
Editorial

Autour et alentour des langages applicatifs regroupe des articles présentant des travaux exposés lors des 16^e et 17^e éditions des Journées françaises des langages applicatifs (JFLA). Ces éditions ont eu lieu en 2005 et 2006 respectivement à Obernai et à Pauillac. Les JFLA ont pour ambition de couvrir le domaine des langages applicatifs, au sens large, en réunissant les concepteurs et les utilisateurs afin de débattre et discuter des avancées de la communauté francophone.

Initialement centrés sur les langages, les thèmes des conférences se sont peu à peu élargis pour refléter la diversité des travaux connexes aux langages applicatifs et inclure les apports d'outils d'autres domaines qui autorisent la construction de systèmes logiciels plus sûrs (assistants de preuves, spécification, prototypage, développement formel d'algorithmes, etc.). S'inscrit dans cette ligne la chronique de S. Blazy sur la réalisation d'un compilateur certifié pour un sous-ensemble du langage C dans le cadre du projet Compcert. Cette réalisation, essentielle pour les applications embarquées, est rendue possible par l'utilisation de l'atelier Coq pour produire automatiquement un compilateur à partir d'une spécification formelle, en assurant que chaque étape de la compilation respecte bien la spécification initiale.

Ce numéro réunit 5 versions étendues et complétées des travaux présentés (parmi 22) aux deux conférences. Ces articles ont tous été soumis à un processus de sélection et révision selon les règles de la revue TSI. Ils reflètent la diversité des thèmes abordés et se répartissent en deux grandes familles :

– des travaux qui concernent des développements autour des langages (articles de J. Cohen, de A. Spicher et O. Michel ainsi que G. Henry, M. Mauny et E. Chailloux),

– des travaux qui concernent des applications comme le travail sur l'implantation d'une politique de contrôle d'accès (article de J. Blond et C. Morisset) ou le travail sur le traitement formel des langues naturelles (article de H. Anoun).

L'article de J. Cohen, « Interprétation par syntaxe abstraite d'ordre supérieur et combinatorisation » propose une méthode originale, utilisant une syntaxe abstraite d'ordre supérieur pour le développement rapide d'interprètes efficaces et extensibles à moindre coût. Elle s'appuie sur le langage fonctionnel hôte en utilisant ses mécanismes de réduction pour évaluer les appels fonctionnels du langage interprété. Un des intérêts de cette méthode est de fournir un point de vue uniforme sur le langage interprété, ce qui le rend facilement extensible.

Le langage MGS propose un point de vue unifié sur les structures de données et leur manipulation en les considérant d'un point de vue topologique. Un programme est ainsi une suite d'opérations de transformations (définies par des règles de réécriture) des collections topologiques. Dans leur article « Représentation et

manipulation de structures topologiques dans un langage fonctionnel », A. Spicher et O. Michel décrivent l'implantation et la manipulation en MGS d'une classe particulière de collections topologiques, les quasi-variétés, et illustrent leurs manipulations à travers la description d'un algorithme classique de subdivision de surfaces en CAO.

L'article de G. Henry, M. Mauny et E. Chailloux, « *Typer la désérialisation sans sérialiser les types* » aborde la problématique de typage statique des fonctions de désérialisation. Cette technique consiste à transformer une valeur du langage en une suite d'octets afin de pouvoir la sauvegarder dans un fichier et la relire ultérieurement. L'article montre comment elle peut être mise en œuvre dans un langage où toute expression reçoit un type synthétisé par le compilateur.

L'article de J. Blond et C. Morisset, « *Un moniteur de référence sûr d'une base de données* » propose la formalisation, dans l'atelier Focal, et l'implantation en SQL d'une politique de contrôle d'accès. Ces travaux répondent au besoin croissant de sûreté dans l'accès aux systèmes de bases de données par l'utilisation de méthodes formelles pour accroître la confiance. L'atelier Focal permet la définition d'une spécification, raffinée ensuite par étapes successives, en un code fonctionnel qui peut être prouvé par le système Coq. On obtient ainsi une implantation sûre d'une spécification formelle, illustrée ici par un moniteur de référence.

L'article de H. Anoun, « *Une bibliothèque Coq pour le traitement des langues naturelles* » présente l'outil ICHARATE, un atelier logique, destiné à la recherche et à l'enseignement, basé sur le système Coq et dédié à la formalisation de grammaires multimodales. Cet atelier permet d'appréhender, de façon interactive, des formalismes logiques complexes tels que les grammaires catégorielles multimodales. On peut avec ICHARATE effectuer des dérivations syntaxiques pour des grammaires données, prouver de nouvelles règles et propriétés universelles portant sur des classes de grammaires afin de rendre compte de phénomènes linguistiques divers.

Nous espérons que le contenu de ce numéro donnera au lecteur curieux l'envie de se plonger plus avant dans les travaux récents de ce domaine de recherche très vivace. Nous remercions les auteurs pour leur contribution et les relecteurs pour la qualité et la pertinence de leurs remarques et conseils.

Thérèse Hardin
Université Pierre et Marie Curie, Paris VI

Olivier Michel
Université d'Évry Val d'Essonne