

Paradigmes objets et synchrones

dans les systèmes temps-réel

Charles André

Laboratoire I3S

Université de Nice-Sophia Antipolis / CNRS

BP 121 - 06903 Sophia Antipolis cédex

email: andre@unice.fr

L'équipe SPORTS (Synchronous Programming Of Real-Time Systems) s'intéresse à la conception de systèmes réactifs sûrs. Nous créons et développons des *modèles*, *outils*, *méthodes* permettant la conception d'applications à contrôle dominant. Nous avons adopté le *paradigme synchrone*, ce qui nous permet d'avoir des modèles formels bien adaptés à une large classe d'applications.

UML est désormais incontournable dans la conception de systèmes complexes. Avec ROOM, l'approche objet entrain dans les applications temps-réel. B.P Douglass, dans ces livres, propose une approche s'appuyant sur UML, en introduisant des profils adaptés aux applications temps-réel. Rational Rose RealTime est commercialisé. Des comités réfléchissent à Java temps-réel... Les approches objets gagnent indéniablement du terrain dans le monde des applications embarquées et temps réel.

Dans cet exposé, nous présentons notre *contribution* dans ce domaine. Nous nous focalisons sur les applications de type contrôle pour lesquelles il est vital d'assurer le bon comportement. Nous montrons ce que notre expérience en modélisation et programmation synchrone peut apporter à l'UML.

En UML, le *comportement dynamique* des objets et des classes est généralement exprimé par les diagrammes de séquences et les Statecharts. Les *diagrammes de séquence* représentent des scénarios, ils sont très informels et constituent des exemples très partiels de cas d'utilisation du système. Les *Statecharts* sont maintenant largement utilisés en UML pour représenter les comportements à l'aide d'un modèle basé sur les états. Les Statecharts exploitent la hiérarchie, le parallélisme et une forme de préemption. Par la diffusion instantanée des signaux, ce modèle est un modèle synchrone.

Nous présentons des alternatives aux diagrammes de séquences et aux Statecharts. Il est clair que le manque de fondement formel rend les diagrammes de séquences peu intéressants pour les systèmes critiques. Nous leur substituons les «Synchronous Interface Behavior» (SIB). Aux Statecharts nous préférons les «SyncCharts». SyncCharts et SIB s'appuient sur des sémantiques mathématiques. Ils sont entièrement compatibles avec les langages synchrones, en particulier avec le langage Estérel. Ils bénéficient en conséquence des outils et environnements développés pour ces langages.

Grandes lignes de la présentation

Rappels sur les objets synchrones de F. Boulanger et les machines d'exécution pour langages synchrones.

Les *SyncCharts*, un modèle d'expression des comportements réactifs complexes.

Exemples de modélisation

Comparaison avec les Statecharts.

Brève introduction à leur sémantique.

Utilisation en validation.

Les *Synchronous Interface Behavior*.

Les idées sous-jacentes.

Exemples et utilisation