

## Hâtons-nous lentement

Tout va très rapidement tant dans la vie quotidienne et spécialement en informatique. Au 21<sup>e</sup> siècle, il est maintenant normal d'espérer trouver toutes les informations après quelques clics, de joindre une personne au bout de son téléphone cellulaire (portable comme on l'appelle en Europe) ou de se sentir délaissé si notre destinataire n'a pas répondu à un courriel au bout de quelques heures.

On cherche des solutions rapides, toutes faites, et *évidemment* gratuites sur le Web. Pourtant sommes-nous bien plus avancés? Nous nous retrouvons alors de plus en plus rapidement aux prises avec de nouveaux problèmes qui entretiennent le cycle infernal de la vitesse. L'informatique n'est pas étrangère à l'accélération du rythme; tout d'abord pour elle-même par cette course aux mégahertz aussitôt engloutis par les nouveaux "features" du système d'exploitation. Le rythme de production du génie logiciel s'accélère à un tel point que les numéros des nouvelles versions des logiciels commerciaux se terminent par ".0.2" quand il ne sont pas carrément erronés à la livraison mais qui sont capables de détecter l'apparition de nouvelles corrections et de nouveaux bugs automatiquement via le Web. On peut maintenant remplir nos formulaires en ligne directement auprès d'une administration qui alors en profite pour les changer constamment et surtout faire en sorte qu'il est impossible de transférer dans la nouvelle version ce que nous avons péniblement rempli dans l'ancienne. L'informatique mobile nous promet la liberté mais n'est-ce pas plutôt une nouvelle forme d'esclavage subtil où personne ne peut se permettre ou prétexter ne pas être joignable en tout temps.

Et pourtant, nous avons tous besoin d'être isolés de temps à autre afin de prendre du recul (eh oui) pour *évidemment* mieux avancer. On peut alors se demander pourquoi des auteurs prennent le temps de soumettre des articles à des journaux-papiers comme TSI alors qu'ils auraient très bien pu les publier sur leur page web ou leur "weblog". Il faut croire que la lenteur a ses vertus que la raison ne peut connaître. Les articles de ce numéro sont un bon exemple des différentes tendances de l'informatique moderne et qui démontre l'intérêt de s'arrêter quelques fois pour mieux repartir. On y passe de l'informatique mobile, aux réseaux, aux logiques temporelles pour la spécification du logiciel jusqu'à une généralisation des systèmes de réécriture; on décrit aussi une expérience simple illustrant que la programmation est un exercice surprenant et plein d'embûches.

*Mobilité et bases de données*, la suite d'un article débuté au numéro précédent, rapporte le résultat d'un travail collectif de chercheurs du CNRS sur la problématique de l'informatique mobile; cet article se concentre sur la gestion de données embarquées, la confidentialité des données et l'adaptabilité en particulier les problèmes de déconnexion.

Hanifa Boucheneb et John Mullins, dans *Analyse des réseaux temporels*, s'attaquent au problème difficile de la vérification des systèmes temps réel à l'aide de réseaux de PETRI temporels. Ils présentent un algorithme en  $O(n^2)$  pour calculer les successeurs de classes dans un graphe fini résultat d'une contraction de l'espace infini d'états des réseaux originaux. Gagner un ordre de grandeur peut transformer un algorithme inapplicable en une méthodologie pratique.

L'article de Radu Mateescu, *Logiques temporelles basées sur actions pour la vérification des systèmes asynchrones*, porte un titre à première vue paradoxal que vient faire le temps dans les systèmes qui n'en tiennent pas compte. Or les systèmes devant toujours aller de plus en plus rapidement, ils ne peuvent se permettre d'attendre les autres, il faut donc les considérer comme asynchrones et non-déterministes. Il est primordial de pouvoir les vérifier de façon formelle afin d'éviter les approximations souvent illusoire. Cet article présente de façon unifiée différentes logiques temporelles basées sur des actions.

Marc Aiguier et Diane Bahrami dans *Une approche générique de la réécriture* réussissent le tour de force de montrer comment la réécriture peut devenir un système de preuve opérationnel indépendamment de la logique sous-jacente. Ce travail de fond montre à quel point le travail théorique permet d'approcher des problèmes classiques sous un nouvel angle et d'en révéler de nouveaux aspects.

P.-A. de Marneffe expose dans *L'erreur est humaine* un petit puzzle inoffensif, mais qui révèle comment la programmation peut être sournoise lorsqu'on se contente d'approches intuitives. Le problème qu'il présente est désespérément simple et pourtant sa solution correcte n'est pas facile à découvrir quoiqu'évidente lorsqu'on nous a mis le nez dessus... Ce problème alimentera sûrement plusieurs conversations autour de pauses-café, mais il mérite sûrement un meilleur sort car il révèle des habitudes ancrées chez les programmeurs.

Bonne lecture mais surtout prenez bien votre temps pour en profiter pleinement !

Guy Lapalme