



HAL
open science

Pérenniser pour transmettre, transmettre pour pérenniser - Destins de l'œuvre mixte interactive

Alain Bonardi

► To cite this version:

Alain Bonardi. Pérenniser pour transmettre, transmettre pour pérenniser - Destins de l'œuvre mixte interactive: Autour de En Echo, de Philippe Manoury. Institut National de l'audiovisuel; Evelyne Gayou. Musique et Technologie - Préserver, archiver, re-produire, Portraits polychromes, hors-série thématique, pp.105-126, 2013. hal-01106912

HAL Id: hal-01106912

<https://hal.science/hal-01106912>

Submitted on 20 Jan 2015

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.

Pérenniser pour transmettre, transmettre pour pérenniser

Destins de l'œuvre mixte interactive – Autour de *En Echo* de Philippe Manoury

Alain Bonardi

CICM – EA 1572 – Université Paris 8
2, rue de la Liberté
93526 Saint-Denis Cedex 02
et
IRCAM
1, place Igor-Stravinsky
75004 Paris

alain.bonardi@ircam.fr

Résumé

Dans cet article, nous rendons compte d'une double démarche autour de l'œuvre *En Echo* de Philippe Manoury pour soprano et électronique temps réel. D'une part, nous avons dans un souci de pérennisation traduit en FAUST un certain nombre de modules du patch Max/MSP, puis généré les documentations mathématiques de ces transformations, et enfin validé artistiquement la démarche. D'autre part, dans un souci de transmission, nous avons élaboré des représentations dynamiques explicitant le fonctionnement du « moteur électronique » de l'œuvre.

Contextes de la création musicale audio-numérique

Les nouvelles conditions de possibilité de la musique devenue numérique

La production sonore et musicale repose par principe sur des outils techniques, ce qui l'oblige à évoluer à chaque révolution technologique. Ainsi, le numérique a reconfiguré tant les techniques que les usages de création sonore et musicale. Que ce soit au niveau des instituts de recherche ou des pratiques grand public, les pratiques musicales et les technologies sont complètement imbriquées, les premières utilisant ou détournant les secondes.

Considérons par exemple l'une des évolutions majeures de l'IRCAM (Institut de Recherche et de Création Acoustique/Musique, Paris) à la fin des années 1980, autour de ce que l'on nomme le « temps réel » : ce développement ne peut se comprendre sans analyser la boucle créative née entre le compositeur Philippe Manoury¹ et le chercheur Miller Puckette. En 1986, ce dernier intègre les premiers modules de suivi de hauteurs et de partitions créés par Barry Vercoe² dans une esquisse de « moteur musical » temps réel sans interface utilisateur, qui constituera plus tard la base du logiciel Max. En 1987, le compositeur valide cette architecture en l'utilisant dans *Jupiter*, première pièce

¹ Manoury, P., *La note et le son*, Paris : l'Harmattan, 1998.

² Logiciel intitulé *The Synthetic Performer*, développé par Barry Vercoe et son équipe en 1984-1985 à l'IRCAM.

interactive temps réel, associant une flûte Midi à un ordinateur. En 1988, Miller Puckette donne naissance à une première interface de contrôle de Max sur Mac. Au cours de cette même année, Manoury crée *Pluton*, pour piano et ordinateur, assimilant directement les dernières innovations du logiciel. En 1991, la pièce *Neptune* pour deux vibraphones Midi, grand tam-tam, marimba et informatique temps réel permet d'établir la nouvelle station de travail ISPW (Ircam Signal Processing Workstation). Un quart de siècle plus tard, ces œuvres de Manoury sont toujours données, mais elles ont été soumises à de nombreuses adaptations au fil des changements de technologie.

Dans le domaine des pratiques grand public, la dépendance aux conditions de possibilité technologiques n'est pas moins grande. De nouveaux usages apparaissent, associant des interfaces informatiques comme les tablettes ou les systèmes de capteurs issus du monde du jeu vidéo à une pratique uniquement gestuelle, ne passant plus par les catégories traditionnelles de la musique³.

Quand l'obsolescence technologique met les œuvres en danger

Parmi les conséquences des nombreuses transformations entraînées par le numérique, l'une des plus préoccupantes concerne la conservation du patrimoine musical contemporain qui est menacé : certaines œuvres ont déjà disparu, faute de représentations permettant de les rejouer tout en s'abstrayant des technologies dont l'obsolescence se fait de plus en plus rapide. Notons que certaines pratiques ont pris en compte ce phénomène en se présentant elles même comme éphémères ; ainsi l'improvisation faisant appel au live-coding⁴ se définit par nature comme devant être vécue dans l'instant, sans pérennité revendiquée.

Nous nous intéressons ici plus particulièrement aux œuvres mixtes interactives, c'est-à-dire à des pièces associant musiciens et machines aussi bien dans l'écriture et la conception que le jeu en concert. La difficulté de leur pérennisation se situe à plusieurs niveaux :

- Ces œuvres comportent un grand nombre de sons d'origine électronique, obtenus soit par transformation de ceux produits par les instrumentistes, soit par synthèse. Les modules permettant leur production, tels que ceux développés dans l'environnement Max/MSP sont soumis à une forte obsolescence du fait de l'évolution des systèmes d'exploitation et des logiciels, et des aléas de la maintenance des composants tiers créés par la communauté des utilisateurs.
- Ces sons sont en général modulés par le jeu des musiciens : il ne suffit donc pas de les enregistrer et de les stocker, car ils seront différents d'une exécution à l'autre. Il faut donc parvenir à des représentations abstraites décrivant leur mode de production et la nature des interactions potentielles.

³ Bonardi, A., « Quand émergent des pratiques musicales a-musicologiques », *Musik – Kontext – Wissenschaft / Musique – Contexte – Savoirs, Perspectives interdisciplinaires sur la musique*, Talia Bachir-Loopuyt, Sara Iglesias, Anna Langenbruch, Gesa zur Nieden (éds), Frankfurt am Main : Peter Lang, 2012, pages 71-81, ISBN 978-3-631-60652-0.

⁴ Brown, A.-R., Code Jamming, *Journal of Media and Culture*, vol. 9, issue 6, december 2006.

L'exemple de *Diadèmes* de Dalbavie incarne cette délicate situation⁵ : il s'agit d'une pièce en trois mouvements pour alto, ensemble instrumental et électronique temps réel créée à l'IRCAM le 16 juin 1986 (Centre Georges Pompidou, Paris, Nathalie Baudoin (alto), Ensemble Itinéraire, direction : Arturo Tamayo). Cette œuvre a été jouée pour la dernière fois en 1992 à l'occasion de l'enregistrement du CD (Adès 205 202 (1996, recorded 1992) Accord 465 313-2 (1996)). Suite à la demande d'un organisateur de concert et de l'éditeur (Lemoine) concernant la création de cette pièce aux États-Unis en décembre 2008 ont été recensés les problèmes posés par l'exécution en concert de la partie technologique de cette œuvre (« âgée » de 23 ans). La situation est pour le moins difficile : les synthétiseurs Yamaha TX 816 utilisés alors ne sont plus fabriqués, ceux que possède l'IRCAM sont dans un état proche de la rupture ; le compositeur a essayé plusieurs solutions d'émulation logicielle, sans qu'aucune ne lui donne satisfaction au niveau du résultat sonore.

En avril 2008, le réalisateur en informatique musicale Serge Lemouton a conçu une nouvelle version de la partie électronique de l'œuvre. Il a choisi de faire « revivre » ces synthétiseurs et les claviers qui les déclenchent en constituant ce que l'on appelle un *sampler*. Il s'agit d'une base de données de sons d'un instrument, correspondant à son enregistrement à différentes hauteurs jouées avec différentes intensités sonores. En l'occurrence ont été enregistrées sur le « vieux » TX 816 (fonctionnant partiellement) les notes demandées par le compositeur dans la partition, avec différents programmes (c'est-à-dire instruments ou sons générés) et différentes vitesses (correspondant à la force d'appui sur la touche du clavier).

Cette situation est tout à fait nouvelle : ces œuvres ne peuvent plus s'appuyer sur les paradigmes culturels stables de ce que nous appelons au sens large la « musique classique occidentale », tels que la notation et l'organisation en familles instrumentales. Si l'on peut parler de recul de l'universalité d'abord en ce qui concerne l'écriture musicale, qui fait face à l'explosion des notations voire à l'émergence de systèmes musicaux sans notation, le phénomène s'étend désormais à la lutherie. Chaque œuvre devient un système en soi.

Ces nouvelles modalités d'œuvres sont d'une certaine manière pénalisées en termes de diffusion. En effet, les organisateurs de concerts hésitent à programmer des pièces telles que *Diadèmes*, pour des raisons techniques. Des exécutions récentes de cette œuvre ont été annulées. Tout un pan de répertoire de la musique électronique est menacé, les œuvres de cette époque sont en péril.

Actions et programmes de pérennisation

Aucune méthode parfaite ne permet actuellement de préserver en l'état une création numérique ; nous n'abordons pas ici l'approche archivistique, plus proche de la conservation, dont l'enregistrement fait partie. Quatre procédés de préservation se dégagent et sont utilisés, parfois combinés^{6, 7, 8}.

⁵ 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.

⁶ Borghoff U.-M., Rodig, P., Scheffcyk, J., Schmitz, L., *Long-Term Preservation of Digital Documents: Principles and Practices*. Springer-Verlag New York Inc, 2006.

⁷ Gladney, H., *Preserving digital information*, Springer-Verlag New York Inc, 2007.

- La sauvegarde muséale : il s'agit de conserver en état de fonctionnement des outils utilisés lors de la création. Cette approche est à proscrire car elle ne fait que retarder la disparition de l'œuvre sans permettre de mise à jour ; mais adossée à un modèle d'archivage (comme celui du projet Mustica⁹), elle offre déjà un accès aux informations et contextes d'exécution.
- L'émulation : elle consiste à simuler sur les environnements disponibles à un moment les outils de lecture des formats d'origine. Séduisante en théorie, car faisant l'impasse sur l'évolution, elle est fragile car une émulation n'est pas toujours parfaite. Nicola Bernardini et ses collaborateurs¹⁰ au centre Tempo Reale citent l'exemple d'*Oktophonie* de Stockhausen, qui nécessite pour son exécution un ordinateur ATARI-1040 ST qui n'existe plus (sauf chez les collectionneurs). Il existe des émulateurs de cet ordinateur sur d'autres machines, mais personne ne sait si le programme *Notator* utilisé par Stockhausen à l'époque fonctionne sur un émulateur. Les communautés d'utilisateurs/collectionneurs d'ATARI sur le Web pourraient être d'une grande utilité dans ces démarches d'expérimentation pour remonter l'œuvre.
- La migration : il s'agit de mettre à jour une ancienne version d'une pièce pour la garder adaptée aux technologies du moment. Les migrations privilégient en général des formats ouverts supposés plus pérennes¹¹. Cette activité est surtout répandue lors d'une adaptation d'une œuvre dans le but de la rejouer. De nombreux compositeurs ont eu leurs œuvres portées d'un environnement technique à un autre. Toutes les institutions dans le champ des arts électroniques font face à des nécessités de migration. À l'IRCAM, les pièces importantes utilisant des ordinateurs Next ont été migrées vers des ordinateurs Macintosh à la fin des années 1990. Avec des résultats jugés de manière très variable selon les compositeurs¹² : la même procédure de migration a ainsi abouti pour certains à un résultat trop proche de l'original (et du coup ne tirant pas assez parti des nouvelles technologies mises en œuvre) ; d'autres le trouvent trop éloigné, n'ayant pas assez à voir avec l'original.
- La virtualisation : elle consiste à rendre les processus musicaux informatiques indépendants d'une plateforme en particulier. Mais il n'est pas facile de se défaire de toutes les dépendances nécessaires au bon fonctionnement de l'exécution de programmes temps-réel. Comme nous le montrerons en détail dans cet article, *En*

⁸ Bonardi, A., Barthélemy, J., The preservation, emulation, migration, and virtualization of live electronics for performing arts : An overview of musical and technical issues. *ACM J. Comput. Cultur. Heritage*, 1(1), 16, 2008.

⁹ Bachimont, B., Blanchette, J.-F., Gerszo, A., Swetland, A., Lescurieux, O., Morizet-Mahoudeaux, P., Donin, N., Teasley, J., Preserving Interactive Digital Music: A Report on the MUSTICA Research Initiative. In *Proceedings of the Third International Conference on WEB Delivering of Music (WEB'03)*, Leeds, England, 2003.

¹⁰ Bernardini, N., Vidolin, A., Sustainable live electro-acoustic music. *Actes de l'International Sound and Music Computing Conference*, Salerno, Italy, 2005.

¹¹ Bullock, J., Caccioli, L., Modernising live electronics technology in the works of Jonathan Harvey. *Actes de l'International Computer Music Conference*, Barcelona, Spain, 2005.

¹² Selon un témoignage personnel du réalisateur en informatique musicale Serge Lemouton.

Echo, de Philippe Manoury a été en partie virtualisée dans le cadre du projet ASTREE.

De nombreux projets, au niveau européen ou français se sont saisis de la question de la pérennisation des œuvres musicales avec technologie. Le projet INTEGRA¹³ a développé une ingénierie logicielle du patch (essentiellement PureData) et proposé dans cette perspective des migrations d'œuvres contemporaines vers des solutions Open Source. Le projet CASPAR¹⁴ s'est penché sur l'implémentation du modèle OAIS¹⁵ auprès de plusieurs communautés, dont le champ de la création numérique. Dans les récents projets français soutenus par l'Agence Nationale de la Recherche, GAMELAN¹⁶ se concentre sur la modélisation des processus de production audio-numérique, permettant de capturer des informations à la source puis d'accompagner au mieux les migrations. Enfin, le projet ASTREE, que nous détaillons ci-dessous, s'intéresse à la pérennisation des traitements temps réel réalisés dans des environnements comme Max/MSP ou PureData grâce au langage FAUST développé par le Grame.

Dans cet article, nous choisissons comme œuvre-support *En Echo* de Philippe Manoury, écrite pour soprano et électronique temps réel. Elle nous servira de fil rouge : nous présenterons les actions de pérennisation menées sur cette œuvre dans le cadre du projet ASTREE. Puis, nous élargirons la question à celle de la transmission, prenant toute la mesure de la portée de la démarche entreprise dans le projet ASTREE, qui ne propose rien de moins qu'une forme de « dé-numérisation » selon le mot de Karim Barkati¹⁷. Enfin, nous présenterons trois formes de représentation du « moteur électronique » de l'œuvre, tentant d'explicitier son fonctionnement.

Comment pérenniser *En Echo* de Manoury ?

Présentation du projet ASTREE

Le projet ASTREE (Analyse et Synthèse de Traitements Temps Réel) a été mené de 2008 à 2011 avec le soutien financier de l'Agence Nationale de la Recherche. Il rassemblait l'IRCAM, le GRAME, l'Ecole des Mines de Paris et le laboratoire CIEREC de l'Université de Saint-Etienne.

Le projet était centré autour du langage fonctionnel FAUST développé par Yann Orlarey au GRAME, permettant une expression concise, proche du langage mathématique de processus synchrones de synthèse et de traitement du signal musical. Ce langage est fondé sur une algèbre de composition de blocs-diagrammes. La figure 1 ci-dessous montre les quatre aspects du projet autour du noyau central que constitue le langage FAUST :

- Dans un premier temps, il s'agit de transformer l'existant en FAUST, que ce soit des patches Max/MSP ou PureData. Un traducteur automatique permettant la conversion sous certaines hypothèses de Max/MSP vers FAUST a été élaboré.

¹³ <http://www.integralive.org>

¹⁴ <http://www.casparpreserves.eu>

¹⁵ notamment l'extraction des informations de représentation et de provenance.

¹⁶ <http://www.gamelan-projet.fr/co/accueil.html>

¹⁷ Barkati, K., Orlarey, Y., Auto-documentation mathématique pour le traitement du signal avec Faust. In *Actes des Journées d'informatique musicale*, Saint-Etienne, 2011.

- La re-synthèse constitue le deuxième enjeu : ayant codé un processeur de signaux en FAUST, est-il possible de générer un objet Max/MSP ou PureData à partir de ce code ? Les facilités de déploiement de FAUST sur de multiples plateformes et environnements offrent cette re-synthèse, ce qui ouvre la porte à la comparaison dans le cadre du logiciel de départ de l'original et de sa version traduite.
- La génération automatique d'une documentation mathématique constitue un point important du projet. L'environnement FAUST produit pour chaque module synchrone traduit une documentation mathématique donnant le code et l'explicitant. C'est à partir de cette documentation que devrait travailler toute personne souhaitant dans l'avenir remettre activé un module de traitement avec les moyens informatiques qui seront disponibles. Il serait assez logique que la description de ces modules soit imprimée et jointe à la partition pour être ensuite archivée en médiathèque dans des conditions idéales de préservation du papier.
- Enfin, le volet 'data mining' vise une première approche, encore modeste, de classification de traitements écrits en FAUST pour lesquels une première recherche d'invariants a été menée.

C'est donc dans le cadre de ce projet qu'a été mené un long travail de traduction en FAUST de modules de *En Echo* de Philippe Manoury, puis leur intégration dans une nouvelle version du patch, avant de valider la démarche en studio puis en concert.

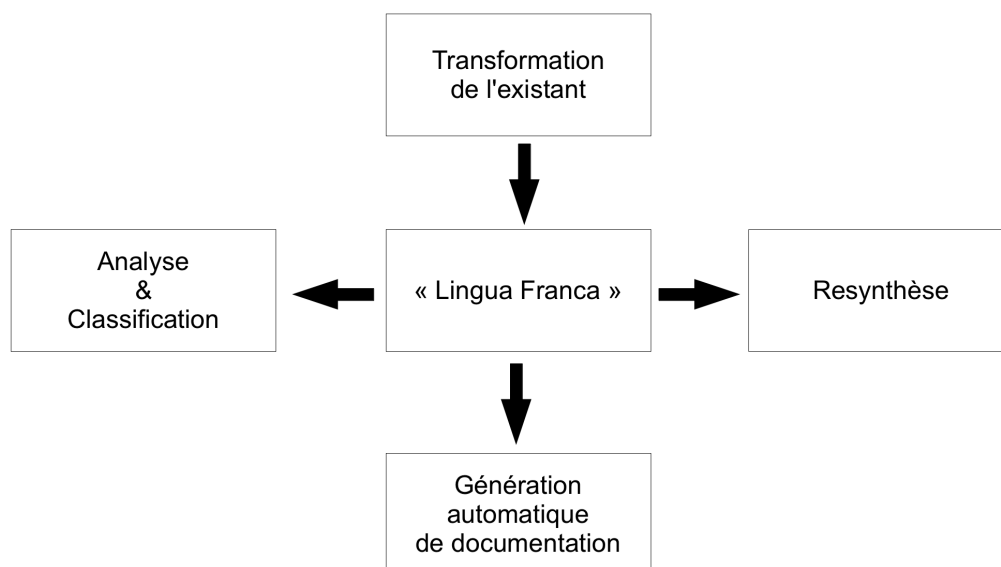


Figure 1. Schéma de principe du projet ASTREE (*Analyse et Synthèse de Traitements Temps Réel*).

Présentation de En Echo de Manoury

Composée en 1993-1994 pour soprano et électronique temps réel¹⁸, sur des matériaux poétiques d'Emmanuel Hocquart, *En Echo* représentait à l'époque pour le compositeur un nouvel enjeu, autour de la question de la voix, de la captation d'un signal sonore extrêmement complexe et variable – en tout cas plus que les instruments – et de sa synchronisation avec un ordinateur.

Dans les notes de programme de la création, le compositeur définit ainsi les interactions voix/machine : « *dans certains cas, le but consiste à pouvoir synchroniser le processeur sur une simple voix parlée, faisant fi de toute connotation de hauteur musicale, dans d'autres, de détecter certains traits caractéristiques (tels que les fricatives ou sons bruités) de l'émission vocale, dans d'autres encore, c'est la mélodie qui se trouve au centre de l'intérêt, etc. Les différentes catégories sont tour à tour sollicitées pour établir une relation sensible entre la voix et la machine. C'est cette polyphonie de comportements qu'il nous importe de détecter.* » Le compositeur a fait le choix d'un suiveur automatique, capable de synchroniser les modules de traitement par rapport aux événements détectés dans la ligne vocale.

Du côté de l'électronique, Manoury fait appel à un patch Max/MSP. L'ensemble du dispositif s'ordonne en modules que nous pouvons classer ainsi :

- Transformation sonore
 - Générateur de bruit (noise)
 - Harmoniseur
- Synthèse sonore
 - Phase-aligned formants ou PAFs¹⁹
- Diffusion/spatialisation
 - Spatialisateur mobile quadriphonique
 - Spatialisateur fixe quadriphonique

La figure 2 est une copie d'écran de la partie centrale du patch Max/MSP, de ce que nous appellerons l'« orchestre numérique » de *En Echo*. Ce n'est pas là une simple figure de style : l'expression « orchestre numérique » trouve ici son sens, car le soutien harmonique de la voix est confié à l'électronique, plus particulièrement aux synthétiseurs formantiques PAFs. De plus, le compositeur a investi l'électronique d'une fonction spatiale qui nous rapproche d'une acoustique orchestrale avec ses différents plans. D'une certaine manière l'œuvre se situe dans le fil de la mélodie pour voix et orchestre.

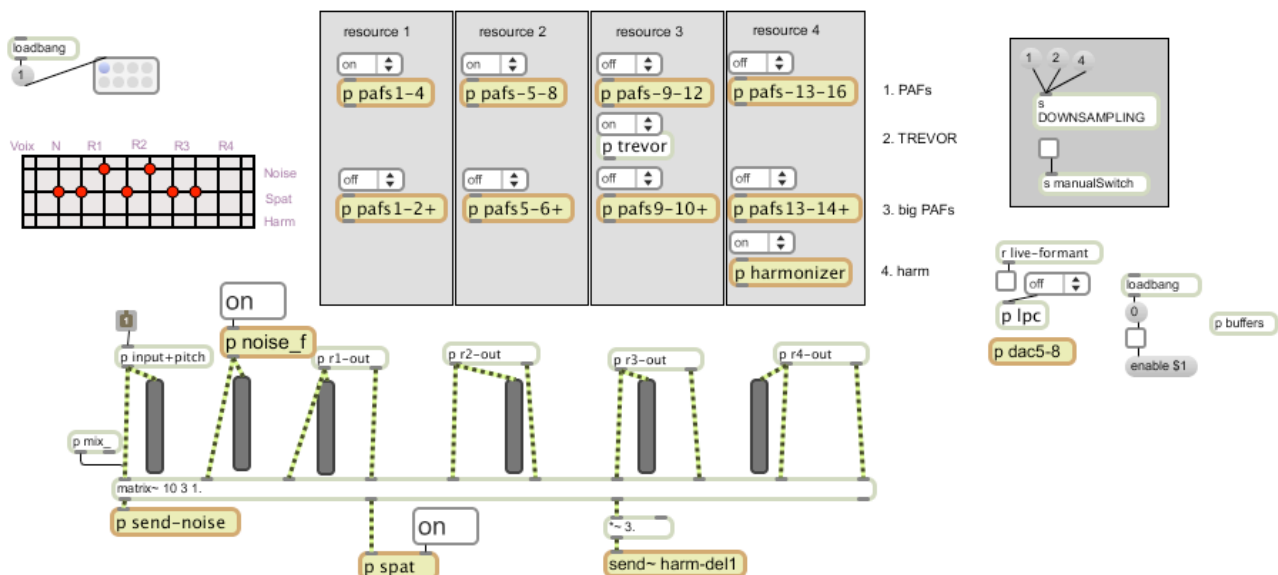


Figure 2. La partie centrale du patch Max/MSP de *En Echo*, présentant l'ensemble des modules utilisés.

Actions de virtualisation dans *En Echo*

Constitution d'une version de référence

Cette première étape paraît évidente, mais ce n'est pas toujours le cas : il s'agit de poser une version de référence qui servira de repère pour toute la suite de la démarche de pérennisation, et de s'assurer que l'on sait en indiquer tous les composants, en l'occurrence logiciels (et accessoirement connaître leur état d'obsolescence actuel).

Dans le cas de *En Echo*, le point de départ du travail entrepris est la version de l'œuvre jouée en avril 2006 à la Maroquinerie à Paris par la soprano Kristina Vahrenkamp. Nommée 2006B, cette version du patch est conçue pour processeur PowerPC sur Macintosh et Max/MSP 4. A la demande des organisateurs du festival de Huddersfield en Angleterre en 2010, Serge Lemouton, réalisateur en informatique musicale à l'IRCAM, a recompilé pour plateforme Intel des objets dont il possédait le code tels que le suiveur de partition **scr-f9** ou le module de reconnaissance de formants vocaux **enecho.formant**. D'autre part, les nouvelles versions, désormais compatibles avec les processeurs Intel, des objets créés par des tiers comme **fiddle~** pour la reconnaissance de la hauteur chantée par la voix (créé par Miller Puckette) ont été incluses. Tout ceci a conduit à une version dite « version 2010 », que Serge Lemouton a mis à disposition des organisateurs anglais ; il n'a malheureusement jamais eu de retour : on ignore donc si cette version 2010 a connu ou pas les feux de la rampe.

La première phase du travail mené a consisté à évaluer cette version 2010 sur processeur Intel et Max/MSP version 5, en utilisant un deuxième ordinateur, plus ancien, tournant sur PowerPC sur lequel fonctionnait la version 2006B, et ainsi d'établir des comparaisons. La version PureData de l'œuvre (surtout utilisée dans les concerts aux Etats-Unis) a également été utilisée. Le fonctionnement « correct » de la version 2010 a été établi : bien qu'elle n'ait peut-être jamais été donnée à Huddersfield²⁰, l'écoute comparative des parties électroniques modulées par l'enregistrement de la partie vocale (voix de Donatienne Michel-Dansac, soprano) ne permet pas de déceler de différence, ce qui confère une forme d'authenticité à cette version 2010.

Travaux menés dans *En Echo*

Les modules qui ont été virtualisés et traduits en FAUST sont :

- Du côté de la transformation sonore : le *générateur de bruit* (noise) et l'*harmoniseur* (harmonizer).
- Du côté de la synthèse sonore : les *PAFs*.
- Du côté de la diffusion sonore : le *spatialisateur mobile* (spat) et le *spatialisateur fixe* (dac5-8).

Pour chacun de ces modules, une préparation a été effectuée : le module a été isolé, son patch d'origine ré-ordonné et commenté pour plus de lisibilité, et un test sonore effectué pour s'assurer que ces opérations n'ont pas affecté le résultat sonore produit. Puis, le code FAUST correspondant a été écrit et compilé, générant un objet Max/MSP (extension .mxo), ainsi que la documentation mathématique automatique associée.

²⁰ En tout cas, le compositeur ne l'a jamais entendue ni validée.

Enfin, le résultat a été intégré dans le patch de *En Echo*, avec la possibilité d'un aiguillage entre l'ancienne version Max et la version FAUST, permettant des comparaisons à la volée en situation. La copie d'écran de la figure 3 montre l'implémentation du générateur de bruit, avec ses deux versions : la version Max/MSP préparée (à droite, *e_cheap_noise_a_faustiser.maxpat*) et la version à partir du code FAUST (à gauche, *e_cheap_noise_faustise.maxpat*, comprenant en fait un seul objet .mxo).

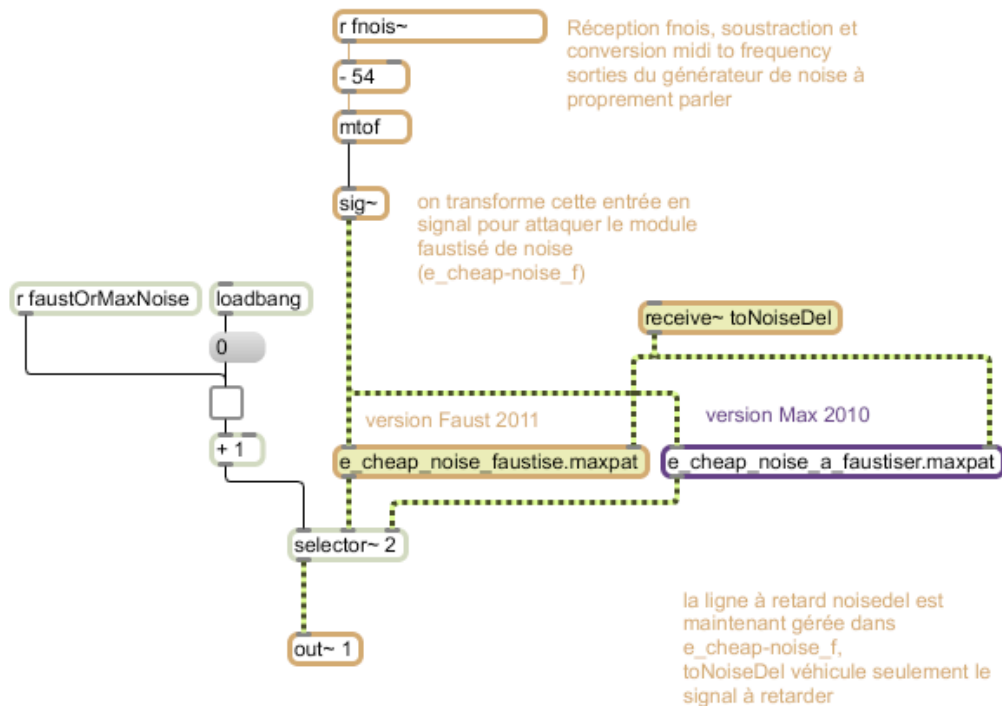


Figure 3. Les deux versions du générateur de bruit (à droite en mauve, version Max 2010 ; à gauche en jaune et marron, version générée à partir du code Faust).

Montrons sur l'exemple de ce générateur de bruit le détail des travaux menés. La figure 4 donne le patch isolé correspondant à la version Max/MSP préparée (*e_cheap_noise_a_faustiser.maxpat*). Pour l'obtenir, nous avons ré-agencé les entrées et sorties, supprimé les abstractions (issues de la bibliothèque Jimmies) du type *delread~*²¹ au profit de *tapout~* pour en faire un module traduisible en FAUST.

Pour établir la traduction FAUST, nous avons commencé par les objets de base tels que *cycle~*²², *phasor~*²³, *tapin~* et *tapout~*²⁴. Alors que le problème paraissait relativement simple à résoudre, il n'en a rien été. En effet, à côté d'un comportement générique bien décrit dans les ouvrages de référence en traitement du signal, les objets possèdent des caractéristiques propres à Max/MSP (notamment au niveau des interpolations) qui ne sont pas documentées. Le code Faust du traitement générique a donc dû être modifié, pour correspondre à ce que Max/MSP produit vraiment (dans sa version 5). Nous avons procédé par « reverse-engineering ». De ce fait, nous avons constitué sur ces objets une documentation explicite des procédés de Max/MSP 5. Il faut noter ici que les évolutions

²¹ Ligne à retard.

²² Oscillateur sinusoïdal.

²³ Produit un signal ascendant en dents de scie, compris entre 0 et 1.

²⁴ *tapin~* et *tapout~* sont les composants d'une ligne à retard. *tapin~* constitue l'espace de stockage, pouvant contenir une certaine quantité d'échantillons sonores. *tapout~* vient lire dans cet espace de stockage, avec un certain retard.

du logiciel Max/MSP (et notamment la nouvelle version Max/MSP 6) entraîneront peut-être des résultats sonores différents. Mais le code Faust permettra de reproduire en le comprenant le fonctionnement de Max/MSP 5.

Les problèmes rencontrés sont les suivants :

- **Phasor~** : comprendre et modéliser la manière dont Max/MSP gère les évolutions de fréquences ainsi que les fréquences négatives.
- **Cycle~** : comprendre deux des modes d'utilisation de l'oscillateur dans Max/MSP, soit avec 512 valeurs, soit avec 513 pour éviter des transitions abruptes lors du retour à la première valeur. De plus, nous avons dû déterminer la manière dont Max/MSP interpole les valeurs données en sortie à partir de la lecture dans la table.
- Ligne à retard **tapin~/tapout~** : comprendre l'interpolation utilisée dans Max/MSP lorsque le retard demandé n'est pas un multiple de la période d'échantillonnage (interpolation)

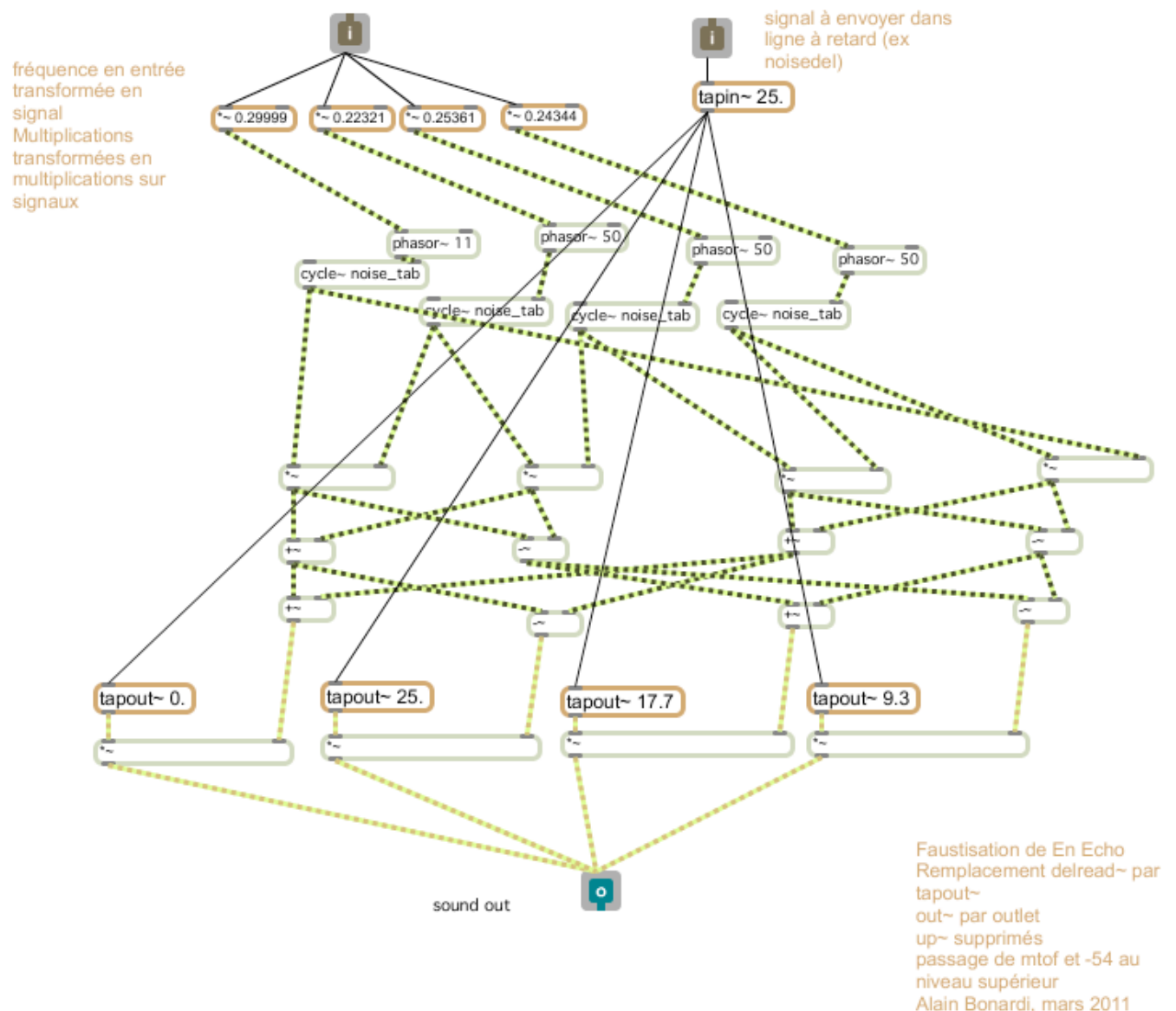


Figure 4. Le module générateur de bruit (noise) de En Echo, ré-agencé avant traduction en FAUST.

Par exemple, voici le code FAUST du *phasor~*, qui doit s'adapter à des fréquences négatives (qui provoquent un renversement des dents de scie devenant décroissantes) :

```
basicPhasor(freq) = (_,1. : fmod) ~ +(float(freq)/float(SR));
myPhasor(freq) = (if(freq>0', basicPhasor(freq), 1.+basicPhasor(freq)));
process = myPhasor;
```

De la même manière, nous avons obtenu la description de l'oscillateur *cycle~* et de la ligne à retard *tapin~/tapout~*.

Le code d'ensemble du générateur de bruit est donné ainsi :

```
//cycle_noise0 décrit l'objet cycle
//myPhasor décrit l'objet phasor~
//variableDelay décrit l'ensemble tapin~/tapout~

d0(x) = variableDelay (x, 0.);
d1(x) = variableDelay (x, 25.);
d2(x) = variableDelay (x, 17.7);
d3(x) = variableDelay (x, 9.3);

eCheapNoise(inlet0, inlet1) = myNoise
with {
//
//On definit 3 permutations agissant sur les 8 signaux arrivant ensemble//
//
p0(a,b,c,d,e,f,g,h) = (a,c,d,e,f,g,h,b);
p1(a,b,c,d,e,f,g,h) = (a,c,b,d,e,g,f,h);
p2(a,b,c,d,e,f,g,h) = (a,e,b,f,c,g,d,h);
//
//Equation du processeur de signaux obtenue
//
myNoise = ((inlet0<:_:_:_),0.29999, 0.22321, 0.25361,
0.24344):p2:(*,*,*,*):(myPhasor,myPhasor,myPhasor,myPhasor):((cycle_noise0<:_:_), (cycle_noise0<:_:_),
(cycle_noise0<:_:_), (cycle_noise0<:_:_)):p0:((*<:_:_),(*<:_:_),(*<:_:_),(*<:_:_)):p1:((+<:_:_),(-
<:_:_),(+<:_:_),(-<:_:_)):p2:(+,+,+,+, (inlet1<:d0,d1,d2,d3)):p2:(*,*,*,*):(+,+):+;
};
process = eCheapNoise;
```

Validation et mise en perspective des travaux de pérennisation

La nouvelle configuration de l'œuvre avec ses modules décrits en FAUST a été validée de deux manières :

- Lors d'une rencontre avec le compositeur Philippe Manoury en juillet 2011 : pour ce dernier, il n'y avait pas de problème d'authenticité.
- En donnant l'œuvre en concert à trois reprises, respectivement en mai 2011 (Journées d'Informatique Musicale, Saint-Etienne), août 2011 (Atelier Musical de Touraine, à la Carrière du Normandoux), octobre 2011 (Atelier Musical de Touraine, Tours). Citons les retours faits par Jean-Claude Risset et John Chowning, présents à Saint-Etienne,, qui connaissent l'œuvre depuis longtemps : ils ont trouvé le son d'ensemble plus clair et plus précis que lors des auditions précédentes (notamment lors d'exécutions aux Etats-Unis auxquelles ils avaient assisté).

Ce travail contribue à l'actualisation de l'œuvre et à sa pérennisation. Il ne s'est pas fait sans difficultés. L'environnement Max/MSP ne dit pas tout sur les caractéristiques de ses objets et ses propres règles de fonctionnement, ce qui oblige à procéder par « reverse-engineering ». Il serait intéressant de capitaliser le travail mené sur les objets de base et

leurs particularismes dans Max/MSP dans une librairie FAUST, ou en complément d'une librairie FAUST existante. D'autre part restent en suspens des questions délicates, comme celle de la phase des oscillateurs : la traduction de l'harmoniseur a par exemple posé problème ; traduits en FAUST séparément, les différents modules qui composent l'harmoniseur fonctionnent correctement ; mais lorsque l'on essaye de générer un seul objet *.mxo* (Max/MSP), des différences de timbre apparaissent. Ce résultat est pour l'instant resté inexplicé : il est peut-être imputable à des problèmes de phase.

Il n'est pas évident d'évaluer l'ampleur de ce travail : tous les modules de synthèse et de traitement sonore ont été transformés (avec pour la plupart le cœur du traitement passé en FAUST, à l'exception des Phase-Aligned Formants pour les raisons expliquées au 2.2.4). Mais aucun traitement concernant la reconnaissance de phénomènes vocaux (hauteur, formants) n'a été porté en FAUST. Ces traitements font appel pour la plupart à des objets développés en C par des programmeurs tiers (comme Miller Puckette), avec par exemple des calculs de transformée de Fourier qui ne peuvent être produits en FAUST. De ce fait, l'intention artistique du compositeur, que nous citons dans la présentation de l'œuvre, et qui concernait avant tout la reconnaissance des phénomènes, ne peut d'une certaine manière être « préservée », en tout cas pas avec le type de méthode que nous avons présenté.

En termes de temps de codage, le travail nous a semblé plus long que ne l'aurait été l'écriture des patches correspondants. Mais entrent ici en compte une bonne connaissance de Max/MSP avec à l'inverse une découverte progressive de FAUST, et également le positionnement du travail, qui est différent : il s'agit de préservation, et il a fallu parfois longuement chercher avant de réussir à « imiter » exactement Max/MSP.

Dé-numériser, pérenniser, transmettre

Vers une nécessaire dé-numérisation ?

La traduction en FAUST permet certes la re-synthèse du patch dans l'environnement d'origine, Max/MSP. Mais elle offre également une opportunité de pérennisation sur support sous la forme de documentation mathématique. L'hypothèse faite par Karim Barkati et Yann Orlarey²⁵ est que le formalisme mathématique possède une stabilité et une pérennité à l'échelle de plusieurs siècles permettant d'espérer qu'une documentation y faisant appel reste lisible à cette échelle temporelle, ce qui théoriquement devrait permettre la réimplémentation du module concerné avec les moyens qui seront alors disponibles.

Examinons ce que contient une telle documentation à partir de l'exemple du *phasor~* déjà examiné préalablement. Dans un premier temps sont rappelées les équations du *process*, c'est-à-dire le traitement synchrone exécuté. Dans un second temps le bloc-diagramme du traitement est donné (cf. figure 5). Puis les notations mathématiques utilisées sont explicitées. Enfin, le code FAUST du module apparaît, ainsi que les librairies FAUST sollicitées par ce code (dans notre cas, la librairie *math.lib* proposant un certain nombre de définitions de fonctions mathématiques, notamment le modulo *fmod* auquel nous avons fait appel).

²⁵ Barkati, K., Orlarey, Y., Auto-documentation mathématique pour le traitement du signal avec Faust. In *Actes des Journées d'informatique musicale*, Saint-Etienne, 2011.

En produisant une telle documentation au format papier, dont la vocation serait d'être jointe à la partition, et conservée dans une médiathèque, nous pouvons parler de dé-numérisation, puisque le format choisi n'est plus numérique et n'est de ce fait plus encodé, mais directement accessible au lecteur.

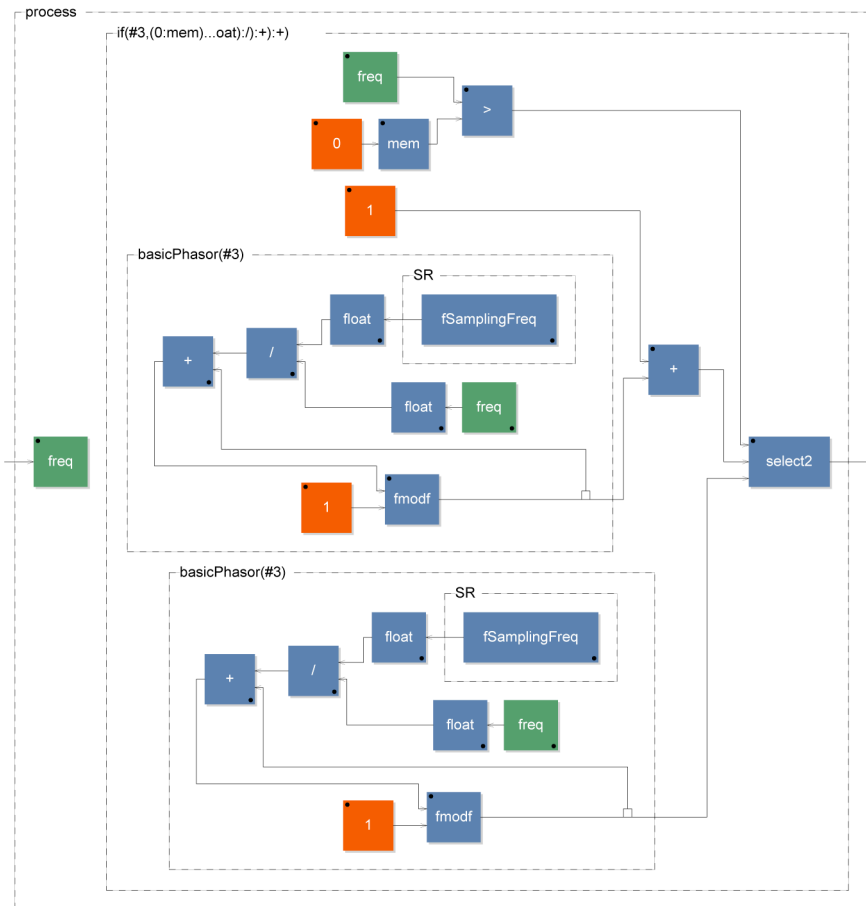


Figure 5. Le bloc-diagramme correspondant au module FAUST du phasor~.

Apprendre et transmettre l'œuvre mixte interactive

Au-delà de la pérennité technique se posent les questions de l'apprentissage et de la transmission de cette œuvre. Le premier postulat que nous posons est que la partie électronique doit être jouée, interprétée comme le serait n'importe quelle partie instrumentale. Nous concevons l'œuvre comme un duo associant une soprano et un réalisateur en informatique musicale et sommes clairement du côté de la musique de chambre.

Il y a donc eu apprentissage de la partie électronique de l'œuvre : d'abord au cours de séances de travail avec Serge Lemouton, réalisateur en informatique musicale²⁶ à l'IRCAM, puis sous la forme de filages avec la partie vocale enregistrée par Donatienne Michel-Dansac. Il s'agissait d'assimiler des caractéristiques « en temps », par exemple :

- Comprendre comment l'œuvre doit sonner à tel ou tel moment ;

²⁶ certainement la personne qui connaît le mieux *En Echo* en dehors du compositeur.

- Apprendre les interventions et gestes précis à faire :
 - Les crescendos/décrescendos doivent être bien calés dans le temps, et il faut soigneusement contrôler leur amplitude.
 - Concernant les déclenchements manuels aux événements (points de synchronisation entre la partie vocale et l'électronique), il convient parfois de débrayer le suiveur automatique et de lancer à la main la partie électronique, parfois pour des raisons techniques (petites notes trop rapides à suivre, ou encore confusion entre la partie de soprano et la texture électronique en raison de notes communes) mais aussi parce que le compositeur le demande à certains endroits, pour obtenir un jeu plus naturel. Ces demandes de Philippe Manoury ne sont pas inscrites sur la partition : nous les avons recueillies du compositeur lui-même mais aussi de Serge Lemouton.

Cette expérience nous a interrogé, et a suscité plusieurs questions :

- Peut-on imaginer des représentations interactives du fonctionnement de l'orchestre numérique pour vraiment jouer la partie électronique ?
- La présence du suiveur automatique de partition ne perturbe-t-elle pas la transmission de l'œuvre, puisque les moments de déclenchement manuel constituent ce que le réalisateur en informatique musicale doit mémoriser et éventuellement transmettre aux futurs interprètes ?
- En termes d'interface de déclenchement, le clavier de l'ordinateur est-il adapté ?

Analyser et représenter l'œuvre mixte interactive

Comment représenter le « fonctionnement » d'une œuvre mixte interactive comme *En Echo* ? La situation musicale associe le patch et la partition. Se placer du point de vue de l'un ou de l'autre serait réducteur :

- Vu de la partition, le patch est « rabattu » sur la notation musicale traditionnelle lorsque cela est possible. Tel est le jeu dans *En Echo* (cf. figure 6). Le timbre et la spatialisation sont en général les grands oubliés de cette projection sur le support-partition : comment les sons sont-ils diffusés dans l'espace ? que doit-on entendre ?
- Vue du patch Max/MSP, la partition n'est représentée que par le truchement des numéros d'événements ou des listes de notes pour le suivi automatique : le formalisme Max ne permet pas *a priori* de représentation musicale.

Nous avons cherché des représentations informatiques implémentées dans Max/MSP et rendant compte du fonctionnement du patch, tout en étant capable d'afficher des informations musicales en notation traditionnelle, pour établir des relations avec ce qui se passe dans la partition. Nous avons ainsi établi trois représentations :

- Une « partition » proposant une synthèse du jeu de l'orchestre numérique dans l'espace sonore octophonique résultat.
- Un vecteur d'état de l'orchestre électronique.
- Un diagramme fonctionnel des modules alimentant le spatialisateur mobile.

Ces trois représentations sont implémentées sous la forme de patches Max/MSP qui récupèrent directement les informations du patch de *En Echo*, notamment par le jeu des objets *send/receive* : ainsi les objets *send* envoient des valeurs numériques, en général

de contrôle et les objets *receive* permettent de recevoir la donnée à distance dans un autre patch. Par exemple l'objet *send niveau* émet des valeurs numériques sous l'appellation *niveau*, que les objets *receive niveau* permettent de recevoir de n'importe quel endroit.

Nous les examinons dans les paragraphes ci-dessous.

The image shows a handwritten musical score for a piece titled 'En Echo'. The score is written on multiple staves. The top staff is labeled 'Voix' (Voice) and features a melodic line with a triplet of notes marked 'mf' (mezzo-forte). Below the voice staff, there are two circled numbers, '34' and '37', with the labels 'TA' and 'MAIN' written above them. The subsequent staves are labeled 'Synth' (Synthesizer) and 'Jamp 2' (Sampler). The 'Synth' staves show complex chordal structures and some tremolo effects. The 'Jamp 2' staves show rhythmic patterns. The score is annotated with various musical notations, including dynamics, articulation marks, and time signatures.

Figure 6. Un extrait de la partition de *En Echo* qui montre comment le compositeur a noté le résultat sonore du patch Max/MSP sur la partition, par module (synthétiseurs PAFs, sampler)

« Partition » dans l'espace sonore octophonique

La figure 7 montre une copie d'écran du patch constituant la représentation de « partition » dans l'espace électroacoustique. On voit tout d'abord quels sont les modules de l'orchestre électronique actifs, comment ils sont spatialisés, ici en l'occurrence sur la quadraphonie fixe. On voit que les PAFs 1 et 5 sortent devant sur le haut-parleur 5, alors que les PAFs 2 et 6 sortent à droite sur le haut-parleur 6, etc. Ce schéma dynamique donne également les notes jouées par les PAFs sous forme de notation en partition, avec le détail par haut-parleur et l'ensemble des notes au centre.

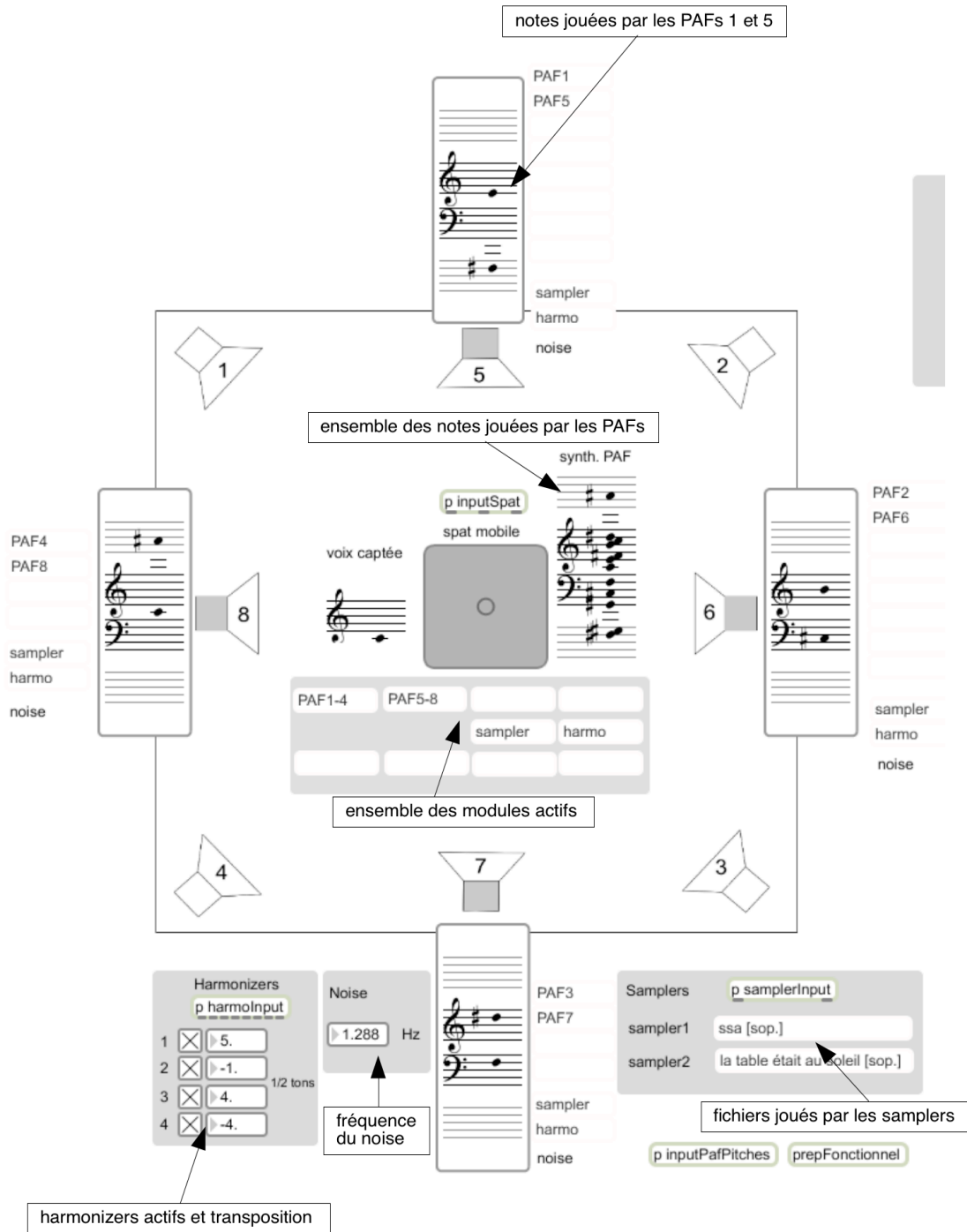


Figure 7. Une représentation proposant une synthèse du fonctionnement de l'orchestre numérique, notamment la visualisation des notes envoyées aux synthétiseurs formantiques et leur spatialisation octophonique. Les annotations encadrées explicitent les différents suivis de modules.

Vecteur d'état de l'orchestre numérique

Comment caractériser à un instant donné l'état de l'orchestre électronique utilisé par Philippe Manoury ? Nous avons imaginé un vecteur à six composantes, chacune d'entre elles correspondant à un bloc du patch. Chaque composante évalue l'activité des modules du bloc et leurs connexions. Chaque valeur correspond à une configuration unique. La figure 8 donne une copie d'écran de ce dispositif. Par exemple, le bloc de ressources R1 qui comprend les PAFS 1 à 4 et PAFS+ 1 et 2, est dans l'état 38.

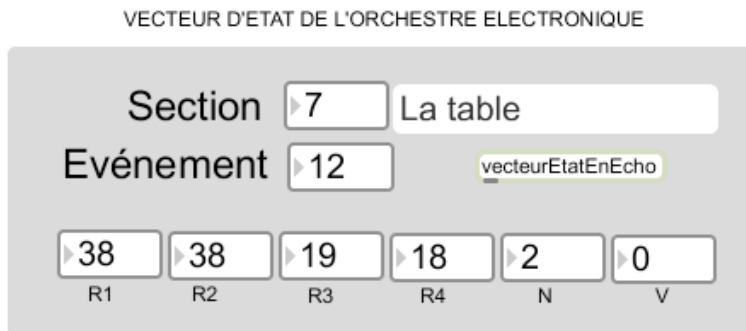


Figure 8. Copie d'écran du vecteur d'état de l'orchestre électronique avec ses six composantes.

Lorsque l'on accumule en mémoire les vecteurs d'état en parcourant l'ensemble de l'œuvre, on trouve environ cent trente changements de vecteurs. Nous avons commencé une recherche de configurations-types. Sur la première séquence, intitulée « La rivière », nous avons rassemblé les configurations semblables et nous obtenons le parcours donné sur la figure 9. Nous avons mis en évidence une configuration de base, fondée sur les PAFs 1 à 4 et 5 à 8 ainsi que le générateur de bruit (noise) : elle est exposée au début de la séquence. A cette situation de base peut s'ajouter soit l'ensemble formé du *sampler* et de *l'harmoniseur* (en orange), soit les PAFs 9 à 12 sans *sampler* ni *harmoniseur* (en mauve) : à partir de l'événement 27 et jusqu'à la fin, la section 1 oscille entre ces deux configurations. Il faudrait évidemment généraliser cette approche aux autres sections de l'œuvre.

En Echo, Philippe Manoury
Section 1, La rivière

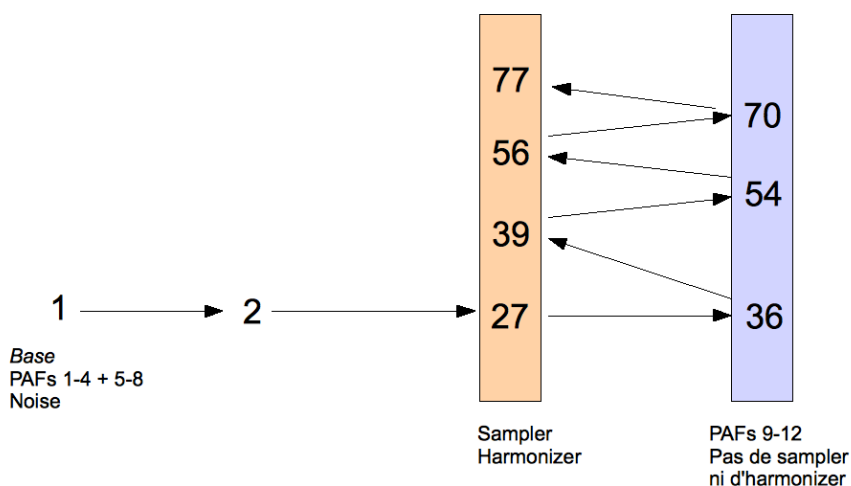


Figure 9. Configurations remarquables d'orchestre numérique au cours de la section 1 « La rivière ».

Diagramme fonctionnel des modules alimentant le spatialisateur mobile

La troisième représentation représente la circulation du son à travers les modules jusqu'au spatialisateur mobile (on ne représente pas ce qui se propage vers la quadriphonie fixe). Dans ce diagramme, les modules actifs apparaissent en noir, alors que ceux qui sont inactifs ne sont pas visibles. La sortie d'un module est représentée par un petit disque noir suivi d'un trait qui indique vers quel autre module le signal est envoyé. La figure 10 montre une copie d'écran instantanée faite dans la première section, « La rivière ».

Cette représentation est non seulement très utile pour suivre à un instant donné ce qui se passe dans un patch, mais aussi pour détecter des changements de conception ou des erreurs partiellement corrigées. La figure 11 représente la situation au début de la deuxième section de l'œuvre : on remarque que la liaison partant du sampler ne va nulle part. En fait, le sampler est bien relié à l'harmonizer, c'est-à-dire que le son sort bien du premier pour entrer dans le second, mais il s'avère que l'harmonizer est éteint à ce moment précis de l'œuvre. Qu'est-ce à dire ? Était-ce une première conception que le compositeur a ensuite abandonnée ? Était-ce une erreur corrigée dans l'urgence, à l'approche d'un concert ? Nous avons interrogé le compositeur le 12 juin 2012 et il nous a confié de ne plus se souvenir de la raison de cette connexion erronée. Une telle représentation pourrait donc contribuer à la génétique d'une œuvre avec patch Max/MSP.

DIAGRAMME FONCTIONNEL=>SPAT MOBILE

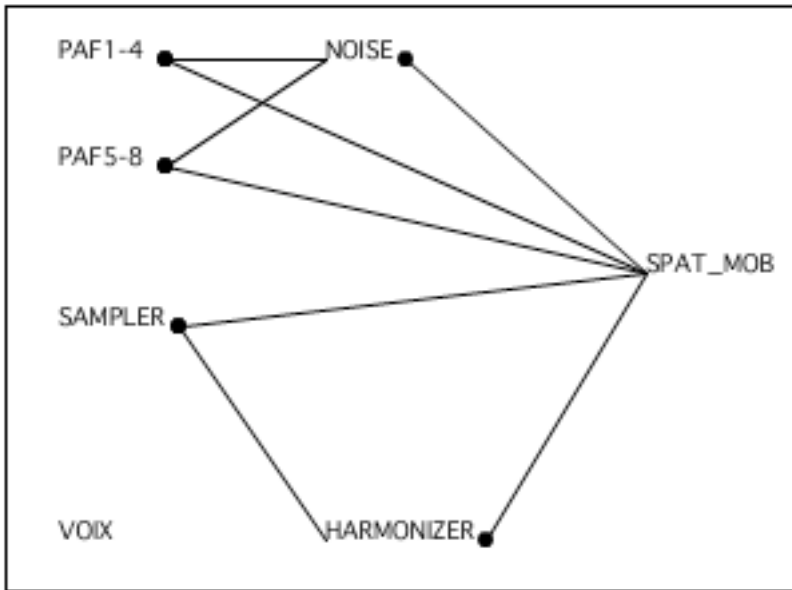


Figure 10. Exemple de diagramme fonctionnel montrant les relations entre modules actifs à un instant donné.

DIAGRAMME FONCTIONNEL=>SPAT MOBILE

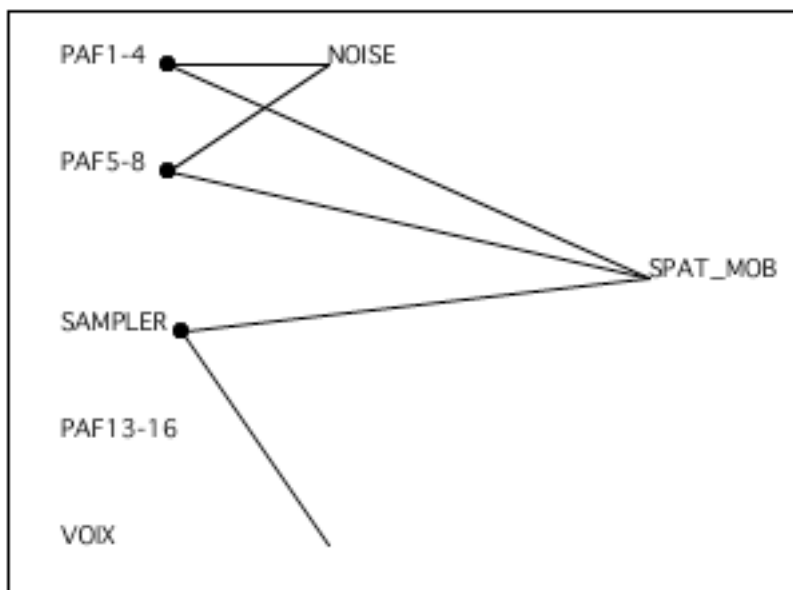


Figure 11. Quand un diagramme révèle une intention abandonnée ou une erreur partiellement corrigée.

Conclusion

Dans cet article, nous nous sommes attaché au destin de la pièce *En Echo* de Philippe Manoury pour soprano et électronique temps réel. En effet, comme toutes les œuvres faisant appel aux technologies, *En Echo* est soumise à l'obsolescence rapide des matériels et logiciels permettant de la donner en concert. Nous avons montré comment nous avons transcrit en FAUST un certain nombre de transformations synchrones du signal sonore, pour lesquelles a été générée une documentation mathématique archivable auprès de la partition sur papier. La nouvelle version de *En Echo* a été validée par le compositeur et en concert en 2011. Malheureusement, l'ensemble du patch Max/MSP de l'œuvre n'a pu être transcrit. En effet, les parties de reconnaissance de phénomènes vocaux (hauteurs et formants) et les parties asynchrones ne sont pas encore traduisibles en FAUST. Toutefois, l'ensemble de ces actions constitue un premier effort de pérennisation pour la transmission de l'œuvre et donc son remontage en concert. Dans cette perspective, citons la sollicitation du flûtiste Nicolas Vallette, qui souhaite monter *Jupiter*, pour flûte et électronique de Manoury, antérieure à *En Echo* (1987). Sachant qu'il y a une certaine continuité dans les procédés électroniques utilisés par le compositeur, il nous paraît intéressant de virtualiser une partie des modules de cette œuvre, ce qui permettra peut-être d'établir une bibliothèque FAUST des traitements employés par Philippe Manoury.

Dans l'autre sens, nous nous sommes interrogé sur la question de la transmission, car cette dernière, sollicitant des passages de connaissances entre spécialistes contribue à la pérennisation de l'œuvre. Cette transmission passe par des supports et formats pérennes tels que des équations mathématiques de transformation sonore inscrites sur papier, mais aussi par l'apprentissage de l'œuvre - ce qui pourrait ne pas être compatible avec l'automatisation complète de sa partie électronique, ou encore des représentations explicitant le fonctionnement de l'œuvre. Dans un même souci d'extension, le travail qui se présente sur *Jupiter* devrait nous amener à généraliser ces représentations, à examiner ce qui est générique et ce qui est propre à chaque œuvre.

Il se dessine ici quelques esquisses d'une musicologie non plus computationnelle au sens de la modélisation d'une partition, mais s'intéressant aux conditions de possibilité désormais informatiques, de l'œuvre, visant la représentation, la pérennisation et la transmission des traitements de l'informatique musicale tels qu'ils se déploient dans le répertoire contemporain.

Références bibliographiques

Bachimont, B., Blanchette, J.-F., Gerszo, A., Swetland, A., Lescurieux, O., Morizet-Mahoudeaux, P., Donin, N., Teasley, J., Preserving Interactive Digital Music: A Report on the MUSTICA Research Initiative. In *Proceedings of the Third International Conference on WEB Delivering of Music (WEB'03)*, Leeds, England, 2003.

Barkati, K., Orlarey, Y., Auto-documentation mathématique pour le traitement du signal avec Faust. In *Actes des Journées d'informatique musicale*, Saint-Etienne, 2011.

Bernardini, N., Vidolin, A., Sustainable live electro-acoustic music. *Actes de l'International Sound and Music Computing Conference*, Salerno, Italy, 2005.

Bonardi, A., Barthélemy, J., The preservation, emulation, migration, and virtualization of live electronics for performing arts : An overview of musical and technical issues. *ACM J. Comput. Cultur. Heritage*, 1(1), 16, 2008.

Bonardi, A., « Quand émergent des pratiques musicales a-musicologiques », *Musik – Kontext – Wissenschaft / Musique – Contexte – Savoirs, Perspectives interdisciplinaires sur la musique*, Talia Bachir-Loopuyt, Sara Iglesias, Anna Langenbruch, Gesa zur Nieden (éds), Frankfurt am Main : Peter Lang, 2012, pages 71-81, ISBN 978-3-631-60652-0.

Borghoff U.-M., Rodig, P., Scheffcyk, J., Schmitz, L., *Long-Term Preservation of Digital Documents: Principles and Practices*. Springer-Verlag New York Inc, 2006.

Brown, A.-R., Code Jamming. In *Journal of Media and Culture*, vol. 9, issue 6, december 2006.

Bullock, J., Coccioli, L., Modernising live electronics technology in the works of Jonathan Harvey. *Actes de l'International Computer Music Conference*, Barcelona, Spain, 2005.

Gladney, H., *Preserving digital information*, Springer-Verlag New York Inc, 2007.

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.

Manoury, P., *La note et le son*, Paris : l'Harmattan, 1998.

Orlarey, Y., Fober, D., Letz, S., An algebra for block diagram languages. *Actes de l'International Computer Music Conference*, 2002.