Académie des sciences

Séance solennelle de réception des Membres élus en 2002
17 juin 2003

L'informatique embarquée
Gérard Berry

Depuis les années 1960, l'informatique est graduellement entrée dans notre paysage, d'abord pour les usages professionnels comme le calcul scientifique et la gestion, puis pour les usages personnels dont bien sûr Internet. L'ordinateur avec son clavier et son écran est devenu aussi familier que la cuisinière ou le téléviseur, et se vend d'ailleurs aux mêmes endroits. Les gens ont même fini par s'habituer à ses dysfonctionnements récurrents, faisant preuve d'une étonnante tolérance pour les défauts des logiciels.

En ce moment, une nouvelle forme rampante et discrète d'informatique est en train d'envahir le monde d'une façon plus insidieuse : l'informatique embarquée, sans écran, ni clavier, enfouie dans des objets allant des plus sophistiqués aux plus banaux. Le grand public n'a pas nécessairement conscience qu'un avion se contrôle entièrement de façon numérique, ni que son téléphone, son lecteur de DVD et son appareil photo contiennent ce qu'on appelait il y a encore peu un puissant ordinateur. L'informatique embarquée est en train de révolutionner des domaines entiers comme la communication, qui évolue vers le sans-fil et l'ubiquité, l'audiovisuel, devenu entièrement numérique, les transports, qui perdent leurs conducteurs, les prothèses médicales, qui deviennent actives, etc. Il se fabrique des milliards de systèmes informatiques noyés dans toutes sortes d'objets, et nous n'en sommes qu'au début.

La première raison de cet essor est que le traitement numérique des données présente des avantages nombreux et décisifs : reproductibilité infinie, facilité de stockage et de transmission, possibilités de compression, de filtrage et de cryptage, etc. Il y a très peu de choses qu'il n'est pas intéressant de numériser. Techniquement, le levier principal est l'extraordinaire progrès de l'électronique : à cause de tout un faisceau de facteurs impossible à détailler ici, on continue depuis vingt ans à doubler la puissance des circuits en gros tous les deux ans. On sait actuellement fabriquer en série et pour un prix très faible des circuits comprenant des dizaines ou des centaines de millions de transistors et fonctionnant à des fréquences très élevées. On sait aussi doter ces circuits de mémoires de grande capacité permettant de stocker des logiciels de grande taille et beaucoup de données. Le degré d'intégration est devenu tel que l'ensemble des composants matériels et logiciels des applications complexes tient sur une seule puce : c'est ce qu'on appelle System on Chip (SoC).

L'informatique embarquée réalise souvent des fonctions critiques pour la fonctionnalité et la sécurité des objets qu'elle commande, et elle est en quelque sorte scellée dans ces objets. Cela la rend très différente de l'informatique classique : si l'on a pris l'habitude de télécharger la nouvelle version d'un logiciel pour corriger ses blocages et ses problèmes, il est exclu que l'on accepte de le faire pour un logiciel de

pilotage d'avion ou de freinage. Il n'y a pas le choix : la qualité de l'informatique embarquée doit être immédiatement parfaite.

Or, la perfection est très difficile à obtenir dans ce domaine. Contrairement aux hommes, les ordinateurs exécutent avec une terrifiante conscience professionnelle et dans tous les détails les ordres qu'on leur donne, y compris ceux qui sont faux. Contrairement à la mécanique, l'informatique est invisible et très discontinue : quelques bits erronés passent facilement inaperçus mais peuvent faire dysfonctionner gravement un circuit ou un programme. Pour prendre une image, faire marcher un grand circuit ou un grand logiciel est à peu près aussi simple que de faire marcher une entreprise d'un million d'employés où chacun ferait exactement ce qu'on lui dit, ni plus ni moins, ce qui n'est évidemment le rêve d'aucun directeur.

En pratique, la vérification de la correction des systèmes informatiques embarqués est beaucoup plus dure que leur conception, et cette difficulté ira en augmentant avec le nombre et la complexité des applications. Pour continuer à faire des systèmes qui marchent, il y a trois clefs complémentaires : premièrement, travailler avec des langages de description et des objets de base de niveau plus élevé, en systématisant une forme de préfabrication logique ; deuxièmement, trouver des méthodes scientifiques permettant de réellement garantir a priori le bon fonctionnement pour simplifier la validation et le test d'un ordre de grandeur ; enfin, mettre en place des processus industriels parfaitement rigoureux pour le développement et la gestion des applications. Je ne parlerai que des deux premiers aspects. Le dernier est bien couvert par des normes industrielles, par exemple pour la certification des logiciels embarqués en aéronautique civile.

J'ai travaillé dans les vingt dernières années sur une approche scientifique originale aux circuits et logiciels embarqués qu'on appelle la « programmation synchrone ». L'idée de base est née quasi-simultanément dans trois labos français mixtes informatique / automatique (Ecole des mines, Institut de recherche en informatique et en automatique, Institut d'informatique et mathématiques appliquées de Grenoble)). Il y a plusieurs façons de l'exprimer, mais j'en retiendrai la plus amusante : quelle serait la bonne façon de programmer un ordinateur « infiniment rapide » pour contrôler un système embarqué ? Cette vue très simple permet de séparer les variables en décrivant le côté fonctionnel à haut niveau indépendamment des contraintes électriques ou logiques liées à l'exécution effective. Elle facilite la programmation parallèle à grande échelle qui est ici indispensable, tout en garantissant le déterminisme des algorithmes de contrôle. Elle s'applique aussi bien aux circuits qu'au logiciel.

Plusieurs langages de programmation de haut niveau ont adopté l'approche synchrone, dont le langage Esterel développé dans mon équipe depuis 1983 et les langages Lustre (Grenoble) et Signal (Rennes). Le modèle mathématique sous-jacent est maintenant bien compris et est conceptuellement beaucoup plus simple que les modèles de la programmation classique. Il permet d'effectuer des calculs très fins et très puissants sur les programmes et leurs comportements, ce qui permet in fine de produire des circuits ou des codes embarqués efficaces et aux performances prédictibles, pour lesquels on peut contrôler la validité de l'approximation synchrone par rapport aux contraintes pratiques. Les techniques synchrones sont systématiquement utilisées dans la conception de grands logiciels de contrôle comme ceux des Airbus.

Le grand enjeu actuel est la vérification automatique de la correction des circuits et programmes. Par exemple, en calculant dans le modèle synchrone, on peut maintenant garantir à 100% qu'un ascenseur ne pourra jamais voyager la porte ouverte, ceci sans avoir à effectuer une campagne de tests jamais vraiment conclusive. La vérification de programmes est un sujet ancien qui n'a pas vraiment abouti dans la programmation classique, mais qui a fait récemment des progrès majeurs pour les circuits et logiciels synchrones, au point de passer maintenant au stade industriel. Elle s'appuie d'une part sur la simplicité du modèle et d'autre part sur de nouveaux algorithmes de vérification de très grandes formules Booléennes et numériques, sujet qui a fait des progrès considérables ces dernières années. On ne peut évidemment pas tout vérifier, mais on peut effectivement assurer des propriétés critiques et surtout trouver des bugs sournois et tout à fait hors de portée des raisonnements humains. L'utilisation systématique de la vérification formelle est une des clefs de la sécurité de l'informatique embarquée dont nous dépendrons toujours plus. Le succès final dépendra des progrès scientifiques que nous pourrons faire sur le coeur scientifique du sujet, qui se situe aux frontières de la logique et de l'algorithmique. Il y a quinze ans, on pensait ce sujet irréalisable. On possède maintenant des heuristiques qui fonctionnent bien, mais on ne sait pas encore vraiment pourquoi. Le comprendre pourrait modifier de façon radicale la façon de concevoir les systèmes embarqués de l'avenir.