

Algorithmes, machines et langages

M. Gérard BERRY, membre de l'Institut
(Académie des sciences) et de l'Académie des technologies, professeur

ENSEIGNEMENT : LE TEMPS ÉLARGI :
HORLOGES MULTIPLES, TEMPS DISCRET ET TEMPS CONTINU

J'ai continué le cycle de cours « Le temps et les événements en informatique » commencé en 2012-2013 dans le cadre de la chaire Algorithmes, machines et langages, créée le 4 juillet 2012 comme première chaire de plein exercice en informatique. Les cours et les séminaires associés se sont déroulés en deux périodes successives. Les trois premiers cours, du 15 au 29 janvier 2014, ont eu lieu à l'Inria Sophia-Antipolis, où j'ai effectué la plus grande partie de mon travail de recherche avant de bifurquer dans l'industrie chez Esterel Technologies de 2001 à 2009. Les six cours suivants, du 4 mars au 9 avril 2014, ont eu lieu au Collège de France. Les vidéos et supports de tous les cours et séminaires sont disponibles en version française originale ou en version anglaise sur le site du Collège de France ^a.

Les trois cours donnés à l'Inria Sophia-Antipolis ont été dédiés à la relation science-industrie dans le développement des langages Esterel et SCADE, présentés en détail en 2012-2013. Ils ont tenté de mettre en lumière les difficultés fondamentales de cette collaboration, souvent mal connues des acteurs, et les solutions qu'on peut leur apporter. Ils ont aussi souligné l'intérêt et la difficulté des urgences scientifiques posées par les applications industrielles. Les six cours donnés ensuite au Collège de France ont été consacrés à des modèles de temps plus variés que ceux vus en 2012-2013, qui étaient essentiellement synchrones mono-horloges. Si le modèle synchrone est certainement le mieux compris dans le domaine, il est loin de couvrir toutes les questions d'intérêt du sujet. De plus, il s'applique essentiellement aux systèmes compacts : circuits mono-horloges, systèmes embarqués centralisés sur un seul processeur, etc. Le cours 2013-2014 et les séminaires associés résumés ici se

a. Pour les cours, voir <http://www.college-de-france.fr/site/gerard-berry/course-2013-2014.htm> et pour les séminaires, voir <http://www.college-de-france.fr/site/gerard-berry/seminar-2013-2014.htm> [NdÉ].

sont intéressés aux systèmes beaucoup plus distribués géographiquement et temporellement. Ces systèmes plus récents mêlent plusieurs bases de temps, dont certaines peuvent être discrètes et d'autres continues. Les temps en question peuvent être définis par des horloges physiques ou logiques, qui peuvent soit rester indépendantes soit être synchronisées de façon plus ou moins précises par des protocoles informatiques spécifiques. Les applications vont des systèmes électroniques sur puces multi-horloges aux grandes bases de données rendues temporellement cohérentes au plan mondial, en passant par toutes les formes de systèmes embarqués répartis.

Comme la science et la technique sont beaucoup moins bien comprises dans ces domaines récents, une présentation unifiée du sujet était prématurée. Les cours se sont donc orientés vers la présentation de points de vue et de résultats cohérents dans des sous-domaines spécifiques. Ils ont étudié des systèmes distribués cadencés par des horloges distantes synchronisées ou non, que ce soit à l'intérieur d'un circuit ou à l'échelle de la planète. Ils ont montré comment coordonner des calculs faits à diverses fréquences, comment synchroniser des horloges à distance, comment parler simultanément de temps continu et de temps discret, comment assurer la correction temporelle de programmes temps-réels distribués, etc. J'ai aussi présenté en détail un de mes résultats personnels préférés, l'étonnante correspondance entre la propagation des courants électriques dans les circuits et la logique intuitionniste. Le dernier cours a été consacré à la réponse aux questions posées pendant l'année.

Tous les cours ont été suivis de séminaires sur les mêmes thèmes, sauf pour les deux séminaires du dernier cours, consacrés au rôle du temps en neurosciences (deux exposés plus généraux de neurosciences avaient déjà eu lieu les années précédentes). Le rôle des oscillations et synchronisations dans le système nerveux apparaît de plus en plus fondamental ; son étude avec les outils de l'informatique me paraît donc constituer un excellent domaine de travail pour l'avenir.

Cours 1 : Esterel, de la recherche à l'industrie : la vision « labo »

Ce cours a présenté l'aventure du développement des langages synchrones selon le point de vue des laboratoires de recherche, en se focalisant sur Esterel et Lustre / SCADE. J'y ai illustré les développements successifs de la théorie et des réalisations logicielles qui ont permis de passer des idées initiales aux langages et systèmes finaux, à travers une succession de découvertes et d'étapes scientifiques. J'ai décrit les nombreux avantages apportés par la collaboration directe avec des industries comme l'avionique, les télécommunications, etc., en particulier de par la fourniture par les utilisateurs industriels d'exemples et de problèmes d'un ordre de grandeur plus grands et plus complexes que ceux qu'imaginent les chercheurs. J'ai ensuite discuté des contraintes associées : la nécessité pour les industriels de disposer d'une chaîne de développement sans maillon faible, ce qui oblige les chercheurs à traiter des points techniques pas toujours excitants pour eux ; la nécessité pour les partenaires et clients industriels d'avoir des produits stables au cours du temps, ce qui les rend méfiants vis-à-vis des avancées scientifiques imprévues en cours de projet, etc.

Séminaire 1 : Le traitement du temps en automatique

Juliette Leblond (Inria Sophia-Antipolis), le 15 janvier 2014

L'automatique est la science de la modélisation, de l'analyse, de l'identification et de la commande des systèmes dynamiques. L'implémentation numérique effective des algorithmes de commande est étroitement liée à l'informatique en temps réel, d'où la liaison avec le cours précédent.

Le séminaire a présenté différents modèles et traitements de problèmes d'automatique dans lesquels le temps joue un rôle essentiel. Après quelques rappels historiques, il a introduit les modèles mathématiques classiques de systèmes dynamiques, qui relient au fil du temps continu ou discret les entrées commandes du système à ses sorties, appelées en termes techniques ses commandes et ses observations. Les descriptions dites « externes » relient directement la sortie du système à ses entrées et au temps, tandis que les représentations internes font intervenir aussi son état. Comme en physique, ces descriptions consistent souvent en des équations différentielles ordinaires (EDO) ou aux dérivées partielles (EDP), avec éventuellement la présence de retards. Dans le cas linéaire, la description externe est une relation de convolution entre l'entrée et la sortie.

Le séminaire a montré comment ces modèles temporels se prêtent à la formulation de propriétés de causalité, d'invariance dans le temps, de linéarité, de stabilité, de commandabilité, d'observabilité et d'optimalité, en insistant sur des exemples linéaires et stationnaires : comment garder son cap en boucle ouverte ou fermée, comment planifier des trajectoires en temps minimal, comment stabiliser des structures flexibles, etc. Pour faire le lien avec le séminaire suivant du 22 janvier, certaines de ces questions ont été formulées dans le domaine fréquentiel.

Cours 2 : Esterel, de la recherche à l'industrie : la vision industrielle

En miroir du cours du 15 janvier, ce cours a présenté les points de vue du fournisseur industriel et de ses clients. Je me suis appuyé sur mon expérience personnelle de directeur scientifique de la société Esterel Technologies, directement impliqué dans le développement et la commercialisation de ses produits : Esterel Studio, atelier formel de développement de circuits synchrones et multi-horloges complexes, et SCADE (*safety critical applications development environment*), dédié aux logiciels embarqués de haute sécurité¹. Sur l'exemple d'Esterel Studio, j'ai insisté sur la qualité et la coordination des rouages nécessaires pour qu'une bonne idée se transforme en produits utilisables pour des développements industriels chez des clients standard, chez qui le pragmatisme technologique l'emporte quasiment toujours sur la vision scientifique. J'ai illustré par l'exemple une difficulté centrale, bien analysée dans le fameux livre *Crossing the Chasm* de Geoffrey A. Moore : les arguments techniques valables pour les industriels visionnaires, et donc promus par les chercheurs, ne le sont en général pas du tout pour les pragmatiques qui forment le gros des acheteurs. De plus, l'introduction de nouveaux langages et de nouvelles méthodes de programmation est mal perçue par ceux qui n'en voient pas immédiatement les avantages, craignent pour

1. SCADE a été initialement développé par le laboratoire Verimag, commun au CNRS et à la société Verilog, à partir du langage Lustre conçu par Paul Caspi et Nicolas Halbwachs au CNRS Grenoble.

la continuité de leur chaîne d'outils, et n'ont pas forcément envie de subir ou d'organiser de nouvelles formations.

J'ai ensuite expliqué pourquoi et comment SCADE, l'autre produit d'Esterel Technologies, a réussi à franchir ce fossé dans le domaine des logiciels critiques en avionique, transport ferroviaire, industrie lourde, etc. : il s'est présenté comme une amélioration sensible des modèles de programmation existants, plutôt que comme une façon de faire radicalement nouvelle, ce qui n'était pas possible pour Esterel Studio. Sa valeur ajoutée est triple : d'abord, il ne change pas fondamentalement les habitudes des utilisateurs ; ensuite, étant claire et formelle, la définition de son langage de programmation est nettement plus simple et plus propre que celle des langages concurrents ; enfin, développée dès les débuts industriels de SCADE, la certifiabilité de son compilateur vis-à-vis des normes avioniques les plus strictes a joué un rôle clé pour son adoption car elle simplifie la validation et permet de raccourcir considérablement le temps de la recertification système lors de modifications du code.

Séminaire 2 : Une fréquence peut-elle être instantanée ?

Patrick Flandrin (ENS Lyon), le 22 janvier 2014

Enregistrer un événement, mesurer l'évolution d'un système, transmettre une information : autant de situations donnant naissance à des séries temporelles qu'il convient d'analyser, de modéliser, de transformer. On sait depuis Fourier ce que de tels traitements peuvent gagner à quitter l'espace direct et naturel du domaine temporel pour aller dans celui des fréquences, mais au prix d'un paradoxe d'interprétation physique. Alors même que l'expérience quotidienne nous conforte dans l'idée que rythmes et oscillations peuvent changer sans cesse, la fréquence mathématique est un concept immuable, éternel et invariant dans le temps. La question est donc de donner sens à l'oxymore d'une « fréquence instantanée » qui réconcilierait la théorie et la pratique – en quelque sorte, peut-on prendre une photo d'un son ? Le séminaire a étudié les limitations intrinsèques de ce type d'approche, en lien avec diverses formes de « principes d'incertitude ». Il a présenté quelques résultats récents sur des réponses possibles à cette question toujours ouverte et a donné des illustrations dans des domaines aussi différents que la détection directe d'ondes gravitationnelles ou l'optimalité du système sonar de la chauve-souris.

Cours 3 : Urgences scientifiques posées par l'industrie : masquages d'horloges, circuits multi-horloges, ECOs et vérification formelle

Le cours du 29 janvier nous a fait revenir à la science à travers trois problèmes scientifiques durs posés pour Esterel par les utilisateurs industriels et dont la solution était vitale pour eux, ce qui la rendait tout aussi vitale pour Esterel Technologies :

- le traitement de la causalité, difficulté intrinsèque des langages synchrones dont la solution a été trouvée dans la relation surprenante entre logique et électricité, présentée en détail dans le cours 7 (voir ci-dessous). J'ai présenté de nouveaux exemples de circuits cycliques naturels pour lesquels ce résultat s'applique, comme le décodeur à la volée de la longueur des instructions du Pentium ;

- le traitement des circuits multi-horloges, maintenant universels en électronique. Nous l'avons résolu en assimilant sémantiquement les horloges à des signaux standard dont l'absence suspend les transitions d'états des modules contrôlés. Cette approche originale dans le domaine ne nécessite aucun changement sémantique du langage noyau d'Esterel. Elle a permis de développer une couche multi-horloge simple au sein du langage Esterel v7, accompagnée bien sûr d'outils de synthèse, de simulation et de vérification formelle complets ;

- le très délicat traitement formel des ECO (*engineering change order*), qui sont des demandes de correction de bugs d'un circuit à faire impérativement sans recompiler son code source. Le coût en temps et en argent de la recompilation / re-synthèse est en effet tel que la seule solution viable pour corriger un bug est de « patcher » directement les masques, mais en prouvant formellement et automatiquement l'identité de la modification physique et de la correction des sources. Notre solution a exigé de développer des outils spécifiques de traçabilité du code compilé par rapport au code source et de relier ces outils de traçabilité à ceux de vérification formelle.

Ces problèmes sont souvent bien différents des questions que se posent spontanément les chercheurs, mais tout aussi scientifiques, tout aussi intéressants, et pas moins difficiles.

Séminaire 3 : Prototypage virtuel de système sur puce pour une simulation rapide et fidèle

**Laurent Maillet-Contoz (ST Microelectronics)
et Matthieu Moy (Verimag Grenoble), le 29 janvier 2014**

La complexité grandissante des circuits intégrés modernes a conduit au développement de nouveaux types de circuits formés de nombreux composants intégrés sur la puce, qu'on appelle « systèmes sur puces ». Il n'est aujourd'hui plus possible d'attendre les premiers prototypes physiques pour valider la bonne intégration de leurs composants et pour développer le logiciel embarqué, souvent de taille considérable. Il est donc nécessaire d'utiliser des prototypes virtuels des circuits tôt dans le flot de conception du système global. Cet exposé bicéphale a présenté les techniques de prototypage virtuel fondées sur l'environnement de programmation standard du domaine : SystemC/TLM (*transaction-level modeling*).

La première partie a présenté les techniques établies et leur utilisation dans le contexte industriel de ST Microelectronics, à travers les différentes plates-formes virtuelles utilisées pour l'anticipation du développement logiciel embarqué et la vérification fonctionnelle de son fonctionnement sur les plates-formes. Elle a décrit leurs caractéristiques et leurs limitations. La représentativité des modèles ainsi construits a ensuite été étudiée, y compris pour les nouveaux besoins liés à la validation de propriétés extra-fonctionnelles devenues essentielles, comme la consommation d'énergie. La seconde partie a présenté les techniques de programmation et d'implémentation développées récemment pour répondre aux nouveaux besoins précités et aussi pour exploiter efficacement le parallélisme des machines hôtes de la simulation. Ces techniques ont été développées au sein d'une collaboration de longue durée entre ST Microelectronics et le laboratoire Verimag (CNRS Grenoble).

Cours 4 : Circuits multi-horloges, métastabilité, synchroniseurs et FIFOs asynchrones

Ce premier cours parisien a d'abord brièvement présenté les divers sujets traités dans l'ensemble des cours restants. Il a été ensuite consacré aux problèmes délicats des circuits multi-horloges.

Alors que les circuits digitaux du XX^e siècle avaient le plus souvent une seule horloge, les systèmes sur puce du XXI^e siècle en comportent en général plusieurs, en particulier pour cadencer des composants de fréquences intrinsèques différentes et pour abaisser la consommation d'énergie en réduisant la vitesse de certaines horloges selon la charge de travail locale des circuits qu'elles cadencent. Les rapports temporels de ces horloges peuvent être variés : synchronisées, en décalage de phase, harmoniques ou encore vraiment asynchrones, ce dernier cas étant le plus complexe en termes d'échange d'information. En effet, le rôle de l'horloge réceptrice est de déclencher par son front montant l'échantillonnage des entrées des registres (échantillonneurs) qu'elle commande. Quand une donnée produite sur une horloge émettrice doit être échantillonnée par un registre cadencé par une horloge réceptrice asynchrone, c'est-à-dire sans liaison avec la précédente, il se peut qu'un front de l'horloge réceptrice commandant l'échantillonnage d'une entrée arrive au moment où l'entrée elle-même est en train de changer. Dans ce cas, le registre peut soit rendre une valeur booléenne aléatoire soit rester métastable entre les deux valeurs booléennes pendant un temps aléatoire à décroissance exponentielle avant de se stabiliser sur une valeur arbitraire. Il n'est donc plus possible d'utiliser un simple registre pour l'échantillonnage fiable désiré. En m'appuyant sur le travail de Ran Ginosar (Technion, Israël), j'ai analysé deux montages fondamentaux qui permettent de contourner cette difficulté intrinsèque à l'aide de protocoles spécifiques : les synchroniseurs multi-horloges et les files d'attente FIFO multi-horloges. J'ai montré que ces montages sont délicats et coûteux, et que diverses optimisations apparemment habiles peuvent s'avérer radicalement fausses.

Séminaire 4 : Architecture à horloges faiblement synchronisées pour les applications d'automatique temps-réel distribuées

Albert Benveniste (Inria), le 5 mars 2014

Ce travail a plusieurs sources. Tout d'abord, une longue amitié avec Paul Caspi, lequel fut longtemps consultant auprès des équipes d'Airbus qui développent les logiciels de contrôle de vol en SCADE. Paul Caspi savait que, lorsqu'ils devaient utiliser une architecture distribuée pour un système critique, les ingénieurs d'Airbus avaient recours à une architecture de communication asynchrone et non bloquante, donc sans conflit d'accès. Chaque entité (calcul, communication, capteurs et actionneurs) y est rythmée par son horloge périodique autonome, non synchronisée avec les autres. Paul Caspi avait étudié la méthodologie de programmation d'Airbus pour compenser les non-déterminismes induits. Dans les années 2000, cette méthodologie a été épurée et réduite à une explication simple. Une autre approche importante est la philosophie TTA (*time-triggered architecture*) de Hermann Kopetz, qui crée un modèle strictement synchrone et temporellement périodique dès le niveau du *hardware* de calcul et de communication, offrant ainsi une cible naturelle au déploiement de programmes SCADE ; cette approche est plutôt favorisée par Boeing. Pourtant, du point de vue de l'automaticien, une boucle de

régulation peut se satisfaire d'une certaine dose d'asynchronisme, tolérable grâce aux marges de robustesse des lois de contrôle : pourquoi s'embarrasser de TTA dans ce cas ? Comme il faut aussi prendre en compte les contrôles discrets tels que la protection et la gestion des modes de marche, une idée naturelle est de tolérer l'asynchronisme pour le contrôle continu distribué tout en combattant son non-déterminisme au niveau du contrôle discret. Enfin, le monde des circuits connaît une problématique similaire pour les circuits asynchrones sans propagation globale d'horloge, comme expliqué dans le cours précédent.

La vision finale qui émerge est simple. La programmation synchrone permet de déployer des applications sur toute machine virtuelle obéissant au modèle des réseaux de Kahn². Le problème est alors de concevoir, au-dessus de l'architecture physique distribuée de systèmes pseudo-périodiques non synchronisés, une couche algorithmique donnant à cette dernière un comportement en réseau de Kahn. Benveniste, Caspi et d'autres collègues ont développé deux techniques pour cela. L'une est un algorithme à jetons qui ne fait pas appel aux dates physiques et s'avère très proche des circuits élastiques à la Cortadella³ ; l'autre utilise des bornes sur la gigue et la dérive inter-horloge pour « épaissir » les événements, mathématisant ainsi la philosophie originale des ingénieurs d'Airbus. Les réseaux de Petri et l'algèbre max-plus permettent d'analyser de manière unifiée ces deux approches.

Cours 5 : Synchronisons nos montres : la synchronisation d'horloges matérielles et logicielles en environnement distribué

Ce cours était directement relié à la conférence donnée le 11 décembre 2013 par Edward Lee (université de Berkeley) et intitulée « Cyber-physical systems: A fundamental intellectual challenge »..

Avoir l'heure et se synchroniser avec d'autres sont longtemps restés des problèmes mal résolus. J'ai d'abord rappelé que les rythmes visibles du soleil et de la lune ne sont pas réguliers, et que les divers instruments inventés au cours des âges étaient peu précis : clepsydre, sablier, horloges à ressort ou à balancier, etc. La première montre vraiment précise (quelques secondes sur une traversée de l'Atlantique) a été conçue par John Harrison en 1759, pour gagner le concours de l'amirauté britannique ouvert en 1714 à la suite du naufrage d'une partie de la flotte anglaise aux îles Scilly – la Royal Society ne reconnaissant cependant sa victoire qu'en 1773. Ce naufrage était directement dû à une mauvaise mesure du temps conduisant à une mauvaise estimation de la longitude. Plus tard, avec l'arrivée du train, des ratages de synchronisations temporelles ont conduit à de graves collisions ferroviaires. La situation ne s'est réellement améliorée que dans la seconde moitié du XX^e siècle avec la généralisation du télégraphe, qui a permis l'harmonisation des heures locales et la création des fuseaux horaires. Aujourd'hui, il est possible d'atteindre une précision bien plus grande grâce aux horloges atomiques, embarquées par exemple dans les satellites GPS, et aux protocoles informatiques subtils qui synchronisent les horloges des ordinateurs ou objets informatisés de toutes sortes.

2. Réseaux de processus déterministes connectés par files d'attente non bornés.

3. Voir les séminaires des 14 et 21 mai 2013 dans le cours de 2012-2013.

Comme l'avait déjà expliqué Edward Lee dans son séminaire du 11 décembre, la synchronisation précise des horloges d'ordinateurs distants joue maintenant un rôle critique dans de nombreux domaines d'application, qui vont des systèmes cyber-physiques contrôlant des objets ou des processus complexes à la réplication en temps réel de grandes bases de données distribuées, en passant par les grands réseaux de capteurs et d'actionneurs ou de gestion d'énergie. Le cours a montré qu'un contrôle fin du temps est essentiel pour assurer le déterminisme global de ces applications distribuées, qui est le plus souvent une condition obligatoire pour elles. Il a présenté les deux protocoles normalisés principaux permettant la synchronisation temporelle à distance : d'abord PTP (*precise time protocol*), protocole très précis car fondé sur la communication avec des horloges atomiques terrestres ou embarquées dans des satellites GPS, mais qui exige des réseaux à temps de transmission prévisible ; ensuite NTP (*network time protocol*), moins précis et plus complexe mais adapté aux réseaux asynchrones de type Internet et maintenant utilisé par pratiquement tous les ordinateurs.

Séminaire 5 : Distribution d'horloges à grande échelle sur une puce François Anceau (CNAM/UPMC LIP6), le 12 mars 2014

La complexité des circuits intégrés croît de manière exponentielle depuis près de 45 ans, selon un rythme qui ne se ralentit pas. La complexité des circuits actuels permet la réalisation monolithique de systèmes informatiques complets appelés « SoCs » (pour *systems on chips*), souvent multiprocesseurs. Ces systèmes sont découpés en zones isochrones qui représentent la taille des blocs qu'il est possible de synchroniser de manière classique à partir des horloges locales à chacun de ces blocs. Il est fréquent, et souhaitable, que les zones isochrones correspondent à des entités fonctionnelles ou à des regroupements de certaines d'entre elles.

Les communications entre ces zones isochrones peuvent être asynchrones (GALS pour *globally asynchronous locally synchronous*) ou synchrones (GSLs pour *globally synchronous locally synchronous*). Le séminaire a montré que, contrairement aux idées reçues, la réalisation de SoCs synchrones de grande taille semble possible. Elle présente de nombreux avantages tels que la communication en un seul cycle entre des zones isochrones voisines, l'assurance d'éliminer tout risque de métastabilité et la possibilité de prévenir le gestionnaire du réseau en cas de risque de sortie du domaine de fonctionnement synchrone. Ceci offre une plate-forme sûre pour des applications de haute fiabilité.

Pour réaliser un système d'horlogerie synchrone, un projet ANR HODISS a été mis sur pied par le Lip6/SOC, avec le CEA-LETI et SupElec⁴. Le système a été réalisé sous la forme d'un réseau de PLL (*phase-locked loops*, oscillateurs asservis en phase) interconnectés par des comparateurs de phase. L'originalité du projet résulte dans la réalisation entièrement numérique des PLL, qui a nécessité le développement de blocs *full-custom*. Un prototype de réseau 4×4 a été réalisé en CMOS 65 nm et a fonctionné dès le premier lot de prototypes. Un réseau 10×10 est en cours de développement. L'étude de la topologie et de la stabilisation du réseau a été reprise postérieurement, en s'appuyant sur une analogie entre le

4. Ce travail n'aurait pas été possible sans la collaboration étroite de Dimitri Galayko, maître de conférences, de Chuan Shan et d'Eldar Zibentov, doctorants. Ils sont tous les trois membres de l'équipe CIAN du département SOC du laboratoire Lip6 de l'UPMC.

comportement du réseau de PLL et la discrétisation d'une surface définie par une équation d'ondes 2D amorties, à l'inverse de ce qui se pratique habituellement. Ceci a permis de comparer le comportement d'un grand réseau de PLL avec la surface d'un liquide dans un récipient ou un lac. En l'absence de perturbations, la surface d'un liquide est parfaitement plane, ce qui permet de dire que les PLL seront tous exactement en phase après une période de stabilisation. L'ajustement des coefficients du filtre des PLL vers l'amortissement optimal minimise le temps de retour à la stabilisation après une perturbation, ainsi qu'au démarrage.

Une perturbation dans une pièce d'eau produit des vagues concentriques qui se propagent avec atténuation jusqu'aux bords où elles se réfléchissent. Le même phénomène se produit dans le réseau de PLL. Un système de « brise-vague » permet de supprimer cette réflexion en simulant une surface infinie. Il revient à dissocier les PLL constituant le bord du réseau pour en faire un anneau qui remplit deux fonctions : l'amélioration de l'excitation du réseau par la fréquence de référence distribuée sur sa périphérie et la suppression de la réflexion des perturbations sur les bords du réseau.

Ces techniques peuvent être facilement adaptées à la synchronisation d'horloges de systèmes macroscopiques tels que les stations GSM, les serveurs informatiques, et autres horloges réparties, sous réserve que ces horloges soient suffisamment stables, c'est-à-dire qu'elles dérivent peu et ne doivent pas souvent changer de réglage.

Cours 6 : Coopération entre modèles de temps et de communication

Les cours précédents avaient présenté plusieurs types de modèles de temps et de communication pour les circuits et logiciels associés : modèles asynchrones sans contrôle de temps, modèles vibratoires où calculs et communications prennent un temps prévisible, modèles synchrones où la communication est supposée conceptuellement instantanée, modèles de temps continu ou de temps discret, etc. Ces modèles ont des caractéristiques et des champs d'application bien différents : les grands réseaux et algorithmes distribués généraux pour l'asynchrone, les circuits, systèmes embarqués et systèmes cyber-physiques pour le vibratoire et le synchrone, la simulation de systèmes physiques pour les modèles de temps continu, etc. Nous avons aussi déjà vu que les applications modernes demandent souvent une conjonction de modèles : circuits globalement asynchrones mais localement synchrones (GALS), orchestration d'activités Web asynchrones par des programmes synchrones (Hop / HipHop), etc.

Ce cours a étudié plus en profondeur la coopération entre modèles de temps et de communication. J'y ai d'abord montré que cette coopération existe déjà dans notre activité de tous les jours, car nous utilisons bon nombre d'outils permettant de passer d'un mode à l'autre : répondeur téléphonique simulant le courrier asynchrone à l'aide du téléphone synchrone, messages répétés en boucle sur la radio ou la télévision simulant les tableaux d'affichage, visioconférence par Internet, etc. J'ai ensuite expliqué que beaucoup de travaux ponctuels ont été faits et beaucoup d'outils développés pour utiliser en coordination plusieurs modèles de temps en informatique, mais de façon disparate et sans beaucoup de relations entre eux.

J'ai ensuite présenté avec plus de détails le système Ptolemy II⁵ développé à l'université de Californie à Berkeley par l'équipe d'Edward Lee. C'est actuellement

5. <http://ptolemy.eecs.berkeley.edu/ptolemyII/>

le seul système ayant pour ambition de développer divers modèles de calcul et leurs compositions hiérarchiques de façon uniforme au sein d'un atelier logiciel unique fondé sur des sémantiques formelles claires des modèles et de leur composition, tout en étant directement applicable dans un grand nombre de domaines pratiques. Par des exemples simples, j'ai illustré les principaux modèles de temps de Ptolemy et la façon de les composer dans les applications : réseaux de flots de données équilibrés ou non, automates finis, réseaux synchrones, simulateurs à événements discrets, systèmes distribués synchrones à temps coordonné, compositions modales ou continues / discrètes de sous-systèmes, etc. Le logiciel Ptolemy II est disponible librement sur le Web, accompagné d'un superbe livre pédagogique dont tous les exemples peuvent être exécutés en ligne ; ces exemples m'ont directement servi de base de cette partie du cours.

Séminaire 6 : La modélisation et la simulation : de la physique aux langages et aux logiciels

Karl-Johan Åström (Université de Lunds, Suède), le 19 mars 2014

Ce séminaire (donné en anglais) a présenté une revue des techniques de modélisation et de simulation selon les angles informatique et automatique. Le sujet a vu le jour suite au développement par Vannevar Bush d'un analyseur différentiel mécanique à la fin des années 1920. À la suite de ce travail préliminaire puis des progrès des composants électroniques, les calculateurs analogiques se sont rapidement développés dans les années 1950 et ont été largement utilisés dans la recherche et l'industrie. Les calculateurs sont ensuite devenus numériques, ce qui a permis d'améliorer l'algorithmique et surtout de supprimer la plupart des bruits induits par les composants des calculateurs analogiques.

Pour passer à l'échelle des vrais systèmes, il a fallu mieux comprendre comment décrire et programmer les simulations. Cela a demandé le développement de langages de modélisation adaptés, qui ont significativement réduit l'effort nécessaire pour construire les simulations. Ces langages, initialement dédiés à des sous-domaines spécifiques, ont été progressivement étendus au champ plus général de la simulation physique. L'idée centrale était de décrire assez directement les phénomènes physiques, avec une forme d'orientation objet pour la composition de leurs composants et des calculs symboliques et numériques pour la simulation et l'optimisation. Un groupe de chercheurs européens a apporté une contribution significative en développant le langage Modelica à la fin des années 1990.

La modélisation et la simulation sont maintenant devenues indispensables pour le développement de systèmes complexes en ingénierie. Elles sont d'usage constant dans des domaines comme l'automobile, l'avionique et l'énergie. Cependant, même si des progrès importants ont été réalisés, il reste beaucoup à faire pour disposer de chaînes de développement complètes, allant de la spécification formelle jusqu'à la génération de code efficace, certifiable, et directement embarquable dans les systèmes visés. Les questions de sémantique des langages, de coopération entre modèles discrets et modèles continus, de correction et d'efficacité des compilateurs et codes générés, d'interface homme-machine et de connexion avec les systèmes de conception 3D posent toutes des difficultés sérieuses sur lesquelles la recherche commence à se pencher.

Cours 7 : L'électricité est constructive : l'équivalence entre la propagation électrique et le calcul booléen constructif pour les circuits synchrones cycliques

Ce cours a présenté la preuve d'un résultat que j'ai mentionné à plusieurs reprises dans des cours précédents (les 13 janvier 2010, 2 avril 2013 et 29 janvier 2014 en particulier) : la caractérisation logique des circuits combinatoires cycliques ayant le même comportement électrique que les circuits acycliques, c'est-à-dire la stabilisation des sorties en temps borné et prévisible. Il a aussi eu pour objet de montrer que les concepts développés dans des domaines fort différents comme la logique mathématique et la preuve informatique de théorèmes sont pertinents même lorsqu'on parle de circuits digitaux.

La preuve en question demande le développement de nouvelles logiques intuitionnistes de la stabilisation électrique. Plusieurs théorèmes intermédiaires montrent l'équivalence du modèle électrique obtenu et d'un système de règles de déduction purement syntaxique qui permet de simuler le fonctionnement des circuits en se limitant à la stabilisation des voltages et en ignorant leurs transitoires. La preuve est essentiellement due à Michel Mandler (université de Bamberg, Allemagne). Elle est plus forte et plus rigoureuse que les preuves précédentes (Shiple-Berry-Touati, 2001), fondées sur des graphes d'événements analysant les transitoires dans un modèle de délai plus contraint et moins naturel (modèle de Brzozowski-Seger).

J'ai d'abord rappelé la source du problème : les circuits cycliques apparaissent naturellement lors de la traduction en circuits de langages de haut niveau comme Esterel. Ils sont tout à fait naturels pour implémenter des algorithmes fondamentalement symétriques, et ils peuvent être plus petits que les circuits acycliques pour la même fonction. Mais l'analyse de leur comportement est non triviale, car les circuits cycliques ne se stabilisent pas électriquement en général. J'ai présenté l'exemple de S. Malik, qui montre que les circuits cycliques peuvent être exponentiellement plus petits que les circuits acycliques pour la même fonction, celui de l'arbitre cyclique pour bus à jetons, et enfin rappelé le nouvel exemple du découpage d'instructions dans les Pentium déjà présenté lors du cours du 29 janvier (cours 3).

J'ai ensuite introduit la nouvelle logique UN (*up-bounded non-inertial delays*). Son modèle représente les histoires électriques des fils, portes et délais en se concentrant uniquement sur la stabilisation de leurs voltages. Il permet de coder de façon simple la logique et les délais des circuits. De façon duale, les règles de déduction de la logique permettent de simuler abstraitement l'exécution électrique effective du circuit. J'ai présenté les théorèmes exprimant l'équivalence entre modèle et système déductif, puis le résultat qui montre l'intuitionnisme de la logique. C'est justement ce caractère intuitionniste qui permet de montrer le déterminisme de la stabilisation des voltages, qui ne peut se faire que sur une seule valeur par sortie.

Une fois montré ces résultats centraux, un simple tour de passe-passe permet de faire disparaître totalement les délais, en montrant que la stabilité des sorties peut se calculer dans le simple calcul booléen constructif (sans tiers exclu), donc hors temps. Le corollaire simple est qu'un circuit qui stabilise ses sorties en logique booléenne constructive le fait électriquement pour tous délais des fils et portes et aux mêmes valeurs, ce qui est le résultat final souhaité.

Cette théorie fournit un algorithme pour déterminer si un circuit cyclique se stabilise ou oscille. Cet algorithme, fondé sur la logique ternaire de Scott, a été présenté au cours suivant du 2 avril (cours 8). Il s'étend aux circuits avec registres, les plus généraux. Il n'est cependant pas encore assez efficace pour passer à l'échelle des grands circuits.

Séminaire 7 : Langages pour la programmation des systèmes hybrides temps discret / temps continu

Marc Pouzet (ENS Paris), le 26 mars 2014

Les langages synchrones rencontrent un succès important dans la programmation de systèmes embarqués tels que les avions, les trains ou les centrales électriques (voir les cours de 2012-2013). L'écoulement du temps y est vu de manière logique comme la suite de réactions instantanées du système à des entrées provenant d'un environnement extérieur. Cette abstraction discrète permet d'écrire des programmes dans un langage mathématique, de garantir statiquement leur déterminisme et de les compiler vers du code s'exécutant en mémoire et temps bornés. La sémantique et la compilation des langages synchrones sont aujourd'hui suffisamment précises et bien comprises pour satisfaire aux exigences imposées par les autorités de certification de l'avionique civile pour le développement des parties les plus critiques (voir le séminaire du 23 avril 2013 par Bruno Pagano sur le logiciel SCADE 6).

Cependant, le développement d'un système embarqué réel demande aussi de savoir modéliser l'interaction entre ses comportements discrets déterminés par circuits et logiciels avec des modèles de ses composants physiques, qu'ils soient mécaniques, électriques ou hydrauliques. Ceci a conduit au développement de langages issus de la simulation numérique comme Simulink, LabView ou Modelica, normes industrielles de fait (voir le séminaire de K-J. Åström du 19 mars). Parce qu'ils permettent de combiner temps discret et temps continu, ces langages de modélisation de systèmes hybrides sont apparemment bien plus expressifs que les langages synchrones. Mais, en y regardant de plus près, on se rend compte que le temps discret n'y est pas un temps logique, indépendant de l'implémentation, mais qu'il peut au contraire dépendre des instants où le solveur numérique décide de rendre le contrôle. Cette absence de frontière étanche induit des comportements difficilement prévisibles et rend délicate la définition d'une sémantique de référence, pourtant indispensable pour garantir que les simulations et les implémentations sur différentes plates-formes sont équivalentes.

L'exposé a montré comment bâtir un langage de modélisation de systèmes hybrides au-dessus d'un noyau synchrone en lui ajoutant des équations différentielles ordinaires (ODEs). Cette extension est conservatrice vis-à-vis d'un langage synchrone existant (SCADE par exemple) : tout programme synchrone y conserve sa sémantique, sa compilation restant inchangée. Mais la partie discrète synchrone permet aussi de modéliser les transitions entre des modes continus. L'analyse non standard de Robinson permet de définir une sémantique synchrone en présence d'un solveur idéal, en se ramenant à un système à temps discret. Cette abstraction nous aide à répondre à des questions clés : quelle discipline imposer dans la combinaison des composants à temps discret et des composants à temps continu ? Que signifie qu'un programme combinant des signaux mixtes discrets / continus est causal ? Sous quelles conditions peut-on produire du code ordonné statiquement pour obtenir

du code embarqué et des simulations efficaces ? Comment s'assurer que les signaux donnés au solveur numériques sont continus durant l'intégration ? La réponse à ces questions est incarnée dans le langage prototype Zélus, qui utilise plusieurs systèmes de types dédiés, généralise la compilation traçable des langages synchrones et interagit avec un solveur numérique efficace.

Cours 8 : Jouer avec le temps

Ce cours a d'abord achevé le traitement des circuits cycliques en abordant l'approche sémantique et algorithmique par la logique ternaire de Kleene et Scott, qui donne une vision sémantique naturelle de la logique booléenne constructive. J'ai d'abord montré qu'un codage simple des valeurs ternaires par des paires de booléens conduit à un algorithme symbolique de détection de la constructivité d'un circuit cyclique utilisant les BDDs (*binary decision diagrams*), tout en permettant aussi la construction d'un circuit acyclique équivalent. J'ai enfin montré comment appliquer un calcul symbolique du même type à la correction constructive d'un circuit séquentiel (avec mémoires) par une exploration temporelle de ses états, ce qui est indispensable en pratique ; je l'ai illustré par le traitement des exemples montrés dans les cours précédents. Cela complète l'étude de la constructivité logique et électrique des circuits, en introduisant un jeu du temps entre états successifs en plus du jeu du temps à l'intérieur de chaque état.

Dans une deuxième partie, j'ai présenté un élégant et efficace synchroniseur mésochrone traitant des horloges de mêmes fréquences et à décalage de phase constant, dû à J.-M. Chabloz et A. Hemani (KTH Stockholm). J'ai expliqué brièvement comment le rendre plésiochrone (décalage de phase non constant). Cela a complété l'étude du calcul multi-horloges.

La fin du cours a été consacrée au modèle Ptides d'Edward Lee *et al.*, intégré dans le système Ptolemy présenté lors du cours du 19 mars (cours 6). Ce modèle permet de réaliser simplement des implémentations temps-réel distribuées de programmes complexes en composant des circuits, programmes et réseaux à temps garantis à l'aide d'une méthode élégante de *timestamping*. Son potentiel applicatif est remarquable.

Séminaire 8 : Programmation de systèmes parallèles temps-réel : défis et réussites des approches synchrones et par réseaux de Kahn

Albert Cohen (Inria) et Marc Duranton (CEA), le 2 avril 2014

Pendant trois décennies, l'observation auto-réalisatrice énoncée par Gordon Moore (le fondateur d'Intel) sur l'augmentation des performances des circuits électroniques a offert des années dorées aux concepteurs de processeurs. Ils ont pu construire des mécanismes de plus en plus sophistiqués pour exploiter un nombre toujours plus grand de transistors calculant de plus en plus vite. Ce mouvement a été accompagné par la conception de langages de programmation portables et sûrs, dotés d'environnements de programmation riches et de compilateurs qui traduisent leurs programmes en codes machine exécutables en appliquant des optimisations fines, le plus souvent à l'insu des programmeurs.

Mais, depuis une dizaine d'années, un coup d'arrêt a été porté à la course à la vitesse des processeurs, par l'atteinte de limites de consommation énergétique et de dissipation

thermique conduisant à des rendements de performance décroissants. Les constructeurs n'ont eu d'autre choix que de s'engouffrer dans une nouvelle course au parallélisme, en intégrant de plus en plus de cœurs de calcul. Mais ce choix rend la conception des systèmes, leur programmation et leur vérification beaucoup plus difficile. Il conduira à une augmentation de productivité bien moindre que précédemment. La domination à venir des multicœurs fondés sur des mécanismes de cohérence de cache et d'optimisation statistique menace également la sûreté des systèmes critiques, car elle rend quasi impossible l'analyse du temps d'exécution maximal d'un code, indispensable à toute preuve de sûreté des applications temps-réel.

Le parallélisme à flot de données, celui des réseaux de Kahn et des langages synchrones, est utilisé depuis longtemps pour la conception de systèmes de contrôle embarqués et de circuits électroniques. Il offre des garanties de sûreté et de portabilité inégalées. Après une présentation des nouveaux défis posés par le parallélisme à grande échelle, les orateurs ont présenté quelques succès de cette approche pour la programmation et l'implémentation de systèmes parallèles sûrs et efficaces, plus particulièrement dans le domaine du traitement vidéo. Ils ont ensuite étudié la parallélisation de langages synchrones et leur adaptation au contexte d'une exécution combinant des aspects temps réel et des aspects calculatoires.

Cours 9 : Discussion et réponse aux questions

Comme dans les années précédentes, j'ai consacré le dernier cours à la réponse aux questions reçues pendant l'année, toutes fort pertinentes. Voici les principales, dans l'ordre de leur traitement dans le cours.

1) Peut-on construire des spécifications vraiment non ambiguës, en particulier dans le monde industriel pour les applications entremêlant homme et machine ? (exemple de l'accident du vol Paris-Rio).

2) J'ai le souvenir d'une logique ternaire pour les relais : oui, non et indéfini = en cours de basculement. Cette logique « floue » a-t-elle été développée ou prise en compte dans vos travaux ?

3) Peut-on construire des logiciels tolérant les fautes comme on le fait pour les matériels ?

4) La logique intuitionniste UN est-elle implémentée en Coq et disponible sur le Web ? Je suppose qu'il y a une limite de taille à ce qui peut être traité avec un outil aussi général mais que les exemples du cours ne sont pas trop gros.

5) Y a-t-il une chaîne logicielle « schéma → solveur SMT » pour les exemples de circuit ? Quelle part est disponible en *open source* ?

6) Je me pose des questions sur le fondement de votre démarche qui postule que le temps d'exécution est nul ou tout au moins négligeable. Y a-t-il une réflexion conduite pour anticiper les conséquences de cas qui pourraient apparaître dans des conditions que je ne sais exclure de façon définitive ? N'y a-t-il aucun cas où cette condition essentielle ne serait pas remplie (boucle logique non détectée ou provoquée par une défaillance, éventuellement fugitive, d'une partie du matériel) ?

7) Observant des fonctionnements étonnants de mes liaisons WiFi sous Windows XP, je me demande si vos méthodes et outils sont utilisés dans le développement des *drivers* (pilotes de périphériques).

8) Il existe plusieurs approches pour représenter un système « synchrone » qui deviendra du matériel intégré avec du logiciel exécuté sur une partie de ce matériel.

Quels sont les algorithmes / méthodes pour passer d'un modèle type Esterel ou Ptolemy à une description matérielle (synthétisable ou synthétisée) et logicielle ?

9) Pourquoi, alors qu'il existe Ptolemy et Esterel, les architectes du matériel utilisent-ils SystemC ?

10) La matière molle a pu gagner ses lettres de noblesse à la fin du XX^e siècle ; envisagez-vous qu'un destin similaire soit possible pour la synchronisation molle dans les circuits ? D'ailleurs, dans les exemples du cours, même les réseaux GALS finissent par être complètement synchrones, et les synchroniseurs que vous avez présentés semblent toujours actifs.

11) À l'inverse, les synchronies enregistrées dans le cerveau (hors sommeil, épilepsie, etc.) sont extrêmement locales et transitoires. Est-ce qu'une telle synchronie « quasi-fluide », ou polychronie, est utilisée dans des circuits électroniques ?

12) Les réseaux neuronaux sont caractérisés par une redondance qui compense les aléas de la biologie. Est-il possible d'utiliser des populations de processeurs asynchrones pour pallier la métastabilité à faible coût énergétique ?

13) Quelle est la relation entre les modèles de comportement de circuit et les articles de Lamport sur la synchronisation par mémoire partagée ?

14) Est-ce qu'on peut par exemple prouver la faisabilité des hypothèses standard en programmation parallèle avec mémoire partagée (cf. par exemple l'article de Lamport, « A Fast Mutual Exclusion Algorithm ») ? Je me pose la même question à propos des multicœurs ou GPU.

15) Quelle logique d'ensemble se dégage des travaux récents reposant sur des modèles formels du parallélisme distribué ? Comment s'insère le développement en cours de TLA+ (Lamport) ? Les résultats classiques (cf. livres de N. Lynch, Raynal) restent-ils valables ou doivent-ils être revus en regard des résultats de la modélisation en logique formelle ?

16) Ne faudrait-il pas améliorer la sensibilité aux questions de sûreté logicielle dans l'industrie, cf. cours du 22 janvier (cours 2) ?

17) *Quid* du nombre astronomique de langages différents ? Une synthèse est-elle possible alors que l'on se préoccupe de l'accès des enfants au « codage » ? Pourrait-on à la fois faire acquérir les bases de la programmation (en n'oubliant pas d'initier aux « événements ») et agir en assemblant des modules préexistants ?

Les réponses écrites à ces questions sont présentées dans les transparents téléchargeables depuis le site Internet du cours^b. Il se trouve que beaucoup d'entre elles sont directement reliées au cycle de cours prévu pour les années suivantes, « Prouver les programmes : pourquoi, quand, comment ».

Séminaire 9.1 : Cerveau, cognition et temps

Virginie van Vassenhove (INSERM Neurospin, CEA Saclay), le 9 avril 2014

Ce séminaire a abordé la question du temps du point de vue des neurosciences cognitives et questionné la manière dont le cerveau crée notre temps psychologique : notre perception consciente du temps repose en effet sur la capacité de notre cerveau à internaliser les variables dynamiques du monde environnant et à les convertir en des représentations mentales accessibles à notre conscience.

b. Voir <http://www.college-de-france.fr/site/gerard-berry/course-2014-04-09-17h30.htm> [NdÉ].

Le cerveau est un organe dont les différentes sous-parties, dites « modules fonctionnels », analysent l'information en parallèle, donc simultanément. Cependant, le traitement de l'information n'est pas nécessairement synchrone et n'opère pas toujours sur les mêmes échelles de temps dans les différentes aires cérébrales. Ainsi, la « chrono-architecture cérébrale » est caractérisée d'une part par des phénomènes d'échantillonnage et de discrétisation de l'information, et d'autre part par des phénomènes d'intégration correspondant à la synchronisation de populations de neurones. Face à la complexité des phénomènes temporels au niveau neuronal, comment pouvons-nous expliquer notre expérience psychologique du temps, qui paraît continue et linéaire ? La représentation mentale d'un temps continu, allant du passé au présent au futur serait-elle une illusion ?

La première partie du séminaire a décrit les échelles de temps pertinentes pour le fonctionnement neuronal, en voyant le temps comme une propriété intrinsèque du fonctionnement cérébral, ainsi que pour l'établissement d'une représentation mentale du temps, en considérant sa dimension psychologique. Les phénomènes de discrétisation de l'information au sein du cerveau et leurs conséquences sur notre perception ont été ensuite discutés. Le séminaire s'est conclu par une discussion des modèles neuroscientifiques du temps et de la manière dont les variables temporelles pourraient être encodées dans le système de représentations mentales à l'origine de la cognition temporelle.

Séminaire 9.2 : Le temps dans le calcul neuronal

Romain Brette (Institut de la vision), le 9 avril 2014

Cette séance de séminaire n'était pas prévue dans le programme initial, mais il m'a paru indispensable de l'ajouter tant les questions posées par la communication interneurones se rapprochent de celles étudiées pour les circuits et programmes. Je remercie Romain Brette et Cyrille Rossant d'avoir accepté de l'organiser au pied levé.

Les neurones communiquent par l'intermédiaire d'impulsions électriques stéréotypées appelées « potentiels d'action ». Traditionnellement, le fonctionnement des neurones a été décrit en termes de calcul algébrique sur les fréquences de ces impulsions, en travaillant essentiellement avec des fréquences moyennes. Cette abstraction permet de se ramener à un cadre mathématique familier. Cependant, les observations expérimentales montrent que les neurones sont très sensibles à la coordination temporelle des impulsions. L'exposé a présenté les limitations théoriques de l'approche traditionnelle, les données empiriques en faveur de l'approche impulsionnelle et les théories de la computation neuronale reposant sur la coordination temporelle des impulsions.

INVITATION DE PROFESSEUR ÉTRANGER

J'ai invité le professeur Edward Lee de l'université de Californie à Berkeley⁶ à donner une conférence dans le cadre de mon cours. Sa visite a eu lieu du 6 au 12 décembre 2013, et sa conférence « Cyber-physical systems: A fundamental

6. Page web personnelle : <http://ptolemy.eecs.berkeley.edu/~eal/>.

intellectual challenge » a été donnée le 11 décembre devant un public nombreux et conquis^c. Edward Lee est l'un des principaux scientifiques opérant dans le domaine des systèmes cyber-physiques, dont l'objet est de contrôler le fonctionnement de dispositifs physiques par des calculs informatiques. Comme la plupart des circuits électroniques sont maintenant intégrés dans ces systèmes, dotés de gros logiciels et mis en réseaux, ce sujet prend une importance toujours croissante. Edward Lee dirige plusieurs importants groupes de recherche à Berkeley et aux États-Unis, dont le partenariat TerraSwarm qui regroupe neuf universités et auquel collaborent également de nombreux industriels. L'objectif de TerraSwarm est de comprendre comment organiser un monde fait de réseaux, à très grande échelle, de capteurs, actionneurs et calculateurs.

Dans son exposé, Edward Lee a montré que les systèmes cyber-physiques posent des problèmes fondamentaux et techniques considérables. En effet, ils exigent de combiner précisément la plupart des méthodes d'abstraction et des techniques d'implémentation et de vérification développées jusque-là séparément en automatique et en informatique. Si chacune d'entre elles a eu beaucoup de succès dans son champ propre, les combiner et les intégrer est loin d'être simple car elles reposent sur des fondements qui peuvent être bien différents, en particulier en ce qui concerne les aspects sémantiques. On peut dire sans ambages que la plupart des méthodes *ad hoc* utilisées actuellement dans l'industrie ont des trous sévères liés à des questions sémantiques posées par la collaboration entre le calcul et les systèmes physiques, comme on l'a vu par exemple pour la simulation lors du séminaire de Marc Pouzet le 26 mars. Les bugs produits par ces trous scientifiques et techniques peuvent être redoutables, comme on le constate à travers le nombre toujours croissant de problèmes liés aux logiciels dans les automobiles.

Un défi fondamental du domaine est la maîtrise des dynamiques temporelles. Le temps joue évidemment un rôle essentiel dans les phénomènes physiques. Pourtant, il ne fait pas partie des considérations fondamentales de l'informatique traditionnelle, comme je l'ai longuement expliqué dans les cours de 2012-2014. La question du déterminisme des comportements, largement débattue dans le cours, est tout aussi centrale pour les systèmes cyber-physiques. Or, la composition brutale d'abstractions et de techniques de développement issues de domaines différents conduit à des non-déterminismes absolument incontrôlables.

Edward Lee a présenté le problème de façon générale, puis les solutions originales et très complètes qu'il propose dans son système Ptolemy II. Ce système unifie dans une présentation graphique très élégante une grande variété de modèles de calcul parallèle atemporels ou temporels : flots de données synchrones et asynchrones, automates, langages synchrones, langages temps réel, etc. Ptolemy II intéresse fortement les industriels et est de plus en plus utilisé dans des applications réelles. Un point central est que les sémantiques de tous ses sous-modèles sont données dans un cadre mathématique unique, ce qui permet de comprendre finement et rigoureusement leurs correspondances et interactions. Composer des sous-modèles issus de paradigmes différents devient une opération bien mieux définie et bien plus sûre, comme je l'ai illustré ensuite dans le cours 8 (« Jouer avec le temps ») pour le beau modèle temps-réel distribué Prides de Ptolemy II. Mais la recherche est loin

c. La conférence est disponible en audio et en vidéo sur le site internet du Collège de France : <http://www.college-de-france.fr/site/gerard-berry/guestlecturer-2013-2014.htm> [NdÉ].

d'être terminée, en particulier en ce qui concerne la génération de code embarquable et la vérification formelle, qui ne sont pas encore disponibles.

Le système Ptolemy II bénéficie d'une documentation particulièrement bien construite, en particulier sous la forme d'un livre disponible aussi bien en papier qu'en PDF sur le Web⁷. La version PDF disponible en libre accès et présente une originalité remarquable : toutes les figures présentant des modèles Ptolemy II sont directement exécutables sur n'importe quel ordinateur.

PUBLICATION

BERRY G. et SERRANO M., « Hop and HipHop: Multitier Web Orchestration », dans NATARAJAN R. (éd.), *Distributed Computing and Internet Technology*, Springer International Publishing, coll. « Lecture Notes in Computer Science », n° 8337, 2014, 1-13, http://link.springer.com/chapter/10.1007/978-3-319-04483-5_1.

CONFÉRENCES DONNÉES À L'EXTÉRIEUR

Cours et conférences du Collège de France à l'étranger

28.08.2013 : dans le cadre de la collaboration Collège de France / Universidade Federal do Rio de Janeiro, « The informatics of time and events ».

04.09.2014 : dans le même cadre, « Circuits and 2-adic numbers ».

19.05.2014 : Maison française d'Oxford, Angleterre, « Informal and formal languages to deal with time and events ».

Conférences académiques invitées

04.10.2013 : Embedded Systems Week, Workshop on Systems Synthesis, Montréal, « The informatics of time and events ».

14.11.2013 : Colloque « Réseaux, savoirs et territoires » à l'École normale supérieure de Paris, « L'informaticien et l'écriture ».

21.11.2013 : Synchron Workshop, Dagstuhl, Allemagne, « Reflections on causality and compositionality ».

22.11.2013 : École Centrale, Chatenay-Malabry, « Pourquoi le monde devient numérique : les racines scientifiques et techniques ».

26.11.2013 : Maison des sciences de Rennes, « Parler du temps, du joli vague de la langue à la précision de l'informatique ».

29.11.2013 : Eden Théâtre des frères Lumière, La Ciotat, dans le cadre des journées de l'Académie des technologies en Provence, « Art et informatique ».

13.12.2013 : École normale supérieure de Lyon, « La photographie numérique : mettre la lumière en boîte, puis l'en sortir ».

28.01.2014 : Centre international de Valbonne (06), « L'informatique : de la révolution technique à la révolution mentale ».

7. http://ptolemy.eecs.berkeley.edu/books/Systems/index.htm#in_browser

03.03.2014 : journée « Limites », Académie des beaux-arts et Académie des sciences, Paris, « L'informatique a-t-elle des limites ? ».

06.02.2014 : International Conference on Distributed Computing and Internet Technologies, Bhubaneswar, Inde, « Hop and HipHop: Multitier web orchestration ».

08.04.2014 : Colloque franco-coréen de bioinformatique, Académie des sciences, « Some computer science challenges raised by computational biology ».

23-24.04.2014 : Institut Henri Poincaré, dans le cadre du trimestre spécial Semantics of Proofs and Programs, « Circuits and 2-adic numbers » et « Cyclic circuit analysis: the intuitionistic character of electricity ».

13.05.2014 : Espaces des sciences de Rennes, dans le cadre des journées de l'Académie des sciences en Bretagne, « Le temps, de la physique à l'informatique et à la musique ».

Autres conférences

21.09.2013 : Congrès des experts de la MACIF, Lille, « L'informatique : de la révolution technique à la révolution mentale ».

26.10.2013 : Centre dramatique national Besançon Franche-Comté, « La conception du monde chez les Shadoks ».

13.12.2013 : Mairie de Crest, Drôme, participation au débat « Révolution numérique : changement technologique ou révolution culturelle ».

16.01.2014 : Société GRTgaz, Paris, « L'informatique : de la révolution technique à la révolution mentale ».

19.03.2014 : Agora des directeurs de systèmes d'informations, Paris « L'informatique : de la révolution technique à la révolution mentale ».

10.04.2014 : Cercle de réflexion « demi-siècle », Paris, « L'informatique : de la révolution technique à la révolution mentale ».

23.05.2014 : Thalès recherche et technologie, campus de l'École polytechnique, Palaiseau « L'informatique, révolution technique ou révolution mentale ? Le cas des systèmes embarqués ».

02.06.2014 : Espace des sciences de Rennes, « Science et conscience chez les Shadoks ».

06.06.2014 : Institut des hautes études pour la science et la technologie (IHEST), Conseil économique et social, Paris, « La pensée informatique, véritable cœur du monde numérique ».

19.06.2014 : Journée « PLM Labs : Back to Basics », Paris, « Spécification, programmation et vérification des logiciels embarqués de haute sécurité ».

ÉMISSIONS DE RADIO

22.02.2014 : Journée organisée à la Sorbonne par France-Culture, débat « Qu'est-ce qu'un algorithme ? ».

12.03.2014 : « Rue des écoles », France Culture, « Pourquoi et comment enseigner l'informatique à l'école ».

01.04.2014 : « Homo Urbanicus », Aligre FM, « Le rôle de professeur au Collège de France ».

02.04.2014 : « La tête au carré », France Inter, « La sécurité informatique ».

23.04.2014 : « Rue des écoles », France Culture, « La pédagogie Montessori aujourd'hui ».

15.05.2014 : « Un temps de Pochon », France Inter, « Qu'est-ce qu'une leçon inaugurale au Collège de France ? ».

ACTIVITÉS DE RECHERCHE

Mes activités de recherche ont concerné deux aspects directement liés au rôle du temps dans les systèmes. J'ai collaboré avec Louis Mandel, qui a rejoint ma chaire en septembre 2013 comme maître de conférences au Collège de France.

Le premier aspect est celui de l'utilisation de la programmation synchrone pour l'orchestration d'activités Web construites par composition de services Web existants. Il est traité en collaboration avec l'équipe-projet Indes de l'Inria-Sophia Antipolis, dirigée par Manuel Serrano. Le Web n'est plus seulement composé de pages lisibles sur un navigateur. On y trouve aussi de très nombreux services, qui sont des programmes situés sur des serveurs et appelables par le protocole http standard, qui rendent non plus une page affichable mais un résultat quelconque codé en chaînes de caractères. De très nombreux objets informatisés sont également connectés au Web, comme les smartphones, chaînes hi-fi et maintenant voitures. Manuel Serrano a défini le langage Hop⁸, qui permet de coordonner ces services bien plus facilement qu'avec la tour de Babel des langages classiques du Web. J'ai ensuite défini avec lui une première version du langage temporel synchrone HipHop⁹ inspiré d'Esterel, dont l'objectif est de traiter les événements et facteurs temporels liés soit au réseau, soit à la physique des objets connectés. HipHop a été implémenté par notre thésard commun Cyprien Nicolas et a fait l'objet d'une communication invitée en 2014 dans un congrès en Inde. Nous travaillons actuellement sur une version Javascript de Hop/HipHop, rendue nécessaire par l'immense popularité de ce langage, devenu le standard de fait du Web.

Le second aspect est la tenue d'un séminaire sur le temps, qui a rassemblé régulièrement une vingtaine de chercheurs de disciplines différentes : circuits, logiciels temps réels, informatique musicale, neurosciences, etc. Ce séminaire a pour vocation de discuter les travaux en cours et les sujets mal compris, fort nombreux dans le domaine. À travers une dizaine de séances, il a permis de commencer à rapprocher utilement les points de vue des différentes disciplines. Deux questions apparaissant comme centrales pour des études et modélisations ultérieures : le rôle des oscillateurs neuronaux pour le transfert de l'information dans le cerveau et leur rôle dans la perception de la musique.

Pour sa part, Louis Mandel a continué en collaboration avec moi sa recherche sur son langage synchrone ReactiveML, en publiant trois articles de conférence et trois articles de *workshops* sur ce sujet.

En 2014, mon groupe de recherche sera renforcé par l'arrivée de Lionel Rieg sur un poste d'ATER Collège de France. Nous allons travailler sur la preuve formelle des algorithmes de compilation d'Esterel et l'extraction automatique du code du compilateur à partir de cette preuve, en utilisant l'assistant de preuve Coq. Ce travail s'inscrira naturellement dans mon nouveau cycle pluriannuel de cours commençant en 2014-2015 et intitulé « Prouver les programmes : pourquoi, quand, comment ».

8. Voir M. Serrano et G. Berry, *Multi-tier programming in Hop*, *Communications of the ACM*, vol. 5, n° 8, août 2012.

9. G. Berry et M. Serrano, « *Hop and HipHop: Multitier Web Orchestration* », *Proc ICDCIT'2014*, Bubhaneswar, Springer-Verlag, coll. « Lecture Notes », n° 8337, 2014.

AUTRES ACTIVITÉS

Je suis président du conseil d'enseignement et de recherches de l'École polytechnique (depuis juin 2014), président du conseil scientifique de l'École d'informatique et électronique ESIEE Paris-Marne-la-Vallée, membre du conseil d'administration de l'Agence nationale de la recherche (ANR), et membre du conseil scientifique de l'Institut pour la recherche et la coordination acoustique et musicale (IRCAM).

J'ai présidé le jury international du prix « Informatics Europe Best Practice in Education Award » participé au jury du prix Milner de la Royal Society, de l'Académie des sciences et de la Leopoldina, ainsi qu'à plusieurs jurys de prix scientifiques de l'Académie des sciences et à celui des bourses françaises L'Oréal-UNESCO pour les femmes et la science.

J'ai participé à deux interviews de l'office parlementaire des choix scientifiques et techniques (OPECST) sur la sécurité informatique.

En juin 2014, j'ai réalisé quatre mini-MOOCs sur l'informatique sur le Web pour Wandida¹⁰, École polytechnique fédérale de Lausanne.

10. <http://wandida.com/fr>

