

## Résumé de carrière de Gérard Berry

### Professeur au Collège de France, chaire Algorithmes, machines et langages

Version 2 du 11 décembre 1014

#### La recherche, 1970-2001

##### Les balbutiements

Je suis tombé dans la science très tôt, ayant appris à lire en bonne partie dans les livres de Chimie, dont j'adorais les formules et les dessins. J'ai ensuite viré vers l'astronomie, puis, adolescent je suis revenu à la chimie avec un labo bien fourni en produits vraiment intéressants que le principe de précaution a bannis depuis longtemps. J'ai rencontré mon premier ordinateur à l'X, à travers une vieille machine bizarroïde qui m'a beaucoup occupé et m'a permis une observation qui allait déterminer ma recherche : celle du gouffre fondamental qui sépare nous humains, intuitifs, lents et assez peu rigoureux de la machine, hyper-rapide, totalement obéissante, mais d'une noire stupidité. J'ai été frappé par la difficulté fondamentale de construire des langages réconciliant ces deux exacts opposés, ainsi que par la grande difficulté d'écrire des programmes qui marchent vraiment. Maurice Wilkes, l'un des créateur d'un des tous premiers ordinateurs, l'a bien expliquée : « As soon as we started programming, we found to our surprise that it wasn't as easy to get programs right as we had thought. Debugging had to be discovered. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs »<sup>1</sup>.

##### L'entrée en recherche

En 1970, je suis entré au Corps des Mines, encouragé par Pierre Lafitte, alors sous-directeur de l'Ecole des mines. J'ai travaillé sur le langage TIF, Traitement et interrogation de fichiers. C'était un projet intéressant, mais ses moyens et son encadrement étaient inadaptés. J'en garde cependant un bon souvenir, en particulier à cause des noms des modules du compilateur, allant d'Apéritif à Digestif en passant par Sportif, Coupetif et bien d'autres.

A la même époque, j'ai retrouvé deux copains de l'X, Jean-Marc Steyaert et Philippe Flajolet, qui m'ont entraîné dans le DEA de Maurice Nivat à Jussieu, puis au bâtiment 8 du tout jeune IRIA, où je devins « chef de projet », titre assez virtuel. En cet endroit se sont rencontrés beaucoup de ceux qui allaient construire la recherche française en informatique (Bruno Courcelle, Gérard Huet, Gilles Kahn, Jean-Jacques Lévy, Jean Vuillemin, etc.), ainsi que certains de ceux qui allaient devenir des acteurs majeurs des logiciels chez NeXT puis Apple (Jean-Marie Hullot et Bertand Serlet). Ce séjour fut extraordinaire, car nous rencontrions tous les chercheurs mondiaux du domaine, dont ceux d'Edimbourg (Robin Milner, Gordon Plotkin, etc.) qui ont été cruciaux pour moi. Nous participions aussi à des écoles de printemps aussi sympathiques qu'efficaces, généralement au fin fond de la campagne française.

---

<sup>1</sup> Dès que nous avons commencé à programmer, nous avons réalisé à notre grande surprise qu'il était bien plus difficile de faire des programmes justes que nous le pensions. Nous avons alors découvert le debugging. Je me rappelle de l'instant exact où j'ai réalisé que j'allais passer une grande partie de ma vie à trouver des erreurs dans des programmes que je croyais justes

## Lambda-calcul et séquentialité

Nourri aux langages de programmations modernes et à leur sémantique mathématique qui définit le sens exact des programmes, j'ai travaillé avec Jean Raymond Abrial puis Bruno Courcelle pour ma thèse de troisième cycle. J'y ai fait ma première découverte scientifique, celle des « fonctions stables », qui allait me mener loin. Un des grands problèmes ouverts de l'époque était de savoir ce que pouvaient réellement calculer les langages de programmation séquentielle classique comme Pascal ou C, et plus généralement le lambda-calcul, formalisme de 1936 père de tous les langages. Plotkin a montré que ces langages étaient incapables d'effectuer des calculs réellement parallèles, autorisés par les modèles mathématiques du calcul comme le modèle de Scott, le meilleur disponible à ce moment. Maurice Nivat m'a alors donné un article à relire de Robin Milner (prix Turing 1992), *Fully Abstract Models of Typed Lambda-Calculi*, qui montrait l'existence et l'unicité d'un bon modèle mathématique de la séquentialité, mais sans dire de quoi il était fait. Partant de mes fonctions stables, sa caractérisation m'a occupé pas moins de neuf ans, longtemps en collaboration avec Jean-Jacques Lévy, mon maître en lambda-calcul.

En 1977, j'ai décidé de migrer à Sophia-Antipolis, dans le nouveau centre initié par Pierre Lafitte au tout début du développement du parc technologique. J'avais quatre raisons : le climat, celui de Paris n'ayant jamais été ma tasse de bruite ; être plus près des montagnes, l'alpinisme étant alors ma passion ; me rapprocher de la mer, la voile me tentant beaucoup ; et, surtout, monter un centre de recherche à partir de rien, car il n'y avait pas pour moi de stimulant intellectuel plus fort. J'ai eu la double chance de prendre comme thésard Pierre-Louis Curien et d'organiser une école d'automne mémorable car toutes les stars du domaine étaient venues. La recherche du modèle de la séquentialité s'affina encore, avec de nouveaux résultats fondamentaux, conduisant à ma thèse d'Etat en 1979 et à la thèse de Curien en 1980 (?). J'ai alors développé un langage très original nommé CDS avec mon thésard Matthieu Devin, rencontré une nuit de Noël déprimée sur UUCP, le précurseur d'Internet. J'avais surnommé ce langage novateur « le langage des années 90 », heureusement sans préciser le siècle – il ne reste plus que 76 ans à attendre !

## Les débuts d'Esterel

J'étais dans un labo bi-disciplinaire, mi informatique, mi automatique, une excellente idée. En 1982, mes amis automaticiens Jean-Paul Marmorat, Jean-Paul Rigault et l'électronicien Jean-Marc Tanzi avaient construit une petite voiture autonome pour un concours organisé par le journal Microsystèmes. Bourrée d'algorithmes, dotées d'accélération et de freinage puissants, elle était impressionnante en virage, mais pas en ligne droite car le labo était trop petit pour de grandes pistes. Elle fut battue en finale par une voiture à vitesse constante développée par des adolescents. Mais nos automaticiens avaient conçu sa conduite à l'aide d'un langage informel original, qui m'a tout de suite fasciné.

Pour moi, il était temps de changer de sujet, et l'idée de travailler sur l'informatique du contrôle des objets physiques m'apportait de l'air frais. De plus, nous sentions tous que l'informatique embarquée prendrait une importance considérable, bien visible maintenant et vouée à exploser encore plus dans les années qui viennent avec l'Internet des objets. A l'époque, la programmation dite « temps-réel » s'attachait surtout à gérer à bas niveau les interférences complexes entre le temps et les événements propres aux systèmes physiques à contrôler et ceux propres au matériel et logiciel de calcul, ce qui rendait les réalisations complexes, peu fiables et peu susceptibles de progrès scientifiques. J'ai rapidement eu une idée iconoclaste : supprimer conceptuellement ces interférences en prétendant que l'ordinateur calculait infiniment vite, ce qui lui donnait par construction un temps de réflexion nul. Comprendre comment programmer un ordinateur infiniment rapide me séduisait par son caractère incontestablement pataphysique. Le sujet s'est immédiatement avéré fécond, et nous avons développé le langage, appelé Esterel en l'honneur des montagnes rouges de ce beau massif et d'une analogie subliminale entre « terel » et « temps réel »

Mais l'implémentation semblait difficile, et, surtout, des problèmes fondamentaux de paradoxes de causalité se posèrent rapidement. Ils furent partiellement résolus dans la thèse de Laurent Cosserat, qui construisit rapidement un prototype permettant de simuler l'exécution des programmes directement à partir de leur sémantique mathématique.

En généralisant le bel algorithme de Brzozowski sur la traduction des expressions rationnelles en automates déterministes, nous comprîmes comment produire des automates à partir de programmes Esterel. Avec Philippe Couronné, je développai alors en Lisp le compilateur Esterel v2, qui nous permit les premières expérimentations sérieuses.

### **Le projet commun Meije entre les Mines et l'Inria**

Deux événements importants se produisirent à ce moment. D'abord, l'arrivée de l'Inria à Sophia-Antipolis, qui nous conduisit immédiatement à construire un projet commun Mines / Inria appelé Meije. Côté Inria, Gérard Boudol s'intéressait aux calculs de processus parallèles récemment développés par Robin Milner, qui étaient les premiers formalismes permettant véritablement de parler de parallélisme général ; devenu chercheur, son thésard Robert de Simone, s'intéressait à la preuve de programmes fondée sur les calculs de processus, bientôt implémentée par ses thésards Valérie Roy et Didier Vergamini. L'interaction avec Esterel fut constante. Ensuite, la rencontre avec deux groupes de recherche français mêlant également automaticiens et informaticiens : celui de Paul Caspi et Nicolas Halbwachs à Grenoble, qui développait le langage Lustre, et celui d'Albert Benveniste et Paul Le Guernic à Rennes, qui développaient le langage Signal. Ces langages différaient d'Esterel par leur style, mais pas par leur fond. Nous les nommèrent rapidement les langages synchrones, et notre collaboration ne s'est jamais arrêtée. Notre façon de faire n'intéressait pas vraiment les chercheurs des autres labos mondiaux qui identifiaient systématiquement parallélisme et asynchronisme, à l'opposé exact de notre hypothèse synchrone, et nous sommes longtemps restés marginaux. Mais nous savions que notre approche était la bonne dans notre cadre des systèmes cyber-physiques, et nous avons tenu bon. Et, surtout, nous avons été financés au bon niveau et au bon moment par le GRECO C3 (Groupe de Recherches Coordonnées Concurrence, Coopération, Communication). Plus tard, j'ai aussi travaillé sur l'asynchronisme, en définissant avec Gérard Boudol la machine abstraite chimique ou CHAM, une autre idée iconoclaste qui a mis du temps à percer avant de devenir un standard du domaine.

En 1985, si le langage Esterel était presque dans sa forme finale, le compilateur Esterel v2 que j'avais développé avec mon thésard Philippe Couronné était très lent et engendrait un code objet explosif en taille. Lors d'un séjour de mon ami Ravi Sethi des Bell Labs (Laboratoires Cloche), nous fîmes une découverte intéressante sur un problème technique ancien mais toujours d'actualité : l'algorithme de Brzozowski<sup>2</sup> que nous avons adapté à Esterel, très élégant mais inefficace, se transformait facilement en l'algorithme de McNaughton-Yamada, très efficace mais difficile à prouver. J'embauchai en thèse Georges Gonthier, qui a considérablement affiné la sémantique mathématique d'Esterel et amélioré ses algorithmes de compilation grâce en particulier à la connexion algorithmique précitée. Georges Gonthier est un chercheur exceptionnel qui, après bien d'autres résultats, conduit maintenant une carrière extraordinaire sur la formalisation sur ordinateur de résultats mathématiques majeurs.

Comme je n'aime pas trop rester au niveau des prototypes, je décidai alors de développer un vrai compilateur nommé Esterel v3 en réunissant toutes les forces du labo<sup>3</sup>, et de le coupler aux outils de

---

<sup>2</sup> Pour la traduction des expressions rationnelles en automate par calcul formel sur les expressions

<sup>3</sup> Frédéric Boussinot, Jean-Paul Rigault, Jean-Paul Marmorat, Annie Ressouche, Jean-Marc Tanzi, permanents, Raphaël Bernhard, Yannis Bres, Xavier Fornari, Georges Gonthier, Frédéric Mignard, Jean-Pierre Paris,

vérification formelle développés par l'équipe de De Simone. Après une paire d'années, le compilateur et le vérifieur fonctionnaient bien. Sous l'impulsion d'Emmanuel Ledinot, avec qui j'ai partagé le prix Science et défense en 1999, Dassault Aviation s'en servit intensivement pour le Rafale, d'autres intérêts industriels apparurent (Bell Labs, Renault, Thomson, etc.), et CISI ingénierie puis ILOG développèrent puis commercialisèrent un atelier de programmation Esterel Studio. A l'université de Nice, Charles André a construit une version graphique très intéressante d'Esterel, les SyncCharts, reprise dans la version industrielle d'Esterel v3, et Daniel Gaffé et Marie-Agnès Peraldi ont étudié les interfaces avec le monde réel, largement fondés sur un principe d'exécution cyclique déjà familier des industriels.

## **Les circuits électroniques**

Mais le problème de l'explosion de la taille du code généré par le compilateur n'était toujours pas résolu. La solution est venue au début des années 1990 d'une rencontre avec les circuits électroniques, dans le groupe de Jean Vuillemin, alors à Digital Equipment où j'étais consultant, suivie d'un voyage en Inde où j'allais déjà régulièrement. Je réalisai alors qu'Esterel peut se compiler directement en circuits, eux-mêmes soit réalisables directement sur silicium soit traductibles en logiciel en évitant complètement l'explosion du code généré. Le compilateur Esterel v3 fut rapidement transformé en Esterel v4 pour intégrer les nouveaux algorithmes, et nous le couplâmes avec les toutes nouvelles optimisations et vérifications formelles développées chez Digital Equipment en utilisant les diagrammes de décision binaire de Bryant, une révolution dans le domaine. Mais le compilateur v4 exigeait des restrictions parfois gênantes sur les programmes compilables dues encore une fois à des problèmes de causalité, ce qui posait de gros problèmes à nos utilisateurs. La solution a été trouvée de façon inattendue, encore une fois à cause de contacts dans le domaine fondamental bien différent de la logique formelle et en particulier de la logique intuitionniste, celle qui n'accepte pas le tiers exclu. Le résultat final, démontré d'abord avec Tom Shiple en 1996, puis considérablement amélioré par Michael Mendler en 2010, établit une relation étonnante entre la propagation des courants électriques et la logique intuitionniste. Pour moi, ce résultat avait complètement bouclé la compréhension théorique du langage, et nous avons continué à développer le système Esterel maintenant nommé v5, en améliorant le compilateur et le vérifieur formel, et en développant des optimiseurs sophistiqués. Mais d'autres chercheurs continuent à explorer d'autres solutions mieux adaptés à des programmes plus généraux que ceux que traite Esterel en particulier en termes de dynamique, comme Frédéric Boussinot avec Reactive C et ses successeurs et Louis Mandel avec Reactive ML.

L'utilisation industrielle s'est considérablement intensifiée dans les années 1990. En 1995, Synopsys, société américaine de CAO de circuits, a acheté une licence source des outils et incorporé Esterel dans un de ses produits. Son concurrent direct Cadence a également adopté Esterel en recherche pour son système Polis de conception conjointe matériel / logiciel. Surtout, nous avons commencé à travailler avec le Strategic CAD Lab d'Intel à Portland, qui s'intéressait à nos avancées conceptuelles. J'étais alors consultant dans ces trois compagnies, ce qui m'apportait beaucoup de nouveaux points de vue.

---

Dumitru Potop, et Horia Toma, thésards, plus Colas Nahaboo, Jean-Bernard Saint sur des projets en collaboration avec l'équipe Centaur de Gilles Kahn à l'Inria Sophia-Antipolis.

## **Esterel Technologies, 2001-2009**

En 2000, la technologie labo d'Esterel atteint son plafond. Mais mon travail chez Intel avec Michael Kishinevsky (mon maître en circuits) nous conduit à développer une nouvelle version d'Esterel appelée Esterel v7 qui traite les calculs sur les données de façon novatrice et bien adaptée aux caractéristiques des circuits modernes. D'autres fabricants de circuits s'intéressent à ce travail, comme Texas Instruments, alors leader des circuits de téléphone, et ST Microelectronics. Je constate alors que le développement en labo de suffit plus, d'autant plus que mon équipe commence à se lasser, ce qui est bien normal. En parallèle, l'industrialisation d'Esterel est reprise par Simulog, la plus ancienne startup de l'Inria, en mauvaise forme. Son PDG, Eric Bantegnie, décide d'en extraire la petite équipe Esterel pour construire un développement de clientèle à beaucoup plus grande échelle. Avec Bernard Dion comme directeur technique et une équipe commerciale nouvelle, il fonde Esterel Technologies. Je propose alors de construire un GIE entre le labo et la nouvelle société, ce qui échouera à cause de la consternante timidité d'une organisation de recherche partenaire (qui n'est ni le CNRS ni l'Inria). Je décide alors de rejoindre Esterel Technologies en tant que directeur scientifique. Nous montons rapidement deux équipes de développement, une pour le langage et son compilateur et l'autre pour l'environnement de programmation Esterel Studio, ainsi qu'une équipe de consultants. Cette période a été exaltante et particulièrement créatrice. Esterel v7 est un langage riche et complexe ; diriger la construction de son compilateur et en programmer moi-même une bonne partie, passer à l'échelle des circuits réels développés par nos clients et résoudre les problèmes scientifiques tout à fait nouveaux qu'ils nous posaient m'ont vraiment poussé au bout de mes possibilités scientifiques et techniques, comme je j'ai expliqué en détail dans mes premiers cours 2014 au Collège de France. Un tel travail aurait été clairement impossible dans un laboratoire, et je remercie tous ses acteurs, et en particulier Xavier Fornari, Marie Mrozciewicz, Bruno Pagano, Marc Perreaut et Gunther Siegel. Cela ne m'a pas empêché de rester en contact avec la recherche : une contribution importante à la simplification de la sémantique et de la compilation d'Esterel a été apportée par Olivier Tardieu, mon dernier thésard encadré en commun avec Robert de Simone à l'Inria.

Mon activité ne se limitait pas au développement technique, car j'étais aussi directement impliqué dans la relation client, travaillant directement avec notre commercial Jean-François Baggioni, notre responsable des consultants Fabrice Capelle, et les consultants parmi lesquels je remercie particulièrement Sylvie Granier. Comme j'adore rencontrer des gens et des points de vue différents de ceux de mon monde, les relations directes avec les clients m'ont énormément apporté.

Esterel v7 devait être initialement appliqué aussi au logiciel embarqué, et nous avons bien sûr gardé nos clients historiques Dassault Aviation, Thalès, etc. Mais, dans ce domaine, nous étions en compétition avec SCADE, l'outil industriel construit autour du langage Lustre de Paul Caspi et Nicolas Halbwachs. Cet outil était développé par l'équipe de Daniel Pilaud au sein de Verimag, structure commune entre le CNRS et la société Verilog, puis chez Telelogic, une société suédoise de télécommunications qui avait racheté Verilog à cause de ses produits télécom. Lors de l'effondrement de la bulle Internet en 2001, Telelogic a eu des problèmes majeurs et nous avons pu lui racheter SCADE et son équipe. SCADE avait déjà de beaux succès, étant utilisé en nucléaire, en ferroviaire, et surtout par Airbus, d'abord pour les commandes de vol de quelques avions puis pour la plupart des logiciels embarqués de l'A380. Son originalité par rapport à la concurrence était son générateur de code certifiable KCG, qui simplifiait considérablement la certification des logiciels embarqués selon la norme internationale DO-178B de l'avionique, qui est de loin l'industrie la plus sérieuse sur ces sujets. Nous avons alors décidé de séparer les activités matérielles et logicielles

d'Esterel Technologies et de développer une nouvelle version SCAD6 de SCAD6 intégrant les points forts d'Esterel et Lustre, jusqu'alors orthogonaux<sup>4</sup>. Esterel Technologies, maintenant rachetée par la société américaine Ansys, poursuit ce développement, et est devenue leader du temps-réel embarqué critique avec plus de 250 clients à travers le monde en avionique, ferroviaire, industrie lourde, etc.

De son côté, en 2008, Esterel v7 avait atteint un succès technique considérable, permettant de développer des circuits plus vite et plus sûrement grâce à la rigueur du langage et aux outils de vérification formelle intégrés. Les circuits produits étaient d'excellente qualité en taille et vitesse et souvent meilleurs que ceux développés par les techniques classiques. Le langage Esterel v7 était en cours de standardisation internationale, avec une bonne collaboration recherche / industrie car tous les laboratoires publics avaient droit à des licences gratuites du langage et de l'ensemble de son outillage. Mais l'industrie des circuits était en pleine évolution. La difficulté grandissante de la conception des circuits, le coût sans cesse croissant des usines et la volatilité de la clientèle rendaient cette industrie de plus en plus coûteuse et les marchés franchement incertains. Nous avons alors vu nos clients abandonner le domaine, d'abord progressivement, puis brutalement lors de la grande crise de 2008. C'est le soir de notre dernière fête clients, où ils exposaient combien les résultats étaient bons, que nous avons tristement compris que nous ne pourrions plus tenir. Il a fallu fermer l'activité circuit d'Esterel Technologies, en licenciant une partie du personnel, mais aussi en renforçant au passage l'activité SCAD6 d'une partie de l'équipe Esterel. J'ai alors décidé de revenir dans la recherche en rejoignant l'Inria et sa commission d'évaluation comme expliqué plus loin.

Esterel v7 et son environnement Esterel Studio ont été ensuite repris par Synfora, une startup américaine, avec laquelle nous collaborions. A son tour, Synfora a été rapidement rachetée par Synopsys, alors leader de la CAO de circuits. Depuis ce jour, Synopsys, avec qui j'avais beaucoup travaillé dans les années 1990, a décidé de mettre le langage et son outillage au congélateur. Je n'ai même pas pu en récupérer l'usage universitaire, que nous donnions gratuitement à toutes les universités et centres de recherche publics quand j'étais chez Esterel. Le deuil a été difficile, même si je j'ai toujours été conscient des risques de l'opération...

## **Retour à l'Inria, 2009-2012**

J'ai intégré l'Inria Sophia-Antipolis de mars 2009 à septembre 2012, d'une part comme président de la Commission d'évaluation de l'institut, avec une action détaillée dans mon discours à la médaille d'or du CNRS, d'autre part en tant que chercheur dans le projet Indes dirigé par Manuel Serrano. Ce dernier développe un langage et un système de programmation très originaux nommés collectivement Hop. Le langage Hop, extension du langage fonctionnel Scheme (et maintenant, dans sa version Hop.js, aussi de Javascript, un standard du web), permet de grandement améliorer la programmation d'applications Web fondées sur la description de documents en HTML et la communication à travers le protocole http pour réaliser le modèle client / serveur du Web. HOP facilite la construction par programmes de documents HTML complexes, et, surtout, permet d'intégrer dans un seul code source les programmes du ou des serveurs et ceux du ou des clients d'un système distribué avec téléchargement automatique de tous les codes dans leur lieu de destination. Le système d'exécution de HOP est remarquablement efficace. Tout ceci permet de remplacer complètement la tour de Babel actuelle des langages et outils du Web par un seul système remarquablement homogène.

---

<sup>4</sup> J'avais auparavant proposé de départager Esterel et Lustre par un combat aérien entre un Rafale programmé en Esterel et un Airbus programmé en Lustre, mais l'équipe Lustre avait obstinément refusé.

Une fois construite cette infrastructure, une nouvelle façon de réaliser des applications distribuées devient naturelle : Les construire e et en composant les innombrables services déjà disponibles sur le Web (un service est un programme callable à distance et qui rend ses résultats sous forme de structures de données au lieu d'afficher des pages Web) et en intégrant n'importe quel objet connecté. Mais l'activité des différents services et objets contactés de façon asynchrone sur le réseau conduit à des échanges de messages et une gestion d'erreurs très complexes. Une couche spécifique d'orchestration est indispensable pour y mettre de l'ordre. Il est rapidement apparu qu'un langage de type Esterel pouvait contribuer à résoudre ce problème, et nous avons développé la couche langage HipHop intégrée dans Hop pour ce faire. HipHop reprend les idées principales d'Esterel, mais leur ajoute un caractère dynamique indispensable pour le Web et déjà expérimenté dans les langages Reactive C de Frédéric Boussinot et Reactive ML de Louis Mandel : si un avion a un nombre fixe d'ailes, de réacteurs et de roues, les applications Web sont tout à fait dynamiques, avec un nombre d'interlocuteurs variables au cours du temps et beaucoup de traitement d'erreurs. Cela pousse nos théories et implémentations à leur limite. Une implémentation prototype a été réalisée, mais il faudra aller beaucoup plus loin.

## **Ma recherche au Collège de France**

J'ai quitté Inria pour le Collège de France en septembre 2012. Je parlerai de mon enseignement un peu plus loin. Pour ce qui est de la recherche, en dehors de la recherche sur HipHop qui continue en coopération, j'ai démarré un nouveau sujet avec Lionel Rieg, ATER attaché à ma chaire. L'idée est de construire un compilateur mathématiquement prouvé du noyau d'Esterel en utilisant Coq, le merveilleux assistant de preuve développé en France à la suite des découvertes fondamentales de mon ami Gérard Huet de son étudiant Thierry Coquand (qui a eu le prix spécial Gödel du centenaire pour ce travail). Coq est utilisé maintenant à travers le monde pour des preuves allant de celles d'un compilateur C par Xavier Leroy à la formalisation complète de grands théorèmes de mathématiques par Georges Gonthier et ses collaborateurs en passant par des preuves de protocoles de sécurité en milieu industriel. Un compilateur Esterel prouvé et directement extrait de sa sémantique formelle serait une première dans mon domaine.

Sur un autre plan, je m'intéresse de plus en plus à la création musicale sur ordinateur. Lors de mon cours 2009-2010, j'ai rencontré Arshia Cont, chercheur à l'IRCAM, qui a développé le remarquable système Antescofo qui permet d'inverser le rapport homme / électronique dans la musique contemporaine. Au lieu de forcer les hommes à suivre les parties électroniques jouées de façon rigide par l'ordinateur, Antescofo permet de décoder en temps réel le tempo et l'articulation avec lesquels les humains jouent, et d'utiliser cette analyse pour asservir la partie électronique aux interprètes. En redonnant le pouvoir aux hommes, Antescofo modifie complètement la donne. Antescofo a été rapidement adopté par les compositeurs à travers le monde. En collaboration avec des compositeurs comme Philippe Manoury et Marco Stroppa, l'équipe Inria / IRCAM / CRNS d'Arshia Cont développe maintenant de nouvelles notions de partitions algorithmiques intégrant les concepts d'Esterel. Ce nouveau sujet est fondé sur l'idée que l'ordinateur n'a aucune raison d'être programmé « à l'ancienne », c'est-à-dire avec des partitions formées de suites linéaires de notes. Comme il peut synthétiser et enchaîner tous les sons possibles notre idée est de les programmer avec des techniques et langages de programmation spécifiques, pour lesquels nos langages synchrones sont une base idéale. Les problèmes à résoudre restent nombreux : il faudra arriver à des notations aussi puissantes mais beaucoup plus précises que celles que les compositeurs développent chacun dans leur coin, tout en restant raisonnablement compatibles avec elles. Il faudra aussi développer des stratégies de rattrapage habile des inévitables erreurs des instrumentistes ou du système de suivi, car, quoi qu'il arrive, « the show must go on ».

Je m'intéresse aussi de plus en plus à d'autres sujets, comme les neurosciences et la biologie computationnelle, dans lesquels les questions liées au temps et les événements sont tout aussi importantes. Toutes ces recherches sont développées dans le cadre d'un séminaire régulier Chronos

soutenu par un réseau ANR, qui regroupe des chercheurs de disciplines variées qui ont envie de s'écouter mutuellement : automatique, informatique, biologie computationnelle, neurosciences et informatique musicale.

## **L'enseignement, de Montessori au Collège de France**

### **L'école Montessori des Pouces Verts**

L'enseignement m'a toujours passionné, ce que je dois tenir de ma mère qui avait la réputation d'être une enseignante exceptionnelle en mathématiques. De 1995 à 2001, j'ai présidé l'Ecole Montessori des Pouces Verts à Mouans-Sartoux, près de Sophia-Antipolis. La méthode introduite par Maria Montessori dans les années 1920 est bien plus qu'une méthode d'enseignement ; c'est une méthode d'éducation globale visant à faire fleurir la personnalité spécifique de chaque enfant. Son étonnant succès à cet égard devrait en inspirer plus d'un. J'ai commencé par intervenir dans la classe des 9-12 ans, mêlant cours de météorologie (la physique avec une expérience par jour) et d'informatique « débranchée », sans ordinateur. J'ai ensuite été promu dans la classe des 6-9 ans, bien plus subtile : autant les 9-12 ans ont déjà une pensée analytique, autant les 6-9 ans fonctionnent par une forme de compréhension globale pas forcément facile à déclencher. Cette expérience a été pour moi extraordinaire, et je me dis toujours qu'il faudra que je l'écrive en détail.

A titre d'exemples, nous avons étudié le binaire en comptant sur le nez au lieu des doigts, les protocoles de communication en transportant des images dessinées par des pions de Go par une variété de méthodes allant de la voix à l'allumage de lampes, la programmation en me transformant en robot stupide capable seulement de comprendre les ordres les plus simples, puis les images et sons numériques, les nombres et les raisons pour lesquelles nous les aimons chacun différemment, etc. Quand on la nourrit bien, la richesse imaginative des enfants de cet âge est confondante – mais elle est aussi facile à détruire par d'une éducation trop rigide qui ne prend pas assez en compte leur personnalité individuelle, n'est pas fondée sur la confiance et provoque l'ennui, l'ennemi absolu de l'apprentissage.

### **La chaire d'Innovation technologique au Collège de France**

En 2007, au fond de mon lit dans un hôtel Californien, j'ai reçu un appel de Pierre-Louis Lions, professeur de mathématiques au Collège de France. Il me proposait de tenir la seconde édition de la chaire annuelle Innovation Technologique Liliane Bettencourt du Collège de France, afin d'y introduire pour la première fois un cours d'informatique. Mon cœur a tout simplement bondi. Industriel à cette époque, j'ai eu immédiatement l'accord de mon PDG Eric Bantegnie. Comme je savais la population française globalement ignorante du sujet, j'ai décidé de transposer mon enseignement Montessori aux adultes, en présentant l'informatique de façon un peu ludique mais tout à fait précise. Plus que d'aligner les points techniques, mon but était de construire chez mes auditeurs un schéma mental adapté à la compréhension des phénomènes informatiques qui forgent le monde moderne. J'espère y avoir réussi, avec une leçon inaugurale quelque peu non-standard et des cours et séminaires présentant chacun un des pans principaux de l'informatique. Je remercie les 12 chercheurs et industriels qui ont donné les séminaires associés à mon cours, ainsi que tous les intervenants de mon colloque final. C'était l'époque du début de la mise en ligne des cours au Collège, et l'effet de mon cours et de ses séminaires a été amplifié par des téléchargements considérables. J'ai rencontré en particulier pas mal de professeurs et d'élèves de lycée qui m'ont dit l'avoir suivi en ligne. Il y a eu plusieurs émissions de radio associées au cours, qui ont probablement constitué les toutes premières sur la science informatique.

Je remercie en particulier la fondation Bettencourt-Schueller, Pierre Corvol, alors administrateur du Collège, pour son soutien constant, Marie Chéron, chargée de la chaire et dont l'action a été magnifique, et toutes les équipes techniques et administratives dont l'efficacité a été parfaite.

Quitter le collège a été difficile pour moi, et mon dernier cours s'est naturellement terminé par « la dernière séance » d'Eddy Mitchell. L'avenir m'a donné tort sur ce point.

### **La chaire « Informatique et sciences numériques »**

Suite au succès de ma première chaire, le Collège de France et l'Inria ont décidé en 2008 de créer une nouvelle chaire annuelle dédiée spécifiquement à l'informatique et de me la confier pour 2008-2009. Obéissant à la loi du Collège qui est de toujours faire un cours nouveau, j'ai construit un cours bien différent du précédent. Son sujet était la modélisation du calcul informatique, directement au cœur de mes préoccupations et de mes papiers théoriques et pratiques depuis toujours. Le succès a encore été au rendez-vous, et j'en remercie chaleureusement mes 11 séminaristes. Les titulaires suivant de cette chaire ont tous donné des cours remarquables dans une grande variété de domaines, et j'espère que cela va continuer.

### **La chaire « Algorithmes, machines et langages »**

En 2011, de plus en plus conscient de l'importance du sujet dans les sciences et dans la société en général, le Collège de France décide de créer une chaire de plein exercice en informatique. Ma candidature est acceptée en 2012. Ma leçon inaugurale et mes cours 2012-2013 ont porté sur le rôle du temps et des événements dans l'informatique, avec les modèles synchrones de type Esterel / Lustre en 2012-2013, puis sur les modèles plus sophistiqués de temps multiples et de temps continus / discrets en 2013-2014. Les trois premiers cours de 2013, donnés à Sophia-Antipolis, ont également présenté ma vision du rapport recherche / industrie. J'ai essayé d'y expliquer pourquoi les chercheurs avaient une vue très partielle des points de vue industriels et réciproquement, et pourquoi comprendre finement ces questions rend la coopération bien plus facile et efficace. Je remercie encore les 20 séminaristes de ces deux années.

Mes prochains cours seront consacrés à la vérification formelle des programmes, sujet où les avancées actuelles sont considérables.

**A suivre....**