'intéresser au lanceur sécurisé OSLO. Ce programme **C** de bas niveau est le premier maillon d'une chaîne de confiance logicielle basée sur un dispositif matériel (appelé TPM – *Trusted Platform Module*). La chaîne de confiance permet de garantir, par construction, que l'environnement d'exécution est sain. En se basant sur ses propriétés de traçabilité et sur les fonctionnalités du TPM, il est alors possible d'identifier de manière sûre le programme qui s'exécute à un moment donné sur une machine distante, ainsi que de préserver des secrets (principe du *coffre fort* – stockage chiffré de données sur le disque et géré par le TPM).

⁹<http://www.lri.fr/~stoulsn/?ZoomSur=Logiciels#Logiciels>

Dans un premier temps nous avons vérifié l'absence de menace du programme Oslo avant d'identifier certaines propriétés critiques réutilisables dans le cadre de PFC. Nous nous intéressons maintenant à l'expression de propriétés temporelles LTL sous la forme d'assertions insérées dans le code C. Tout l'enjeu est de réussir à fournir suffisamment d'hypothèses pour que les assertions soient prouvables. L'objectif est ensuite de fournir un outil permettant de décorer automatiquement un programme C à partir d'une propriété temporelle, de telle sorte que les obligations de preuve soient déchargées, aussi souvent que possible, de manière automatique.

3.3 Encadrement et participation à des jurys

Au cours de ma thèse, j'ai proposé un certain nombre de sujets de stages dont quatre ont été pourvus. Le but était alors en même temps d'avoir une assistance dans mes recherches et de m'investir plus avant dans les filières d'enseignement. Voici les 4 stages que j'ai encadrés :

- Ounayda Mohamed (2003–2004) Étudiante en Licence 3, je me suis chargé à 80% de son encadrement, avec Didier Bert, durant son stage d'été portant sur la *mise en place d'une interface graphique pour GénéSyst*. J'ai été son principal interlocuteur et ce sont mes besoins qui ont défini ses directions d'avancée.
- Xavier Morselli (2003–2004) Étudiant en Master 1, je l'ai également co-encadré à 80% avec D. Bert durant son stage d'ouverture aux travaux de la recherche portant sur l'*optimisations de l'outil GénéSyst*. Ici aussi, j'ai été son principal interlocuteur puisque son stage consistait à vérifier la correction et à optimiser le premier prototype de *GénéSyst*. Ce stage a été marqué par la rédaction d'un article qui a été soumis et accepté en session outil d'une conférence nationale [C.7].
- Evelyne Altariba (2003–2004) Élève ingénieur en seconde année d'ENSIMAG, je l'ai co-encadrée à 50% avec M-L Potet durant son stage de fin d'année portant sur la *compression d'un projet B*.
- Amal Haddad (2004–2005) Étudiante en Master 2R, je l'ai co-encadrée à 25% avec M-L Potet durant son stage portant sur la *Modélisation et la vérification de politiques de sécurité*. J'ai principalement épaulé Amal dans sa découverte de la méthode B. J'ai également participé à son jury de DEA en qualité d'encadrant. Elle a maintenant commencé une thèse au LIG, au sein de l'équipe VASCO.

4 Publications

Les travaux menés durant cette thèse ont donné lieu à un certain nombre de communications. Cette section résume les principales d'entre elles. Une liste plus complète, incluant posters et séminaires, est disponible sur ma page personnelle (<http://www.lri.fr/~stoulsn/?ZoomSur=publications>).

– Thèse

- [Th.1] N. Stouls. *Systèmes de transitions symboliques et hiérarchiques pour la conception et la validation de modèles B raffinés*, IPG (Institut Polytechnique de Grenoble), 2007.
<http://www.lri.fr/~stoulsn/index.php?ZoomSur=publications>

– Revues en langue française d'audience internationale

- [A.1] F. Badeau, D. Bert, S. Boulmé, C. Métayer, M-L. Potet, N. Stouls et L. Voisin. *Adaptabilité et validation de la traduction de B vers C - Points de vue du projet BOM*, série TSI 7/2004, Hermès-Lavoisier. *Article pré-sélectionné dans AFADL'03 pour une édition spéciale du journal.*

– **Conférences d’audience internationale**

[CI.2] N. Stouls and M-L. Potet. In J. Julliand and O. Kouchnarenko, editors *Security Policy Enforcement Through Refinement Process*, B’2007, the 7th Int. B Conference, LNCS 4355 p.299–318, Springer. *Taux d’acceptation : 48%*

[CI.3] D. Bert, M-L. Potet et N. Stouls. *GeneSyst : a Tool to Reason about Behavioral Aspects of B Event Specifications. Application to Security Properties*, LNCS 3455 p.299–318, Springer-Verlag, ZB 2005. *Taux d’acceptation : 40%*

– **Conférences en langue française**

[CF.4] N. Stouls et V. Darmaillacq. *Développement formel d’un moniteur détectant les violations de politiques de sécurité de réseaux*, AFADL’06, pages 179–193. *Taux d’acceptation : 45%*

[CF.5] N. Stouls et M-L. Potet. *Explicitation du contrôle de développement B événementiel*, AFADL 2004, pages 13–27. *Taux d’acceptation : 46%*

[CF.6] F. Badeau, D. Bert, S. Boulmé, C. Métayer, M-L. Potet, N. Stouls et L. Voisin. *Traduction de B vers des langages de programmation*, AFADL’03, pages 87–102. *Taux d’acceptation : 56%*

[CC.1] N. Stouls et V. Darmaillacq. *Développement formel d’un moniteur*, Majecstic 2005, pages 397–401. *Taux d’acceptation : ?%*

[C.7] X. Morselli, M.-L. Potet et N. Stouls. *GénéSyst : Génération d’un système de transitions étiquetées à partir d’une spécification B événementiel*, AFADL 2004, pages 317–320. *Session outils, sans comité de sélection*

– **Rapports de recherche**

[RR.1] N. Stouls. *Introduction aux cartes à puce*, 2006
<http://www.lri.fr/~stoulsn/Productions/CartesAPuce/CartesAPuce.ps.gz>

[RR.2] N. Stouls. *Introduction aux critères communs*, 2004.
<http://www.lri.fr/~stoulsn/Productions/CC/CriteresCommuns.ps.gz>

5 Charges collectives et prises de responsabilités

5.1 Responsabilités liées à l’activité d’enseignement

En accord avec mes activités d’enseignement, je me suis investi dans un certain nombre d’activités :

- Participation à la journée de présentation des Master 2 de 2004, ainsi qu’à la journée des doctorants de 2006 en présentant¹⁰ de manière décalée la vie d’un doctorant.
- Participation à l’organisation (Notamment avec la mise en place de moyens informatiques) et au déroulement des journées portes ouvertes 2007 et 2008 de l’université d’Orsay.
- Participation aux jurys d’entretien de sélection des candidats à la formation continue d’ingénieurs en informatique de l’IFIPS en 2007.
- Participation à la fête de la science 2007 dans l’atelier de vulgarisation de la logique organisé par Florence Plateau et Yannick Moy.

¹⁰<http://www.lri.fr/~stoulsn/Exposes/2006-11-PresentationVieDoctorant.pdf>

- De manière générale, j’ai participé à toutes les réunions pédagogiques et les jurys auxquels j’ai pu me rendre.

5.2 Responsabilités liées à l’activité de recherche

Durant ma thèse, je me suis intéressé au fonctionnement d’un laboratoire de recherche en assistant de manière régulière aux réunions bimensuelles de mon équipe de recherche et en prenant à ma charge certaines missions en rapport avec l’équipe ou le laboratoire :

- Durant ma thèse, j’ai été chargé de la gestion de l’inventaire informatique de l’équipe VASCO.
- J’ai également participé de manière occasionnelle à la mise à jour du site web de l’équipe.

De manière plus large, j’ai également participé à l’animation de la vie scientifique ou non des doctorants du LSR et, plus récemment, du LIG :

- J’ai mis en place un groupe, nommé *Sorties-LSR*, qui a pris la forme d’une liste de diffusion et d’une page web. J’y ai organisé et diffusé un certain nombre d’activités ludiques ayant permis la fédération des doctorants du LSR venant d’équipes différentes et ne se connaissant pas. A mon départ, l’infrastructure regroupant 35 personnes a été transmises à Frédéric Dadeau, puis à Charlotte Hug, pour devenir l’actuel groupe *Sorties.LIG*.
- Avant mon départ de Grenoble, j’ai également participé à la mise en place de moyens de communication (site web et liste de diffusion) pour promouvoir les actions de l’*association des doctorants du LIG* qui organise des séminaires scientifiques à destination des doctorants.

5.3 Prises de responsabilités annexes

- Membre actif de l’école de plongée universitaire de Grenoble durant 9 ans : Organisation des cours théoriques et pratiques, rédaction des sujets d’examens, aide à la préparation des sorties, aide à la gestion du site web, etc.
- Création en 1997 du club informatique du collège Tipaerui, qui est ensuite devenu celui du lycée Paul Gauguin (Tahiti Polynésie Française). Définition de projets motivant l’existence du club et participation aux journées portes ouvertes du lycée avec démonstration de nos projets.

6 Bilan

J’espère que ma candidature en tant qu’enseignant-chercheur retiendra votre attention. En effet, je pense que l’aspect enseignement d’un dossier est aussi important que son aspect recherche et que le temps passé à définir la bonne manière de présenter un cours ou le taux de réussite de ses étudiants sont valorisant au même titre que des publications. C’est dans cet esprit d’effort pédagogique que j’ai commencé mes enseignements durant ma seconde année de magistère et c’est encore dans cet esprit que je candidate aujourd’hui chez vous.

Références

- [Abr96] J.R. Abrial. *The B-Book*. Cambridge University Press, 1996.
- [BC00] D. Bert and F. Cave. Construction of Finite Labelled Transition Systems from B Abstract Systems. In W. Grieskamp, T. Santen, and B. Stoddart, editors, *Integrated Formal Methods*, volume 1945 of *Lecture Notes in Computer Science*. Springer-Verlag, 2000.

- [TH02] K. Trentelman and M. Huisman. Extending JML Specifications with Temporal Logic. In *Algebraic Methodology And Software Technology (AMAST '02)*, LNCS 2422, pages 334–348. Springer-Verlag, 2002.