DOSSIER DE CANDIDATURE À UN POSTE DE MAÎTRE DE CONFÉRENCE

Guillaume Vauvert

Thèmes :

Systèmes multi-agents

Systèmes multi-agents large échelle et tolérance aux fautes

Systèmes multi-agents et aide à la décision

Systèmes multi-agents et génie logiciel

Systèmes multi-agents et simulation

Systèmes multi-agents et traitement automatique de la langue

7 avril 2005

Table des matières

Chapitre 1

Curriculum Vitae

1.1 État Civil

Nom	Vauvert
Prénom(s)	Guillaume
Date naissance	8 décembre 1971
Adresse	8 avenue Paul Fleury
	95170 Deuil La Barre
Tél	0139 831 301
Courriel	`guillaume@vauvert.net`

1.2 Situation professionnelle

Grade	Post-Doc
Affectation	équipe REGAL de l'INRIA sur le projet ALSACE.
Adresse	LIP6
	8, rue du Capitaine Scott
	75015 Paris

REGAL est une équipe jointe entre l'INRIA, le CNRS et le LIP6.
ALSACE (Adaptation dynamique de Logiciels par des Composants Evolutifs) est un projet commun au LIP6 (équipes Oasis et SRC) et l'INRIA (équipe REGAL) en collaboration avec le LIH (Laboratoire d'Informatique du Havre).

1.3 Études

2004	**Docteur en Informatique de l'Université Paris 13**	LIPN, Univ. Paris Nord
15/01	Titre : Atteinte de consensus basée sur l'échange de préférences entre agents rationnels et autonomes : application à la formation d'alliances Direction : Amal El Fallah - Seghrouchni, HDR, LIPN, Univ. Paris Nord Rapporteur : Richard EMILION, Professeur, MAPMO, Univ. Orléans Rapporteur : Suzanne PINSON, Professeur, LAMSADE, Univ. Paris-Dauphine Examinateur : Philippe DAGUE, Professeur, LIPN, Univ. Paris Nord Examinateur : Jean-Daniel ZUCKER, Professeur, LIM&BIO, Paris Nord Examinateur : Patrick TAILLIBERT, Directeur de Recherche, Thalès Systèmes Aéroportés Examinateur : Henry SOLDANO, Maître de Conférences, LIPN, Univ. Paris Nord	
1998	**DEA d'Intelligence Artificielle**	LIPN
	stage : Dassault Électronique ; resp. : Brigitte Biébow, LIPN	Univ. Paris Nord
	sujet : Utilisation de graphes conceptuels pour résumer des dépêches AFP	
1997	*SM : Officier Insertion Professionnelle et Promotion sociale*	*Armée de l'Air*
1996	**Maîtrise d'Informatique**	Univ. Lyon I
	Mention AB, 4ème sur 60	69622 Villeurbanne Cedex
1995	**Maîtrise de Mathématiques Discrètes**	Univ. Lyon I
1995	**Licence de Mathématiques Appliquées**	
1993	**DEUG A filière Mathématiques**	69622 Villeurbanne Cedex
1989	**Baccalauréat A**	Lycée Claude Bernard
		69400 Villefranche-sur-Saône

1.4 Postes

Post-Doc	LIP6 (Univ. Paris 6)	Projet DARX, INRIA	oct.	2004	1 an
Post-Doc	Institut Galilée (Univ. Paris 13)	Projet Européen ALVIS	mars	2003	6 mois
ATER	Institut Galilée (Univ. Paris 13)	Univ. Paris 13	sept.	2002	1 an
ATER à mi-temps	Institut Galilée (Univ. Paris 13)	Univ. Paris 13	sept.	2001	1 an
Moniteur	Institut Galilée (Univ. Paris 13)	Univ. Paris 13	sept.	1998	3 ans

Superpeer Semantic Search Engine (ALVIS) (`http://www.alvis.info`) est un contrat STREP de IST dans le 6ème Programme Cadre européen sur le développement de moteurs de recherche spécialisés et distribués sur le web (2004-2006).

1.5 Sujets de recherche

- Systèmes multi-agents : l'autonomie et les problématiques associées (coopération, interopérabilité, interactions, organisations).
- Systèmes multi-agents large échelle : tolérance aux fautes, QoS adaptative par mobilité et réplication notamment, grille computing.
- Systèmes multi-agents et aide à la décision : modélisation des préférences, aide à la décision multi-critères, gestion du flux informationnel.
- Systèmes multi-agents et génie logiciel : intégration d'agents-objets et agents-services, adaptation dynamique du système.
- Systèmes multi-agents et simulation : approche distribuée multi-échelle pour la modélisation, l'analyse et la simulation de systèmes biologiques, physiques ou humains.
- Systèmes multi-agents et traitement automatique de la langue : approche SMA pour le TAL pour résoudre les problèmes liés à l'approche séquentielle.

Chapitre 2

Publications

Les publications sont téléchargeables sur `http://www-src.lip6.fr/homepages/Guillaume.Vauvert/index.php?cat=rec`. Sauf pour [DHNV05], je suis l'auteur principal de tous les articles.

- Communications effectuées ou acceptées
  - Audience internationale, avec comité de sélection : [VE00a], [VE01a], [VE01b], [VE02]
  - Audience nationale, avec comité de sélection : [DHNV05]
  - Audience nationale, sans comité de sélection : [Vau00]
- Démonstration
  - [VE00b]
- Articles en préparation :
  - [Vau05b]
- Collaborations en cours :
  - [VÉ05]
  - [VC05]
- Rapports
  - Projet ALVIS : [Vau04]
  - Projet DARX : [Vau05a]

À noter que MAAMAW (référence [VE01a]), bien qu'étant un workshop européen, est considéré par les chercheurs du domaine comme étant de la qualité d'une conférence internationale. Voir le Comité de Programme à `http://www-leibniz.imag.fr/MAAMAW2001/cfp.html`.

Bibliographie

[DHNV05] Julien Derivière, Thierry Hamon, Adeline Nazarenko, and Guillaume Vauvert. Développement d'une plate-forme d'enrichissement des documents textuels : l'expérience du projet alvis. In *Journée ATALA : thème "Articuler les traitements sur corpus"*, Février 2005.

[Vau00] Guillaume Vauvert. Formation de coalitions pour agents rationnels. In *Proceedings des JLIPN'2000*, Villetaneuse, France, 11-12 Septembre 2000. VIIIèmes Journées du L.I.P.N. - Systèmes Multi-Agents & Spécifications Formelles et Technologies Logicielles.

[Vau04] Guillaume Vauvert. Alvis : format of annotation. http ://www-src.lip6.fr/homepages/Guillaume.Vauvert/alvis/alvis_format.pdf, 2004.

[Vau05a] Guillaume Vauvert. A qos approach to agents replication. 2005.

[Vau05b] Guillaume Vauvert. A qos approach to decide who and how to replicate agents in large-scale mas. 2005.

[VC05] Guillaume Vauvert and Cosmin Carabella. Autonomies : several concepts in one term ? 2005.

[VE00a] Guillaume Vauvert and Amal El Fallah - Seghrouchni. Coalition formation among egoistic agents. In *Proceedings of MAMA'2000*, Wollongong, Australie, 11-13 Décembre 2000. International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce. Récompensé par le prix de "L'un des trois meilleurs papiers" (parmi les 92 papiers acceptés).

[VE00b] Guillaume Vauvert and Amal El Fallah - Seghrouchni. Démonstration : Formation de coalitions pour des agents libres ; application à la gestion de transport aérien. In *Proceedings de JFIAD-SMA'2000*, Saint-Jean-La-Vêtre, France, 2-4 Octobre 2000. 8èmes Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents.

[VE01a] Guillaume Vauvert and Amal El Fallah - Seghrouchni. Coalition formation among strongly autonomous and weakly rational agents. In *Proceedings de MAAMAW'2001*, Annecy, France, 2-4 Mai 2001. Modelling Autonomous Agents in a Multi-Agent World. 10th European Workshop on Multi-Agent Systems. Taux d'acceptation des papiers : 13/73=18%.

[VE01b] Guillaume Vauvert and Amal El Fallah - Seghrouchni. A distributed algorithm for coalition formation among strongly autonomous and weakly rational agents. In *Proceedings de IAT'2001*, Maebashi City, Japan, 23-26 Octobre 2001. The 2nd Asia-Pacific Conferences on Intelligent Agent Technology. Taux d'acceptation des papiers : 25/134=19%.

[VE02] Guillaume Vauvert and Amal El Fallah - Seghrouchni. Formal specification of opinions applied to the consensus problem. In *Proc. of the VIIIth IBERAMIA'2002*, Seville, Spain, 2002. Comité de programme : http ://www.lsi.us.es/iberamia2002/organizacion_eng.html.

[VÉ05] Guillaume Vauvert and Richard Émilion. Preference modelisation and classification. 2005.

Chapitre 3

Activités administratives

3.1 Fonction électives ou administratives

- 2000 : Élu représentant au CIES
- 2001 : Responsable des Séminaires Juniors
- 2002-2003 : Élu représentant des Doctorants au Conseil du Laboratoire
- 2003-2004 : Élu représentant des Non-Permanents au bureau du département

  Participation active à l'élaboration de la nouvelle mouture du LMD, notamment par ma vision de l'étudiant d'aujourd'hui.

  2004 : Participation à l'élaboration de proposition au SLR : responsable du groupe de reflexion "Statut des non-permanents - LIPN" (voir `http://www.univ-paris13.fr/egrpn/cr.php`, compte-rendu du 18 juin 2004, thème "Statut des non-permanents - LIPN").
- 2004 : Membre de l'équipe système (dont co-responsable des sauvegardes des comptes utilisateurs)

3.2 Rapporteur

- Rapporteur auxiliaire pour ICMAS'2000.
- Rapporteur pour la Revue d'Intelligence Artificielle n 4, 2003.

Chapitre 4

Activités d'enseignement

4.1 Responsable de matière

– Étude de cas (2001-2002 et 2002-2003).
Remaniement du guide projet (voir `http://www-src.lip6.fr/Guillaume.Vauvert/enseign/etudK/guide/guide.ps`)
Sujets de projet : `http://www-src.lip6.fr/Guillaume.Vauvert/enseign/etudK/sujet/descriptif.ps` et `http://www-src.lip6.fr/Guillaume.Vauvert/enseign/etudK/sujet/sujet.ps`
Pour plus de détails, voir `http://www-src.lip6.fr/Guillaume.Vauvert/index.php?cat=ens`.

4.2 Enseignements dispensés

Pour plus de détail sur les activités d'enseignement, voir `http://www-src.lip6.fr/homepages/Guillaume.Vauvert/index.php?cat=ens`.

Étude de cas	DEUG MIAS2	19,5hTD	2002-2003
Initiation au génie logiciel + Projets	DEUG MIAS2	19,5hTD	2001-2002
	DEUG MIAS1	7*2hTD+6*2hTP	1998-1999
Paradigme Objet	Lic. Informatique	9hTD+15hTP	2002-2003
Spéc. formelles, Prog. par Contrat, POO, TP en Java	ILOG 1	9hTD+15hTP	2001-2002
Tâche:*Nouveaux sujets de TDs et de TPs, notamment pour intégrer les spécifications fonctionnelles*			
Programmation Impérative	DEUG MIAS1	39hTD+19,5hTP	2002-2003
Programmation Impérative en langage C	DEUG MIAS1	39hTD+19,5hTP	2001-2002
Programmation Impérative en langage C et Pascal			
Tâche:*Création de TDs et de TPs (4 sur 20)*			
Programmation Fonctionnelle	DEUG MIAS2	7*1,5hTD+6*2hTP	2001-2002
Initiation à CAML + Projets	DEUG MIAS2	7*1,5hTD+6*2hTP	1999-2000
Tâche:*Création de sujets de TDs, de TPs et de projets, suivi de projet*			
Principes des Ordinateurs et de la Programmation	DEUG MIAS1	6*(3hTD+1,5hTP)	2002-2003
Notions de base matérielles et logicielles	DEUG MIAS1	6*(3hTD+1,5hTP)	2001-2002
Tâche:*Création de TDs et de TPs (3 sur 12)*			
Algorithmique de graphes	Lic. Informatique	7*3hTD	2002-2003
Notions de base de théorie des graphes + Projets			
Tâche:*Adaptation de TDs, projet (sujet : plus court chemin entre deux stations de métro)*			
Programmation Système	Lic. Informatique	7*3hTP	2002-2003
Notions de base.			
Tâche:*Suivi des TPs.*			
Méthodologie de Travail Universitaire	DEUG MIAS1	6*(1,5hTD)	2001-2002
Résumé de Cours, animation de groupe + Fiches	DEUG MIAS1	6*(1,5hTD)	2001-2002
Tâche:*Résumé de Cours, animation de groupe + Fiches*			
Enquête Industrielle	ILOG 1	5hTD	2001-2002
Dossier et un exposé sur les ingénieurs informaticiens			
Tâche:*Suivi d'étudiants, réunion de groupe, suivi de contact avec l'entreprise*			
Architecture	Lic. Informatique	12hTD+12hTP	2001-2002
ALU, Chemin de données + Fiches résumé et techniques			
Tâche:*Création de sujets de TDs, de TPs et d'examen*			
Théorie des Langages	DEUG MIAS2	7*2hTD+6*2hTP	1999-2000
Grammaires, Lex et Yacc + Projets			
Tâche:*Installation logicielle, création de sujets de TDs, de TPs et de sujets de projets, suivi de projet*			
Calculabilité	DEUG MIAS1	7*2hTD+6*2hTP	1998-1999
Fonctions récursives, totales, programme universel			
Tâche:*Utilisation de sujets existants*			
Algorithmie	DEUG MASS1	6*3h(TD/TP)	1998-1999
Initiation au Turbo Pascal			
Tâche:*Création de quelques TDs et TPs*			
Cours d'accueil à l'Informatique	DEUG MIAS1	2hTP	2001-2002
Apprendre à se servir d'un ordinateur			
Tâche:*Explication rapide du livret*			

– Enseignement au niveau Bac+3 : 77h eq. TD
– Enseignement au niveau Bac+1 et Bac+2 : 403h eq. TD

Dans tous mes enseignements sauf les TP de programmation système, j'ai participé à l'élaboration des sujets de TD, de TP et d'examen, ainsi qu'à la correction de ces derniers. Dans toutes les matières où il y avait un projet, j'ai élaboré des sujets de projets et encadré un nombre conséquent d'étudiants.

4.3 Co-encadrement

– Co-encadrement de deux mémoires de Master "Systèmes et Applications Réprtis" (le binôme Gabriel Chatrousse/Teddy Turkay, et le binôme Arnaud Mangin/Catherine Kong), 2005, sur le sujet "HELP : Highly-distributed Electronic Library of (scientific) Publications".

4.4 Projet d'enseignement

J'ai enseigné beaucoup de matières différentes à des publics de niveaux et d'origine différente (quelques étudiants venant de Math Sup en ILOG). Enseigner des concepts qui commencent à être complexes aux Licences et aux Ingénieurs est très plaisant.
Mais j'ai obtenu autant sinon plus de plaisir à faire découvrir aux étudiant de première année la face cachée de l'Informatique – la face visible étant Internet et les jeux vidéos. L'initiation aux fondamentaux de l'informatique est souvent ardue, surtout par rapport à leur idée initiale. Parvenir à conserver leur attention, à leur donner l'envie d'apprendre et de comprendre, bref à les motiver, est ce qui m'a apporté le plus de satisfaction. Je pense que la première année de l'Université est un tournant pour la plupart, un virage souvent fatal d'ailleurs. J'espère pouvoir apporter davantage à ces jeunes étudiants en imaginant de nouvelles façons de les faire pénétrer dans le monde de l'informatique.
Parallèlement, le monde des réseaux est un monde que je connais peu et mal, alors qu'il joue un rôle de plus en plus important dans notre société de l'information, et que semble naitre un manque d'enseignants connaissant bien ce domaine (tout comme pour celui des bases de données). De plus, ce thème est proche de ma problématique de recherche. J'envisage donc de m'investir dans ce domaine afin de pouvoir l'enseigner.
J'aimerais aussi transmettre à des étudiants de Master-Rechercher le plaisir que j'ai à travailler dans le domaine des Systèmes Multi-Agents.
Depuis ma Maîtrise de Mathématiques Discrètes, je n'ai cessé d'apprendre l'informatique en autodidacte. Je me sens donc tout à fait capable d'assurer n'importe quel enseignement en informatique, du moins pour le cycle Licence, afin de m'adapter aux besoins en enseignement.

4.5 Détails

J'ai choisi de détailler mes enseignement par ordre de niveau universitaire, afin de montrer la logique des enchaînement - le cas échéant.

4.5.1 Calculabilité

Niveau : DEUG MIAS1 – Horaire : 7*2hTD+6*2hTP
Cette matière consiste à initier les étudiants aux bases théoriques de l'informatique : fonctions calculables, aux problèmes de décidabilité et de Machine de Turing universelle ...
Lors de ma première année d'enseignement, ce fût mon premier TD. Enseigner l'informatique comme on a enseigné les mathématiques à une époque (en commençant par les bases théoriques), cela ne réussît pas beaucoup aux étudiants, dont beaucoup ne connaissaient presque rien de l'informatique. Si ça avait le mérite de mettre en selle certains étudiant, la grande majorité était désarçonnée, et certains mettaient pied à terre.
Cet enseignement a été déplacé en option de deuxième année.

4.5.2 Algorithmie

Niveau : DEUG MASS1 – Horaire : 6*3h(TD/TP)
Il s'agissait d'une introduction à la programmation impérative, en Turbo Pascal. Des petits exercices, que l'on essayait d'adapter à leur domaine, avec plus ou moins de réussite. Les séances de 3 heures trop espacées (toutes les deux semaines) ne facilitaient ni l'apprentissage ni leur intérêt pour l'informatique.

4.5.3 Cours d'accueil à l'Informatique

Niveau : DEUG MIAS1 – Horaire : 2hTP
Lors de ce premier contact avec un ordinateur (même si cela est de moins en moins vrai, sauf en ce qui concerne les étudiants étrangers), il n'est pas facile de leur faire assimiler tant de notions en si peu de temps : maîtrise de la souris, des boutons, des fenêtres, des dossiers et fichiers, du clavier. Il leur faudra parfois plusieurs mois pour apprivoiser le clavier, ce qui est très handicapant ; beaucoup d'étudiants demandent des cours de dactylographie.

4.5.4 Principes des Ordinateurs et de la Programmation

Niveau : DEUG MIAS1 – Horaire : 6*(3hTD+1,5hTP)
L'objectif de cette matière est d'expliquer, avant de commencer à programmer, comment fonctionne un ordinateur. Pour cela, on utilise un modèle simplifié de l'ordinateur. Ce modèle permet d'expliquer les concepts fondamentaux sans les dénaturer : composition de l'UAL, horloge, bus, adressage mémoire. On continue ensuite à utiliser ce même modèle pour expliquer le codage des instructions et des données, et on parvient même à une très bonne introduction à la compilation.
Cette introduction est extrêmement profitable : elle permet de démystifier l'ordinateur, et de créer un langage et des concepts de référence qui nous serviront à communiquer tout au long de leur scolarité.

4.5.5 Méthodologie de Travail Universitaire

Niveau : DEUG MIAS1 – Horaire : 6*(1,5hTD)

Cette matière a pour but de donner aux étudiants des outils méthodologiques (qui leur manquent) leur permettant de réussir leurs études universitaires. Cette matière étant récente, il fallait inventer ... et expérimenter. J'ai participé à la mise en place de sujets communs (fiches résumé de cours, exposés thématiques), et inventé de nouveaux. Le premier a été la réalisation d'un dossier professionnel personnalisé, avec deux objectifs : leur montrer l'étendue du champs et des débouchés en informatique (autre que "Internet" et les "Jeux Vidéos") ; et surtout, espérer qu'une profession attrayante constitue pour eux une motivation de travail, le manque de motivation étant selon moi la principale cause d'échec pour ce public. Le deuxième sujet a consisté à définir ensemble le profil du "parfait étudiant". Très loin de leur pratique habituelle, ce profil a néanmoins contribué à améliorer leur méthode de travail, avec par exemple la mise en place de groupes de travail sérieux et motivant.

4.5.6 Programmation Impérative

Niveau : DEUG MIAS1 – Horaire : 39hTD+19,5hTP

Il s'agit de leur premier contact avec la programmation ; au cours de l'année, nous voyons les types de base, les structures de données, les instructions de contrôle, les fonctions et les procédures, la récursivité, les fichiers. À la fin de l'année, les étudiants sont capables d'écrire des programmes assez complexes utilisant nombres de ces nouvelles notions, dont la preuve est le projet qu'ils rende en Étude de cas (voir ci-après).
C'est aussi le temps des désillusions : la programmation de jeux en 3D ne leur est pas encore accessible. Nous tentons de créer des sujets de TD intéressants, mais tous les thèmes ne s'y prêtent pas facilement.
Au début, le langage utilisé était le Pascal, fruit d'une longue tradition universitaire, et duquel j'étais partisan. Mais ce langage n'est guère utilisé dans les entreprises, ce qui semble désintéresser les étudiants. Nous décidames donc collégialement de passer au C, mais en l'enseignant comme du Pascal. C'est peut-être finalement un meilleur choix que le Pascal : C est un langage très utilisé dont ils comprennent l'utilité (par exemple, pouvoir lire le code du noyau de Linux leur donne une grande satisfaction), enseigné avec une certaine rigueur qui leur sera grandement profitable (c'est du moins l'objectif). Une parfaite osmose avec la chargée de Cours (Adeline Nazarenko) a permis une excellente adéquation entre le cours, le TD et les étudiants.

4.5.7 Étude de cas

Niveau : DEUG MIAS1 – Horaire : 19,5hTD

Il s'agit d'une initiation au génie logiciel, afin de leur donner le plus tôt possible une méthodologie de gestion de projet. D'abord enseigné au deuxième semestre de la première année, puis, comme je l'avais proposé, elle fût déplacé au premier semestre de la deuxième année, notamment afin que les concepts comme les fichiers soient déjà assimilés en phase d'analyse.

J'ai été responsable de cette matière et j'ai donc largement contribué à sa mise en œuvre. J'ai remanié par deux fois le "Guide de Projet" initialement élaboré par Marc Champesme afin de mieux répondre aux

difficultés rencontrées par les étudiants. Celui-ci est téléchargeable à l'adresse : http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/guide/guide.ps.
J'ai proposé des sujets de projet, d'abord sous forme simplifiée pour permettre aux étudiants de choisir (une version trop fouillée les rebute), puis une version détaillée sur laquelle ils travailleront effectivement (http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/sujet/descriptif.ps et http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/sujet/sujet.ps). Afin d'éveiller au maximum l'intérêt des étudiants, j'ai proposé une liste large (une vingtaine de sujets) et diversifiée (mathématique, jeux, utilitaires).
Avant d'appliquer le cours de génie logiciel sur leur projet proprement dit, quelques TDs d'introduction étaient proposés. Par exemple, voici un sujet pour plusieurs séances :

- l'énoncé
  http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/tdp/td1/td1e.ps
- la correction
  http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/tdp/td1/td1c.ps
  http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/tdp/td1/concArchiFonct.eps.pdf
  http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/tdp/td1/concArchiObjet.eps.pdf

Puis nous travaillons sur un (tout) petit projet, analyse et conception d'abord (http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/tdp/td2/td2.ps), puis implémentation (http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/tdp/td2/td3.ps).
Ensuite, une fois les bases comprises sur des exemples simples, les étudiants appliquent la méthodologie (fournie par le "Guide de Projet") à leur propre projet. J'ai défini un planning à suivre de rendu de documents par les étudiants et de rendu de correction par les enseignants afin d'assurer que le projet soit termné dans les temps (voir http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/planning/planning.ps).
J'ai aussi constitué une bibliothèque qui permet un accès facile à la programmation en mode console sous Microsoft Windows. La documentation est téléchargeable à : http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/librairie/console.ps.
J'ai élaboré, en me basant sur les expériences passées de mes collègues et de moi-même, une grille de notation précise, qui prend en compte toutes les dimensions du projet, afin de ne ni léser ni avantager des étudiants qui se seraient focalisés sur une partie précise. Cette grille est disponible à http://www-src.lip6.fr/homepages/Guillaume.Vauvert/enseign/etudK/notation/notation.sxc.

4.5.8 Programmation Fonctionnelle

Niveau : DEUG MIAS2 – Horaire : 7*1,5hTD+6*2hTP

Nouveau paradigme de programmation, la programmation fonctionnelle oblige les étudiants (et moi avec eux) à déconstruire ce qu'ils ont forgé en programmation impérative pour se familiariser avec cette nouvelle façon de programmer. Le langage choisi était CAML, dont le typage strict oblige là-aussi à une certaine rigueur.
J'ai proposé des nouveaux sujets de TD et de TP, ainsi qu'un sujet de projet. J'ai ensuite suivi tous les étudiants qui avaient choisi mon sujet (une vingtaine de trinômes).

4.5.9 Théorie des Langages

Niveau : DEUG MIAS2 – Horaire : 7*2hTD+6*2hTP
Après une introduction aux grammaires au début (partie préférée par les plus mathématiciens), place aux aspect plus concrets avec la compilation et les inévitables Lex et Yacc (partie préférée par les plus informaticiens).
J'ai pris en charge une bonne partie de l'installation des logiciels nécessaires aux TPs, ce qui ne se fit pas sans mal. J'ai proposé quelques sujets de TDs et de TPs, ainsi qu'un sujets de projets. Tout comme pour les sujets en CAML, j'ai suivi tous les étudiants ayant choisi mon projet.

4.5.10 Paradigme Objet

Niveau : ILOG 1 – Horaire : 9hTD+15hTP

J'ai enseigné deux années le Paradigme Objet, pendant que Marc Champesme en assurait le cours. La première année, Marc remodela totalement le cours, afin d'en faire un vrai cours de programmation objet, et non un cours de Java. Ce changement en profondeur obligea à changer tous les TDs, et amena aussi beaucoup de discussions.
Afin d'associer une méthodologie à la Programmation par Objet, Marc Champesme décida de commencer par des spécifications formelles en CASL, puis de la Programmation par Contrat, avant de parler des notions Objet (classes, polymorphisme, héritage, ...). Le langage d'implémentation était le Java.
Les sujets de TDs et d'examen ont été élaborés à deux, à partir de discussions longues et nombreuses. Le paradigme objet a été présenté en utilisant une syntaxe proche du langage Eiffel, comme le fait Bertrand Meyer dans son livre "Object-Oriented Software Construction, Second Edition".

4.5.11 Algorithmique de graphes

Niveau : Lic. Informatique – Horaire : 7*3hTD
Le cours commençait par des nNotions de base de théorie des graphes (les aspects mathématiques), puis continuaient sur les algorithmes classiques de parcours et de recherche de meilleur chemin. Finalement, un projet leur était proposé.
J'ai repris presque entièrement les TDs existants. Par contre, j'ai proposé un sujet de projet nouveau et qui a semblé les intéresser, car ayant un intérêt pratique. Il s'agissait de trouver le plus court chemin entre deux stations de métro, les stations et leurs coordonnées ainsi que les lignes leur étaient fournies dans un fichier texte.

4.5.12 Architecture

Niveau : Lic. Informatique – Horaire : 12hTD+12hTP
L'objectif de cette matière est de donner une compréhension profonde de l'architecture matérielle d'un ordinateur : processeur (portes logiques, additionneur, UAL, pipelines), mémoire (caches, temps d'accès), disque dur ...
J'ai beaucoup collaboré avec le chargé de cours pour proposer des TDs et des TPs concrets, basés sur l'étude de documentation technique. Le travail des étudiants consistait à faire des fiches résumé de cours et des fichers techniques sur des articles que nous leur donnions. Le passage de la théorie à la pratique était ainsi très naturel.

4.5.13 Programmation Système

Niveau : Lic. Informatique – Horaire : 7*3hTP
Il s'agissait ici d'aborder la programmation système : processus, threads, sockets, pipes, mutex, ... Je n'ai participé qu'en tant que critique a posteriori à la conception des sujets TPs. Pour l'essentiel, j'ai tenté de faire le lien entre ce qu'ils voyaient en Cours et en TD avec ce qu'on faisait en TP et ce qui était utilisé dans le monde réel.
De formation plutôt mathématique, la programmation système manquait à ma culture informatique ; j'ai un peu comblé ce vide.

4.5.14 Enquête Industrielle

Niveau : ILOG 1 – Horaire : 5hTD
L'enquête industrielle permet aux étudiants d'avoir un premier contact avec le monde de l'entreprise dans des conditions plus faciles que la recherche d'emploi. Ils apprenent à se conduire en professionnel, et font connaissance avec leur futur potentiel monde du travail.
Par groupe de quatre, les étudiants doivent ainsi réaliser un dossier sur l'occupation d'un ingénieur en informatique dans une entreprise. Ils gèrent leur contact, leurs rendez-vous (discrètement contrôlés par moi). Je les rencontre réglièrement pour les guider dans la constitution de leur dossier et de leur exposé.

Expérience à renouveler, ne serait-ce que pour garder une vision souvent trop imprécise de ce qui se trame dans les entreprise, destination privilégiée de nos étudiants.

Chapitre 5

Activités de recherche

5.1 Participation à des conférences avec communication orale

- 8èmes Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents, 2000
- VIIIèmes Journées du L.I.P.N. - Systèmes Multi-Agents & Spécifications Formelles et Technologies Logicielles, 2000
- International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce, 2000
- Modelling Autonomous Agents in a Multi-Agent World. 10th European Workshop on Multi-Agent Systems, 2001
- The 2nd Asia-Pacific Conferences on Intelligent Agent Technology, 2001
- VIIIème Ibero-American Conference on Artificial Intelligence, 2002

5.2 Participation à des séminaires, colloque, groupe de travail

- Journée ATALA : Agents et langage, 2004
- Journée ATALA : Méthodes et outils pour l'évaluation des analyseurs syntaxiques, 2004
- Colloque ACI Cognitique : "Invariants et variabilité dans les sciences cognitives" (Interactions sociales dans les SMA : impact de la variabilité des préférences d'agents hétérogènes), 2000
- Groupe de travail GDR-PRC-I3 7.3 : "systèmes multi-agents", 1999

5.3 Co-organisation de colloques

- VIIIèmes Journées du L.I.P.N. - Systèmes Multi-Agents & Spécifications Formelles et Technologies Logicielles

5.4 Collaboration avec des entreprises

- mars 1998 - sep. 1998 : Stage de DEA chez Dassault Électronique (Utilisation de Graphes Conceptuels pour résumer des dépêches AFP et permettre d'intéroger cette base par des Graphes Conceptuels)
- oct. 2003 - mars 2003 : projet exploratoire avec Thalès (5 personnes).
  Passer d'une conception orientée programmation déclarative à une conception système multi-agents
  Le projet a abouti à une nouvelle architecture du logiciel de diagnostic de Thalès.
- mars 2004 (6 mois) : en Post-Doc sur le projet européen ALVIS (Superpeer Semantic Search Engine - `http://www.alvi.info`), j'ai été amené à interagir avec les partenaires industriels (Index Data Aps) et académiques (Unite Mathématique, Informatique et Génome, Institut National de la Recherche Agronomique) pour la conception d'un formalisme de représentation des annotations textuelles en XML et d'une chaine de traitement appropriée.

Chapitre 6

Résumé des résultats obtenus

6.1 Systèmes Multi-Agents

Ces résultats proviennent pour l'essentiel de mon travail de thèse.

6.1.1 Introduction

L'évolution de la programmation vers des composants de plus en plus puissants et indépendants a conduit à l'émergence des Systèmes Multi-Agents (SMA). Ils ont aussi conquis une place importante dans les systèmes de prise de décision où ils remplacent les humains, notamment en commerce électronique. Ma thèse ce concentre sur ce deuxième domaine, mais de nombreux résultats s'appliquent au premier.
Dans un contexte de forte concurrence économique, les agents doivent être considérés comme totalement autonomes, c'est-à-dire capable de prendre n'importe quelle décision à n'importe quel instant. La difficulté est alors de parvenir à faire en sorte que ces agent atteignent un consensus. C'est le problème que nous avons attaqué, mais il nécessite une longue approche.

6.1.2 Autonomie

Bien qu'étant un concept central dans la définition d'un agent, l'autonomie a longtemps été définie de façon floue. Elle a souvent été considérée plus comme un fil conducteur que comme un contrainte forte, ce qui a amené une forte divergence dans la conception même de la notion de SMA. Pourtant, l'autonomie est fondamentale pour permettre une adaptation dynamique des agents au contexte, en leur laissant toute liberté de prise de décision.
Nous distinguons trois niveaux d'autonomie.
Au niveau agent, il existe aujourd'hui plusieurs définitions de l'autonomie, mais beaucoup sont en fait des autonomies partielles, se basant sur l'argument que l'autonomie totale ne permet pas de contrôle du système. En fait, le comportement des agents n'est pas prédictible, parce qu'ils sont hétérogènes et ont leur propre stratégie, parce qu'ils peuvent tricher, et parce qu'ils peuvent avoir des fautes matérielles ou logicielles. La question est alors : comment contrôler le comportement d'un agent totalement autonome ? Pour cela, il faut que les contraintes spécifiées par les protocoles ne reposent pas sur le processus de décision interne qui est caché, mais sur les données observables, c'est-à-dire les messages échangés. On peut alors contrôler ces messages échangés pour vérifier qu'ils respectent bien le protocole, et en cas de fraude, une sanction doit être appliquée. Cette sanction est basée sur la rationalité de l'agent (ex : une amende pour une rationalité économique).
Au niveau organisationnel, les agents doivent avoir la liberté de choisir leurs interactions : période, partenaires, cause. Les alliances répondent à ces contraintes : ce sont des groupes d'agents qui se forment et se défont dynamiquement en fonction des besoins.
Au niveau système, les agents devraient aussi avoir la liberté de choisir leur protocoles, de les créer, de les adapter. Mais ces modification doivent se faire selon une certaine règle, car changer un protocole peut avoir des conséquences sur l'issue des interactions. Si l'on compare les protocoles de niveau habituels à des lois, cela revient à dire qu'il faut une constitution. Nous n'attaqueront pas ce problème dans cette thèse.

6.1.3 Consensus

Dans un système d'agent totallement autonomes, une prise de décision qui touche plusieurs agents ne peut se faire que par consensus. Pour respecter l'hétérogénéité des agents, nous avons cherché un protocole qui ne nécessite pas d'importantes capacités de calcul, ni de compétences particulières (cognitives par exemple).
Pour atteindre un consensus, les agents doivent s'échanger de l'information pour tenter de modifier les opinions des autres. L'information la plus simple à échanger est la position sur laquelle ils sont actuellement.
Afin d'inciter les agents à faire évoluer leur position, un blocage déclenche une procédure qui fait fusionner des agents au sein de coalitions.
Le protocole utilise en outre une diffusion parallèle cryptée pour s'assurer que tous ont accès à l'information en même temps. La complexité de l'algorithme est exponentielle dans le pire des cas, mais les expérimentations montrent que la convergence est assez rapide en pratique.

6.1.4 Opinion

Notre protocole nécessite un formalisme de représentation d'opinions, vus comme des ensembles de préférences, et de plusieurs opérateurs. Tous les formalismes proposés contiennent une part plus ou moins importante de transitivité. Pour nous, une préférence est une notion locale, de comparaison entre deux options, qui ne doit pas faire référence aux autres préférences.
Nous avons donc proposé un formalisme original de représentation des opinions qui permet de plus de calculer de façon très naturelle l'opinion d'un groupe, nécessaire à notre protocole.
Les expérimentations montrent que la satisfaction à une allure gaussienne (peu de mécontents et de très satisfaits, beaucoup de moyennement satisfaits), alors que le système de vote produit une satisfaction à allure uniforme (tous les degrés de satisfaction sont à peu près représentés avec le même poids).

6.1.5 Alliance

L'autonomie au niveau organisationnel nous oblige à permettre aux agents de s'organiser selon des structures souples et dynamiques, les alliances.
Le problème de la formation d'alliance entre agents totalement autonomes peut être résolu en le considérant comme un problème d'atteinte de consensus sur la structure organisationnelle à atteindre.

6.1.6 Implementation

Afin de tester les protocoles, j'ai développé une plateforme qui permet de les tester rapidement, en facilitant leur implémentation. Le code source pèse 145Ko pour la partie agent et opinion, et 32Ko pour la partie Réseau de Petri qui permet la simulation des protocoles.

6.1.7 Expérimentations

Pour expérimenter notre protocole, j'ai effectuer des tournois entre différentes stratégies, en faisant varier les paramètres.
Les résultats montrent que les agent les plus rigides ne sont pas gagnants, ce qui tend à les rendre plus flexibles, et qu'une forte compétition conduit à un consensus plus rapide.

6.1.8 Conclusion

Dans un contexte d'agents totalement autonomes, nous avons proposé un protocole dont les règles sont basées sur de données observables, qui respectent l'autonomie et l'hétérogénéité et dont les utilisateurs

sont motivés selon leur rationalité. Les expérimentations montrent que les stratégies les plus néfastes ne sont pas payantes.

6.2 Traitement Automatique de la Langue

Ces résultats proviennent de la recherche que j'ai effectuée lors de mon Post-Doc sur le projet ALVIS. Voir `http://www-src.lip6.fr/homepages/Guillaume.Vauvert/alvis/alvis_format.pdf` pour plus de détails.
J'ai proposé une architecture de traitement qui encapsule chaque outil existant dans des traducteurs afin qu'il n'existe qu'un seul format d'échange entre outils.
Après avoir étudié en profondeur de nombreux formats d'annotation de document, notre choix s'est porté sur la création d'un nouveau format en XML, qui permette une représentation fine des information lingusitiques, et qui autorisent les structures non-arborescentes. De plus, à travers ce format, nous avons dû faire un choix de la plus petite unité représentable.
À travers ce Post-Doc, j'ai découvert XML et tous les outils atenants. J'ai aussi plongé dans les outils de Traitement Automatique de la Langue, renouant ainsi avec mon stage de DEA.
J'en sors convaincu que beaucoup de difficultés seraient résolues par une approche distribuée, par les Systèmes Multi-Agents.

6.3 Systèmes Multi-Agents large échelle

Ces résultats sont issus de la recherche que je mène actuellement au sein du projet DARX. Voir `http://www-src.lip6.fr/homepages/Guillaume.Vauvert/inria_report.pdf` pour plus de détails. Il s'agit d'un travail en cours, mais déjà des résultats intéressants ont vu le jour.

DIMAX est une plateforme pour Système Multi-Agents, qui tourne au-dessus de DARX, un middleware qui autorise la réplication des processus selon plusieurs stratégies. La puissance de calcul étant de plus en plus disponible sous la forme de machines en réseau (grappe, grille), les applications réparties large échelle – notamment les Systèmes Multi-Agents large échelle – utilisent un grand nombre de ressources sur des durées telles que la probabilité de panne est non négligeable.
Il est donc nécessaire d'anticiper la panne par la réplication des processus, la seule réponse fiable. Étant impossible de répliquer tous les agents, il est nécessaire de choisir quels sont les plus critiques. Les approches existantes consistaient à transmettre des indices de criticité de la couche SMA à la couche système, qui devait alors décider qui et comment répliquer les agents. Estimant qu'il s'agissait d'un goulet informationnel, j'ai cherché à définir plus précisément la criticité. J'en suis arrivé à la considérer comme les conséquences d'une faute en terme de Qualité de Service (QoS) ; la probabilité de faute est aussi prise en compte. Ce modèle est très général, mais prédire les conséquences de la faute est très dépendant de l'application. Un vaste chantier s'ouvre pour trouver des solutions réutilisables pour permettre à des applications large échelle de fonctionner le mieux possible du point de vue de l'utilisateur (la QoS du système), en prenant en compte ces possibilités de faute. Il semble nécessaire d'inclure dans le système des connaissances du concepteur de l'application, des préférences de l'utilisateur de l'application, des connaissances dynamiques de l'application (par observation en cours d'exécution), des connaissances dynamiques sur l'utilisation des ressources et d'autres connaissances. L'objectif est alors de permettre l'intégration de toutes ces connaissances pour prendre la meilleure décision.
Cette approche semble des plus prometteuses de par sa généralité, mais elle nécessite encore de la recherche pour être opérationnelle. L'utilisation d'agents pour superviser l'exécution de l'application permet une flexibilité et une richesse d'échange informationnelle qui devraient donner des meilleurs résultats que l'approche classique.

Chapitre 7

Programme de recherche

7.1 Axes de recherche

7.1.1 Le problème central des systèmes multi-agents : l'autonomie

Formalisation de l'autonomie

Les systèmes multi-agents constituent le dernier pas de l'évolution des logiciels vers des systèmes complexes à composants de plus en plus autonomes. Alors que depuis les débuts du domaine l'autonomie est considérée comme une propriété fondamentale de l'agent, le concept n'a jamais été défini de façon très précise. Comme le montre le thème de la récente conférence AAMAS'03 (`http://csce.uark.edu/~hexmoor/AAMAS-03/AAMAS-03-cfp.htm`), le sujet acquiert aujourd'hui une place prépondérante, et à juste titre il me semble.
Je considère ici que les agents sont plutôt des super-composants, de taille et de complexité supérieure à celle des composants. Si composants et agents se ressemblent sur bien des points, je pense que vouloir faire au niveau composant ce qu'on fait au niveau agent rend difficile le contrôle au niveau agent. Certes, on bénéficierait de tous les avantages de l'autonomie des composants, mais on perdrait du contrôle au niveau agent, ce qui rendrait le système encore plus difficilement contrôlable.
Mon sujet de thèse initial sur la formation de coalitions, une fois plongé dans un contexte de commerce électronique, a évolué vers la problématique plus fondamentale : que peut-on contrôler dans un système dans lequel les agents sont totalement autonomes ? Pour respecter l'autonomie, il faut que le protocole ne pose de contraintes sur des données observables (les communications) afin de les rendre vérifiables. On ne contrôle pas ce qui se passe dans l'agent, seulement ce qui en sort.
Cette élégante solution a cependant un inconvénient majeur : la vitesse de convergence est très lente. Plus d'autonomie, cela signifie certes plus de flexibilité, d'évolutivité, d'adaptativité, un système plus ouvert, mais aussi moins de contrôle.
Mon objectif est donc de parvenir à une formalisation de la notion d'autonomie afin d'être capable de fixer le degré d'autonomie en fonction du niveau de contrôle désiré. On rogne un peu sur l'autonomie pour gagner en contrôle.
La validation du système se ferait alors en deux temps :

- valider le protocole spécifié dans un certain formalisme (Réseau de Petri par exemple), en faisant l'hypothèse que l'agent respecte les contraintes spécifiées ;
- valider l'agent en environnement fermé (génie logiciel) :

  validation off-line : vérifier que le comportement de l'agent n'entre pas en conflit avec les spécifications du protocole.

  validation on-line : vérifier à l'exécution, d'une par observation des communications par le système, et d'autre part en incluant dans l'agent un "composant-conscience" qui contrôle qu'en interne, l'agent respecte bien les contraintes données.
- valider l'agent en environnement ouvert (commerce électronique, web-services) :

  validation on-line : vérifier à l'exécution, d'une par observation des communications par le système, et d'autre part en incluant dans l'agent un "composant-législateur" qui informe l'agent ce qu'est la norme, qui ne repose que sur les communications.

En environnement fermé, on a une autonomie un peu restreinte, mais avec plus de contrôle. En environnement fermé, l'autonomie est plus grande, mais avec moins de contrôle.
Toute la difficulté est alors de spécifier les protocoles et les composants "composant-conscience" et "composant-législateur" pour le rendre le moins intrusif possible.
De mon point de vue, l'autonomie intervient beaucoup plus dans la conception des interactions (et donc protocoles) que dans celle des agents.
Un groupe de travail francophone (avec industriels) sur le sujet est en cours de constitution.

L'autonomie au niveau interaction

Originellement, l'autonomie s'appliquait à un groupe de personnes qui se régissent par leurs propres lois. Transposé au monde des agents, l'autonomie au niveau interaction doit permettre aux agents de choisir (librairie de protocoles), adapter (paramétrage de protocoles), modifier (fusion de morceaux de protocoles) ou créer les protocoles qu'ils utilisent.
Plusieurs travaux exploratoires ont montré la complexité du sujet (travaux de Jean-Luc Koning et d'Alexandre Pauchet en France), mais l'intérêt me semble majeur : apporter de l'autonomie au niveau interaction introduira c'est certain un degré supplémentaire de flexibilité et d'adaptabilité.
Cet axe me semble fondamental pour tout ce qui est lié aux systèmes multi-agents : la coopération, l'interopérabilité, les interactions, les organisations, ...

7.1.2 Systèmes Multi-Agents et Génie Logiciel

Dans les Systèmes Multi-Agents, le génie logiciel occupe une place importante. Comme le souligne Jean-Pierre Briot, les agents sont le dernier pas de l'évolution de la programmation sur le plan de l'abstraction (intentions, plans), sur le plan du découplage (après les composants) et sur le plan de la liberté d'action (autonomie).
Les approches habituelles de génie logiciel pour les systèmes multi-agents adoptent la vision classique externe du système : les agents sont conçus, puis implémentés et enfin exécutés sans qu'ils interviennent dans le processus. Or, des capacités d'introspection pourraient permettre aux agents de s'observer en train de se construire, de s'exécuter, d'interagir, de s'organiser. Ils ne seraient plus alors une marionnette dans les mains de leur concepteur, mais un acteur de leur conception.
Cette approche aurait peut-être encore plus de succès en Extreme Programming, où l'entité développée est très liée à son concepteur/développeur, et où l'expérimentation et l'amélioration continuelle nécessite de nombreuses interactions entre l'humain et le logiciel.

7.1.3 Représentation des préférences

Durant ma thèse, j'ai eu besoin d'utiliser un formalisme de représentation des préférences, mais tous reposaient sur une hypothèse de transitivité plus ou moins forte. Estimant qu'une préférence est une comparaison locale entre deux options qui ne doit pas tenir compte des autres options, j'ai proposé un nouveau formalisme qui ne se base pas sur cette hypothèse. Ce formalisme permet notamment de calculer de façon naturelle la préférence d'un groupe.
Les préférences sont très utilisées en théorie de la décision, et ce nouveau formalisme s'avère prometteur pour définir de nouveaux opérateur. De même, en classification, on utilise un concept proche de celui de la préférence, avec les mêmes difficultés. J'ai décidé d'approfondir ma recherche dans ce domaine en reprenant contact avec un de mes rapporteur, Richard Émilion. Une collaboration tripartite avec Amal El Fallah est envisagée.

7.1.4 Systèmes large échelle et tolérance aux fautes

La puissance de calcul est de plus en plus disponible sous la forme de machines en réseau (grappe, grille). Les applications réparties large échelle – notamment les Systèmes Multi-Agents large échelle – utilisent un grand nombre de ressources sur des durées qui rendent la probabilité de panne non négligeable.
iquer tous les agents, il est nécessaire de choisir quels sont les plus critiques. Les approches existantes

consistaient à transmettre des indices de criticité de la couche SMA à la couche système, qui devait alors décider qui et comment répliquer les agents. Estimant qu'il s'agissait d'un goulet informationnel, j'ai cherché à définir plus précisément la criticité. J'en suis arrivé à la considérer comme les conséquences d'une faute en terme de Qualité de Service (QoS) ; la probabilité de faute est aussi prise en compte. Ce modèle est très général, mais prédire les conséquences de la faute est très dépendant de l'application. Un vaste chantier s'ouvre pour trouver des solutions réutilisables pour permettre à des applications large échelle de fonctionner le mieux possible du point de vue de l'utilisateur (la QoS du système), en prenant en compte ces possibilités de faute. Il semble nécessaire d'inclure dans le système des connaissances du concepteur de l'application, des préférences de l'utilisateur de l'application, des connaissances dynamiques de l'application (par observation en cours d'exécution), des connaissances dynamiques sur l'utilisation des ressources et d'autres connaissances. L'objectif est alors de permettre l'intégration de toutes ces connaissances pour prendre la meilleure décision.
Cette approche semble des plus prometteuses de par sa généralité, mais elle nécessite encore de la recherche pour être opérationnelle. L'utilisation d'agents pour superviser l'exécution de l'application permet une flexibilité et une richesse d'échange informationnelle qui devraient donner des meilleurs résultats que l'approche classique.

7.1.5 Simulation de phénomènes complexes

L'approche Système Multi-Agents

Les simulations utilisent généralement deux types de modèles : soit un modèles global sous forme d'équations qui doit être résolut mathématiquement avant d'être simulés ; soit des modèles locaux de comportement des éléments et de leurs interactions qui peuvent être implémanté directement. Indéniablement, la première approche est beaucoup plus rapide sur une machine isolée ; mais elle présente de nombreux inconvénients. 1) La puissance de calcul est aujourd'hui beaucoup plus disponible sous la forme d'un large réseau de machines (grappes, grilles, ...) que sous celle d'une unique ordinateur puissant. La possibilité de distribuer le calcul sur le réseau est alors un facteur primordial qui détermine la vitesse d'exécution. La deuxième approche étant distribuée par nature, elle peut reprendre l'avantage sur des topologies matérielles adéquates. 2) Dans l'approche globale, le modèle mathématique doit être calculable, ce qui contraint et bride les possibilités de modélisation. Dans les approches distribuées, il n'y a pas de recherche de solution globale ; la limite est alors la calculabilité en un temps raisonnable des modèles individuels et des interactions. 3) Une modification même mineure d'un système nécessite dans le premier cas de recommencer tout le calcul, après avoir avoir trouvé la solution du nouveau système. Au contraire, dans le deuxième cas, la modification, le remplacement ou la suppression d'éléments du système peut se faire sans arrêter le système, ce qui facilite les expérimentations.
L'approche distribuée permet d'analyser le comportement du système, à n'importe quel instant, à n'importe quel endroit, et à n'importe quelle échelle.
Les calculs par approche distribuée utilisant un grand nombre de ressources (processeurs, mémoire, réseau), et ayant une durée de vie imporante, il est nécessaire de prendre en compte la possibilité d'erreur et de crash d'un agent (voir tolérance aux fautes).

Simulation multi-échelle

La modélisation d'un système se fait à plusieurs échelles ; selon le point de vue choisi, les structures ne sont pas les mêmes. L'utilisation simultanée de modèles à des échelles différentes – un modèle multi-échelle – présente plusieurs avantages. 1) Lors de la modélisation, il est difficile de se concentrer sur tous les niveaux en même temps. Le fait d'avoir plusieurs modèles à différentes échelles permet de se focaliser sur une certaine partie du modèle, de la raffiner, puis de se tourner sur une autre, sans perte ni de cohérence, ni d'information. 2) En simulation, il est parfois trop coûteux de simuler toutes les éléments de niveau le plus bas. Or, certaines parties du modèle n'ont qu'une faible influence sur la partie étudiée ; ou le fait de considérer un modèle de niveau supérieur (moins précis) ne produit pas un résultat moins plausible. 3) Le modèle d'un niveau $n+1$ commet généralement une erreur par rapport au niveau n. Il se peut aussi que le niveau $n+1$ ne soit pas connu. On peut alors contruire le niveau $n+1$ en apprenant par observation le niveau n, ce qui lui permet de se corriger ou de se construire. La phase d'apprentissage peut se faire en début de simulation uniquement, ou de temps en temps en cours de simulation : les niveaux $n+1$ et n

coexistent seulement le temps nécessaire.
La simulation multi-échelle facilite la modélisation (ce qui concerne une grande partie de l'informatique), permet de simuler des systèmes beaucoup plus complexes, et autorise l'adaptation dynamique des modèles. La simulation de systèmes complexes me semble un angle d'attaque prometteur pour la compréhension du génome (on sait aujourd'hui simuler les interactions rigides, mais pas les niveaux au-dessus) ou pour la simultation de phénomènes physiques et sociaux.
Ce thème est à la rencontre de plusieurs de mes passions : les systèmes multi-agents d'une part, et les fondements de la Nature (la mécanique quantique) et de la Vie (la cellule et l'ADN). Mon postdoc porte sur la tolérance aux fautes de systèmes multi-agents large échelle, ce qui s'applique tout à fait à la simulation en physique des particules et en biologie cellulaire.

7.1.6 Systèmes Multi-Agents et Traitement Automatique de la Langue

L'augmentation extraordinaire de la quantité d'information disponible sous forme électronique nécessite et autorise des outils d'extraction d'information qui accèdent au niveau sémantique. Pour cela, de nombreux outils ont été développé, chacun s'attaquant à une phase spécifique du traitement.
Leur utilisation séquentielle pose cependant de nombreux problèmes. 1) La transmission de l'information d'une étape à l'autre n'est pas standardisée, ce qui nécessite une traduction - parfois avec perte - de la sortie d'un outil vers l'entrée du suivant. 2) Le traitement est organisé de façon séquentielle ; les données traversent les outils de traitement, comme dans l'approche fonctionnelle, alors que l'approche objet a montré tout son intérêt. 3) La principale difficulté est que le traitement séquentiel des données est très sensible aux erreurs, surtout si celles-ci ont lieu dans les premières étapes ; leur propagation peut engendrer des erreurs sur des structures plus grandes, comme par exemple le rôle d'un point. Même si ces erreurs sont détectées à des étapes ultérieures, il est très difficile de corriger les effets négatifs produits.
Je pense qu'une approche Système Multi-Agents (SMA) permettrait d'aider à répondre à ces problèmes. Un SMA est un ensemble d'agents qui s'exécutent en parallèle en collaborant pour la réalisation d'une tâche, en interagissant via l'échange de messages de haut niveau (actes du langages). Un agent est un processus (ou un thread) qui a en charge une donnée (SMA objet) ou un service (SMA fonctionnel), et qui possède des capacités d'intelligence artificielle comme le raisonnement, l'apprentissage (pour améliorer son comportement, apprendre de nouveaux concepts, ...).
Ces approches ont déjà été tentées, mais leur succès relatif peut être expliqué. Les approches SMA fonctionnel ne permettent pas une gestion fine des erreurs, notamment parce qu'il est difficile d'attacher des informations aux données, et d'y faire des traitements spécificiques, comme par exemple la dépendance causale entre connaissances. Les approches SMA objet ont des difficultés à réutiliser les outils existants, ce qui donne des résultats mitigés. Enfin, les approches réactives, pour lesquelles les agents ont un état simple et réagissent aux entrées en produisant une sortie par une fonction de calcul ne permettent pas non plus la réutilisabilité, mais de plus, leur convergence est très lente.
Pour ma part, je pense qu'une approche hybride, mélangeant agents-donnée et agents-service permettrait de bénéficier des avantages des deux approches. Une sélection statique ou dynamique des agents serait alors possible en observant la dynamique du système.

Chapitre 8

Charges collectives

8.1 Travaux

- 1999 : Participation à l'organisation des Journées du LIPN (séminaire sur 2 journées, sur le thème des Systèmes Multi-Agents et des Spécifications Formelles)

  Temps : 3 semaines (non consécutives)

- 1999 : séminaire sur les Doctoriales

  Temps : 1 semaine

  Résultats : divers documents produits : le programme détaillé (`http://www-src.lip6.fr/homepages/Guillaume.Vauvert/doctoriales/doctoriales_activites.pdf`), l'Entreprise et l'innovation (`http://www-src.lip6.fr/homepages/Guillaume.Vauvert/doctoriales/doctoriales_entreprise_innovation.pdf`), la présentation faite au labo (`http://www-src.lip6.fr/homepages/Guillaume.Vauvert/doctoriales/doctoriales_transparents.pdf`)

- 2000 : Élu représentant au CIES (`http://www-rocq.inria.fr/assoc-thesards/Past/cies.html`)

  Temps : Quelques réunions

  Résultats : Aucun : mes propositions ont toutes été refusées.

- 2001 : Responsable des Séminaires Juniors

  Temps : Quelques heures

  Résultats : Un seul séminaire organisé (tous les doctorants du laboratoires étaient soir nouveau soit en cours de rédaction)

- 2002-2003 : Élu représentant des Doctorants au Conseil du Laboratoire

  Temps : 1 réunion mensuelle de 2h

  Résultats : Participation à toutes les décisions du Conseil lorsque je connaissais le sujet

- 2003-2004 : Élu représentant des non-permanents au bureau du département

  Temps : 1 réunion mensuelle

  Résultats : Participation active à l'élaboration de la nouvelle mouture du LMD, notamment par ma vision de l'étudiant d'aujourd'hui.

- 2004 : Participation à l'élaboration de proposition au SLR : responsable du groupe de reflexion "Statut des non-permanents - LIPN"

  Temps : 15 jours (difficilement calculable), en prenant en compte tout le temps passé en lectures, discussions, réunions, manifestations et leur préparation ..)

  Résultats : Les propositions du groupe de travail (`http://www-src.lip6.fr/homepages/Guillaume.Vauvert/slr/nonpermanent/`) ont été reprise par le groupe de travail de l'Université (`http://www.`

univ-paris13.fr/egrpn/cr.php, compte-rendu du 18 juin 2004, thème "Statut des non-permanents - LIPN").

- 29/04/2004-31/08/2004 : Membre de l'équipe système (dont co-responsable des sauvegardes des comptes utilisateurs)

  Temps : Très variable : 1 à 6 heures par semaine

  Résultats : Des sauvegardes bien effectuées, des utilisateurs heureux.