



**HAL**  
open science

# UN EXEMPLE DE CONVERGENCE DE L'ANALYSE ET DE L'INFORMATIQUE MUSICALES

Alain Bonardi, Guillaume Pellerin, Emilie Zawadzki

► **To cite this version:**

Alain Bonardi, Guillaume Pellerin, Emilie Zawadzki. UN EXEMPLE DE CONVERGENCE DE L'ANALYSE ET DE L'INFORMATIQUE MUSICALES: Enrichir le projet Analyses par le Web Audio. Journées d'Informatique Musicale 2017, May 2017, Paris, France. hal-01582653

**HAL Id: hal-01582653**

**<https://hal.science/hal-01582653>**

Submitted on 6 Sep 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# UN EXEMPLE DE CONVERGENCE DE L'ANALYSE ET DE L'INFORMATIQUE MUSICALES

## Enrichir le projet Analyses par le Web Audio

*Alain Bonardi*  
CICM - MUSIDANSE  
Université Paris 8 et IRCAM  
alain.bonardi@univ-paris8.fr

*Guillaume Pellerin*  
Centre de Ressources IRCAM  
guillaume.pellerin@ircam.fr

*Emilie Zawadzki*  
Centre de Ressources IRCAM  
emilie.zawadzki@ircam.fr

### RÉSUMÉ

Cet article rend compte d'un exemple de convergence entre analyse et informatique musicales : nous proposons aux internautes de découvrir un aspect de *En Echo* de Philippe Manoury, pour soprano et électronique en temps réel, en manipulant un module de transformation du son programmé en Faust pour le Web audio. Le projet ouvre à d'autres environnements que ceux utilisés par le compositeur, notamment le Web, et à l'appropriation du code par le lecteur. Dans un premier temps, nous présentons le projet Analyses mené à l'Ircam autour d'œuvres faisant appel aux technologies de l'informatique musicale, qui est le cadre de cette expérimentation. Nous montrons ensuite plusieurs exemples d'applications pédagogiques fondées sur l'apprentissage par la manipulation en lui associant souvent la découverte de la programmation, notamment avec le langage Javascript. Inspirés par ces réalisations, nous proposons un premier harmoniseur Web Audio écrit en langage Faust et destiné à la manipulation et l'appropriation en ligne de ce traitement tel que conçu par le compositeur dans *En Echo*. Nous concluons en suggérant des pistes de développement pour l'analyse des musiques mixtes, en étendant cette approche à d'autres modules et en développant des partitions augmentées.

### 1. INTRODUCTION

Les œuvres de musique mixte, associant instruments acoustiques et dispositifs électroniques ou informatiques (en temps réel ou sur support fixé) interrogent l'analyse musicale. La partition y perd son statut de document privilégié, rejointe par les programmes informatiques et schémas d'implémentation et de connexion. La description du dispositif électronique entre en compte ; mais elle ne ressort pas de la seule lutherie, car elle est inséparable de son séquençement temporel [6]. La conférence « Analyser la musique mixte » organisée en 2012 à l'Ircam montrait l'étendue du domaine, des problématiques et des approches : les questions transversales de notation, d'hétérogénéité des supports, d'orchestration, de lutherie, d'interprétation

rencontraient les propositions d'analyses d'œuvres ou de corpus.

Si l'analyse musicale s'est ouverte à d'autres objets d'étude que la partition, et à d'autres supports que le papier, ses productions restent en grande majorité des textes. En préambule de l'ouvrage *Analyser la musique, pourquoi, comment ?* de François Delalande [3], Daniel Teruggi donne comme but à cette activité de « *décrire l'audible dans ses multiples dimensions à travers l'écrit* ». Cet attachement à l'écrit peut surprendre venant d'auteurs issus du GRM, abordant la musique par son versant acousmatique. La recherche présentée dans cet article expérimente une analyse d'œuvre mixte temps réel faisant appel à des programmes informatiques comme support. Le but n'est pas seulement de se substituer au support papier mais bien de déployer de nouvelles modalités d'analyse tirant parti des trois fonctions de l'ordinateur [4] :

- l'ordinateur comme *processeur de calcul* (en temps réel ou non) des synthèses et transformations sonores ;
- l'ordinateur comme *mémoire*, c'est-à-dire le stockage des sons soit à la volée soit à partir d'enregistrements antérieurs ;
- l'ordinateur comme *interface*, qui vise une certaine « jouabilité » du dispositif électronique, aspirant implicitement à devenir instrument.

Notre approche souhaite renforcer l'appropriation de l'analyse par le lecteur : dans la perspective dessinée il y a quelques années autour de l'écoute active ou *active listening* [5], se dessinent des approches de l'analyse musicale sur support numérique que nous pouvons qualifier d'actives grâce à des supports interactifs. Remarquons que la musique mixte, pensée comme association interactive entre deux mondes musicaux, incite à des représentations également interactives de ses processus.

Dans un premier temps, nous présenterons le projet Analyses mené à l'Ircam, qui sert de support à cette recherche. Dans un deuxième temps, nous donnerons des exemples de réalisations pédagogiques associant la manipulation à la compréhension. Nous montrerons ensuite l'harmoniseur temps réel Web Audio que nous

avons élaboré dans le cadre de l'analyse de *En Echo* de Philippe Manoury. Nous évoquerons enfin les perspectives d'évolution que nous imaginons dans le cadre du projet Analyses.

## 2. LE PROJET ANALYSES À L'IRCAM

Le projet Analyses est une série d'analyses musicales proposées depuis 2015 par le Centre de Ressources de l'Ircam. Il s'inscrit dans un mouvement plus large de documentation et présentation des œuvres musicales contemporaines faisant appel à la technologie, dont témoignent des réalisations comme les Portraits Polychromes du GRM<sup>1</sup> ou les analyses du projet TaCEM (évoquées dans la section 3.2 du présent article). Le but du projet Analyses est de mettre à disposition du public une documentation de référence sur un ensemble d'œuvres « canoniques » (œuvres mixtes temps réel ou sur support fixé, œuvres acousmatiques), réalisées à l'Ircam ou en connexion directe avec cet institut. Toutes les analyses multimédia (incluant extraits sonores, partitions, etc.) sont accessibles gratuitement en ligne sur le portail (cf. figure 1) : <http://brahms.ircam.fr/analyses/>

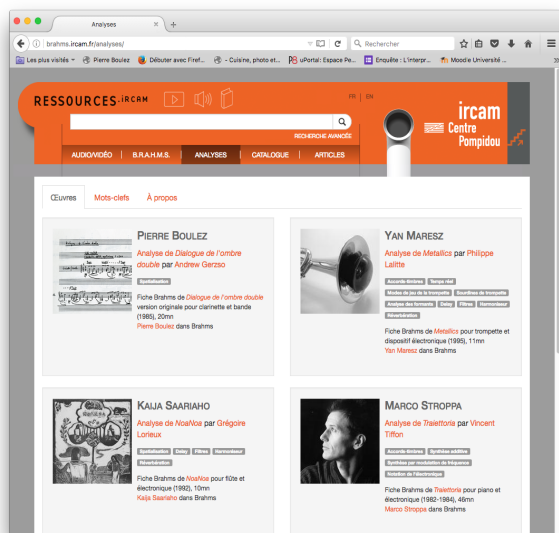


Figure 1. Le portail du projet Analyses de l'Ircam.

Les analyses s'adressent à plusieurs publics : interprètes, musicologues, étudiant-e-s en musique et en informatique musicale — et toute personne curieuse de ces musiques, au-delà de motifs professionnels.

Plusieurs objectifs sont poursuivis par ce projet :

- faire connaître les œuvres produites à l'Ircam à un public plus large,
- montrer la relation entre l'idée musicale et les technologies utilisées,
- identifier les nouveaux éléments du vocabulaire musical qui émergent à travers ces œuvres,
- offrir un support d'information aux interprètes.

Chaque analyse est structurée en trois parties :

- une description générale de l'œuvre,
- une analyse d'extraits de l'œuvre avec mise en relation de l'idée musicale et de l'écriture électronique (cf. figure 2),
- une liste de ressources spécifiques (type de problème musical abordé, technologies utilisées, œuvres abordant le même type de problématique) et générales (biographiques, historiques, techniques).

Les analyses seront également mises en relation avec :

- Brahms : une base de données encyclopédique en ligne de compositeurs de musique contemporaine de toutes les nationalités dont les œuvres ont été créées après 1945. Cette base contient actuellement environ 600 références. Pour chaque compositeur, on y trouve une partie biographique accompagnée de sources d'information, et une autre partie qui situe l'orientation esthétique, les phases principales et le contexte historique de l'œuvre.
- Images d'une œuvre : une collection d'interviews filmés des compositeurs.
- Sidney : une base de données qui contient les éléments techniques (programmes informatiques, sons etc.) nécessaires à l'exécution des œuvres avec technologie.

L'*harmoniseur* effectue des opérations de translation sur la fréquence du signal d'entrée. Il permet donc de transposer un son, vers l'aigu ou le grave, sans modifier sa durée. Si le taux de transposition est de quelques degrés, le timbre n'est pas altéré, mais au-delà le son est déformé, et prend un timbre caractéristique. On peut affecter un *delay* à chaque transposition pour obtenir un effet d'arpège. Un exemple d'utilisation de l'*harmoniseur* se trouve à la fin de la section *Cup* (mes. 89-91), où le trompettiste joue un *si3* en valeurs brèves. Lors de l'événement 43 (mes. 89), les valeurs de transposition sont respectivement de 10 cents, -300 cents, -1000 cents, -1300 cents. L'effet produit est un accord plaqué constitué des notes *la #2, do #3, sol #3* et *si3*.

L'événement 44 (mes. 89) illustre le rôle que peut jouer l'affectation d'un *feedback* à l'*harmoniseur* en produisant des transpositions en chaîne par réinjection des valeurs vers le signal d'entrée. Un délai de 500 ms. est d'abord affecté à l'*harmoniseur*, l'accord joué par l'électronique est donc plaqué un demi-temps après la note jouée (noire = 60). Après un second délai de 2000 ms, une transposition (-200, -708, -897, -1289) est appliquée avec un *feedback* (fb 50) qui entraîne un glissement vers le grave pour chaque note de la phrase mélodique après un court délai de 50 ms.

micro-to-bounce 0 10;  
 harmon1 del 500;  
 harmon2 del 500;  
 harmon3 del 500;  
 harmon4 del 500;

del 2000

micro-to-harm 120 10  
 harm1 fb 50, win 25, del 50, trsp -200;  
 harm2 fb 50, win 25, del 50, trsp -708;  
 harm3 fb 50, win 48, del 50, trsp -897;  
 harm4 fb 50, win 10, del 50, trsp -1289;

© Editions Durand / Universal Music Publishing Classical

Figure 2. Description du rôle de l'harmoniseur, extraite de l'analyse de *Metallica* de Yan Maresz.

<sup>1</sup> [ww.inagram.com/accueil/collections/portraits-polychromes](http://ww.inagram.com/accueil/collections/portraits-polychromes)

Ces analyses se présentent d'emblée comme technico-musicales, intégrant dans leur propos la présentation du dispositif électronique comme contribution à la conception musicale élaborée par le compositeur. Les analyses sont actuellement essentiellement fondées sur des descriptions textuelles corroborées par des exemples multimédia. La figure 2 montre un exemple tiré de l'analyse de *Metallics* de Yan Maresz, correspondant à la présentation du rôle de l'harmoniseur dans cette œuvre, avec l'écoute d'un extrait significatif.

### 3. MANIPULER POUR COMPRENDRE

Nous donnons ici trois exemples d'applications pédagogiques de simulation permettant la compréhension par la manipulation. Nous avons choisi trois approches différentes, allant de l'apprentissage de contenus scientifiques à la création musicale, proposant trois démarches de compréhension : des lois physiques, du travail du compositeur par l'écoute, et enfin de la génération musicale en Javascript. Remarquons que la créativité proposée au lecteur ne relève pas seulement de l'objet et du contenu proposés mais également de l'ouverture de l'application à la programmation ou reprogrammation.

#### 3.1. Manipulation et programmation de simulations en sciences physiques

Depuis quelques années, de nombreuses applications interactives proposent l'implémentation de simulations permettant de comprendre des principes et lois scientifiques par la manipulation. Les sciences physiques se sont tournées assez vite vers la création d'environnements permettant de saisir la portée des équations et constantes modélisant les phénomènes. Fondé en 2002 par le prix Nobel de physique Carl Wieman, le site PHET (<https://phet.colorado.edu/fr/>) propose des simulations interactives en mathématiques et sciences codées en HTML 5, directement utilisables dans un navigateur. L'objectif peut également dépasser la manipulation de simulations : dans certains cas, le lecteur peut accéder au code informatique de ces réalisations et progresser en langages de programmation. Ainsi le logiciel Phision (<http://phision.net/>) permet à la fois de créer et pratiquer des simulations en utilisant des modèles déjà inclus dans le logiciel, mais aussi d'en programmer de nouvelles en écrivant des scripts en Javascript.

#### 3.2. Manipulations de contenus sonores

Le projet TaCEM, mené à l'Université de Huddersfield par Michael Clarke [2] et son équipe, propose des analyses interactives fondées sur l'écoute de sons extraits (en stéréo) de l'œuvre étudiée (en anglais, *interactive aural analysis*). Le projet présente notamment l'analyse de *Wind Chimes* de Denis Smalley réalisée

grâce à l'application standalone IAA conçue en Max (2010). Les manipulations possibles concernent :

- un zoom d'écoute permettant de sélectionner et d'écouter une partie d'un son pré-défini en filtrant son contenu sur mesure ;
- le parcours d'une taxonomie de fragments sonores significatifs dans l'œuvre, des plus brefs aux plus longs ;
- la variation des vitesses de lecture de trois fragments superposés, avec la possibilité de se recalculer sur celles adoptées par le compositeur.

Le lecteur qui souhaiterait aller plus loin, en confectionnant sa propre analyse interactive fondée sur l'écoute peut utiliser le logiciel TIAALS développé par la même équipe. Il n'est toutefois pas possible de programmer cet environnement en utilisant par exemple des scripts.

#### 3.3. Manipulations musicales génératives

Le site Javascript Systems Music<sup>2</sup> propose un apprentissage du Web Audio guidé par la reconstruction d'œuvres de Steve Reich et de Brian Eno. L'approche est progressive : avec *It's Gonna Rain* de Steve Reich, le but est de découvrir la Web Audio API en jouant des sons; *Ambient 1 : Music for Airports* de Brian Eno introduit des traitements audio tels que la réverbération; enfin *Discreet Music* toujours de Brian Eno présente la synthèse audio et le séquençement. L'utilisateur programmeur est guidé par l'idée de reconstitution. Il s'agit de générer de la musique en appliquant les principes de construction élaborés par les compositeurs, avec la possibilité de partager non plus seulement le résultat musical mais surtout le système génératif utilisable sur de très nombreux appareils.

## 4. VERS UN HARMONISEUR INTERACTIF WEB AUDIO

### 4.1. Ouvrir à la manipulation

Dans leur état actuel, les analyses Ircam présentent les modules de traitement temps réel de manière statique au niveau des mots-clés communs à toutes les pièces, donnant des définitions des procédés de synthèse ou de transformation sonore accessibles par des liens hypertextuels. Le corps de l'analyse est également statique, que ce soit le texte ou les extraits audio donnant à entendre le résultat produit.

Dans le cadre de ce projet, notre premier axe de travail sur l'appropriation de l'analyse par le lecteur concerne la mise à disposition en ligne des modules temps réels utilisés par les compositeurs dans les pièces mixtes qui sont analysées. Comme dans les projets cités dans la partie 3, nous souhaitons enrichir la compréhension en associant la manipulation et la perception du résultat, dans notre cas sonore. Contrairement au projet TaCEM et au site Javascript Systems Music, nous ne visons pas

---

<sup>2</sup> <http://teropa.info/blog/2016/07/28/Javascript-systems-music.html>

l'analyse de parties précises d'une œuvre ou leur reconstruction générative, mais bien la présentation d'un traitement sonore par sa prise en main. Nous sommes également intéressés par l'appropriation du code informatique, ici en Faust, qui sera distribué avec l'analyse.

## 4.2. Spécifications d'un module d'harmoniseur

Nous présentons ici le prototype que nous développons concernant un harmoniseur et sa mise en œuvre dans *En Echo* de Philippe Manoury pour voix et électronique en temps réel (1993). Rappelons qu'un harmoniseur est un module de traitement sonore qui permet de transposer en temps réel un son vers l'aigu ou vers le grave. La transposition peut être micro-tonale ou au contraire s'inscrire dans un travail classique d'écriture des hauteurs (maximum de 2 octaves vers l'aigu ou vers le grave).

Nous sommes partis de la version 2010 du patch Max de l'œuvre, précédemment étudié lors de travaux de recherche sur la préservation des œuvres mixtes [1]. De plus l'analyse de cette pièce pour le portail Ircam est en cours de préparation, et représente une première étape d'intégration d'un module actif.

Nous avons isolé un sous-patch qui réalise un harmoniseur (cf. figure 4) représenté sous la forme d'un système avec ses entrées, ses sorties, ses paramètres et son mode de calcul d'une transposition d'un son par effet Doppler grâce à deux retards variables avec un recouvrement ou *overlapping* d'une demi-période. Le module permet des transpositions des sons en entrée ainsi que des décalages temporels. Il comporte une entrée pour le son d'origine, une sortie pour le son transformé et un jeu de 4 paramètres de pilotage :

- l'étendue de la transposition exprimée en *midicents* (100 signifie un demi-ton vers l'aigu, -1200 un octave vers le grave);
- la largeur de la fenêtre d'effet Doppler exprimée logarithmiquement entre 0 et 127;
- l'éventuel délai à ajouter à la transposition, exprimé en millisecondes;
- le niveau de sortie exprimé logarithmiquement entre 0 et 157.

Dans un deuxième temps, nous avons traduit ce patch en langage Faust<sup>3</sup> (cf. extrait du code figure 3) pour permettre sa compilation vers de multiples cibles : logiciel standalone, traitement pour Max ou Pure Data, plugin, mais aussi vers le Web.

```
//-----//
//ELEMENTARY TRANSPOSITION BLOCK
//-----//
transpoBlock(moduleOffset, midicents, win,
extradel, amplitude)
= dopplerDelay with {
```

```
freq = midicents : +(6000) : *(0.01) :
ba.midikey2hz : -(261.625977) : *(-3.8224)
/(float(win));

adjustedPhasor = freq : pdPhasor :
+(moduleOffset) : ma.decimal;

th_trigger = (adjustedPhasor > 0.001) *
(adjustedPhasor@1 <= 0.001);
trig_win = win : ba.sAndH(th_trigger);
trig_del = extradel : ba.sAndH(th_trigger);

delayInSamples = adjustedPhasor : *(trig_win) :
+(trig_del) : *(millisec);
variableDelay = de.fdelay(262144,
delayInSamples);
cosinusEnvelop = adjustedPhasor : *(0.5) :
sinusEnvelop;

dopplerDelay = (variableDelay, cosinusEnvelop)
: * : *(amplitude);
};
```

Figure 3. Extrait du code Faust décrivant un bloc élémentaire de transposition, sans recouvrement, dans *En Echo* de Manoury.

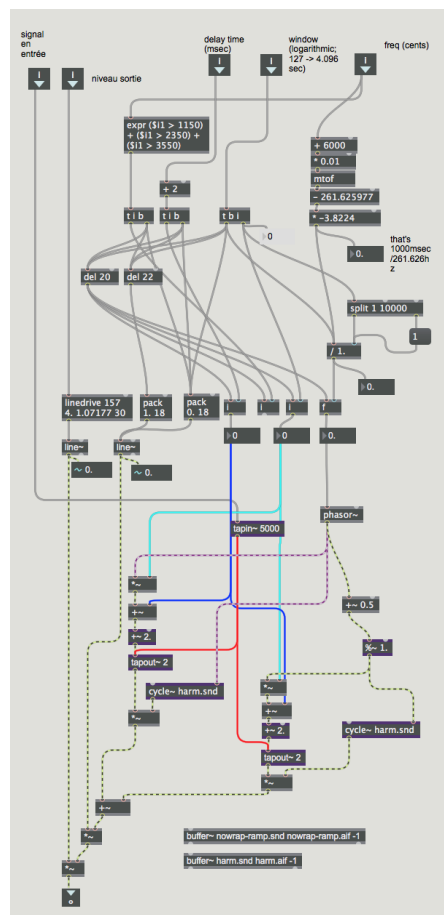


Figure 4. Copie d'écran d'un harmoniseur extrait du patch de *En Echo* de Manoury. Source : version 2010 du patch pour Max 5, communiqué aux auteurs par Serge Lemouton.

<sup>3</sup> Langage fonctionnel développé par le Grame à Lyon. <http://faust.grame.fr>

### 4.3. Interface de manipulation du module

Remarquons tout d'abord que le compositeur utilise quatre harmoniseurs en parallèle dans *En Echo*. Le code Faust complet comporte donc 8 modules de retard variable, avec 4 harmoniseurs comportant chacun 2 lignes à retard. La traduction du code informatique ne suffit pas : il est nécessaire que le lecteur de l'analyse puisse accéder aux paramètres de pilotage choisis par le compositeur et à un ou plusieurs exemples musicaux en entrée choisis dans l'œuvre. Il doit également pouvoir utiliser le microphone de son ordinateur pour produire des sons qui seront transformés par le module.

Nous avons choisi un extrait de la partition dans lequel Manoury utilise l'harmoniseur directement sur la voix pour l'épaissir (effet chorus). Il est présenté à la figure 5.



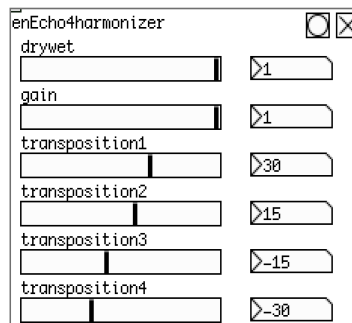
**Figure 5.** Extrait de la partition de la 4<sup>ème</sup> section « Mea Lux » de *En Echo*, chiffres 19 et 20.

Il est tout d'abord nécessaire de récupérer les valeurs des paramètres de contrôle utilisés par le compositeur à ce moment de la partition, en explorant les paramètres de contrôle du patch de l'œuvre. Les résultats obtenus sur ce passage musical pour les 4 harmoniseurs sont :

	Harmo 1	Harmo 2	Harmo 3	Harmo 4
transposition (midicents)	30	15	-15	-30
fenêtre (de 0 à 127)	50	50	50	50
retard (millisecondes)	0	0	0	0
Amplitude (de 0 à 127)	127	127	127	127

A ce niveau, nous devons choisir les paramètres de contrôle que nous ouvrons à la manipulation par le lecteur de l'analyse. Les paramètres d'amplitude et de largeur de fenêtre ne sont pas déterminants ici, nous choisissons donc de les fixer dans le code. En revanche, les transpositions en *midicents* constituent des moyens de contrôles simples à comprendre et faire varier. Nous choisissons de les initialiser dans la configuration prévue par le compositeur et de permettre à l'utilisateur de les varier dans un ambitus de +/-1 demi-ton (de -100 à +100

midicents). Pour des raisons de facilité de manipulation, nous ajoutons à ces 4 contrôles celui du gain général du module et celui du *drywet* variant entre 0 (son d'origine non transposé) à 1 (uniquement le son transposé, plus de son d'origine). La figure 6 montre un exemple de génération de *canvas* et traitement sous Pure Data à partir du code Faust.

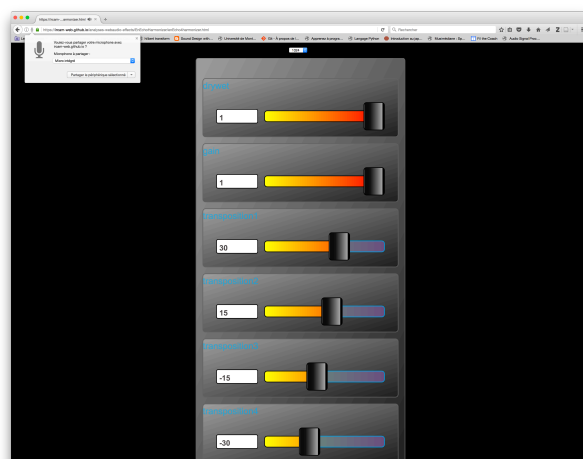


**Figure 6.** Exemple de *canvas* et traitement généré sous Pure Data à partir du code Faust.

A partir de ce même code Faust, nous avons ensuite déployé une première interface expérimentale utilisant le Web Audio en utilisant le compilateur *faust2webaudioasm* fondé sur *asm.js*, un moteur Javascript optimisé. La démonstration fonctionnelle est disponible à l'URL suivante : <https://ircam-web.github.io/analyses-webaudio-effects/EnEchoHarmonizer/enEcho4harmonizer.html>

L'ensemble du code est accessible sur le dépôt github : <https://github.com/Ircam-Web/analyses-webaudio-effects>

La figure 7 montre une copie d'écran de l'interface du module Web Audio développé, avec le menu de validation de l'entrée audio sélectionnée. Dans un second temps, nous personnalisons l'interface, ici générée automatiquement avec l'outil *faust2webaudioasm*.



**Figure 7.** Copie d'écran de la fenêtre du module Web Audio d'harmoniseur de *En Echo*.

## 5. CONCLUSION

Nous avons situé l'analyse des œuvres mixtes dans le paysage en expansion de l'analyse musicale en général, notamment des œuvres faisant appel aux procédés de l'informatique musicale. Le projet Analyses à l'Ircam propose ainsi un portail d'analyses musicales en ligne d'œuvres faisant appel aux technologies. Nous avons montré des exemples d'applications pédagogiques fondées sur l'apprentissage par la manipulation en lui associant souvent la découverte de la programmation, notamment avec le langage Javascript. Enfin, nous avons développé un module Faust d'harmoniseur issu d'une modélisation d'une partie du patch de *En Echo* de Manoury, puis extrait les paramètres significatifs de son contrôle. La version Web Audio de ce module offre les avantages de cet environnement, comme l'absence d'installation spécifique (un navigateur suffit) ou la mise à disposition d'un objet à part entière que l'on s'approprie et que l'on réutilise. A ces points positifs s'ajoutent ceux apportés par Faust, qui propose un code efficace, multi-cibles, multi-plateformes.

Nous envisageons donc de tester ce dispositif dans l'analyse de *En Echo* de Manoury destinée au portail Ircam. Nous construirons ensuite une bibliothèque de modules avec leurs interfaces d'appropriation (fichiers sons du compositeur, possibilité de tester avec le microphone de l'ordinateur support, manipulation des paramètres de contrôle), qui seront présents à la fois dans le thésaurus d'informatique musicale que nous élaborons et dans les diverses analyses. Si cette proposition concerne dans un premier temps une œuvre mixte temps réel, elle nous semble être généralisable à d'autres œuvres fondées sur des codes informatiques.

Une autre piste de travail portera sur la conception et la réalisation de partitions augmentées : actuellement présentées sous la forme d'extraits de la partition instrumentale sous la forme d'images fixes (non vectorielles) avec parfois un curseur de suivi de l'extrait musical, les partitions d'analyse en ligne doivent évoluer non seulement dans leurs formats techniques (formats vectoriels) mais aussi dans le choix des informations proposées au lecteur et le degré d'interaction avec elles. Nous réfléchissons à des conteneurs temporels interactifs qui ouvriraient à ces possibilités.

A terme la présentation des analyses des œuvres avec technologie devrait s'appuyer sur un ensemble de modules jouables et de conteneurs temporels, auxquels seront reliés les différents éléments documentant l'œuvre.

## 6. REFERENCES

- [1] Bonardi, A., Pérenniser pour transmettre, transmettre pour pérenniser - Destins de l'œuvre mixte interactive - Autour de *En Echo*, pièce de Philippe Manoury. In *Musique et Technologie - Préserver, archiver, re-produire*, Portraits polychromes, hors-série thématique, Paris : Institut National de l'audiovisuel, pages 105-126, 2013.
- [2] Clarke, Michael (2012) *Analysing Electroacoustic Music: an Interactive Aural Approach*. Music Analysis, 31 (3). pp. 347-380. ISSN 0262-5245.
- [3] Delalande, F., *Analyser la musique, pourquoi, comment ?*, Paris : INA Editions, 2013, 248 pages.
- [4] Lemouton, S., Ciavarella, R., Bonardi, A., Peut-on envisager une organologie des traitements sonores temps réel, instruments virtuels de l'informatique musicale ? *Actes de la Cinquième Conférence de Musicologie Interdisciplinaire (CIM'09)*, pages 118-121, Paris, octobre 2009.
- [5] Pachet, F., *Active Listening : What is in the Air?* Sony CSL Internal Report, 1999. Disponible à <https://www.csl.sony.fr/downloads/papers/uploads/pachet-99a.pdf> [URL vérifiée le 29/12/2016].
- [6] Schnell, N., Battier, M., Introducing Composed Instruments, Technical and Musicological Implications, *Actes de la Conférence New Interfaces for Music Expression 2002 (NIME02)*, Paris, 2002.