Mars 2002

Dossier de candidature à un poste de Maître de Conférences

M. Claude MICHEL

Claude.Michel@sophia.inria.fr
http://www-sop.inria.fr/coprin/Claude.Michel/dossier.html
Tél: 04.92.38.79.12
Projet COPRIN (I3S - INRIA)
INRIA - Sophia Antipolis
2004, route des Lucioles - BP 93
06902 - Sophia-Antipolis cedex - France

Contenu

1 Introduction **2**

2 Choix de travaux à évaluer **2**

3 Synthèse **2**

4 Curriculum vitae **4**
4.1 Activités Professionnelles 4
4.2 Formation 4

5 Activités de recherche **5**
5.1 Contraintes sur les flottants (11/2000 - ...) 5
5.2 Modèles et implémentation d'interprètes réflexifs (1992-1999) 5
5.3 Représentation des connaissances et langage d'acteurs (1990) 7
5.4 Projet de recherche 7

6 Activités d'enseignement **9**
6.1 Descriptions des enseignements 9
6.1.1 Formation initiale 9
6.2 Maîtrise 10
6.2.1 DEA 10
6.2.2 Formation continue 11
6.3 Supports de cours 11

7 Activités d'administration et autres responsabilités collectives **12**
7.1 Administration de la recherche 12
7.2 Administration de l'enseignement 12

8 Autres activités **12**

9 Liste de publications et réalisations techniques **13**
9.1 Revue avec comité de lecture 13
9.2 Conférences internationales avec comité de lecture 13
9.3 Conférences nationales avec comité de lecture 13
9.4 Thèse et DEA 14
9.5 Réalisations techniques 14

1 Introduction

Ingénieur de recherche au sein du projet COPRIN (I3S-INRIA) depuis novembre 2000, mon parcours se caractérise par un changement de thématique de recherche important.

Mes travaux de recherches en thèse ont porté sur la réflexivité. Ces travaux avaient en commun avec mon stage de DEA, les langages d'acteurs. D'utilisateur de ces langages, je suis passé (après une interruption de deux années due aux obligations militaires) à implémenteur de ces langages en m'intéressant à la réflexivité. J'ai alors accompli un important travail de réflexion sur ce thème, travail caractérisé par de nombreuses implémentations qui avaient pour but de valider mes idées.

L'emploi d'ingénieur de recherche que j'occupe actuellement est financé par le projet RNTL-INKA floats. Ma tâche principale concerne la réalisation d'un solveur de contraintes portant sur les nombres flottants. Cette thématique de recherche, nouvelle pour moi, a nécessité l'apprentissage du fonctionnement des systèmes de contraintes ainsi que de l'arithmétique des nombres flottants. Bien qu'une telle reconversion ait représenté un réel défi, elle a déjà porté ses fruits. Plusieurs résultats ont ainsi fait l'objet de publications dans des conférence internationales (CP 2001, 7th AMAI, TSI).

Depuis mes débuts, j'ai assuré des activités d'enseignements devant des public variés et dans des contextes différents. J'ai pris à cœur mes fonctions d'enseignant et je me suis attaché à effectuer des cours clairs et rigoureux.

En plus de mes activités de recherche et d'enseignement, j'ai souvent participé à l'administration de l'enseignement (par exemple, j'ai été responsable des enseignements d'Excel en DEUG 2B) et à l'administration de la recherche (par exemple, je suis membre du comité d'organisation des JFPLC'2002 et des JNPC'2002). J'ai aussi assuré au sein de l'équipe contraintes des fonctions d'ingénieur système qui, entre autre choses, se sont traduites par l'installation de machines sous Unix (PC sous Linux et Sun sous Solaris).

L'ensemble de ces activités, qui montrent mon intérêt pour la carrière universitaire, m'amène à candidater à un poste de Maître de Conférences.

2 Choix de travaux à évaluer

1. Manuscrit de la thèse
2. Article de recherche: *Solving constraints over floating point numbers*, Claude Michel, Michel Rueher, Yahia Lebbah, Constraint Programming 2001, Paphos, Chypre, Novembre 2001.
3. Article de recherche: *Exact projection functions for floating point number constraints*, Claude Michel, 7th AIMA, Fort Lauderdale, Floride, Janvier 2002.

L'ensemble de ces documents est accessible sur le Web à l'addresse :
http://www-sop.inria.fr/coprin/Claude.Michel/dossier.html.

3 Synthèse

- **État civil**
  - M. Claude Michel, né le 12/07/66 à Marseille, célibataire.
- **Situation professionnelle**
  - Ingénieur de Recherche, projet COPRIN (I3S - INRIA) (11/2000 - 06/2002)

- **Formation**
  - Doctorat informatique (1997)
  - DEA informatique (1990)
  - Ingénieur ESSI (1990)
- **Thèmes de recherches**
  - Contraintes sur les flottants
  - Réflexivité dans les langages fonctionnels et d'acteurs
- **Publications**

	Intitulé	Titre	Auteurs	Pgs	Année
Revue	TSI	*Génération automatique de cas de test structurels avec les techniques de programmation par contraintes*	B. Botella, A. Gotlieb, C. Michel, M. Rueher, P. Taillibert	26	2002
Conférences Internationales	AMAI	*Exact projection functions for floating point number constraints*	Claude Michel	11	2002
	CP2001	*Solving constraints over floating point numbers*	C. Michel, M. Rueher, Y. Lebbah	15	2001
Conférences Nationales	LMO99	*Support linguistique à la constitution de bibliothèques de méta-composants*	Claude Michel, J.P. Regourd	14	1999
	IAD& SMA	*Un exemple de raisonnement distribué: DEMACT*	Claude Michel, J.P. Regourd	12	1993
	SEIGE	*Représentation des connaissances et programmation par acteurs*	Claude Michel, J.P. Regourd	12	1992

- **Enseignements**

Matières	Cursus	Années	Nb Heures
Contraintes	Maîtrise UNSA	2002	12h
Assembleur	IUT GTR 1ère année	2002	30h
Java	IUT GTR 1ère année	2002	36h
Unix	IUT GTR 1ère année	2000-2001	60h
Réflexivité	DEA	2002 & 2001 & 1998	9h
Excel	DEUG B 2ème année	1994-1995 & 1992-1993	128h
Pascal	DEUG A 2ème année	1993-1994	64h
C	Formation Professionnelle	1991	30h
		Total	369h

- **Administration**

Intitulé	Années
Participation à la gestion du RNTL	2001-2002
Membre du comité d'organisation des JFPLC'2002 et JNPC'2002	2001-2002
Fonctions d'ingénieur système	2000-2002
Administration de l'enseignement d'Excel (DEUG 2B)	1994-1995

- **Autre**

Intitulé	Années
Participation aux travaux liés à une convention de recherche passée entre AMADEUS et l'I3S	2001-2002
Participation à l'encadrement d'un stage de DEA	1994-1995

4 Curriculum vitae

4.1 Activités Professionnelles

- **Ingénieur de recherche :** (11/2000-06/2002) (CDD[1] UNSA de 19mois)
  projet COPRIN (I3S - INRIA),
  Projet :

  RNTL INKA-floats : extension du générateur automatique de jeu de test INKA aux flottants. (en collaboration avec THALES (division Systèmes Aéroportés), AXLOG, le LSR (IMAG - Grenoble) et le LIFC (U. de Besançon))

- **Doctorat en informatique :** (1992-1997) (mention très honorable, soutenu le 5/12/1997)
  Statut : moniteur et allocataire MESR, Université de Nice-Sophia Antipolis
  Sujet : *Modèles et Implémentations d'Interprètes Réflexifs*
  Jury :

  M. Jean-François PERROT, Université de Paris VI, **président** et rapporteur
  Mme Danièle HÉRIN, Université de Montpellier II, rapporteur
  M. Jacques MALENFANT, Université de Bretagne Sud, rapporteur
  M. Pierre COINTE, École des Mines de Nantes
  M. Jacques FERBER, Université de Montpellier II
  M. Olivier LECARME, Université de Nice-Sophia Antipolis
  M. Michel RUEHER, Université de Nice-Sophia Antipolis
  M. Jean-Pierre REGOURD, Université de Nice-Sophia Antipolis, **directeur de thèse**

- **Ingénieur de recherche et développement :** (1990-1992)
  Entreprise : R&D labs, Deutsche Thomson Brandt Gmbh, Villingen, Allemagne.
  Durée : 24 mois (dont 16 au titre du service militaire)
  Responsabilités :

  Développement et mise en œuvre d'une plate-forme matérielle et logicielle pour le prototypage de téléviseurs.

4.2 Formation

- **D.E.A.** Informatique (1990), Université de Nice-Sophia Antipolis, mention bien,
  Sujet de stage :

  Un modèle de raisonnement distribué
  (encadré par Mme Danièle Hérin et M. Jean-Pierre Regourd)

- **Ingénieur** de L'École Supérieure en Sciences Informatiques (1990)
  (ESSI - Université de Nice-Sophia Antipolis).

- **D.E.U.G**. S.S.M. option Math, Physique, Mécanique (1987),
  (Faculté des Sciences et Techniques de Saint Jérôme, Aix-Marseille III).

- Baccalauréat série C (1985) (Académie d'Aix-Marseille).

---
[1]financé par le RNTL INKA-floats

5 Activités de recherche

5.1 Contraintes sur les flottants (11/2000 - ...)

Mes travaux de recherches dans ce domaine portent sur la réalisation d'un solveur pour les systèmes de contraintes définies sur les nombres flottants. Ce travail s'inscrit dans le projet RNTL INKA[2] qui a pour objectif la réalisation d'un système automatique de génération de jeux de test structurel pour les programmes écrits dans des langages impératifs tels que C ou C++. Dans ce contexte, le traitement d'expressions utilisant des nombres flottants est un problème majeur. Le point délicat réside dans la recherche de données d'entré qui garantit que, lors de l'exécution, le programme C ou C++ passera effectivement par un point préalablement sélectionné. La résolution de ce problème nécessite la prise en compte du modèle de calcul propre aux flottants lors de la définition et de la résolution des contraintes.

- **Filtrages conservatifs pour les contraintes sur les flottants**

Problème Élaborer des techniques de filtrages qui conservent l'ensemble des solutions de contraintes sur les flottants.

Difficultés La difficulté provient du fait que les techniques utilisées sur les domaines continus (i.e. les réels) font appels à des propriétés qui ne sont pas conservées par l'arithmétique sur les flottants. Par exemple, si la relation $16.1 = 16.1 + x$ a pour unique solution sur les réels 0, elle admet n'importe quel flottant appartenant à l'intervalle $[-1.77635683940025046e-15, 1.77635683940025046e-15]$ comme solution sur les flottants. Par ailleurs, les techniques de filtrage sur les domaines finis se révèlent être beaucoup trop lourdes pour filtrer des contraintes sur les flottants (il y a plus de 10^{18} flottants sur 64 bits entre -1.0 et 1.0). Il est donc nécessaire de développer des techniques adaptées au traitement des contraintes sur les flottants.

Contributions

∗ Définition d'une technique de filtrage basée sur la *box*-consistance et qui conserve l'ensemble des solutions sur les flottants (en collaboration avec Michel Rueher et Yahia Lebbah).
Publication : Cette technique à fait l'objet d'un article long publié à CP2001 (7th International Conference on Principles and Practice of Constraint Programming).

∗ Définition de fonctions de projections correctes et complètes pour les contraintes sur les flottants.
Publication : Ces fonctions de projections ont fait l'objet d'article long au 7th AIMA (7th International Symposium on Artificial Intelligence and Mathematics).
Perspectives :

∗ Développement d'une véritable $2B$-consistance qui conserve l'ensemble des solutions de contraintes sur les flottants.
Application des ces travaux à la vérification de la cohérence entre les spécifications d'un programme et son implémentation (Cette application fait l'objet d'un second RNTL (j'ai participé à l'élaboration de ce second RNTL)).

5.2 Modèles et implémentation d'interprètes réflexifs (1992-1999)

Ces travaux traitent de différents problèmes liés à la réflexivité dans les langages fonctionnels et à objets distribués. Durant mon stage de DEA, j'ai eu à utiliser un langage d'acteurs minimaliste dont les limitations auraient pu être contournées à l'aide de la réflexivité. La réflexivité, i.e. la

[2] Les partenaires du projet sont : THALES (systèmes aéroportés), AXLOG, le LSR (IMAG) et le LIFC (U. de Besançon). Ces travaux font suite à la thèse de Arnaud Goetlieb dans laquelle il montre comment construire à partir d'un programme dans un langage impératif tel que C, un système de contraintes qui assure de passer par un point sélectionné dans le programme. Pour ce faire, il s'appuie sur la forme SSA du programme ainsi que sur un graphe de flot de contrôle.

capacité qu'a l'interprète d'un langage de se prendre pour objet de son propre calcul, permet de modifier et d'enrichir les mécanismes d'un interprète. Si son potentiel semble illimité, elle soulève cependant de nombreux problèmes. Lors de ma thèse, je me suis attaqué à certain d'entre eux qui vont de la relation entre implémentation et modèle, au problème de la réutilisation de code réflexif en passant par le problème de l'inefficacité des interprètes réflexifs. L'ensemble de ces travaux se caractérise par un effort important d'implémentation.

- **Réflexivité globale**

Problème Clarification des liens entre la "philosophie" de la réflexivité et son implémentation dans les langages fonctionnels.

Difficultés Dès ses débuts, la réflexivité a bénéficié d'une définition claire. Cependant, si de nombreux interprètes sont venus illustrer différentes implémentations possibles, aucun n'a vraiment pris pour objectif de suivre au plus près la philosophie de la réflexivité. En particulier, la notion de tour d'interprètes dont l'interprète de niveau n interprète celui de niveau $n-1$ n'a pas été complètement implémentée. Par exemple, les différents interprètes disponibles ne propagent pas les effets d'une modification de la définition l'interprète de niveau $n \geq 2$ sur le niveau 0 (ou niveau utilisateur).

Contributions Introduction du concept de réflexivité globale; définition et réalisation d'un interprète réflexif basé sur une tour virtuellement infinie d'interprètes et qui permet de modifier de manière permanente le fonctionnement de l'interprète, par opposition à la modification temporaire du cours de l'interprétation proposée par les lambda réflexives.
Thèse : chapitre 4 (sert de base à un cours de DEA).
Réalisation : un interprète réflexif, I_{r^∞} (utilisé pour illustrer un cours de DEA) (610 l de code Scheme).

- **Évaluation partielle et réflexivité**

Problème Utilisation l'évaluation partielle afin de pallier au problème de l'efficacité des interprètes réflexifs.

Difficultés L'évaluation partielle est une technique de transformation de programme qui permet de spécialiser un programme à partir d'une spécification partielle des ses paramètres d'entrée. Ainsi, l'évaluation partielle d'un interprète en fonction du programme à interpréter (mais sans la connaissance des données passées au programme) peut se traduire par la suppression de l'appel à l'interprète. Il est naturel d'en déduire que son application à un interprète réflexif devrait permettre, en supprimant plusieurs niveaux d'interprétation, de réduire le coût de la réflexivité. Plusieurs recherches ont donc tenté d'utiliser l'évaluation partielle pour optimiser les interprètes réflexifs. Mais aucun de ces travaux ne s'est attaqué au problème de la réduction d'une tour virtuellement infinie d'interprètes, en particulier lorsque celle-ci permet une réflexivité globale.

Contributions Définition et implémentation d'une technique d'utilisation de l'évaluation partielle qui réduit effectivement le coût de la réflexivité, qu'elle soit locale (lambdas réflexives) ou globale.
Thèse : chapitre 5 (sert de support à un cours de DEA).
Réalisation : un interprète réflexif, I_{rpe} (1760 l de code Scheme pour l'interprète et 1998 l de code Scheme pour l'évaluateur partiel).

- **Réflexivité dans les langages d'acteurs**

Problème Introduction de la réflexivité dans une couche acteur minimaliste basée sur le modèle de Gul Agha.

Difficultés Si un certain nombre d'interprètes réflexifs pour des langages d'acteurs ont été écrit, ces travaux se sont concentrés sur les problèmes liés au modèle laissant de coté les aspects pratiques ainsi que la problématique de l'implémentation. Or, pour obtenir un interprète effectivement utilisable, il fallait aussi traiter de ces problèmes.

Contributions Définition et développement d'un interprète réflexif pour un langage d'acteurs basé sur un découpage fin des mécanismes de l'interprète, LActE/R.
Thèse : chapitre 7 (modèle), 8(implémentation) et 9 (mise en œuvre).
Réalisation : LActE/R, un interprète réflexif (utilisé par M. Jean-Christophe Pazzaglia dans sa thèse) (6580 l de code Lisp).

- **Bibliothèques de méta-composants**

Problème Permettre de réutiliser les programmes réflexifs.

Difficultés L'un des freins à l'utilisation de la réflexivité est l'accroissement de la complexité qu'elle introduit. Nous proposions la mise en place de bibliothèques de méta-composants. Cependant les langages d'acteurs ne disposent pas des outils linguistiques nécessaire à la constitution de telles bibliothèques. L'objet de ces recherches était donc d'introduire dans les langages d'acteurs les outils qui permettent la constitution de telles bibliothèques.

Contributions Définition et implémentation d'un système de spécification des éléments réifiés par l'entité méta qui permet de définir des méta-composants.
Publications : Article long publié dans LMO'99 (Langages et Modèles à Objets).
Réalisation : Un interprète réflexif pour un langage d'acteurs, SActE/R (2915 l de code C (dont 589 l de macros M4) et 2314 de code Scheme).

5.3 Représentation des connaissances et langage d'acteurs (1990)

Lors de mon stage de DEA, j'ai travaillé sur la modélisation et l'implémentation d'un système de représentation des connaissances à l'aide d'un langage d'acteurs.

- **Représentation des connaissances et langages d'acteurs**

Problème Proposer une modélisation et une implémentation du système de représentation des connaissances DEMO à l'aide des acteurs.

Difficultés DEMO est un système de représentation des connaissances basé sur les réseaux sémantiques qu'il enrichie d'une typologie de nœuds et de liens. Il bénéficie d'une gestion dynamique de la cohérence des connaissances représentées assurée par des actions lancés sur les nœuds (gestion appelée raisonnement par la suite). Une première version de DEMO avait été implémenté à l'aide du langage à objet Loops. Elle avait soulevé un certain nombre de problèmes liés au choix du langage :

* L'architecture de DEMO est basée sur trois niveaux de connaissances. Son implémentation en terme de classes et d'objets nécessitait une double hiérarchie de classes duales des niveaux de connaissances;
* Bien que le raisonnement dans DEMO soit par nature distribué, l'implémentation en Loops avait adoptée une gestion centralisé basée sur des "tableaux noirs".

Ces recherches avaient donc pour objectifs d'offrir une implémentation du modèle DEMO plus en adéquation avec son architecture et de distribuer son raisonnement en s'appuyant sur le parallélisme inhérent aux langages d'acteurs.

Contributions Définition d'un modèle de raisonnement distribué basé sur les acteurs.
Publications : Article long publié dans IAD&SMA'93 (Intelligence Artificielle et Systèmes Multi-agents),
Article long publié dans SEIGE'92 (Systèmes Experts en Informatique de Gestion).

5.4 Projet de recherche

Les travaux effectués dans le cadre du projet RNTL Inka peuvent trouver de nombreux prolongement de recherches. Les contraintes présentent un potentiel important pour l'analyse de programme et font d'ailleurs l'objet de plusieurs recherches dans ce cadre. La mise au point d'un solveur de contraintes sur les flottants ouvre de nombreuses perspectives et je pense naturellement en exploiter un certain nombre. Les thèmes de recherche que j'envisage poursuivre sont les suivants :

- **Détection automatique d'erreurs de programmation** :
Les programmes C contenant des expressions sur les flottants sont la source d'un certain nombre d'erreurs de programmation dont la détection n'est pas une tâche facile. De telles erreurs sont parfois causées par le comportement subtil des calculs sur les flottants. Par exemple, même lorsque $x \neq 0$, l'expression $x * x$ peut être égale à zero[3] et être la cause d'une division par 0 (e.g. pour l'expression $1/(x * x)$). Les contraintes sur les flottants fournissent un cadre théorique et pratique clair pour traiter ces cas difficiles. Dans l'exemple précédent, l'utilisation d'une $FP - 2B - consistance$ permet de savoir, à partir de la connaissance du domaine de définition de la variable x, si l'expression $x * x$ peut conduire ou non à un résultat égal à 0. Ces recherches auront donc pour objet de fournir un système de détection des erreurs potentielles d'un programme C faisant appel à des expressions sur les flottants. Si de tels systèmes existent déjà, ils laissent échapper des subtilités telles que celle de l'exemple précédent et il leur est particulièrement difficile de remonter d'une source d'erreur à sa cause, une caractéristique propre à la résolution de contraintes.

- **Détection de non-conformité aux spécifications** :
La vérification de logiciel est aujourd'hui une voie de recherche importante qui utilise de nombreuses techniques allant de l'inspection et de la relecture des spécifications et du code à la preuve de propriétés en passant par des techniques de tests. Chacune de ces approches a montrer ses limites. Une voie complémentaire peut être ouverte en s'appuyant à la fois sur les recherches menées sur l'utilisation des contraintes dans l'analyse de programme (e.g. l'outil Inka) et sur celle menées sur l'utilisation des contraintes dans les spécifications (e.g. l'outil BZ-testing-tool). La confrontation de ces deux systèmes de contraintes ouvre de nouvelles possibilités. Une voie possible est la détection de non-conformités aux spécification d'un programme. Il s'agit de montrer qu'il existe des valeurs pour lesquelles le système de contraintes généré à partir du programme est insatisfiable alors que le système de contrainte généré à partir des spécifications est satisfiable pour ces même valeurs. Or, si la transformation de spécifications ou d'un programme en un système de contraintes est une technique aujourd'hui maîtrisée, l'utilisation des systèmes de contraintes produits pour la détection de non-conformités est un domaine aujourd'hui encore inexploré. L'un des problèmes soulevé concerne la confrontation de spécification de propriétés mathématique avec les expressions sur les flottants qui les implémentent. L'utilisation de solveurs sur les réels et de solveurs sur les flottants devraient apporter un certain nombre de réponses à cette question.

Si ces thèmes de recherche représentent le prolongement naturel de mes travaux actuels, je reste, comme j'ai pu le démontrer par le passé, ouvert à d'autre thèmes. Plus particulièrement, je pense pouvoir profiter avantageusement de mes diverses compétences pour travailler sur les langages pour les contraintes.

[3]Par exemple, un des effets de l'arrondi est que le produit du plus petit flottant différent de 0 par lui même donne 0.

6 Activités d'enseignement

J'ai exercé des activités d'enseignement dans différents contextes :

- en formation initiale (premier cycle universitaire),
- en maîtrise d'informatique (second cycle universitaire),
- en D.E.A.,
- en formation continue.

Que ce soit durant mes années de monitorat ou durant les périodes où j'occupai un emploi d'ingénieur, j'ai toujours pris un réel plaisir à enseigner. Le partage de mes connaissances, mais aussi les relations avec les étudiants, me sont, dés le début, apparu comme des activités fort enrichissantes. Un des aspects positif de l'enseignement est son interaction avec la recherche. L'enseignement m'a en effet permis, en m'obligeant à tenir un discours clair, de consolider mes connaissances. La recherche a contribué à concevoir des cours sur des sujets porteurs qui ont sus capter l'intérêt de mes étudiants. J'envisage donc l'enseignement comme une activité constructive et je suis prêt à m'investir dans des cours différents et variés.
De part mon expérience, je me sens aujourd'hui plus particulièrement près à assurer les enseignements suivants :

- en premier cycle
  - la plupart des enseignements de base
- en licence, maîtrise
  - langages applicatifs et à objets.
  - compilation et interprétation
  - système d'exploitation (Unix)
  - algorithmique
- en DEA
  - réflexivité
  - programmation par contraintes
  - langage d'acteurs et à objets concurrents

6.1 Descriptions des enseignements

6.1.1 Formation initiale

- 2002: Enseignements de programmation en assembleur (Cours et T.P.) en IUT GTR 1ère année.
  Effectif : deux groupes de 25 étudiants.
  Heures : 30
  Statut : vacataire
  Ces cours et T.P. ont pour objectifs de familiariser les étudiants avec la programmation en assembleur INTEL. Les notions introduites sont complétées par une présentation de l'architecture d'un PC et des microprocesseurs.
- 2002: Enseignements de programmation en Java (T.P.) en IUT GTR 1ère année.
  Effectif : deux groupes de 25 étudiants.
  Heures : 36
  Statut : vacataire
- 2000-2001 : Enseignement d'Unix (T.P.) en IUT GTR 1ère année.
  Effectif : deux groupes de 25 étudiants.
  Heures : 60h
  Statut : vacataire

- 1994-1995 : Enseignement d'Excel (Cours, T.D. et T.P.) en D.E.U.G. deuxième année de Sciences Naturelles
  Effectif : deux groupes de 25 étudiants
  Heures : 64h
  Statut : moniteur
  Ces enseignements étaient similaires à ceux que j'ai pu effectuer durant l'année universitaire 1992-1993. Outre ma tache d'enseignant, j'assurais la coordination et l'administration des enseignements pour l'ensemble des intervenants de ce module.

- 1993-1994 : Enseignement de Pascal (T.D. & T.P.) en D.E.U.G. deuxième année de Sciences Physiques
  Effectif : deux groupes de 25 étudiants
  Heures : 64h
  Statut : moniteur
  Ces TD & TP avaient pour objectif de permettre à des étudiants de deuxième année de DEUG de pratiquer, à travers différents excercices, les notions apprises en cours.

- 1992-1993 : Enseignement d'Excel (Cours, T.D. et T.P.) en D.E.U.G. deuxième année de Sciences Naturelles
  Effectif : deux groupes de 25 étudiants
  Heures : 64h
  Statut : moniteur
  Ces cours/TD/TP avaient pour public des étudiants en deuxième année de DEUG biologie. L'objectif était de les familiariser avec l'outil Excel en leur montrant son potentiel dans le cadre de leur travaux. Certain des étudiants n'ayant jamais eu encore à utiliser l'outil informatique, il fallait donc enseigner les bases de l'informatique jusqu'aux macros Excel.

6.2 Maîtrise

- 2002 : T.D. de contraintes en Maîtrise d'informatique (Université de Nice-Sophia Antipolis)
  Effectif : 25 étudiants
  Heures : 12h
  Statut : vacataire
  Ces T.D. mettent en application les notions introduites dans le cours de contraintes. Différents exercices de programmations par contraintes sont mis en œuvre à l'aide du solveur de contraintes et du simplex d'Ilog.

6.2.1 DEA

- 2002 : Intervention de 3h dans le D.E.A. d'informatique de l'Université de Nice-Sophia Antipolis
  Titre : Réflexivité dans les langages fonctionnels
  Effectif : 15
  Statut : vacataire
  Ces cours présentent aux étudiants de DEA les principaux aspects de la réflexivité dans les langages fonctionnels (Ils s'appuient sur les chapitres 4 et 5 de ma thèse).

- 2001 : Intervention de 3h dans le D.E.A. d'informatique de l'Université de Nice-Sophia Antipolis
  Titre : Réflexivité dans les langages fonctionnels
  Effectif : 15
  Statut : vacataire

- 1998 : Intervention de 3h dans le D.E.A. d'informatique de l'Université de Nice-Sophia Antipolis
  Titre : Réflexivité dans les langages fonctionnels
  Effectif : 15
  Statut : vacataire

6.2.2 Formation continue

- 1991 : Cours de langage C à la Deutsche Thomson Brandt GmbH
  Effectif : 10 ingénieurs
  Heures : 30h
  Statut : ingénieur de recherche et développement
  Ces cours s'adressaient à un public d'ingénieurs. La plupart d'entre eux avaient une formation d'ingénieur en électronique et programmaient en assembleur. Ces cours avaient donc pour objectif de leur enseigner la programmation dans un langage de "haut" niveau en insistant sur certains problèmes comme le code généré par le compilateur et l'interfacage avec des routines écrites en assembleur.

6.3 Supports de cours

1. *Réflexivité dans les langages fonctionnels.*
   Support de cours,
   Objet : Présentation des notions fondamentales et des mécanismes qui régissent la réflexivité dans les langages fonctionnels (20p).

2. *Le jeu de la vie.*
   TP d'Excel,
   Objet : Réalisation du jeu de la vie (Conway, Scientific American, octobre 70 et septembre 71), un exemple d'automate cellulaire, à l'aide du tableur Excel.

7 Activités d'administration et autres responsabilités collectives

7.1 Administration de la recherche

- (2001-2002) **Membre du comité d'organisation des JFPLC'2002 et JNPC'2002**
  Les Onzièmes Journées Francophones de Programmation Logique et par Contraintes et les Huitièmes Journées Nationales de Programmation par Contraintes sont organisées par le projet COPRIN (I3S - INRIA) et auront lieu à Nice. Le comité d'organisation est composé de Gilles Trombettoni, Michel Rueher et moi-même. Je suis, entre autre, chargé de la mise en place et de la gestion du Web.
  (http://www.sop-inria.fr/coprin/jfplc2002).

- (2001-2002) Participation à la gestion du RNTL.

- (2000-2002) Fonctions d'**ingénieur système** au sein de l'équipe *contraintes* de l'I3S. (Installation et maintenance de PCs sous Linux et de Sun sous Solaris 8).

7.2 Administration de l'enseignement

- (1994-1995) **Administration de l'enseignement** d'Excel de 2e année de DEUG B.
  En complément de la coordination des enseignements, j'ai assuré l'organisation des examens (Conception des épreuves, surveillance, établissement des procès verbaux, correction des copies et remise des notes). J'ai été aidé dans l'ensemble de ces tâches par les autres moniteurs.
  L'effectif de cette seconde année de DEUG de biologie était supérieur à 200 étudiants.

8 Autres activités

- (2001-2002) Participation aux travaux liés à une convention de recherche passée entre AMADEUS et l'I3S.

- (1994-1995) **Encadrement du stage de DEA de M. Hussein Ramadan**.
  Titre : Mécanismes de distribution pour langage d'acteurs
  Taux d'encadrement : 30%
  L'objet de ce stage était d'introduire des mécanismes de distribution au sein de LActE afin de permettre sa distribution sur un réseau de stations de travail. Ce travail s'est traduit par une version de l'interprète LActE capable de fonctionner sur un réseau de Sun.

9 Liste de publications et réalisations techniques

9.1 Revue avec comité de lecture

1. *Génération automatique de cas de test structurels avec les techniques de programmation par contraintes*
   Bernard Botella, Arnaud Gotlieb, Claude Michel, Michel Rueher, Patrick Taillibert,
   Article accepté pour publication à la revue Technique et Science Informatique, Hermes Sciences Publications (26p).

9.2 Conférences internationales avec comité de lecture

1. *Exact projection functions for floating point number constraints*
   Claude Michel,
   Article accepté au Seventh international symposium on Artificial Intelligence and Mathematics (7th AIMA, supporté par "Annals of Mathematics and Artificial Intelligence"), (11p), Fort Lauderdale, Floride (US), Janvier 2002,
   Mots clés : fonctions de projections sur les flottants, 2B consistance sur les flottants.

2. *Solving constraints over floating point numbers*,
   Claude Michel, Michel Rueher, Yahia Lebbah,
   Article publié dans les actes de 7th International Conference on Principles and Practice of Constraint Programming 2001 (CP'2001), LNCS 2239 (Toby Walsh Ed., Springer Verlag), pp 524-538 (15p), Paphos, Chypre, Novembre 2001,
   Mots clés : contraintes sur les flottants, filtrage conservatif sur les flottants,
   Taux d'acceptation : 33% (44 sur 135).

9.3 Conférences nationales avec comité de lecture

1. *Support linguistique à la constitution de bibliothèques de méta-composants.*
   Claude Michel, Jean-Pierre Regourd,
   Article publié dans les actes de Langages et Modèles à Objets 1999 (LMO'99, J. Malenfant & R. Rousseau Eds., Hermes Sciences), pp. 25-38 (14p), Villefranche-sur-mer, 27-29 janvier 1999,
   Mots clés: acteurs, réflexivité, bibliothèques de méta-composants

2. *Un exemple de raisonnement distribué: DEMACT.*
   Claude Michel, Jean-Pierre Regourd,
   Article publié dans les actes des Premières Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents 1993 (IAD&SMA 93), pp 193-204 (12p), Toulouse, 7-8 avril 1993,
   Mots clés: raisonnement, distribution, langage d'acteurs, représentation des connaissances.

3. *Représentation des connaissances et programmation par acteurs.*
   Claude Michel, Jean-Pierre Regourd,
   Article publié dans les actes de Systèmes Experts en Informatique de Gestion 1992 (SEIGE'92), pp. 16-27 (12p), Nice, 17-18 septembre 1992,
   Mots clés: intelligence artificielle distribuée, programmation par acteurs, représentation des connaissances, raisonnement.

9.4 Thèse et DEA

1. *Modèles et Implémentations d'Interprètes Réflexifs.*
   Claude Michel,
   Thèse de doctorat en informatique, mention très honorable
   Soutenue le 5 décembre 1997, à l'Université de Nice-Sophia Antipolis devant le jury composé de :

   M. Jean-François PERROT, Université de Paris VI, président et rapporteur

   Mme Danièle HÉRIN, Université de Montpellier II, rapporteur

   M. Jacques MALENFANT, Université de Bretagne Sud, rapporteur

   M. Pierre COINTE, École des Mines de Nantes, examinateur

   M. Jacques FERBER, Université de Montpellier II, examinateur

   M. Olivier LECARME, Université de Nice-Sophia Antipolis, examinateur

   M. Michel RUEHER, Université de Nice-Sophia Antipolis, examinateur

   M. Jean-Pierre REGOURD, Université de Nice-Sophia Antipolis, directeur de thèse

   Mots clés : réflexivité, réflexivité globale, connexion causale, évaluation partielle, méta-acteur, efficacité, uniformité linguistique.

2. *Un modèle de raisonnement distribué.*
   Claude Michel,
   Rapport de DEA,
   Mots clés : réseau sémantique, représentation des connaissances, langage d'acteurs, raisonnement, distribution.

9.5 Réalisations techniques

1. **SActE/R**, un interprète réflexif pour un langage d'acteurs, SActE/R (2915 l de code C (dont 589 l de macros M4) et 2314 de code Scheme).
2. **LActE/R**, un interprète réflexif pour un langage d'acteurs (6580 l de code Lisp).
3. I_{rpe}, un interprète réflexif basé sur l'évaluation partielle, (1760 l de code Scheme pour l'interprète et 1998 l de code Scheme pour l'évaluateur partiel).
4. I_{r^∞}, un interprète réflexif pour un sous ensemble de Scheme (610 l de code Scheme).
5. **LActE** v1.5, une version optimisée de LActE basée sur la compilation des comportements (1594 l de code Lisp).