

Amphithéâtre de l'Opéra National de Lyon



■ 12h30

LE PROJET FAUST

Nouvelles approches pour le «calcul» du son Conférence de Yann Orlarey

Lancement du site autour de FAUST (langage de programmation et compilateur pour la synthèse et le traitement du signal)

Faust

Le projet FAUST, mené à Grame depuis quelques années, vise à développer un ensemble d'outils et de techniques très innovantes au service du «calcul» du son. S'adressant à un large public, curieux de l'état de l'art dans le domaine des techniques numériques, la conférence a pour ambition de présenter les derniers développements du projet FAUST de manière simple et accessible à tous.

La notion de «calcul» (du son, de la musique, des images, des interactions temps-réel) revêt aujourd'hui une importance toute particulière pour les musiciens et les artistes qui cherchent à exploiter d'une manière personnelle et innovante les outils numériques. Elle est en réalité au cœur du processus de création numérique. L'ambition du projet Faust est, dans le domaine du son, de rendre ces calculs à la fois plus faciles, plus accessibles et plus performants.

Pour que la machine puisse réaliser nos calculs, nous avons besoin d'un «vocabulaire», d'un «langage», qui va nous permettre de l'instruire des choses à faire. On conçoit facilement que le choix du langage ne soit pas neutre. Il va en réalité avoir un double impact. D'une part il va conditionner la facilité avec laquelle nous allons penser, décrire et comprendre ces calculs. D'autre part il va influencer sur la performance et l'efficacité de la réalisation de ces calculs par la machine.

La réponse proposée dans le cadre du projet FAUST à ce double défi, s'articule autour de trois points. Tout d'abord le langage FAUST lui-même, spécialement conçu pour décrire des calculs sonores de manière simple, précise et concise. Deuxièmement des techniques de synthèse de programme qui vont, à partir de cette description, engendrer automatiquement un programme informatique capable de réaliser ces calculs de la manière la plus efficace possible. Troisièmement la possibilité, là aussi automatique, d'adapter ce programme informatique aux principaux standards du traitement audio numérique (MaxMSP, VST, JACK, ALSA, OSS, PD, SC, etc...).

La présentation des différents aspects du projet FAUST sera accompagnée de démonstrations et d'exemples concrets. Signalons que les sources et les logiciels du projet FAUST sont accessibles gratuitement sur sourceforge (<http://sourceforge.net/projects/faudiostream/>). Il est également possible d'utiliser FAUST en ligne, sans avoir à l'installer sur sa machine, à partir de l'adresse suivante : <http://faust.grame.fr>. Le site comporte tout un catalogue d'exemples que l'utilisateur peut utiliser tel quels ou adapter à ses propres besoins.

Entrée libre



© M. Guefferat

Yann Orlarey

compositeur, directeur scientifique (Grame)

Yann Orlarey est né en 1959. Parallèlement à des études universitaires en sciences économiques et en informatique, il a suivi la classe de musique électroacoustique du Conservatoire de Saint-Étienne. Membre du Grame depuis 1983, il est actuellement directeur scientifique de cet organisme. Ses travaux de recherche portent principalement sur les langages formels pour la composition musicale et les systèmes d'exploitation musicaux temps-réel. Il est l'auteur ou le co-auteur de différents systèmes et logiciels musicaux. Son répertoire musical comporte des pièces sur bande, des pièces interactives et des pièces instrumentales pour solistes, petites formations et orchestres. Pour la plupart, ses œuvres font appel à des moyens informatiques, soit au niveau des situations de jeu instrumental proposées aux interprètes, soit au niveau de la composition musicale proprement dite. Plusieurs pièces sont le résultat d'un travail de co-écriture réalisé dans le cadre de Grame avec le clarinetiste et compositeur Jérôme Dorival, par exemple. Ses œuvres ont été jouées en Europe, mais aussi aux Etats Unis, au Canada et en Chine.

Faust

Online Compiler

Summary

New Faust source code.

Compilation

options

Name

Architecture

Other

Faust Code

```
declare name "default";
declare version "1.0";
declare author "Grame";
declare license "GPL";
declare copyright "© 1999-2006";

// ...
// ...
// ...

makeFile "input1",
input("music1.fst");

makeFile "output1" = makeFile "input1" | makeFile "lib" | 20 | 100;
makeFile "output2" = makeFile "input2" | makeFile "lib" | 40 | 100;
makeFile "output3" = makeFile "input3" | makeFile "lib" | 60 | 100;
makeFile "output4" = makeFile "input4" | makeFile "lib" | 80 | 100;
```