

Langages Synchrones

Applications, Fondements et Évolutions

Charles ANDRÉ

Projet AOSTE - Laboratoire I3S/INRIA
Université de Nice-SophiaAntipolis/CNRS
06903 Sophia Antipolis cédex
andre@unice.fr

Dans les années 1980, des études indépendantes ont conduit à l'apparition de *langages spécialisés* dits *langages synchrones* comme Esterel, Lustre et Signal, ainsi qu'à celle d'un formalisme graphique appelé Statecharts. Ces langages et formalismes répondaient à des besoins de programmation et de modélisation des *systèmes réactifs*, c'est à dire des systèmes qui sont en interaction permanente avec leur environnement. La nature des applications considérées explique que ces langages aient été initialement conçus par des équipes mixtes comprenant des automaticiens et des informaticiens.

Visant des applications industrielles, il est vite apparu que ces langages devaient être expressifs et compilables de façon efficace afin de pouvoir traiter de "gros" programmes. Les techniques de compilation des langages synchrones ont fait de grands progrès dans les années 1990, permettant ainsi à ces langages de *passer du statut académique au statut industriel*. Leur champ d'application s'est également élargi. Ils sont à présent largement utilisés dans le développement de logiciels embarqués ainsi que dans la conception de circuits numériques complexes.

Les langages synchrones sont spécialisés dans l'expression rigoureuse des réactions. Un programme synchrone doit apporter des réponses non ambiguës aux questions : quelles sont les actions qui composent une réaction, quand les exécuter, quelles sont les liens de dépendance entre ces actions, ... Une caractéristique essentielle des langages synchrones est de pouvoir spécifier des *comportements déterministes* même en présence de parallélisme et de communications. La chaîne de transformation depuis la spécification jusqu'à l'implémentation préserve ce déterminisme.

En fonction de la classe d'application visée, l'utilisateur peut choisir son style de programmation : un *style impératif* (pour les applications dans lesquelles les aspects relatifs au contrôle sont prépondérants) ou un *style déclaratif* (pour des applications avec une part importante de traitements de données en temps réel). Cette distinction a tendance à s'estomper avec les nouvelles plates-formes de développement synchrones qui permettent de mélanger les deux styles de programmation. La plupart des langages offrent une syntaxe *textuelle* et une syntaxe *graphique*. Dans tous les cas, les langages restent simples avec un nombre limité d'entités et de constructeurs. Malgré leur simplicité, ces langages permettent l'utilisation de concepts de haut-niveau comme le parallélisme de conception, les communications par diffusion, et les descriptions hiérarchiques.

La *sémantique intuitive* des langages est simple et naturelle (tout au moins pour les concepteurs de systèmes réactifs). Les évolutions du système sont perçues comme une séquence de réactions disjointes. Une réaction comprend une phase de lecture des informations d'entrée, une phase de calcul et une phase d'écriture des informations de sortie.

Durant la réaction, le programme est complètement isolé de son environnement. La connaissance de la sémantique intuitive est suffisante pour l'utilisateur. Cette sémantique est compatible avec une *sémantique formelle* (définie en termes mathématiques) qui est la sémantique de référence pour le langage. L'existence d'une sémantique mathématique fait que les compilateurs et les générateurs de codes sont plus faciles à développer et à vérifier. La *vérification formelle* de certaines propriétés des programmes, comme les propriétés de sûreté, est également simplifiée.

Dans cette présentation, nous aborderons les points suivants :

- Modèles synchrones, abstractions, sémantiques
- Spécificités du synchrone : Calculs d'horloges, Causalités
- Évolutions (Plates-formes industrielles, Recherche)

Langages et modèles traités :

- Estérel, Lustre, Signal, Lucid synchrone, SyncCharts
- Environnements : Esterel Studio, SCADE, Polychrony

Documentation :

- Benveniste A., Caspi P., Edwards S. A., Halbwachs N., Le Guernic P., de Simone R., « The Synchronous Languages 12 Years Later », Proceedings of the IEEE, vol. 91, n° 1, p. 64-83, 2003.
- André C., « Comparaison des styles de programmation de langages synchrones », rapport de recherche, n° ISRN I3S/RR-2005-13-FR, I3S, Sophia-Antipolis, (F), Juin, 2005. (<http://www.i3s.unice.fr/~mh/RR/2005/RR-05.13-C.ANDRE.pdf>)