Le compilateur en ligne de Faust : un IDE en ligne pour le langage de programmation FAUST

Romain Michon, Yann Orlarey

LE COMPILATEUR EN LIGNE DE FAUST : UN IDE EN LIGNE POUR LE LANGAGE DE PROGRAMMATION FAUST

Romain MICHON et Yann ORLAREY
GRAME, Centre national de création musicale
9 rue du Garet
69202 Lyon,
France,
rmnmichon@gmail.com et orlarey@grame.fr

RÉSUMÉ

FAUST est un langage de programmation fonctionnel pour le traitement du signal et la synthèse de sons en temps réel. Grâce à un système de fichiers d'architectures, un seul et unique programme FAUST peut être utilisé pour générer du code pour un ensemble de types d'applications et de plug-ins.

Le compilateur en ligne de FAUST ici présenté est une application Web écrite en `PHP` et en `JavaScript` offrant un environnement de développement multiplateforme et multiprocesseur pour le langage FAUST. Cet outil rend possible l'utilisation de la plupart des fonctionnalités de FAUST dans un navigateur Web et intègre un catalogue d'exemples évolutif faisant de lui une plate-forme pour utiliser et échanger facilement tout objet FAUST.

Le fonctionnement du compilateur en ligne de FAUST est présenté en détail dans cet article. Les possibilités offertes par cet outil sont discutées et une brève ouverture sur les enjeux de l'utilisation des technologies Web pour l'informatique musicale est faite.

1. INTRODUCTION

FAUST [4] est un langage de programmation fonctionnel pour le traitement du signal et la synthèse de sons en temps réel. Grâce à un système de fichiers d'architectures, un seul et unique programme FAUST peut être utilisé pour générer du code pour un ensemble de types d'applications et de plug-ins.

Le compilateur en ligne de FAUST est une application Web écrite en `PHP` et en `JavaScript` offrant un environnement de développement multi-plateforme et multiprocesseur pour le langage FAUST. Cet outil rend possible l'utilisation de la plupart des fonctionnalités de FAUST dans un navigateur Web. Nous pensons qu'il permet de résoudre l'une des principales limites de FAUST : le nombre important de dépendances requises pour l'utilisation de ses architectures. En effet, bien que FAUST soit totalement indépendant de toute bibliothèque externe pour pouvoir être compilé, l'utilisation de certaines de ses architectures se montre assez complexes à cause du grand nombre de dépendances sur lesquelles leur fonctionnement repose. Le problème des architectures système que certains utilisateurs travaillant par exemple sous Windows ont pu rencontrer est également solutionné.

Le compilateur en ligne de FAUST intègre un catalogue d'exemples éditable par ses utilisateurs ce qui en fait également une plate-forme d'échange. Grâce à l'interaction entre ses différents éléments, il peut être vu aussi comme un outil pédagogique dans lequel il est possible de visualiser un algorithme de traitement du signal écrit en FAUST, son équivalent en `C++`, sa représentation sous forme de digramme, sa documentation et son résultat.

Le compilateur en ligne de FAUST a été implémenté dans le cadre du développement du nouveau site internet de FAUST [1] . Il peut être intégré dans toute page `HTML` et installé facilement sur un serveur Apache.

2. INTERFACE

Le compilateur en ligne de FAUST est basé sur un block de taille fixe dont les proportions ont été définies en fonction de la taille du nouveau site Web de FAUST. Toutefois, il peut être aisément mis à l'échelle et intégré dans n'importe quelle `div` d'une page `HTML` en utilisant un système de `iframe` (Cf. 4). Un mode "plein écran" est également disponible afin d'offrir aux utilisateurs une zone de travail plus large.

La navigation entre les différents états d'un objet FAUST se fait au travers d'un ensemble d'onglets accessibles à l'aide d'une barre horizontale (Cf. cadre n° 1 de la figure 1).

2.1. Editer le code

Le code FAUST peut être édité de deux manières différentes dans le compilateur en ligne. La plus évidente consiste à utiliser l'éditeur de code placé dans l'onglet `Faust Code`. Le programme `JavaScript` CodeMirror [2] est utilisé pour mener à bien cette tâche. Ce dernier a été choisi pour sa compatibilité avec un grand nombre de navigateurs Web (Firefox, Safari, Opera, Internet Ex-

1. `http://faust.grame.fr`
2. `http://codemirror.net`

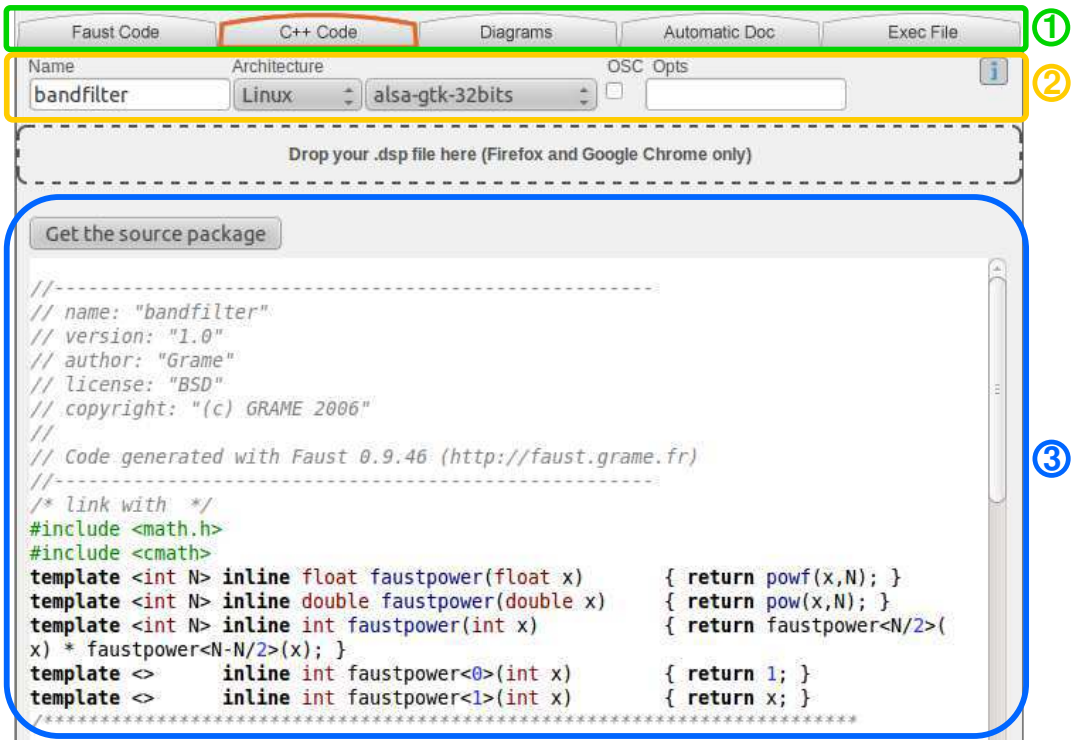

Figure 1. Vue d'ensemble des différentes zones constitutives du compilateur en ligne de FAUST. Le cadre 1 contient la barre de navigation, le cadre 2 le menu des options de compilation et le cadre 3 les différents types de résultats demandés ou l'éditeur de codes affichés en fonction de l'onglet sélectionné.

plorer, Google Chrome, etc.). Il utilise un modèle de coloration de code spécifique à FAUST.

Grâce a un script `AJAX`, le compilateur en ligne de FAUST autorise également le glissé/déposé de tout fichier `.dsp` contenant un code FAUST dans une zone placée juste au dessus de l'éditeur de code (Cf. figure 1). Cette zone est présente dans chaque onglet. Cela permet donc de compiler un code FAUST créé en local rapidement et de manière très interactive.

2.2. Compilation

2.2.1. *Options de compilation*

Le menu d'options de compilation est visible dans les onglets `C++ Code` et `Exec File`. Il est situé juste en dessous de la barre de navigation (cf. cadre 2 sur la figure 1) et permet de définir le type d'exécutable qui sera généré lors de la compilation `C++`. La plupart des architectures FAUST sont disponibles et une option "OSC" peut être sélectionnée si l'architecture choisie le permet. D'autres options peuvent être passées en argument à l'aide de la zone de texte `Opts`.

Le serveur Linux (Ubuntu) sur lequel est installé le compilateur en ligne de FAUST à GRAME permet de compiler le code `C++` généré par FAUST en 32 bits et en 64 bits pour les plate-formes suivantes :

– Linux ;
– Windows, en utilisant le cross compilateur `MinGW` [3] ;
– OSX 10.6, en effectuant la compilation sur un serveur Mac distant à l'aide du système d'exécution de commandes de `SSH` [4] et en important le fichier obtenu avec `scp`.

Si l'une de ces options est changée dans les onglets `C++ Code` ou `Exec File`, le résultat est automatiquement mis à jour.

2.2.2. *Compilation*

Afin de mener à bien les différentes étapes de compilation, un dossier temporaire contenant le fichier FAUST a traiter et un Makefile généré en fonction des options et de l'architecture choisies par l'utilisateur est créé sur le serveur. Il sera utilisé pour produire le code `C++`, les diagrammes, la documentation de l'objet, l'exécutable et les différents packages.

2.3. Générer le code C++

Lorsque l'onglet `C++ Code` est sélectionné, le compilateur de FAUST est appelé sur le serveur par le Makefile. Le code généré est alors traité par `Highlight` [5] pour le colorer. Si une architecture autre que *none* est sélectionnée par un utilisateur, le Makefile créera également

3. `http://www.mingw.org`
4. `http://www.openssh.com`
5. `http://www.andre-simon.de/doku/highlight/en/highlight.html`

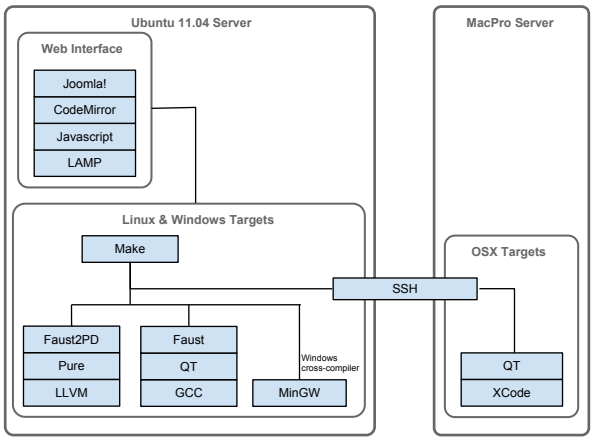

Figure 2. Schéma des différents composants du serveur.

un package contenant tous les éléments nécessaires pour qu'il puisse compiler le code `C++` sur sa machine sans que FAUST n'y soit installé. Il est toutefois important de noter qui si l'option `OSC` est sélectionnée, le package généré lors de l'étape décrite précédemment ne sera compilable que si FAUST est installé sur le système de l'utilisateur car les bibliothèques alors requises font partie de la distribution de FAUST.

Si le code FAUST entré contient des erreurs, le message retourné par le compilateur sera affiché au lieu du code `C++`. Le compilateur en ligne réagira de la même manière dans le cas des onglets `Diagrams` et `Automatic Doc`.

2.4. Visualiser les différents types de diagrammes

Plusieurs types de diagrammes du code FAUST donné au compilateur en ligne peuvent être visualisés dans l'onglet `Diagrams` :

- block-diagramme (Cf. figure 3) ;
- schéma des signaux (Cf. figure 4) ;
- schéma des tâches.

Comme le montrent les figures 3 et 4, le type de diagramme affiché est sélectionné à l'aide d'un menu déroulant. Les images générées sont dans un format vectoriel (SVG), il est donc possible de zoomer sur ces dernières sans voir apparaître de crénelage. De plus, les block-diagrammes peuvent être parcourus en cliquant sur leurs différentes boîtes.

2.5. Visualiser la documentation générée automatiquement d'un objet FAUST

Grâce a de récents travaux menés à GRAME dans le cadre du projet ASTREE (voir *Remerciements*), le compilateur de FAUST est maintenant en mesure de générer automatiquement la documentation mathématique d'un objet écrit avec le langage FAUST [1]. Le compilateur en ligne est compatible avec cette fonction et permet d'afficher le fichier `pdf` produit à l'aide de Google Document

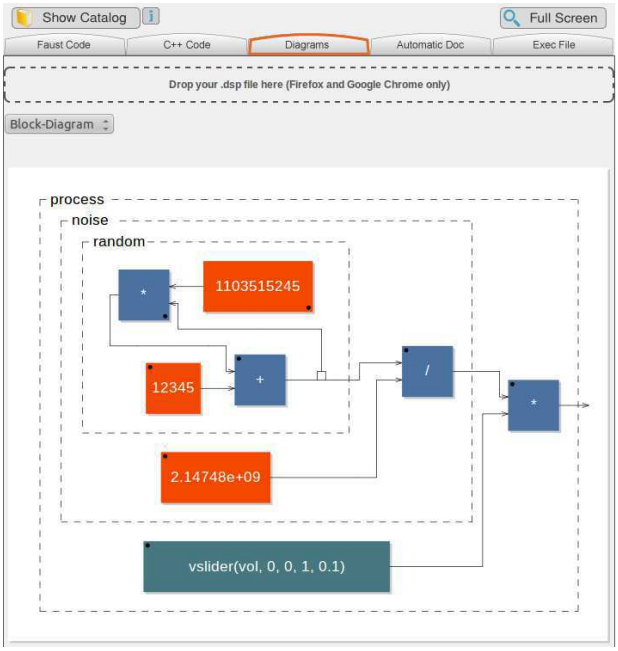

Figure 3. Capture d'écran de l'onglet `Diagram`. L'option *Block-Diagram* est sélectionnée.

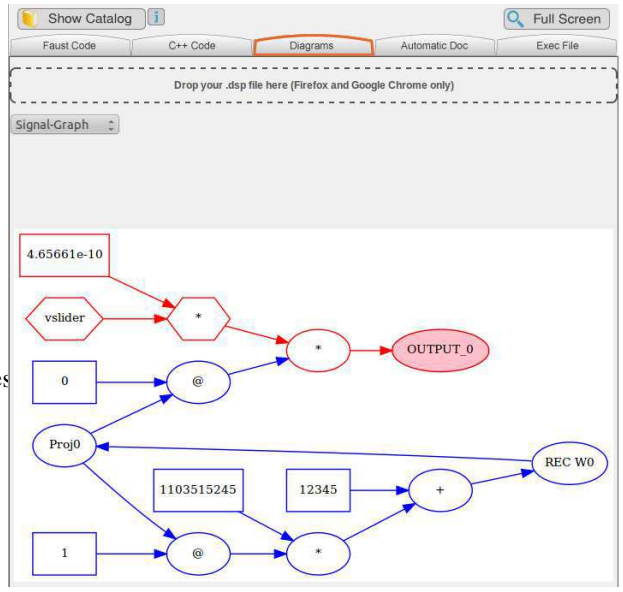

Figure 4. Capture d'écran de l'onglet `Diagram`. L'option *Signal-Graph* est sélectionnée.

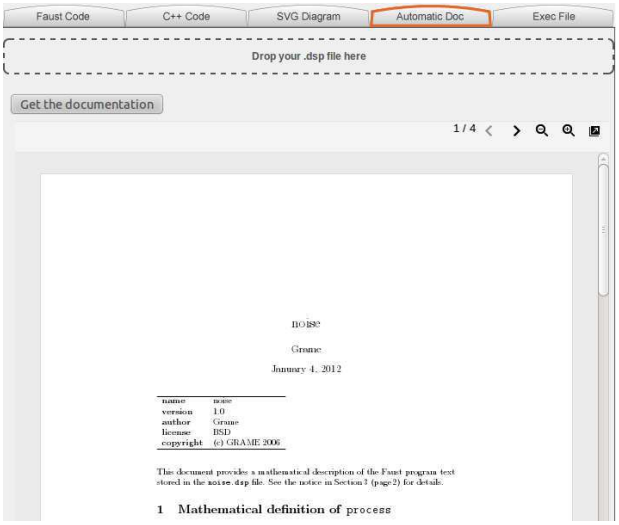

Figure 5. Capture d'écran de l'onglet `Automatic Doc`.

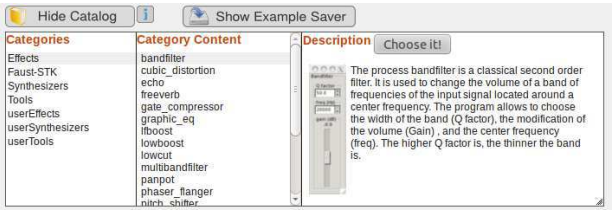

Figure 7. Le catalogue d'exemples.

Figure 8. L'enregistreur d'exemples.

Viewer [6]. Le Makefile décrit dans 2.3 est utilisé pour mener à bien cette opération.

Une capture d'écran de l'onglet `Automatic Doc` est visible dans la figure 5.

2.6. Générer un exécutable

Lorsque l'onglet `Exec File` est choisi, la compilation `C++` est effectuée en fonction du système d'exploitation et de l'architecture processeur sélectionnée. Si cette opération s'est déroulée correctement, un bouton de téléchargement apparaît. Sinon, le message d'erreur retourné par le compilateur `C++` est affiché. Le fichier généré sera, en fonction de l'architecture FAUST choisie, soit un paquet contenant plusieurs fichiers, soit un exécutable.

3. CATALOGUE D'EXEMPLES

Le compilateur en ligne de FAUST contient une plateforme pour échanger facilement et de manière interactive des objets FAUST : le catalogue d'exemples. Deux éléments le constituent : le catalogue lui-même et un enregistreur d'exemples.

3.1. Le catalogue

La catalogue à la forme d'un explorateur de dossiers dans lequel chaque code FAUST est placé dans différentes catégories (Cf. figure 7) :

- Effects ;
- Faust-STK [3] ;
- Synthesizers ;
- Tools.

Un ensemble de catégories modifiables "utilisateur" (user) sont également disponibles.

6. `http://docs.google.com/viewer`

Lorsqu'un objet FAUST est sélectionné, une courte description et une capture d'écran de ce dernier sont affichées dans la colonne `Description`. Il peut alors être envoyé au compilateur en ligne soit en le double-cliquant, soit poussant le bouton `Choose it!`.

La catalogue d'exemples peut être affiché dans tous les onglets du compilateur en ligne de FAUST. Ainsi, un utilisateur peut choisir de visualiser simplement le diagramme d'un objet ou sa documentation.

3.2. Enregistrer un objets Faust dans le catalogue

Comme cela a été dit précédemment, le compilateur en ligne de FAUST se veut être une plate-forme pour l'échange d'objet FAUST. Ceci est rendu possible par l'enregistreur d'exemples qui permet à un utilisateur de sauvegarder son code FAUST dans une des catégories "utilisateur" du catalogue. Aucune identification n'est requise pour effectuer cette tâche : n'importe qui peut enregistrer son travail sur le catalogue à condition qu'il s'agisse d'un code FAUST compilable correct. Lorsqu'un nouvel exemple est placé dans le catalogue, l'utilisateur peut choisir d'y ajouter une courte description ainsi qu'une capture d'écran.

Si un exemple existant est modifié par un utilisateur différent de celui qui l'a créé, chaque utilisateur qui a contribué à cet exemple sera informé de cette modification par un e-mail. Enfin, n'importe qui peut choisir de supprimer un exemple contenu dans le catalogue. Tout comme dans le cas précédent, l'ensemble des contributeurs de cet objet sont alors avertis.

Dans le but de conserver un bon ordre dans le catalogue et d'éviter les spams et les duplicatas, un système de vote a été implémenté. Cet outil peut aussi permettre de mettre en valeur les meilleurs exemples utilisateurs.

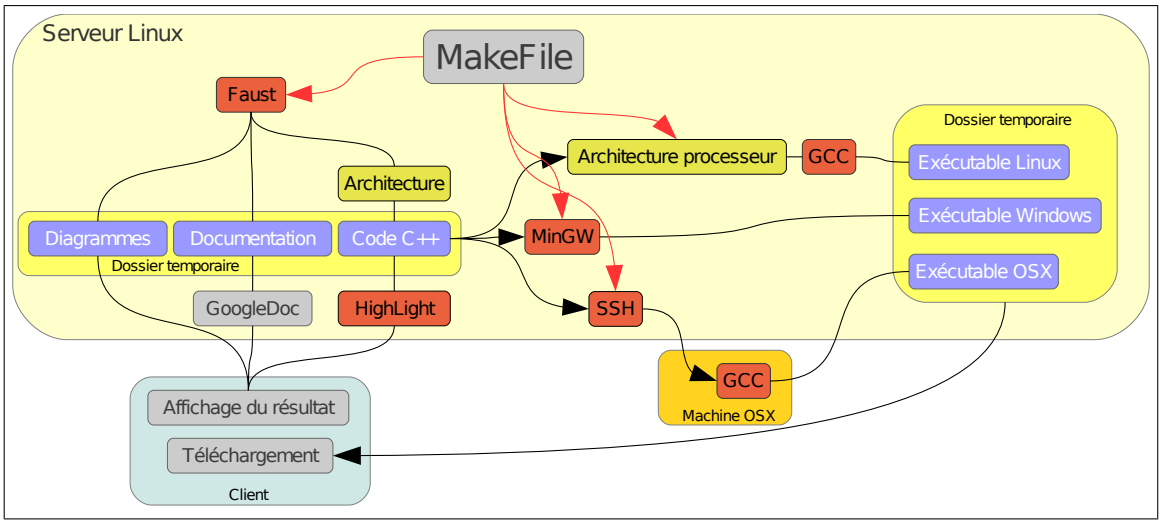

Figure 6. Fonctionnement du compilateur en ligne de FAUST.

4. EXPORTER LE COMPILATEUR EN LIGNE DE FAUST

Le compilateur en ligne de FAUST peut être implanté très facilement dans un page HTML en utilisant une `iframe` :

```
<iframe src="http://faust.grame.fr/
    compiler" style="border:none" height="
    1200px" width="722px"></iframe>
```

Il peut aussi est installé sur un serveur Apache [7] simplement en copiant son code source, disponible dans la distribution de FAUST [8].

5. TRAVAUX FUTURS

De récentes améliorations apportées au compilateur de FAUST lui permettent de générer du code JAVA et LLVM [9] en plus de C++. Ainsi, il pourrait être intéressant d'implémenter une architecture FAUST qui permette de générer des applets JAVA directement utilisables en ligne depuis un navigateur Web. Nous espérons donc pouvoir compléter assez rapidement le compilateur en ligne de FAUST avec un nouvel onglet dans lequel un applet JAVA correspondant au code FAUST édité serait utilisable. Nous avons bon espoir d'implémenter un système dans lequel les changements apportés au code seraient visibles presque immédiatement au niveau de l'applet JAVA qui resterait active jusqu'à la fermeture de la session. Par conséquent, une forme de "live coding" pour FAUST deviendrait disponible.

Le compilateur en ligne de FAUST semble offrir une bonne solution aux problèmes de plate-formes et de dépendances impliqués par l'utilisation de certains fichiers d'architecture. Il pourrait ainsi être intéressant de mettre en place un système de service Web RESTful[5] pour FAUST accessible à l'aide d'un API simple. Le programme FaustWorks [10] pourrait alors utilisé un tel système pour offrir la possibilité aux utilisateurs de compiler leur code FAUST pour des plate-formes différentes de la leur et avec des architectures qu'ils n'arrivaient pas a utiliser sur leur système.

7. http://www.apache.org
8. http://sourceforge.net/projects/faudiostream
9. http://llvm.org/

6. CONCLUSION

Le compilateur en ligne de FAUST simplifie significativement l'utilisation de FAUST en mettant à disposition un environnement de programmation prêt à l'emploi et libre de tout problèmes de compatibilité et d'installation.

Il est certain que les technologies Web jouent un rôle de plus en plus important dans le domaine de l'informatique musicale. L'avènement de nouveaux standards tel que HTML5 ou la nouvelle API web-audio [11] ouvre la voie vers de nouveaux domaines d'exploration tel que le traitement numérique du signal en ligne.

7. REMERCIEMENTS

Ce travail a été effectué dans le cadre du projet ASTREE (ANR-08-CORD-003) sur la préservation des pièces de musique électronique en temps réel.

8. REFERENCES

[1] Barkati, K., Fober D., Letz, S. et Orlarey, Y. "Two recent extensions to the Faust compiler", *Proceedings of the Linux Audio Conference (LAC-2011)*, National University of Ireland, Maynooth, Irlande, 2011.

10. FaustWorks est un IDE pour FAUST implémenté avec Qt.
11. https://dvcs.w3.org/hg/audio/raw-file/tip/webaudio/specification.html

[2] Fober, D., Orlarey, Y. et Letz, S. "Faust architecture design and OSC support", *Proceedings of the Conference on Digital Audio Effects (DAFx-11)*, IRCAM, Paris, France, 2011.

[3] Michon, R. et Smith, J. "Faust-STK : a set of linear and nonlinear physical models for the Faust programming language", *Proceedings of the Conference on Digital Audio Effects (DAFx-11)*, IRCAM, Paris, France, 2011.

[4] Orlarey, Y., Fober, D. et Letz, S. "An algebra for block diagram languages", *Proceedings of the International Computer Music Conference (ICMA)*, Gothenburg, Suède, 2002.

[5] Richardson, L. et Rubu, Y. *Services Web RESTful*. O'Reilly, Paris, 2007.