PROJET EFFICACE : DÉVELOPPEMENTS ET PERSPECTIVES EN COMPOSITION ASSISTÉE PAR ORDINATEUR

Jean Bresson, Dimitri Bouche, Jérémie Garcia, Thibaut Carpentier,
Florent Jacquemard, John MacCallum, Diemo Schwarz*
UMR 9912 STMS IRCAM-CNRS-UPMC / * CNMAT, UC Berkeley
{*prénom.nom*}@ircam.fr

RÉSUMÉ

EFFICACe est un projet de recherche centré sur les outils de composition assistée par ordinateur (CAO), explorant les rapports entre calcul, temps et interactions dans les processus de composition musicale. Nous présentons différents travaux en cours dans le cadre de ce projet, concernant l'application de processus interactifs pour le traitement et l'ordonnancement des séquences temporelles, le traitement et la spatialisation sonore, ou encore les interfaces homme-machine.

1. INTRODUCTION

Le projet EFFICACe [1] propose une orientation de développement pour les outils de composition assistée par ordinateur (CAO), explorant de nouveaux rapports entre calcul, temps et interactions articulés autour de l'environnement OpenMusic (langage de programmation visuelle dédié à la CAO [8]) et d'autres technologies développées à l'Ircam et au CNMAT. L'ambition de ce projet est de réduire certains antagonismes critiques de l'informatique musicale, en liant de manière formelle les processus de composition musicale assistés par ordinateur à des domaines tels que l'interaction temps-réel, la synthèse ou la spatialisation sonore. Nous abordons le rapprochement des domaines « signal » et « symbolique » dans le traitement et la représentation de l'information musicale en regard de la dualité entre les systèmes temps différé et temps réel. En décloisonnant les processus de CAO du domaine strictement « hors-temps » (ou temps « différé ») cette démarche a pour but de les intégrer dans une interaction structurée avec leur contexte, dans une optique compositionnelle (c'est-à-dire dans les processus qui conduisent à la création de matériau musical) mais également dans une logique de performance ou d'exécution musicale.

Nous souhaitons par cette démarche produire un contexte informatique permettant de traiter différentes dimensions des structures et processus musicaux dans un cadre cohérent et expressif. Les directions privilégiées concernent la production, le traitement dynamique et le rendu des structures temporelles et rythmiques (relations entre le temps déroulé d'exécutions musicales et les processus de calcul hors-temps), ainsi que les processus de synthèse et de spatialisation sonore à travers lesquels ces structures seront déployées, et pour lesquels un nouveau cadre paradigmatique est proposé au sein de l'environnement de CAO.

2. CAO ET PROCESSUS RÉACTIFS

Lors de précédents travaux nous avons introduit un nouveau paradigme de calcul et de programmation dans l'environnement OpenMusic, combinant les approches formelles existantes, basées sur le style fonctionnel, aux approches réactives inspirées des systèmes interactifs temps-réel [9, 6]. La mise en place de chaînes réactives dans les programmes visuels engendre une interactivité localisée : un changement ou une action de l'utilisateur (*évènement*), dans le programme ou les données qui le composent, produit une série de réactions conduisant à sa mise à jour (ré-évaluation). Un tel évènement peut provenir d'une source extérieure (typiquement, un port MIDI ou UDP ouvert et attaché à un élément du programme visuel) ; ainsi, une communication bidirectionnelle peut être établie entre les programmes visuels et des applications ou dispositifs externes. L'environnement de CAO se trouve alors inséré dans la temporalité d'un système plus large, et potentiellement régi par les évènements et interactions produits par ou dans ce système. Cette temporalité peut être celle du processus de composition [2], ou celle de la performance. La figure 1 montre un exemple de patch OpenMusic utilisé en *live* [3] pour l'analyse et l'affichage dynamique des caractéristiques harmoniques du jeu d'un piano MIDI.

1 . ANR-13-JS02-0004-01 – `http://repmus.ircam.fr/efficace/`

2 . Ce système a été éprouvé par exemple lors de la création de la pièce *Quid sit musicus ?* de Philippe Leroux, en relation avec un dispositif d'entrée graphique [13].

3 . Concert *Livre Digital – Colóquio Franco-Brasileiro de Criação et Análise Musicais com Suporte Computacional* (Campinas, Brésil).

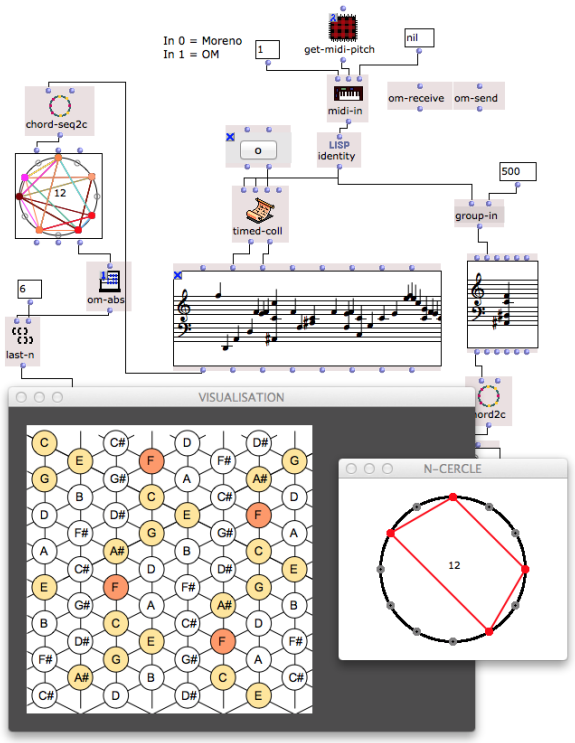

Figure 1: Processus réactifs dans OpenMusic : réception de données via MIDI et représentations graphiques.

3. SÉQUENCES TEMPORELLES ET ORDONNANCEMENT

La notion de composition musicale est étroitement liée à la construction de structures temporelles. Les processus compositionnels traitent et produisent des séquences, rythmes ou autres structures micro- ou macro-temporelles de façon subtile et parfois complexe. Dans ce contexte le paradigme « temps différé » de la CAO permet de manipuler le temps des structures musicales comme un paramètre compositionnel, pouvant à la fois subir des calculs et produire des formes complexes [15]. Le temps y est calculé sans restriction : passé, présent et futur peuvent être pris en compte dans le calcul d'un instant, et la complexité de ce calcul n'est pas soumise aux contraintes du temps réel. Dès lors la richesse expressive d'un environnement de CAO tient en grande partie aux croisements qu'elle permet de mettre en œuvre entre les relations temporelles et fonctionnelles au sein des structures musicales.

Les aspects dynamiques et l'entrée de données dans le flux temporel de la création ou de la performance questionne certaines des caractéristiques fondamentales des systèmes de CAO. Dans cette section nous abordons successivement la question de la transcription des structures temporelles dans le domaine de la notation rythmique, puis celle des mécanismes d'ordonnancement sous-jacents à notre environnement.

3.1. Expression rythmique des flux temporels

Un certain nombre de directions de recherche ont été abordées sur la question de la quantification rythmique, soit celle du passage de séries de données temporelles vers les représentations pulsées et structurées de la notation musicale [3]. Sur le versant théorique, l'utilisation de formalismes tels que les automates d'arbres pondérés ou la réécriture de termes pour la manipulation de données temporelles structurées en arbres [16, 17] permet d'envisager de nouveaux algorithmes de transcription, et des méthodes d'évaluation pour l'adéquation des arbres rythmiques à certaines caractéristiques visées.

Le problème de la quantification rythmique se pose en effet comme insoluble de façon univoque : il est nécessaire de « calibrer » un système de résolution afin de l'adapter au contexte musical, déterminant par exemple un optimum entre précision de la transcription et simplicité/lisibilité des partitions produites. Nous nous tournons donc vers des systèmes semi-supervisés, où les interactions du compositeur sont déterminantes dans le déroulement du processus. Ici la programmation visuelle se présente comme un paradigme pertinent, et les extensions réactives mentionnées précédemment, couplées à des interfaces dédiées, se montrent décisives pour la mise en œuvre des méthodes formelles développées (voir Figure 2). Par ailleurs, les interactions et choix de l'utilisateur pourront être mis à profit dans une logique d'apprentissage pour l'optimisation des processus algorithmiques [22].

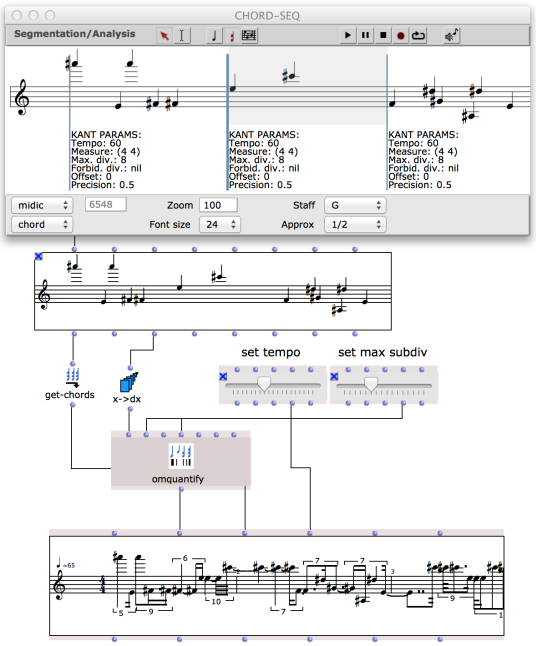

Figure 2: Interaction dans les processus de quantification rythmique. L'utilisation d'éditeurs pour la supervision du processus, couplée à des composants réactifs, favorise la découverte et l'expérimentation.

3.2. Ordonnancement dynamique

L'ordonnanceur dans un système musical a pour but de déclencher sur demande des séquences d'actions a_i à leurs temps planifiés respectifs t_i (typiquement, pour « jouer » une partition ou une structure musicale donnée). Dans une première approche, une simple file d'actions lue par un processus permet de réaliser cette opération. Dans un système performatif temps réel, cette file est « alimentée » et consommée en continu (voir Figure 3). Fusionnant le calcul dans le déroulé du temps musical, un tel système contraint le temps de calcul des actions, qui doit respecter des échéances tout en autorisant une articulation entre les flux synchrones d'entrée/sortie et des séquences d'évènements asynchrones résultant notamment d'interactions avec un utilisateur (ou avec un interprète).

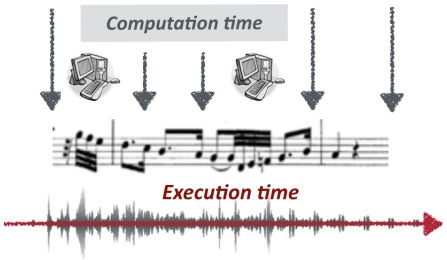

Figure 3: Relations entre calcul (*planning*) et exécution musicale dans un système temps réel.

Dans le cadre « temps différé » classique de la CAO, le calcul produit et articule des évènements musicaux dans le temps (notes, sons, etc.) : cette phase d'*ordonnancement*, qui détermine et ordonne la séquence d'actions, est préalable et indépendante de la phase d'*exécution* qui lira cette même séquence (voir Figure 4).

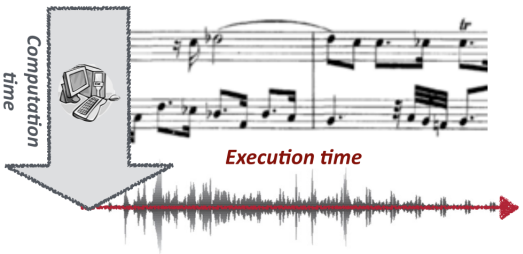

Figure 4: Relations entre (*planning*) et exécution musicale dans un système temps différé.

En considérant l'intégration d'évènements dans les phases de calcul et d'exécution musicale (que ces évènements soient eux-mêmes issus de calculs ou d'interactions avec le monde réel), s'esquisse un modèle d'ordonnancement hybride, décloisonnant les catégories précédentes. Dans cette perspective nous développons une architecture permettant de lier les processus implémentés dans l'environnement de CAO à la structure responsable de l'ordonnancement et des exécutions musicales. Cette architecture repose sur une hiérarchisation des objets musicaux intégrée dans leur conversion et leur représentation en liste d'actions (Figure 5). Les évènements

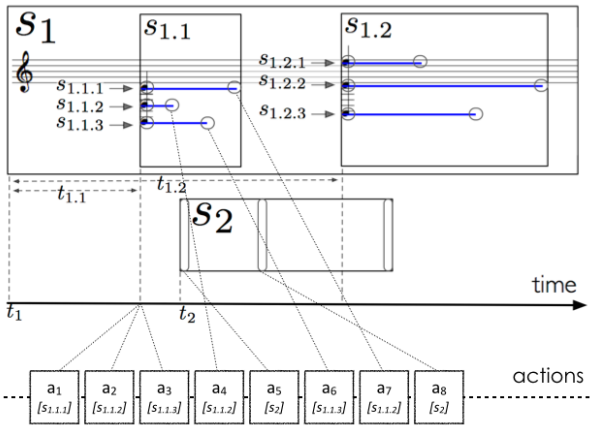

Figure 5: Exemple d'une structure musicale hiérarchique et conversion en actions. S_1 est une structure à 3 niveaux (séquence/accords/notes). S_2 est une séquence de 3 actions.

et interactions de l'utilisateur opérant sur les structures musicales peuvent ainsi reconfigurer le processus d'ordonnancement pendant l'exécution, permettant la modification en temps réel d'une partition en cours de lecture. Un moteur de calcul dédié gérant les calculs longs optimise l'exécution et fournit un retour d'information vers la partition. En admettant par ailleurs qu'une action exécutée puisse elle-même reconfigurer l'ordonnancement d'autres actions, des processus potentiellement complexes et autonomes peuvent alors être mis en œuvre au sein de la partition, ouvrant le champ à la programmation de scénarios et structures temporelles interactifs, non linéaires voire non déterministes.

4. CHAÎNE DE TRAITEMENT AUDIO

L'intégration des méthodes d'analyse, de traitement et de synthèse audio dans des processus compositionnels est un aspect déterminant des évolutions actuelles de la CAO. En complément des approches *off-line* existantes, mises en œuvre par l'intermédiaire d'outils externes de type « ligne de commande » et des fichiers d'entrées/sorties [5], une nouvelle architecture audio favorisera les aspects dynamiques dans la manipulation de ressources sonores, et une meilleure interopérabilité des outils.

Plusieurs bibliothèques entrent en jeu dans cette architecture (voir sections suivantes). L'interface de ces bibliothèques avec le langage Lisp d'OpenMusic est réalisée grâce à un « emballage » sous forme de bibliothèques dynamiques ANSI-C, et par l'entremise du *package* Common Foreign Function Interface (CFFI for Common Lisp).

L'allocation de *buffers* d'échantillons sonores dans l'environnement nous permet de manipuler les ressources audio et de les transférer librement entre les différentes couches et composants logiciels, favorisant le couplage des processus d'écriture, de synthèse et de spatialisation sonore.

A la suite de cette section, la figure 6 montre un exemple de procédé pouvant être réalisé en combinant les différents outils et bibliothèques présentés ici.

4.1. LibAudioStream (LAS)

La bibliothèque *LibAudioStream* (LAS) du Grame est utilisée pour la préparation et le rendu des ressources audio dans OpenMusic. Cette bibliothèque multi-plateforme, dans sa version la plus récente, permet de gérer les fichiers audio multi-canaux [21]. Tout en faisant abstraction des problématiques liées aux architectures et au rendu, elle offre un contrôle à l'échantillon près sur le traitement et l'ordonnancement dynamique des flux audio. En outre, LAS embarque un compilateur Faust [24], permettant de programmer et d'appliquer des traitements sur ces flux audio [4].

4.2. ISMM Audio Engine (IAE)

ISMM Audio Engine (IAE) est un moteur audio versatile pour la synthèse sonore basée sur le contenu, développé par l'équipe ISMM (interaction son–musique–mouvement) de l'Ircam [11]. Ce moteur a été intégré en tant que bibliothèque C++ portable dans des environnements divers tels que Max, Unity3D (IAEOU), iOS, et récemment dans OpenMusic.

IAE effectue la synthèse granulaire (possiblement *pitch-synchrone*) et la synthèse concaténative par corpus [27], basée sur du matériel audio annoté. Les annotations sont attachées à des segments de fichiers ou buffers audio, et contiennent des descriptions de caractéristiques perceptives de chaque segment, tels que sa brillance, son énergie, son timbre, ou son contenu fréquentiel. Les annotations peuvent être extraites automatiquement du matériel sonore, chargées à partir de fichiers créés par des logiciels externes, ou spécifiés par l'utilisateur. L'analyse automatique des annotations s'effectue via le framework *PiPo* (*plugin interface for processing objects*), par exemple par le module *pipo.ircamdescriptor* qui calcule environ 70 descripteurs audio.

La synthèse, rendue sur un buffer audio ou connectée à un périphérique, peut ensuite être paramétrée selon un grand nombre de paramètres de jeu granulaire (enveloppes, volume, transposition, filtrage).

4.3. OM-Spat

Le *Spatialisateur* [18] est un moteur audio développé à l'Ircam et dédié à la spatialisation sonore. Conçu de façon modulaire, il combine un réverbérateur artificiel multicanal [19] et un étage de panoramisation configurable (*panning* d'intensité ou de temps, encodage/décodage ambisonique, reproduction binaurale ou transaurale, etc.). Le *Spatialisateur* s'appuie sur un paradigme de spatialisation « orientée objet » (i.e. la manipulation d'entités sonores, indépendamment du dispositif de restitution) et il peut être piloté par une interface de contrôle perceptif [20] permettant une paramétrisation intuitive de l'espace acoustique virtuel créé.

Au niveau logiciel, le *Spatialisateur* se présente sous la forme d'une bibliothèque C++/objective-C de composants audio et d'interfaces graphiques. Cette bibliothèque est déployée dans divers environnements de production musicale (objets externes Max, plugins VST/AU/AAX, etc.).

Précédemment utilisé sous forme d'utilitaire en ligne de commande dans la bibliothèque OM-Spat, le *Spatialisateur* est ici lié dynamiquement et accessible de façon modulaire depuis OpenMusic. Les interfaces graphiques de contrôle, *monitoring* et *authoring* du *Spatialisateur* peuvent également être ré-exploitées et incluses dans les éditeurs de l'environnement. Jusqu'alors utilisées dans des environnements temps réel (Max ou autre), ces interfaces sont essentiellement destinées à une représentation « instantanée » de la scène sonore ou de l'état du processeur de spatialisation. Nous présentons dans la Section 5 une extension de ces outils offrant un contrôle de l'ordonnancement et de la temporalité des paramétrages spatiaux.

4.4. Contrôle (OSC)

L'encodage OSC (Open Sound Control [28]) est utilisé dans une grande partie des outils logiciels dédiés à la musique et au son comme support de communication pour les données de contrôle. Le système d'adressage, simple et hiérarchique, permet l'organisation des données sous forme de messages, typiquement transférés via UDP. La notion de *bundle* permet de regrouper les messages dans une structure unique, associée à un *time-tag* pour une mise en contexte temporel [25].

La bibliothèque *libo*, développée au CNMAT, permet la manipulation (encodage, décodage, traitement) de messages et *bundles* OSC. Cette bibliothèque est notamment utilisée comme support du système *odot* [12] pour Max, qui permet de manipuler les données structurées sous forme de *bundles* OSC et étend le format d'un langage intégré pour le traitement de ces données.

L'encodage et la transmission de messages OSC peut être effectuée dans OpenMusic en Common Lisp (à travers l'implémentation *cl-osc* [4]) ou via la bibliothèque *libo*. L'objet *o.bundle*, construit à partir de messages formatés comme des simples listes (adresses OSC suivies de données) cache un pointeur sur une version sérialisée du *bundle* OSC, compatible avec les fonctionnalités de *libo*. Ses données pourront être transmises à des applications externes (à travers le réseau), ou en tant que pointeur aux autres bibliothèques C liées à l'environnement.

4. https://github.com/hanshuebner/cl-osc

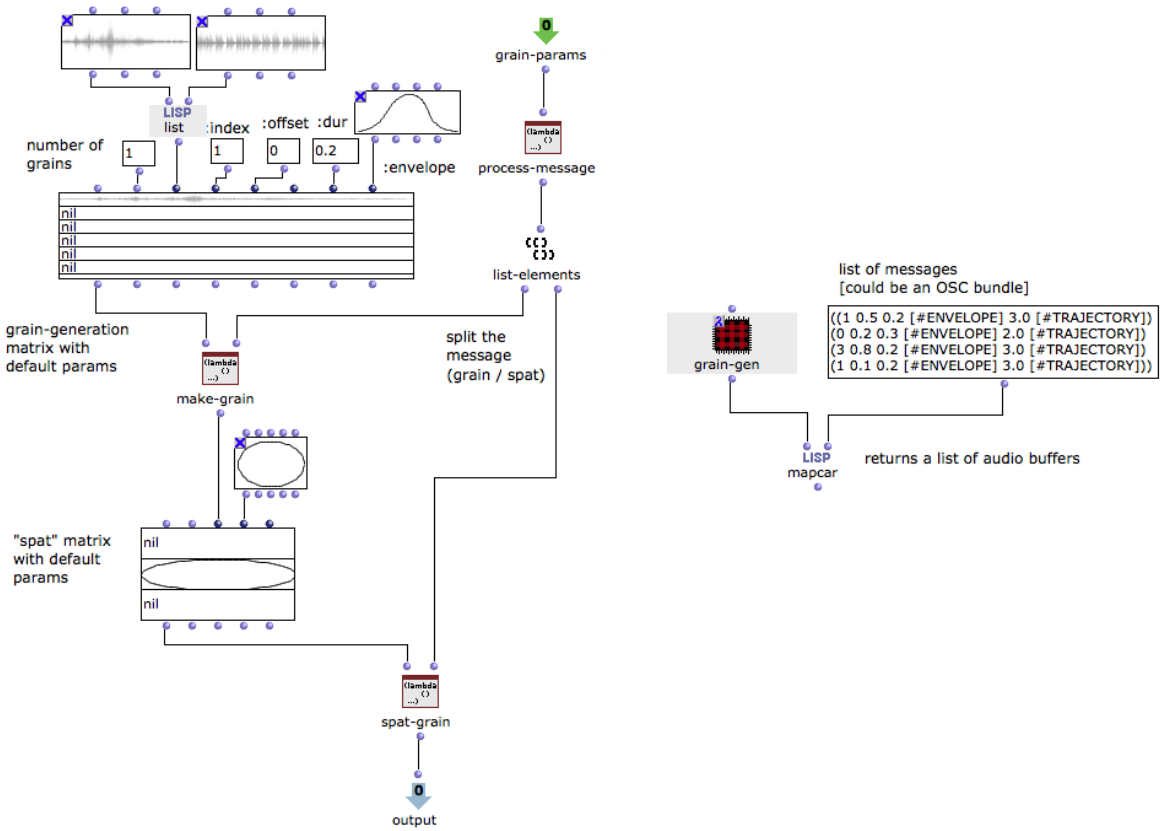

Figure 6: Schéma de principe d'un processus couplé de synthèse granulaire et spatialisation, sous forme d'un programme visuel OpenMusic. Le moteur IAE est sollicité pour la production de grains sonores, dont le traitement spatialisé peut-être contrôlé de façon algorithmique ou dynamiquement via des messages OSC.

La particularité d'une utilisation dans un contexte compositionnel par rapport à l'usage (généralement temps réel) du format OSC, tiendra dans l'accent nécessaire mis sur la structuration temporelle des messages et des évènements de contrôle. L'utilisation des *time-tags* et le groupement de messages en ensembles structurés permettront le stockage et le parcours de données de contrôle en tant que séquences temporelles (sur le modèles des séquences d'évènements MIDI).

5. INTERFACES UTILISATEUR

Les interactions de l'utilisateur dans la spécification et l'exécution des calculs jouent un rôle majeur dans le cadre applicatif envisagé avec les outils décrits précédemment. En complément des outils de programmation visuelle, nous proposons ici des interfaces permettant d'appréhender de manière dynamique la complexité et le caractère temporel des données musicales.

Nous nous intéressons dans cette perspective aux données de contrôle pour la spatialisation sonore. Un certain nombre de travaux récents dans ce domaine laissent en effet entrevoir des perspectives innovantes pour la CAO [23, 26, 10].

Cependant, le caractère statique et relativement abstrait des approches programmatiques de la CAO tend à s'opposer aux démarches plus empiriques et sensibles généralement mises en œuvre dans l'appréhension du son spatial. Ainsi l'usage de la spatialisation dans les environnements de CAO demeure souvent décorrélé des processus compositionnels à proprement parler.

5.1. Observations préliminaires

Nous avons réalisé plusieurs entretiens avec des compositeurs ayant une expérience dans le domaine de la spatialisation, afin de mieux comprendre leurs besoins et proposer des outils facilitant le contrôle et l'écriture de la spatialisation. Nous avons ainsi étudié les représentations et les logiciels employés, et cherché à identifier les problèmes rencontrés lors de leur utilisation. Ces observations tendent à montrer que les compositeurs travaillent principalement avec des représentations visuelles de l'espace sonore, pour les esquisses comme pour les partitions finales. La figure 7 montre des exemples d'esquisses représentant des indications de trajectoires spatiales ainsi que la salle ou la disposition des haut-parleurs.

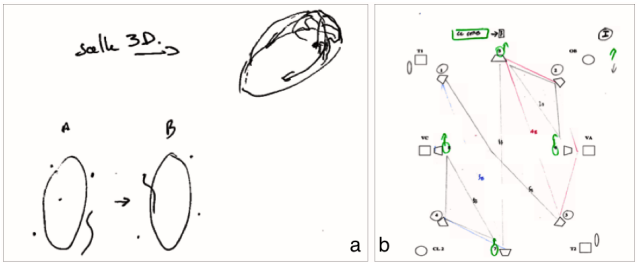

Figure 7: Esquisse de trajectoires spatiales. *a*) Dessins représentants une salle 3D et des vues de dessus avec différentes trajectoires. *b*) Schéma représentant les haut-parleurs et les trajectoires de sources sonores.

Malgré le caractère singulier de chaque démarche associant des paramètres graphiques à des notions variées et plus ou moins abstraites, la connexion de l'espace physique avec les représentations musicales traditionnelles (par exemple rythmiques) est également une constante parmi les problématiques rencontrées auprès des différents compositeurs. Plus généralement ces problématiques concernent la dimension d'écriture temporelle des processus, ainsi que les moyens pour de saisie, de visualisation et de rendu des trajectoires et autres paramètres spatio-temporels.

L'objectif des interfaces proposées est de présenter une vision à la fois formelle et intuitive, adaptée aux spécificités de l'environnement logiciel, des rapports temps-espace dans la spécification des scènes sonores.

5.2. Contrôle des scènes spatiales temporelles

Nous proposons un objet SPAT-SCENE permettant de créer, visualiser et modifier des scènes sonores et leurs évolutions dans le temps [14]. Cet objet est construit dans OpenMusic à partir d'un ensemble de sources et de trajectoires spatiales. La figure 8 illustre l'éditeur de cet objet, divisé en deux zones principales : la scène sonore (en haut) et la *time-line* (en bas).

La vue de la scène sonore permet de manipuler les sources sonores et les haut-parleurs à un instant donné. Cette interface provient du *Spatialisateur*, lié à notre environnement et avec lequel sont partagées un certain nombre de données et d'interactions. Nous avons fait le choix d'intégrer cette vue « classique » dans notre environnement, afin de faciliter la transition pour des utilisateurs ayant une expérience avec cette interface. La *time-line* contient, pour chaque source, des positions temporelles correspondant à un changement d'état ou de position. Ces positions temporelles sont équivalentes aux images-clés que l'ont peut retrouver dans les éditeurs de vidéo. Selon les cas, elle pourront être interpolées et/ou ré-échantillonnées lors du rendu.

En tant qu'objet « temporel », SPAT-SCENE pourra être utilisé au sein d'autres objets permettant d'agencer les éléments dans le temps (par exemple l'objet *maquette* existant

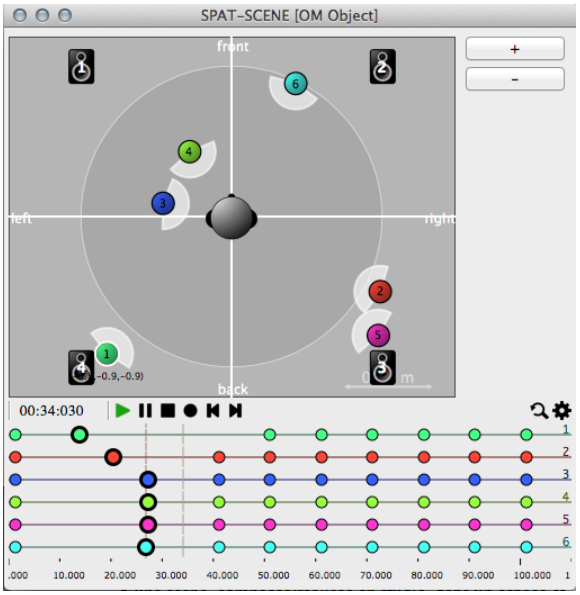

Figure 8: Interface graphique pour la visualisation et la manipulation de sources sonore spatialisées. *Haut* : interface de visualisation du *Spatialisateur*. *Bas* : *time-line* représentant les positions définies pour les sources dans le temps.

dans OpenMusic [7], ou le système d'ordonnancement dynamique décrit dans la section 3). Dans ce cas, l'objet graphique s'adaptera à la durée des trajectoires qu'il contient, et pourra présenter une ou plusieurs images-clés donnant un aperçu de l'évolution de la scène sonore. La figure 9 montre trois images-clés d'un objet SPAT-SCENE contenant trois sources.

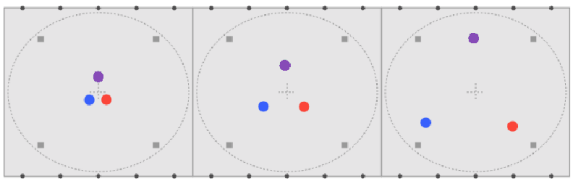

Figure 9: Visualisation par images-clés de l'évolution de la scène sonore. Les points colorés représentent les sources sonores pour trois instants différents.

Lors de la « lecture » de cet objet, l'évolution des paramètres et des descriptions spatiales peut être transmise via OSC à des processus externes réalisant la spatialisation (typiquement, l'objet Spat disponible dans Max). Ce type de procédé, transférant le résultat de processus compositionnels formalisés dans une relative interactivité au niveau du traitement et du rendu spatialisé, devrait favoriser l'apprentissage et l'exploration d'alternatives musicales. Une connexion plus directe pourra également être établie avec le moteur de rendu du *Spatialisateur* (cf. Section 4.3).

6. DISCUSSION ET PERSPECTIVES

Les récents développements présentés dans cet article esquissent un environnement de CAO combinant interfaces, structures de contrôle et moteurs de traitement audio, intégrés dans la temporalité dynamique d'un contexte réactif.

Bien qu'étant axée sur ces aspects interactifs, nous nous sommes gardés de confondre notre approche à celle d'un système « temps-réel ». Si la vitesse de calcul des ordinateurs actuels tend à confondre la perception du temps différé et du temps réel (du moins du point de vue de l'utilisateur), l'orientation « temps-différé » des outils de CAO, où le calcul des structures musicales n'est pas asservi à leur propre exécution, reste en effet selon nous une caractéristique fondamentale dans le développement de processus formels liés à l'écriture musicale. L'extension de cette approche dans un contexte réactif est ainsi complémentaire aux travaux récents tels que les bibliothèques *bach* et *cage* dans Max [1, 2], où le paradigme temps-réel structure des processus étendus au domaine symbolique de la CAO.

A l'heure de l'écriture de cet article, la plupart des travaux présentés existent à l'état de prototype. Le moteur réactif est intégré dans les versions récentes de l'environnement OpenMusic (version 6.9).

Les prochaines perspectives se concentrent sur les questions de représentation et transfert des données de contrôle entre environnements et dispositifs externes de rendu ou de captation, ainsi que sur les interfaces (graphiques, gestuelles) qui permettront un usage optimal et expressif des outils et technologies développées.

Le projet EFFICACe est financé par l'Agence Nationale de la Recherche (Ref : ANR-13-JS02-0004-01).

7. REFERENCES

[1] Agostini, A. , Ghisi, D. « Real-time computer-aided composition with bach », *Contemporary Music Review*, 32(1), 2013.

[2] Agostini, A. , Ghisi, D. « cage : a high-level library for real-time computer-aided composition », *International Computer Music Conference*, Athens, Greece, 2014.

[3] Assayag, G., Hanappe, P., Agon, C., Fineberg, J., « Problèmes de quantification et de transcription en composition assistée par ordinateur », *in* Genevois, H., Orlarey, Y. (dir.) *Musique & mathématiques*, Aléas éditeur, 1997.

[4] Bouche, D., Bresson, J., Letz, S., « Programmation and Control of Faust Sound Processing in OpenMusic », *International Computer Music Conference*, Athens, Greece, 2014.

[5] Bresson, J. « Sound Processing in OpenMusic », *International Conference on Digital Audio Effects – DAFx'06*, Montréal, Canada, 2006.

[6] Bresson, J., « Reactive Visual Programs for Computer-Aided Music Composition », *IEEE Symposium on Visual Languages and Human-Centric Computing – VL/HCC*, Melbourne, Australia, 2014.

[7] Bresson, J., Agon, C., « Temporal Control over Sound Synthesis Processes », *Sound and Music Computing Conference*, Marseille, France, 2006.

[8] Bresson, J., Agon, C., Assayag, G., « OpenMusic. Visual Programming Environment for Music Composition, Analysis and Research », *ACM MultiMedia (Open-Source Software Competition)*, Scottsdale, USA, 2011.

[9] Bresson, J., Giavitto, J.-L., « A Reactive Extension of the OpenMusic Visual Programming Language », *Journal of Visual Languages and Computing*, 25(4), 2014.

[10] Bresson, J., Schumacher, M., « Representation and Interchange of Sound Spatialization Data for Compositional Applications » , *International Computer Music Conference*, Huddersfield, UK, 2011.

[11] Cahen, R, Schwarz, D., Boissarie, X., Ding, H., Jacquemin, C., Schnell, N., Savary, M. Bourgeois, M.-J., Perennez, P., .Tanant, J., *Topophonie Research Project*, ENSCI, Les Ateliers, Paris, France, 2012.

[12] Freed, A., MacCallum, J., Schmeder, A. « A Dynamic, Instance-Based, Object-Oriented Programming in Max/MSP using Open Sound Control Message Delegation », *Proceedings of the International Computer Music Conference*, Huddersfield, UK, 2011.

[13] Garcia, J., Leroux, P., Bresson, J., « pOM – Linking Pen Gestures to Computer-Aided Composition Processes », *International Computer Music Conference*, Athens, Greece, 2014.

[14] Garcia, J., Bresson, J., Carpentier, T. « Towards Interactive Authoring Tools for Composing Spatialization », *3DUI'15 : IEEE 10th Symposium on 3D User Interfaces*, Arles, France, 2015.

[15] Giavitto, J.-L., « Du temps écrit au temps produit en informatique musicale », *in* Vinet, H. (dir.) *Produire le temps*, Hermann, 2014.

[16] Jacquemard, F., Bresson, J., Donat-Bouillud, P., « Rhythm Tree Rewriting », Meeting of the IFIP WG 1.6 on Term Rewriting, VSL, Vienna, Austria, 2014.

[17] Jacquemard, F., Donat-Bouillud, P., Bresson, J., « A Structural Theory of Rhythm Notation based on Tree Representations and Term Rewriting », *International Conference on Mathematics and Computation in Music – MCM 2015*, London, UK, 2015.

[18] Jot, J.-M., « Real-time Spatial Processing of Sounds for Music, Multimedia and Interactive Human-Computer Interfaces », *ACM Multimedia Systems Journal (Special issue on Audio and Multimedia)*, 7(1), 1997.

[19] Jot, J.-M., Chaigne, A., « Digital delay networks for designing artificial reverberators », *90th Convention of the Audio Engineering Society*, Paris, France, 1991.

[20] Jullien, J.-P., « Structured model for the representation and the control of room acoustic quality », *International Congress on Acoustics*, Trondheim, Norvège, 1995.

[21] Letz, S. « Spécfication de l'extension LibAudioStream », rapport technique projet INEDIT, 2014.

[22] Maire, A., « Quantification musicale avec apprentissage sur des exemples », mémoire de M1, École Normale Supérieure de Cachan, 2013.

[23] Nouno, G., Agon, C.,« Contrôle de la spatialisation comme paramètre musical », *Journées d'Informatique Musicale*, Marseille, 2002.

[24] Orlarey, Y., Fober, D., Letz, S., « Faust : an Efficient Functional Approach to DSP Programming », *in* Assayag, G., Gerzso, A. (dir.) *New Computationals Paradigms for Computer Music*, Editions Delatour, 2009.

[25] A. Schmeder and A. Freed, "Implementation and Applications of Open Sound Control Timestamps," in *Proc. Int. Computer Music Conf.*, Belfast, 1998.

[26] Schumacher, M., Bresson, J., « Spatial Sound Synthesis in Computer-Aided Composition », *Organised Sound*, 15(3), 2010.

[27] Schwarz, D., « Corpus-based concatenative synthesis », *IEEE Signal Processing Magazine*, 24(2), 2007.

[28] M. Wright, "Open Sound Control : an enabling technology for musical networking," *Organised Sound*, 10(3), 2005.