

# Numéros / n° 6 - Techniques et méthodes innovantes pour l'enseignement de la musique et du traitement de signal

## « Créer, enseigner, chercher : MUSICOLL »

**Anne Sèdes**

Résumé

La filière « composition assistée par ordinateur » animée par l'équipe du CICM (université Paris 8) propose aux étudiants un enseignement faisant interagir recherche et création. Ces dernières années, quelques thématiques ont permis à l'équipe de faire converger et interagir enseignement, recherche et création : la pratique de la composition mixte, l'élaboration de la bibliothèque de spatialisation HOA, le projet MUSICOLL. Ces trois thématiques sont présentées en indiquant comment elles interfèrent sur le plan de l'enseignement, de la recherche et de la création au fil de la Licence et des parcours du Master arts, mention musicologie et pratiques musicales et du doctorat « esthétique, sciences et technologies des arts, spécialité Musique ».

## Introduction

Au département de musique de l'université Paris 8, les outils et méthodes innovants sont au centre de nos approches, aussi bien en tant qu'enseignants-chercheurs que créateurs. Dans le texte qui suit, après un rappel du contexte académique et de nos récentes activités, nous présentons le projet MUSICOLL, qui concerne la musique temps-réel, collaborative et nomade, dans un contexte de formation par la recherche et par la création associant informatique et musique, comme un exemple intéressant de mise en œuvre de notre démarche.

## 1. Enseigner, créer, chercher

Notre approche s'inscrit dans le domaine de la musique sur ordinateur, dans un cursus de musique et musicologie, qui, en interaction avec les politiques scientifiques de l'université Paris 8, promeut largement les arts, les sciences et les technologies, la formation par la recherche, les méthodologies de la recherche-création, ou encore les humanités numériques.

En tant que spécialité mineure dans un cursus standard de musique et musicologie en licence, master et doctorat, les enseignements qui sont offerts à nos étudiants concernent la composition musicale, l'introduction à la programmation, l'audio-numérique, ainsi que des approches académiques faisant converger approches théoriques et créations pratiques. La pratique de la musique électroacoustique, de l'électronique en temps-réel et des musiques mixtes étendues à l'intermédiaire sont au cœur de notre pédagogie, où chaque étudiant trouve sa démarche singulière afin de construire son propre profil de spécialité, étant supervisé par une équipe d'étudiants avancés, de doctorants et d'enseignants-chercheurs.

Les musiques mixtes constituent un champ précieux pour l'expérimentation, l'enseignement, la recherche, la création et le développement logiciel. Dans ce champ, grâce à des environnements tels que Max <sup>(1)</sup> et Pure Data <sup>(2)</sup>, la thématique de la spatialisation du son, grande dimension de la composition que

le xx<sup>e</sup> siècle et ses technologies ont fait émerger, a permis de développer le projet HOA <sup>(3)</sup>. Ce projet a été soutenu pendant quatre années par le Labex Arts H2H, en développant conjointement des projets artistiques et logiciels. Il a donné lieu à plusieurs projets de thèse, à la livraison de bibliothèques logicielles qui sont maintenant bien connues de la communauté internationale, ainsi qu'à de nombreuses créations expérimentales dans un cadre universitaire. Le projet est toujours en cours de développement et résiste bien aux évolutions des systèmes d'exploitation et des environnements logiciels concernés.

Le projet HOA visait un objectif relativement modeste : une librairie en C++ pour permettre aux musiciens d'accéder à de l'ambisonie d'ordre élevé pouvant être mise en œuvre dans différents logiciels, pour développer des interfaces graphiques nouvelles dans le champ de la visualisation du son, pour offrir une approche 3D, et pour donner accès à des transformations du son à l'intérieur du domaine mathématique des harmoniques sphériques et de ses formalismes, etc.

Un certain nombre d'ateliers ont permis à des étudiants en musique de créer des espaces sonores au sein de projets musicaux dans le cadre de pièces pour l'atelier de composition du département de musique, qui, placé sous la direction du compositeur José-Manuel Lopez, accueille chaque année des ensembles professionnels reconnus, tels que les Percussions de Strasbourg, les solistes de l'ensemble de l'Itinéraire, d'Aleph, de Sillages?

Concernant la création logicielle, des doctorants tels que Julien Colafrancesco, Pierre Guillot et Elliott Paris ont contribué aux différentes versions et révisions de la bibliothèque, sous la direction d'Anne Sèdes et d'Alain Bonardi au CICM.

De façon informelle, les méthodes étaient basées sur l'apprentissage par la *faire*, par la modélisation, la boucle essai-erreur, le test et la reformulation, en interaction avec un retour des utilisateurs, soit en atelier, soit via l'internet.

Les limitations de Max, dont le code source propriétaire est relativement fermé et dont les changements significatifs et les évolutions sont assez imprévisibles au fil des nouvelles versions, aussi bien que celles de Pure Data, *open source*, qui implique toujours des révisions sérieuses (Guillot, 2014), ont amené l'équipe à considérer de nouvelles approches, au moment où les démarches collaboratives et nomades, ou encore le *cloud* projettent des perspectives d'avenir. En grande partie basé sur le sujet de recherche doctoral d'Elliott Paris <sup>(4)</sup>, le projet MUSICOLL a été lancé, héritant de fait du bilan méthodologique du projet HOA. Son but : renouveler les pratiques des musiques en temps réel, viser la programmation de patches collaboratifs, multiplateformes et pérennes.

Les objectifs de MUSICOLL sont :

- Produire un nouvel environnement de traitement audio numérique temps réel collaboratif et nomade.
- Étudier et faciliter la prise en main de cet environnement par des musiciens, créateurs ou apprenants.
- Examiner des méthodes nouvelles d'enseignement de la musique en temps réel et de ses environnements logiciels.
- Disséminer en terrain scientifique, artistique et professionnel.

Dans ce qui suit, nous allons présenter l'état du projet quant à ses développements logiciels, au cours de ses 18 derniers mois <sup>(5)</sup>.

## 2. MUSICOLL : Le projet et ses premiers résultats après dix-huit mois.

La collaboration entre des musiciens répartis dans divers endroits du monde pour faire de la musique

ensemble via l'internet est déjà populaire (Renaud *et al.* 2007). L'étape suivante aura consisté à mettre réunir plusieurs personnes pour des séances de travail et d'enregistrement, par exemple grâce au séquenceur collaboratif Ohm Studio, développé par OhmForce (6). De la même manière, avec MUSICOLL, nous abordons le design collaboratif du traitement du son en temps réel par les moyens d'un langage graphique, de type Max ou Pure Data.

Le projet MUSICOLL est financé par l'ANR (7), et accueilli au sein de la Maison des Sciences de l'Homme Paris Nord, les partenaires en sont le CICM et la Compagnie OhmForce, spécialisée dans l'audio numérique collaboratif. Il vise à transformer la pratique musicale du temps réel dans une perspective plus coopérative et nomade. Le projet a commencé en 2016 et s'étend jusqu'à décembre 2018, avec une valorisation et un développement en communauté qui pourra courir bien au-delà de cette date. Il propose le développement d'une première maquette d'un environnement musical collaboratif en temps-réel permettant à plusieurs utilisateurs de travailler simultanément sur un traitement hébergé en ligne et accessible d'un terminal connecté ; nous avons nommé cet environnement Kiwi (8).

Une fois la maquette réalisée, nous allons étudier sa prise en main dans le cadre de nos cours de licence à l'université, dédiés aux débutants en programmation audio numérique en temps réel, afin d'observer le potentiel de renouvellement pédagogique d'un tel environnement. Nous allons également mettre cet environnement à la disposition d'enseignants et d'étudiants en collège et lycée, afin d'examiner l'intérêt qu'il peut susciter, et afin de partager et populariser une telle pratique.

Le premier résultat attendu est la production du prototype d'une application Kiwi fournissant les briques essentielles à la construction de transformations sonores et de synthèse pour répondre à un premier niveau d'attente des compositeurs et des apprenants. Kiwi est pour le moment une application multiplateforme, fonctionnant en réseau dont l'approche collaborative est basée sur la bibliothèque Flip développée par OhmForce. Il est décrit dans la troisième partie du présent article. À l'issue de ce projet, on pourrait envisager une version plugin de l'application qui pourrait être intégrée dans des stations de travail audio numériques comme OhmStudio, permettant d'intégrer des processus en temps réel conçus de façon collaborative dans un séquenceur.

Parallèlement au développement du logiciel Kiwi, deux études d'utilisation commenceront dès la rentrée 2017, dirigées vers deux communautés importantes à l'université Paris 8 : les compositeurs d'un côté et les étudiants inexpérimentés en traitement de son en temps réel de l'autre. Des tests en collège, en conservatoire ou en médiathèque de quartiers sont également en cours d'étude en Île-de-France, afin d'atteindre un public plus jeune et plus large.

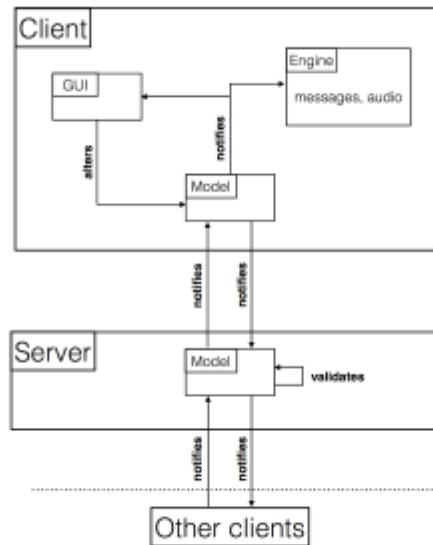
Figure . Festival La Démo, Médiathèque Don Quichotte et Maison des Sciences de l'Homme Paris Nord



Saint-Denis, 31 mai 2017 : atelier de patching collaboratif pour collégiens sur 4 ordinateurs de la Médiathèque, puis restitution musicale à l'auditorium de la MSH Paris Nord.

Ce projet de recherche soulève plusieurs problèmes : le premier, d'ordre technique, concerne le traitement du signal et sa synchronisation à travers les réseaux. D'autres questions concernent la conception dans les approches collaboratives ou bien encore l'appropriation et la diffusion. On tente également d'encourager les utilisateurs à travailler différemment, à partager leurs retours.

Figure . Schéma de l'architecture actuelle de l'application Kiwi



### 3. L'application KIWI : l'architecture du prototype

Kiwi est une application qui permet de créer des effets audio en temps réel en connectant des boîtes graphiques appelées « objets » à l'intérieur d'un graphique appelé « patch ». Cette approche, similaire à celle offerte par les logiciels Max et Pure Data, prend une autre dimension en raison de ses aspects collaboratifs et nomades. En effet, ce projet permet aux utilisateurs de concevoir et de créer collectivement des moteurs audio en partageant et testant leurs idées en temps réel dans le même « patch » et dans un flux de travail commun. Les fonctionnalités collaboratives ont été mises à disposition par OhmForce via Flip <sup>(9)</sup>, un framework C++ pour créer des applications collaboratives dans un modèle architectural de type modèle-vue-contrôleur (MVC) et gérer des transactions sur le réseau. Le rendu graphique et audio est réalisé par JUCE <sup>(10)</sup>, un framework multiplateforme couramment utilisé dans l'industrie des applications audio.

Lorsque la partie client modifie son modèle (en modifiant un lien ou en ajoutant un objet à un patch par exemple), ce modèle envoie automatiquement une transaction au serveur qui distribue la modification à tous les autres clients. La partie serveur peut également invalider une modification si, pour des raisons de concurrence, cette altération ne peut pas être appliquée (par exemple, si un lien a été créé par un utilisateur et qu'un autre utilisateur a supprimé l'un des objets liés à ce lien en même temps). Dans ce cas, le serveur peut choisir de restaurer le modèle de données utilisateur à un état valide précédent et obliger tous les utilisateurs à appliquer la même modification (l'objet sera supprimé et le lien ne sera pas créé par exemple).

Comme cela a été suggéré plus haut, la partie client utilise le modèle et gère les autres composants : l'interface utilisateur graphique (GUI) et le moteur. La partie GUI est composée d'une vue et d'un contrôleur. La partie vue de l'interface graphique représente le modèle en affichant le patch, les objets et les liens à l'écran. Lorsque la partie modèle du patch change, la vue reçoit une notification et s'actualise. La partie contrôleur reçoit les interactions de l'utilisateur, venant de la souris ou du clavier, par exemple, et les utilise pour modifier le modèle. Par conséquent, dans notre appropriation du patron de conception MVC, la partie GUI est à la fois une vue et un contrôleur. D'autre part, le moteur est le noyau calculatoire de l'application, mais il peut aussi être assimilé à une vue au sens du patron de conception MVC bien qu'il ne comporte aucune fonctionnalité graphique. Comme expliqué précédemment, le modèle est autonome et n'est pas directement lié au moteur. Ainsi, en tant que vue, le moteur reçoit les notifications lorsque le modèle change et les interprète. Le moteur définit la fonction opératoire des objets. Par exemple, un objet appelé « + » dans le modèle peut se résumer à un nom, que le moteur interprète comme opération d'addition. Lorsqu'un lien est créé dans le modèle, le moteur va lier ces deux objets, ce qui leur permet de communiquer par des messages.

Il existe deux types d'opérations qu'un objet peut traiter. Les premiers sont les messages pouvant résulter des interactions avec la souris ou le clavier. Le calcul de ces opérations se produit à un taux relativement faible. Les autres opérations sont effectuées sur des signaux audio numériques à un taux d'échantillonnage élevé (entre 44 100 et 196 000 Hz). Le moteur, qui réalise les calculs, procède aux vérifications nécessaires en triant les unités calculatoires. Il optimise également les processus et gère les relations entre les messages et les signaux audio des opérations. Par conséquent, le moteur permet de créer des effets, des synthétiseurs et des sons encore plus complexes en reliant les processeurs et en transcrivant des messages en opérations mathématiques.

Ainsi, l'application possède une architecture multicouche avec des liens complexes entre ces couches qui lui permet de gérer les nombreuses implications résultant des aspects collaboratifs et nomades du projet. Ces aspects seront abordés dans la section suivante.

## 4. Les implications des aspects collaboratifs et nomades

Après avoir fonctionné sur un serveur local avec un ordinateur qui hébergeait le patch avec plusieurs utilisateurs pouvant le modifier, l'application fonctionne désormais hors réseau local, via l'internet et un serveur centralisé.

Pour l'instant, l'application offre uniquement des interactions telles que : création, destruction et déplacement d'objets graphiques et de liens. Ces modifications apportées au modèle sont gérées par Flip. Ainsi, une fois que le serveur les a prises en compte, il avertit automatiquement tous les clients connectés des modifications. Néanmoins, nous commençons tout juste à gérer ces interactions et avons déjà rencontré des problèmes de concurrence. En effet, que se passe-t-il si quelqu'un supprime un objet alors que quelqu'un d'autre le relie à un autre objet ? L'objet devrait-il survivre et le lien doit-il être créé ? Ou le lien doit-il être ignoré et l'objet est-il supprimé ? En effet, une de ces interactions devrait être ignorée. Les deux ne peuvent pas coexister. Pour l'instant, toutes ces questions concurrentes restent non résolues. Anticiper et prévoir les problèmes sera l'une des tâches principales du projet. Pour répondre à ces problèmes, nous prévoyons de définir des cas d'utilisation, d'offrir un ensemble de solutions et de les évaluer en utilisant les commentaires des utilisateurs (l'organisation des tests est présentée dans la section suivante de cet article). À terme, nous espérons définir un système prioritaire où chaque action est hiérarchisée par rapport aux autres. Par exemple, si la suppression d'un objet a une priorité plus élevée que le déplacement d'un objet, la suppression d'un objet alors qu'un autre utilisateur le déplace entraîne son élimination.

Au-delà de la modification simultanée du modèle, d'autres concepts sont directement liés à un flux de travail collaboratif. La conception des interfaces utilisateur doit être revue dans des environnements collaboratifs, en imaginant comment afficher les interactions des autres utilisateurs afin qu'ils puissent appréhender rapidement ce qui se passe sans augmenter l'entropie de l'information du patch.

Un exemple de ces interactions est la sélection d'un objet. Pour le moment, l'hypothèse est la suivante : savoir si un ou plusieurs autres utilisateurs sélectionnent un objet en même temps est plus important que de savoir qui sélectionne quel objet. L'équipe OhmForce nous a conseillé de définir un schéma qui met en évidence le contour de l'objet à l'aide de différentes couleurs en fonction des règles suivantes : une couleur est affectée à la sélection de l'utilisateur local et une autre est utilisée pour montrer qu'un objet a également été sélectionné par quelqu'un d'autre sur le réseau. L'un des avantages de cette méthode est qu'on n'utilise que deux couleurs, afin d'utiliser d'autres couleurs pour afficher différentes informations dans le patch. Cela améliore également la lisibilité des patches. Néanmoins, cette information est partielle. On pourrait vouloir savoir quel utilisateur sélectionne les objets comme dans l'application Google Docs. Cela peut être réalisé en utilisant un ensemble de couleurs distinctes où chaque utilisateur est affecté à une seule et ses sélections sont personnalisées à l'aide de cette couleur, mais cette approche peut rendre difficile la lecture d'un patch. Une autre idée serait d'afficher une fenêtre avec des informations sur les utilisateurs qui effectuent la sélection alors que la souris survole les objets. Cette approche qui semble satisfaisante à première vue, pose en fait un problème pour les interfaces tactiles. Par conséquent, nous allons de nouveau faire des suggestions à un ensemble de testeurs pour trouver les solutions qui correspondent le mieux aux besoins de l'application Kiwi.

Ce type de sélection d'objet est lié à l'aspect collaboratif de l'édition de patch. Par exemple, l'expérience

de l'équipe OhmForce dans le domaine collaboratif a introduit un problème que nous n'aurions jamais imaginé : si un utilisateur supprime un objet ou déplace l'objet au-delà des limites de la fenêtre du patch, ces deux interactions apparaîtront aux autres utilisateurs comme la disparition de l'objet et aucun d'entre eux ne pourra déterminer rapidement ce qui s'est produit réellement. Dans l'application OhmStudio, OhmForce a résolu le problème en créant des animations sur l'interface utilisateur graphique pour informer les utilisateurs passifs de ce type d'interaction (en glissant ou en décolorant l'interface graphique).

Ces problèmes sont gênants, au point de rendre l'application inutilisable. Par exemple, l'utilisateur a besoin de temps pour tester et écouter les résultats audio intermédiaires dans le processus de création de patch. Cela peut prendre beaucoup de temps, et d'autres utilisateurs voudront peut-être modifier le patch pendant la session d'écoute de leur co-créateur. Cependant, toute modification du patch entraînera probablement des problèmes audio et des artefacts. Une solution que nous avons considérée, mais qui reste à démontrer, offre une option qui verrouille le patch et désactive sa mise à jour pendant que le son est activé. Après l'écoute, le patch récupérerait l'information et serait à nouveau synchronisé.

Ainsi, de nombreux problèmes ne peuvent apparaître qu'en utilisant et en testant les premières versions de l'application, en expérimentant à la fois les aspects collaboratifs et de portabilité.

## 5. Cas d'usage, éléments de démonstration et de validation

Nous avons commencé à travailler sur les usages, avec un premier scénario de démonstration et de validation. Ce scénario a été mis en place en collaboration avec OhmForce et deux utilisateurs de la scène, chacun d'eux travaillant avec Kiwi sur un patch commun. Ce scénario simule un court atelier où l'utilisateur A explique à l'utilisateur B comment gérer un oscillateur simple et plusieurs oscillateurs par duplication. Il fournit des situations où la collaboration se produit simultanément ou successivement et met en évidence de nombreux problèmes collaboratifs spécifiques aux environnements graphiques en temps réel.

L'environnement en temps réel de Kiwi nécessite une validation par différents types d'utilisations par des communautés spécifiques intéressées par des applications particulières. Un premier exemple est l'enseignement de langages graphiques tels que Max ou Pure Data aux débutants et, dans notre cas, les étudiants de premier cycle au département de musique de l'université Paris 8. Son but sera de concevoir et tester les fondamentaux d'un cours basé sur la plateforme collaborative de Kiwi, en comparant les pratiques éducatives antérieures et nouvelles pour introduire à la programmation audionumérique en temps réel. L'ensemble des pratiques d'enseignement et d'apprentissage sera documenté avec des enregistrements vidéo.

## Conclusion

Le projet MUSICOLL et son logiciel Kiwi n'existent que depuis dix-huit mois. Nous avons actuellement un petit nombre d'objets disponibles dans Kiwi, et d'autres devront être ajoutés pour permettre aux compositeurs et aux étudiants de tester l'application de façon significative. Afin d'être efficaces, les développements futurs devront tenir compte des retours des beta-testeurs, pour mieux ajuster les spécifications du logiciel à leurs demandes. Une version beta sera distribuée début 2018.

À travers l'exemple MUSICOLL, nous avons essayé de montrer comment une équipe travaillant, enseignant, recherchant et créant au sein d'un département de musique dans une université en sciences humaines très orientée par les arts et le numérique, peut générer un environnement favorable au développement inventif de nouveaux outils de création, offrant des compétences nouvelles aux étudiants, à divers niveaux, depuis la licence jusqu'au doctorat. Cela peut se faire via un projet de développement logiciel basé sur des méthodologies informelles et expérimentales, en toute autonomie, en dehors des modèles traditionnels portés par l'industrie et son ingénierie. Nous espérons partager ce type de contributions avec des communautés potentielles adaptées à l'enseignement, à la création et à la formation par la recherche et le développement, dans le but de partager des savoirs et savoir-faire

communs et mutualisables. C'est bien là un des objectifs de l'université.

---

1. <http://cycling74.com/products/max>

2. <http://puredata.info/>

3. [www.mshparisnord.fr/hoalibrary/](http://www.mshparisnord.fr/hoalibrary/)

4. Contrat doctoral financé par le Labex Arts H2H, arts et médiations humaines.

5. Notons que dans l'article initial paru en avril 2017 aux Presses des Mines, nous revenons sur les 10 derniers mois. La présente édition en français et en ligne nous permet de proposer une mise à jour des 20 derniers mois. Rappelons au passage que cet article fait suite à la conférence d'Anne Sèdes dans le cadre du colloque de Saint-Étienne en novembre 2015. C'est à la suite de ce colloque que Jean Millot a candidaté à l'offre de développeur pour le projet MUSICOLL, et dont il est désormais l'un des principaux développeurs.

6. <http://www.ohmstudio.com>

7. Le numérique au service des arts, du patrimoine, des industries culturelles et éditoriales, 2015.  
[http://www.agence-nationale-recherche.fr/projets-finances/?tx\\_lwmsuivibilan\\_pi1\[Programme\]=2120150703](http://www.agence-nationale-recherche.fr/projets-finances/?tx_lwmsuivibilan_pi1[Programme]=2120150703)

8. En parallèle à cet article, nous renvoyons à la publication publiée à l'occasion des Journées d'Informatique Musicale 2017. « Kiwi : Vers un environnement de création musicale temps réel collaboratif », E. Paris, J. Millot, P. Guillot, A. Bonardi & A. Sèdes, actes des JIM 2017. Cette publication et sa démonstration ont permis à ses auteurs d'obtenir le prix « jeunes chercheur » 2017 de l'AFIM (association francophone d'informatique musicale).

[https://jim2017.sciencesconf.org/data/Eliott\\_Paris2017aa.pdf](https://jim2017.sciencesconf.org/data/Eliott_Paris2017aa.pdf).

9. <http://irisate.com>

10. <https://www.juce.com/>

---

**Pour citer ce document:**

Anne Sèdes, « Créer, enseigner, chercher : MUSICOLL Â », *RFIM* [En ligne], Numéros, n° 6 - Techniques et méthodes innovantes pour l'enseignement de la musique et du traitement de signal, Mis À jour le 14/06/2018

URL: <http://revues.mshparisnord.org/rfim/index.php?id=510>

Cet article est mis À disposition sous [contrat Creative Commons](#)