

Langages synchrones

Séminaire MaMuX

Mathématiques, Musique et relations avec d'autres disciplines

Vendredi 3 février 2012, 14h30-18h00

IRCAM, Salle I. Stravinsky

I, place I. Stravinsky 75004 Paris

Entrée libre dans la mesure des places disponibles

Cette séance, organisée à l'initiative des équipes Parkas (Ecole Normale Supérieure, département d'informatique) et Représentations Musicales (IRCAM/UMR STMS/INRIA), aura pour but de présenter un panorama des travaux et outils actuels dans le domaines des langages synchrones et leur utilisation dans la conception de systèmes réactifs, notamment dans le domaine musical.

Page Web de la séance :

<http://repmus.ircam.fr/mamux/saisons/saison11-2011-2012/2012-02-03>

Programme:

14h30-15h15 **ReactiveML et aspects dynamiques dans les langages synchrones**
Louis Mandel (Ecole Normale Supérieure / LRI / Université Paris Sud)

Les langages synchrones ont été conçus pour programmer des systèmes temps-réels critiques et ont eu des succès majeurs dans ce domaine. Leur pouvoir expressif a été volontairement réduit pour garantir des propriétés essentielles: exécution en temps et mémoire bornée, absence de blocage à l'exécution, déterminisme.

Plusieurs travaux dont le modèle réactif de Frédéric Bousinot et le langage Lucid Sychrone ont cependant montré que le parallélisme synchrone ne se limite pas au cadre des systèmes temps-réels. Il peut être utile pour programmer des applications tels que des interfaces graphiques ou des jeux vidéo.

Dans cet exposé, je présenterai le langage ReactiveML qui a servi par exemple à la modélisation de réseaux ad hoc. Ce langage est une extension du langage généraliste OCaml avec des primitives synchrones. Il mélange ainsi l'expressivité des langages fonctionnels (polymorphisme, ordre supérieur, etc.) avec le modèle concurrent et déterministe du synchrone.

15h15-16h00 **FAUST : programmation synchrone et fonctionnelle pour le traitement du signal**
Yann Orlarey (Grame)

FAUST est un langage de programmation synchrone spécialement conçu pour facilement développer des applications de synthèse et de traitement du signal audio numérique. Basé sur une approche fonctionnelle de la programmation, centré sur la notion de composition de fonctions, FAUST propose une notation concise, de haut niveau, permettant de décrire des applications ou des plugins audio souvent en quelques lignes de code seulement.

Le compilateur FAUST se charge ensuite de traduire automatiquement ces descriptions en programmes C++ optimisés. Il offre des fonctionnalités avancées telle que la parallélisation automatique (OpenMP ou Work-Stealing) et la génération automatique d'une documentation mathématique décrivant précisément, à des fins de préservation à long terme, tous les calculs mis en oeuvre par un programme.

Grâce au concept d'architecture, un même programme FAUST peut être déployé sans modifications sur de nombreuses plateformes audio : Max/MSP, Puredata, VST, Csound, Supercollider, Jack, CoreAudio, etc. FAUST dispose de son propre IDE, FaustWorks, qui facilite grandement l'utilisation du langage et accélère le cycle de développement. Un compilateur en ligne est également disponible. Il permet aux utilisateurs de produire aisément leurs propres applications audio ou plugins (aussi bien pour Linux, Mac et Windows) sans avoir aucun outil à installer.

16h15-17h00 **Combiner du temps discret et du temps continu dans un langage synchrone**
Marc Pouzet (Ecole Normale Supérieure / Université Pierre et Marie Curie)

Les langages synchrones permettent de programmer des systèmes réactifs embarqués à la fois très complexes et très sûrs, et on en trouve des exemples d'utilisation très significatifs dans l'avionique, le transport terrestre ou l'énergie.

Ils sont fondés sur le modèle du "parallélisme synchrone" qui introduit une notion de temps discret global dans les programmes et donne la possibilité d'écrire des programmes parallèles déterministes. Le compilateur d'un langage synchrone garantit des propriétés de sûreté importantes pour le logiciel critique: déterminisme, absence de blocage (deadlock), génération de code séquentiel s'exécutant en temps et mémoire bornés, etc. En somme, les langages synchrones permettent de programmer dans un langage de haut niveau, le code final embarqué étant produit directement par compilation.

Dans cet exposé, je montrerai comment et pourquoi étendre un langage synchrone existant pour programmer des systèmes hybrides combinant des signaux mixtes continus et discrets. Le résultat est un langage qui permet de programmer, dans une source unique, le contrôleur discret et son environnement physique continu. La combinaison de signaux mixtes soulève des questions nouvelles de sémantique, de typage et de compilation. Je montrerai les solutions que nous avons proposées et les illustrerai, à l'aide d'un prototype en cours de développement, et qui étend Lucid Synchrone avec des équations différentielles ordinaires (ODEs).

17h00-17h45 **Aspects synchrones dans les langages de programmation appliqués à la composition et à la performance musicales**
Karim Barkati et José Echeveste (IRCAM)

A travers une analyse de plusieurs langages de programmation dédiés à la création musicale (composition et performance), cet exposé abordera les différentes stratégies adoptées par ces langages, en particulier au niveau de la syntaxe et de la sémantiques du langage, des politiques d'ordonnement, de la modélisation du temps musical, des applications visées (traitement audio, live-coding, accompagnement automatique...), et de leurs relations aux langages synchrones.

I ème saison du Séminaire MaMuX **Mathématiques, musique et relations avec d'autres disciplines**

Les séances de la saison 2011-2012 visent à renforcer les échanges dans les domaines des sciences et technologies de l'information et les études à la fois formelles et appliquées des relations entre langages, systèmes et paradigmes informatiques, logique, calcul, rapportés à une perspective musicale.

Trois séances (les 7/10/2011, 2/12/2011 et 9/03/2012) seront dédiées aux systèmes évolutifs à mémoire (SEM) développés par A.C. Ehresmann et J.P. Vanbreemsch, en introduisant en détail ce modèle, et en l'illustrant, aussi bien avec les motivations originales que sont la cognition et les sciences sociales, qu'avec la musique.

Calendrier de la saison :

7 octobre 2011: *Modélisation de la hiérarchie et de l'émergence : autour des systèmes évolutifs hiérarchiques.*

4 novembre 2011: *Programmation par contraintes en musique.*

2 décembre 2011: *Systèmes évolutifs à mémoire (2): Auto organisation, apprentissage.*

6 janvier 2012: *Théories du jonglage et applications musicales.*

3 février 2012: *Langages synchrones.*

9 mars 2012: *Systèmes évolutifs à mémoire (3).*

30 mars 2012: *Arbres et structures hiérarchiques.*

4 mai 2012: *Représentation des structures harmoniques.*

Le Séminaire MaMuX est organisé par L'Equipe Représentations Musicales de l'**IRCAM**, avec le soutien du **CNRS** (UMR STMS – Sciences et technologies de la musique et du son) et du Réseau National des Systèmes Complexes (**RNSC**).
Comité scientifique : Carlos Agon, Moreno Andreatta, Gérard Assayag, Jean Bresson, Jean-Louis Giavitto, John Mandereau.

Pour tout renseignement, contacts et propositions d'intervention ou de séance thématique:

John Mandereau: john.mandereau@ircam.fr

Jean Bresson: jean.bresson@ircam.fr

Plus d'informations sur <http://repmus.ircam.fr/mamux/>