

PROCESSUS COMPOSITIONNELS ET OPÉRATEURS MUSICAUX DANS ML-MAQUETTE – LES OUTILS DE TRAITEMENT DU SIGNAL

Jean Bresson
IRCAM - CNRS STMS
Représentations Musicales

RÉSUMÉ

ML-Maquette est une application informatique dédiée la pédagogie musicale, développée à partir de l'environnement de composition assistée par ordinateur OpenMusic. Cet article présente les modalités de transfert d'outils et processus musicaux issus de la composition (ou de l'analyse) musicale créés dans OpenMusic vers un cadre d'utilisation pédagogique dans ML-Maquette. Ces modalités seront illustrées pour le cas particulier d'outils relevant du domaine du traitement du signal sonore (analyse/synthèse), qui montrera comment des processus a priori complexes se retrouvent intégrés de façon relativement simple dans ML-Maquette sous la forme d'opérateurs élémentaires.

1. INTRODUCTION

Musique Lab Maquette (ou ML-Maquette) est une application dédiée à l'enseignement musical réalisée dans le cadre du projet Musique Lab 2 [15] [9]. Celle-ci est développée à partir de l'environnement de composition assistée par ordinateur (CAO) OpenMusic [1] [5], et permet à un utilisateur (professeur ou élève de classe de musique) de reconstituer dans un document pédagogique des procédures relevant de formalismes divers tels que la manipulation des différents paramètres des structures musicales (hauteurs, rythmes, dynamique), les transformations et opérations sur ces structures, l'harmonie, l'intégration temporelle des objets musicaux [3].

Une description détaillée de ML-Maquette pourra être trouvée dans [7]. Des aspects et utilisations pédagogiques de l'application et des autres modules composant l'environnement Musique Lab 2 sont également exposés dans [11].

Après un rapide aperçu de l'application dans la section 2, nous essaierons plus précisément ici de présenter les modalités de l'intégration de techniques et modèles issus de la CAO, développés dans OpenMusic, vers un cadre pédagogique dans ML-Maquette. Le protocole syntaxique permettant cette intégration, basé principalement sur des fonctionnalités relevant de programmation par meta-objet disponibles dans

OpenMusic, sera présenté dans la section 3. La section 4 illustrera ensuite cet exposé en introduisant par la même occasion un ensemble d'outils dédiés au traitement du signal sonore intégrés dans l'application.

2. ML-MAQUETTE

ML-Maquette permet de créer des documents dynamiques concrétisant des situations pédagogiques, dans lesquelles sont étudiés des concepts ou extraits musicaux. La figure 1 montre une fenêtre de l'application représentant ce type de document.

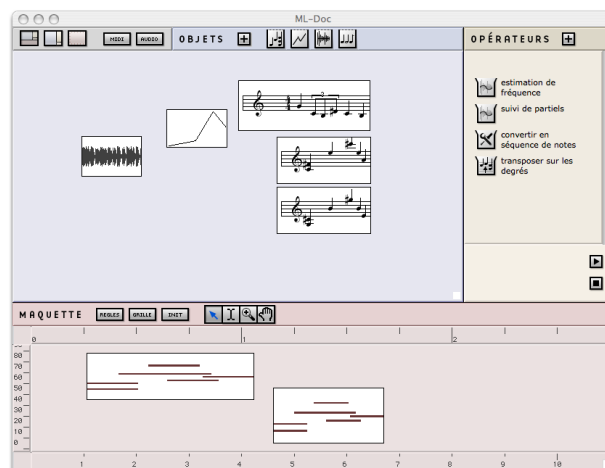


Figure 1. Un document dans ML-Maquette.

Trois zones composent le document. La zone *objets*, située dans la partie supérieure gauche, permet d'instancier différentes classes d'objets musicaux (accords, séquences, courbes, sons.)

La zone *opérateurs*, située dans la partie supérieure droite, contient un ensemble de fonctions capables d'opérer des traitements sur les objets. Ces fonctions (opérateurs), qui font l'objet principal du présent article, sont sélectionnées par l'utilisateur parmi une bibliothèque, en fonction de l'application visée dans chaque document.

En glissant un objet sur l'un de ces opérateurs, et généralement après la saisie de différents paramètres nécessaires à sa fonction, on obtient un nouvel objet

pouvant être à son tour inséré parmi les autres dans la zone *objets* (voir figure 2). Un cycle de création et de transformation d'objets musicaux peut ainsi être développé, selon une démarche expérimentale et pédagogique visée.

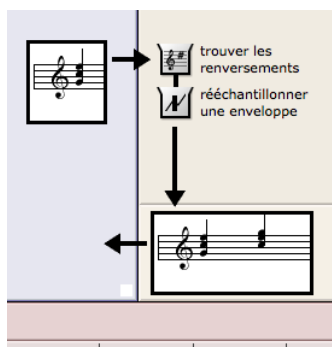


Figure 2. Application d'un opérateur et création d'un nouvel objet.

Enfin la zone *maquette*, située dans la partie inférieure du document, permet d'insérer dans un contexte temporel les objets préalablement constitués. Cet espace permet, outre l'assemblage et le montage des objets musicaux, d'affecter à ceux-ci des comportements fonctionnels avancés selon l'opérateur qui les a générés.

Dans l'ensemble d'un document peuvent ainsi se développer des scénarios et pédagogies variés liés à la création et la manipulation de matériaux et structures musicales [11].

3. LES OPÉRATEURS ML-MAQUETTE

La bibliothèque d'opérateurs disponibles dans ML-Maquette constitue un réservoir d'outils pour la réalisation des documents pédagogiques. Ces opérateurs sont classés en différents groupes, se rapportant à autant de domaines d'application.

En règle générale, un opérateur correspond à une méthode écrite en Lisp/CLOS [16] et enregistrée dans l'application selon certaines règles syntaxiques. Le traitement de cette méthode, notamment par le protocole de meta-objets de CLOS [13], permet à l'application d'intégrer automatiquement celle-ci parmi les opérateurs, en particulier pour la création des composants adaptés dans l'interface graphique.

3.1. Typage et spécialisation des méthodes

Un nombre limité de classes d'objets musicaux sont supportés dans l'application (accord, séquences musicales, courbe 2D, ou fichier audio). Pour être utilisée sous forme d'un opérateur dans ML-Maquette, le premier argument d'une méthode doit donc être spécialisé par l'une de ces classes. Une fonction générique peut cependant regrouper plusieurs méthodes

spécialisées chacune par une classe différente. La première opération réalisée par le système lors de l'application d'un opérateur (i.e. lorsque l'on glisse un objet dans la zone *opérateurs*) est en effet de déterminer, parmi les méthodes correspondant à une fonction générique, les types d'objets acceptés par celle-ci.

Les arguments supplémentaires de l'opérateur (s'il y en a) doivent eux aussi se soumettre à des contraintes de typage afin de permettre la construction d'un formulaire, présenté à l'utilisateur lors de son application. Ce formulaire permettra la spécification des valeurs de chaque argument d'une opération. Des classes sont prédéfinies à cet effet : entier, liste à choix multiples, booléen, voire des notions musicales comme l'intervalle, la tonalité, le monnayage rythmique, etc. Elles permettront, par l'intermédiaire de méthodes spécialisées, de mettre en place les composants graphiques du formulaire qui transmettront la valeur des paramètres correspondants dans le corps de la méthode-opérateur.

Enfin, l'utilisation de meta-classes pour la création des méthodes et fonctions génériques dans OpenMusic (c'est à dire, par exemple, le fait d'utiliser une sous-classe de `standard-method` pour définir des méthodes aux propriétés et comportements étendus [2]) permet d'attacher des informations complémentaires aux opérateurs : une icône particulière, des valeurs par défaut ou indications textuelles pour les arguments, etc.

L'exemple suivant pourrait être le prototype d'une méthode Lisp destinée à être convertie en opérateur ML-Maquette.

```
(defmethod! mon-operateur ((objet chord)
                           (param1 integer)
                           (param2 listofvalues)
                           (param3 intervalle))
  :menuins '((2 ("choix1" t) ("choix2" nil)))
  :initvals '(nil 4 "choix2" nil)
  :indoc '(nil "nombre d'iterations" "type" "")
  ;; [... corps de la fonction ...]
)
```

Le respect de ces règles d'écriture définit ainsi une méthode intégrant l'information nécessaire à son insertion dans l'application et à la création des interfaces adéquates (voir figure 3).

3.2. Corps des opérateurs

Le corps d'une méthode-opérateur ML-Maquette a pour seule contrainte de retourner une instance de l'une des classes supportées dans l'environnement. Ainsi, et notamment du fait que les deux environnements partagent le même langage et les mêmes structures de données, les outils développés dans OpenMusic (à des fins compositionnelles ou analytiques) peuvent-ils être aisément adaptés sous forme d'opérateurs dans ML-Maquette.

La correspondance directe entre les patches OpenMusic (programmes visuels) et le code Lisp permet

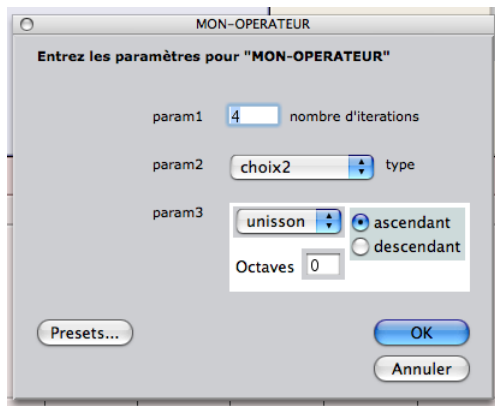


Figure 3. Formulaire de paramétrage apparaissant lors de l'application de *mon-operateur*.

dès lors d'envisager l'intégration des processus compositionnels sous forme d'opérateurs élémentaires. Une fonction primitive (prédéfinie en Lisp dans OpenMusic ou dans une bibliothèque) est incluse directement dans la syntaxe générale présentée ci-dessus ; et un patch plus complexe peut généralement être traduit relativement facilement : chaque boîte graphique correspond à un appel à une fonction Lisp, et les connexions permettent de reconstituer directement la composition fonctionnelle du programme représenté.

Par ailleurs, il est possible de tirer profit des propriétés de persistance de OpenMusic pour réutiliser dans ML-Maquette du code visuel créé dans l'environnement de CAO. Une méthode définie graphiquement dans OpenMusic (voir [2]) est en effet enregistrée sous forme d'un fichier de code Lisp, qui peut être importé et évalué dans ML-Maquette. La fonction ainsi définie peut alors être utilisée de façon transparente dans un opérateur (insérée dans le protocole de typage décrit précédemment).

4. LES OPÉRATEURS DE TRAITEMENT DU SIGNAL SONORE

L'exemple des outils de traitement du signal sonore va nous permettre d'illustrer les propos précédents.

Ces outils sont dérivés de processus réalisés dans OpenMusic, relevant notamment de l'analyse du signal sonore et de la conversion des données d'analyse en structures musicales de haut niveau, propices au traitement musical, ou de la démarche symétrique consistant à produire des signaux sonores à partir de structures musicales symboliques issues des traitements compositionnels.

Ces processus sont eux-mêmes implémentés à l'aide d'un ensemble de fonctionnalités spécialisées constituées à partir d'outils externes d'analyse et de synthèse sonore [6]. La figure 4 est un exemple de ce type de processus mis en oeuvre dans un patch OpenMu-

sic, réalisant l'analyse additive (*partial tracking* [14]) d'un signal sonore, dont le résultat, stocké provisoirement dans un fichier, est finalement converti en une représentation symbolique sous forme de partition [12].

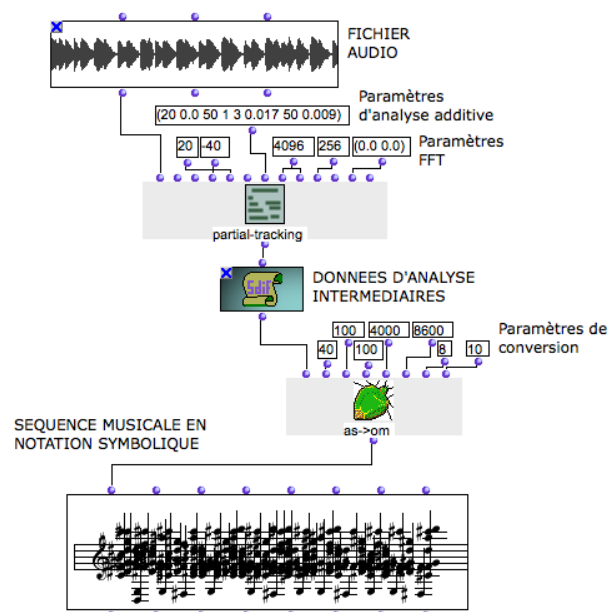


Figure 4. Un processus d'analyse du signal sonore implémenté dans OpenMusic.

Ce processus fait généralement partie d'un modèle compositionnel plus général, mais il représente une entité de traitement cohérente dans celui-ci ; en effet, il permet de passer d'une structure de donnée initiale (un son numérique) à une autre (un séquence de notes et/ou d'accords).

Pour créer un opérateur ML-Maquette à partir de ce type de processus, il suffit donc de traduire le patch correspondant en une fonction Lisp, selon l'un des procédés décrits plus haut. Il importe alors cependant d'effectuer un choix des variables dans le processus, qui seront les paramètres présentés à l'utilisateur dans ML-Maquette, lui permettant d'agir sur différents aspects de ce processus.

Dans l'exemple de la figure 4, on observe en effet, hormis le fichier audio donné en amont du processus, de nombreux paramètres correspondant aux différentes fonctions appelées (caractéristiques de la fenêtre d'analyse pour la transformée de Fourier, seuils de détection des partiels, lissage, et autres paramètres pour l'analyse, et durée minimale entre accords, filtre en fréquence, polyphonie maximale, etc. lors de la conversion en notation symbolique). Pour l'opérateur ML-Maquette correspondant, nous avons uniquement conservé les bornes fréquentielles exprimées en hauteur symbolique pour le filtrage du résultat, ainsi que la polyphonie maximale, ou nombre de notes pouvant être considérées simultanément dans un même accord. Les valeurs des paramètres restent

sont donc fixées dans le corps de l'opérateur.

L'utilisation de cet opérateur dans ML-Maquette est illustrée sur la figure 5. On observe en effet que seules ces variables sélectionnées dans le processus sont demandées à l'utilisateur dans le formulaire intermédiaire.

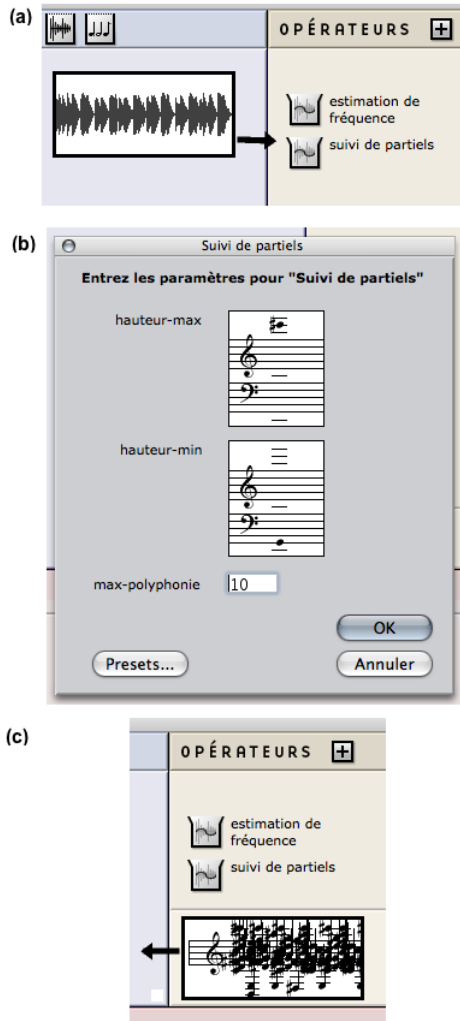


Figure 5. Utilisation de l'opérateur de suivi de partiels dans ML-Maquette : (a) un son est glissé depuis la zone *objets* sur l'opérateur ; (b) un formulaire demande d'entrer certains paramètres pour effectuer cette opération ; (c) un nouvel objet apparaît, qui peut être glissé dans la zone *objets*.

Différents opérateurs ont ainsi été élaborés selon ce même principe, s'appuyant pour la plupart sur le moteur d'analyse/synthèse additive PM2 de l'Ircam, notamment utilisé dans le logiciel AudioSculpt [4]. Parmi ceux-ci, on trouvera notamment le processus inverse, qui réalise une synthèse additive à partir des notes d'une séquence musicale ou d'un accord considérées comme des partiels (voir figure 6). On trouvera aussi l'analyse par accord, qui utilise une segmentation préalable du signal sonore pour réaliser une analyse additive plus structurée (figure 7),

ou encore l'estimation et le suivi de fréquence fondamentale des sons supposés harmoniques [8] (figure 8).

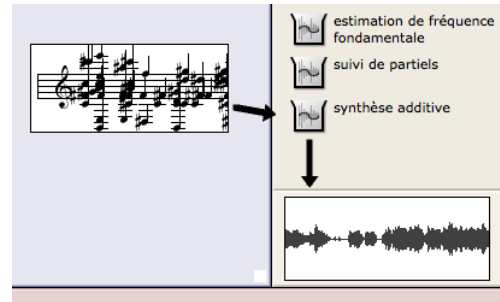


Figure 6. Opérateur de synthèse additive.

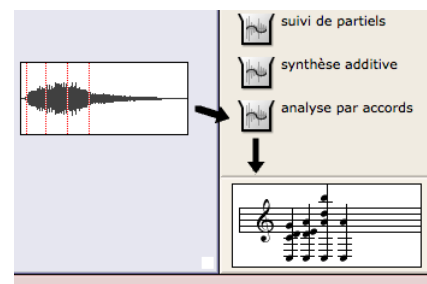


Figure 7. Opérateur d'analyse additive par accords.

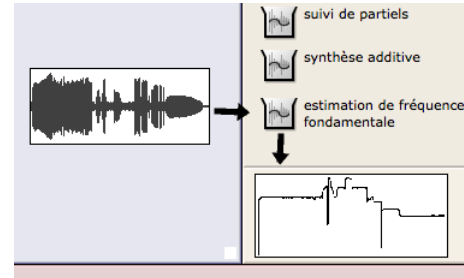


Figure 8. Opérateur d'estimation de fréquence fondamentale.

5. CONCLUSION

Nous avons présenté le protocole permettant la création d'opérateurs musicaux dans l'application ML-Maquette, et particulièrement les moyens de transfert de processus issus de l'environnement de composition OpenMusic vers cette application. Cette possibilité d'augmenter et de personnaliser la bibliothèque des opérateurs constitue selon nous une ouverture sur les évolutions à venir de l'application, et un aspect important du dynamisme général du projet Musique Lab 2.

L'exemple choisi avec les opérateurs de traitement du signal est caractéristique par le contraste entre la

complexité des processus initiaux (mettant en jeu des méthodes de traitement du signal et de CAO avancées) et la simplicité de leur mise en oeuvre dans ML-Maquette, notamment par le choix réduit des paramètres présentés par les opérateurs correspondants, mais aussi par la boucle opératoire élémentaire (*objet - opérateur - paramètres - nouvel objet*) de l'application.

Ces opérateurs de traitement du signal sont ainsi des outils très utilisés dans les expériences pédagogiques menées jusqu'ici avec ML-Maquette. Ils mettent en effet rapidement en évidence des notions fondamentales de la musique actuelle telles que les relations entre le son acoustique et sa représentation en hauteurs, ou encore entre le son naturel enregistré et le son de synthèse (dans une démarche d'analyse/resynthèse, par exemple).

Les expériences d'analyse et de synthèse sonore, suivant l'approche compositionnelle de la CAO, permettent ainsi un recul sur le matériau sonore et musical, complémentaire avec l'interaction directe permise par les environnements de traitement du signal en temps réel, dont notamment ML-Audio, le module correspondant dans le projet Musique Lab 2 [10].

6. REFERENCES

- [1] Agon, C. *OpenMusic : Un langage visuel pour la composition musicale assistée par ordinateur*. Thèse de Doctorat, Université Pierre et Marie Curie (Paris 6), 1998.
- [2] Agon, C. et Assayag, G. "OM : A Graphical extension of CLOS using the MOP", *Proceedings of ICL'03*, New York, 2003.
- [3] Assayag, G. et Bresson, J. "OpenMusic : de la composition à l'enseignement", *L'inouï*, #2. Ircam - Editions Leo Scheer, 2006.
- [4] Bogaards, N. and Röbel, A. "An interface for analysis-driven sound processing", *AES 119th Convention*, New York, USA, 2004.
- [5] Bresson, J., Agon, C. et Assayag, G. "OpenMusic 5 : A Cross-Platform release of the Computer-Assisted Composition Environment", *Proceedings of the 10th Brazilian Symposium on Computer Music SBCM'05*, Belo Horizonte, Brésil, 2005.
- [6] Bresson, J. "Sound Processing in OpenMusic", *Proceedings of the 9th International Conference on Digital Audio Effects DAFx-06*, Montreal, Canada, 2006.
- [7] Bresson, J., Guédy, F. et Assayag, G. "Musique Lab Maquette : approche interactive des processus compositionnels pour la pédagogie musicale", *Revue STICEF (Sciences et Techniques de l'Information et de la Communication pour l'Éducation et la Formation)*, Volume 13, 2006.
- [8] Doval, B. et Rodet, X. "Estimation of Fundamental Frequency of Musical Sound Signals", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing - ICASSP'91*, Toronto, Canada, 1991.
- [9] Guédy, F. "Musique Lab 2, un environnement pour la pédagogie musicale", *L'inouï*, #2. Ircam - Editions Leo Scheer, 2006.
- [10] Guédy, F. "Le traitement du son en pédagogie musicale", *L'inouï*, #2. Ircam - Editions Leo Scheer, 2006.
- [11] Guédy, F., Bresson, J. et Assayag, G. "Musique Lab 2 - Un environnement d'aide à la pédagogie musicale", *Actes des Journées d'Informatique Musicale - JIM'07*, Lyon, 2007.
- [12] Hanappe, P. et Assayag, G. "Intégration des représentations temps/fréquence et des représentations musicales symboliques", in Chemillier, M. et Pacht, F. (dir.) *Recherches et applications en informatique musicale*. Hermes, 1998.
- [13] Kiczales, G., des Rivières, J. et Bobrow, D. G. *The Art of the Metaobject Protocol*. MIT Press, 1991.
- [14] McAulay, R. J. et Quatieri, T. F. "Speech analysis/synthesis based on a sinusoidal representation", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4), 1986.
- [15] Puig, V., Guédy, F., Fingerhut, M., Serrière, F., Bresson, J. et Zeller, O. "Musique Lab 2 : A Three-Level Approach for Music Education at School", *Proceedings of the International Computer Music Conference*, Barcelone, Espagne, 2005.
- [16] Steele, G. L. *Common LISP the Language. Second Edition*. Digital Press, 1998.