



K-Box : programmation d'une extension logicielle pour instrument percussif

Karim BARKATI
CICM - Université Paris 8
barkazik@club-internet.fr

Résumé : Cet article présente d'abord le logiciel K-Box, un instrument informatique sensible à l'intensité des attaques percussives et uniquement à ce critère, puis brièvement l'opéra *Richter, un opéra documentaire de chambre* pour lequel il a été développé, en collaboration avec son auteur, Mario LORENZO. Il décrit ensuite techniquement les algorithmes utilisés et leur organisation au sein du programme, en exposant les raisons des choix effectués au cours du développement de ce logiciel, sous l'environnement de programmation Max/MSP. Il propose enfin quelques axes de réflexion qui me paraissent fondamentaux en ce qui concerne la programmation d'outils numériques pour les compositeurs : la collaboration entre programmeur et compositeur, la notation en rapport avec ces nouveaux outils, le degré d'abstraction nécessaire à l'acte compositionnel, et le problème de la pérennité de ce type d'œuvres.

Mots-clés : interface logicielle, exécution musicale, temps réel, méta-instrument.

1. Introduction

Le développement du logiciel présenté ici a entre autres été motivé par la problématique suivante : plutôt que d'utiliser un clavier ou de construire une nouvelle interface de contrôle pour le temps réel, comment utiliser directement l'instrument du musicien ? En effet, un instrumentiste maîtrise son propre instrument bien mieux qu'une interface spécifique, et jouerait donc plus précisément une partie synthétique contrôlée directement par son instrument [1]. Par ailleurs, la grosse caisse ne permet de considérer que les nuances, paramètre à la fois hautement musical et plus aisé à détecter que la hauteur par exemple. Cette stratégie directe présenterait cependant un inconvénient majeur : la grosse caisse est un peu bruyante en tant que simple interface de contrôle, sauf quand le compositeur souhaite lui laisser son statut d'instrument acoustique pour profiter d'un nouveau rôle double. C'était précisément l'intention de Mario LORENZO dans son œuvre *Richter, un opéra documentaire de chambre*, dans le cadre de laquelle a été développé le logiciel K-Box, grâce à une intense collaboration. Ce logiciel apporte une dimension électroacoustique en sus de l'instrumentarium acoustique, par un système de reproduction d'échantillons en réponse au jeu de la grosse caisse. L'ensemble constitué de cette grosse caisse et du dispositif électroacoustique est dénommé le « Sol Cantante », ou « Soleil Chantant » en français. Nous présentons ici les fonctionnalités et la programmation du logiciel K-Box, afin de montrer les concepts, principes, et notions nécessaires à la programmation d'outils logiciels pour les compositeurs, et les intérêts compositionnels de tels outils.

Après un résumé *Richter, un opéra documentaire de chambre*, la première partie de cet article décrira l'architecture de ce logiciel, c'est-à-dire l'organisation technique des traitements informatiques et audionumériques ; l'environnement de programmation Max/MSP est propice



à ce type d'analyse, grâce à ses orientations graphiques et modulaires. La seconde partie mettra en évidence les méthodes de programmation impliquées dans ce type de développement logiciel, afin de dégager certains principes qui me paraissent fondamentaux.

Résumé de *Richter, un opéra documentaire de chambre* :

En 1948, le physicien autrichien Ronald Richter persuade le général Perón qu'il est capable de maîtriser la fusion nucléaire. Le président argentin, sensible au prestige des savants du Troisième Reich, accueille avec enthousiasme ce projet fantastique : reproduire dans un laboratoire le fonctionnement du soleil. La petite île Huemul, dans un coin paradisiaque de la Patagonie, est transformée en véritable bunker, où vont avoir lieu de mystérieuses expériences. Bientôt, l'Argentine annonce au monde la solution de tous les problèmes énergétiques de l'humanité. Les résultats, cependant, se font attendre, et Richter commence à agir de manière extravagante. Un groupe de scientifiques essaye de prouver que le soi-disant savant est un incapable et un mythomane. Leur chef, le docteur Balseiro, lui rend visite sur son île. Il y trouve un homme à la dérive dans un espace sombre et vide, qui prétend atteindre la température du soleil grâce aux sons produits par une série de haut-parleurs. Richter sera déclaré fou par un psychiatre et son centre de recherche tombera en ruine, mais ses appareils acoustiques seront récupérés par les créateurs du premier laboratoire de musique électronique d'Amérique latine.

Ces faits constituent la base de *Richter, un opéra documentaire de chambre*. Les expériences acoustiques du « savant », rapportées par des documents historiques, y sont développées dans le sens d'une parabole, avec, au centre, un instrument imaginaire : *El sol cantante*. Certains homonymes de Richter – notamment, le pianiste et l'inventeur de l'échelle sismologique – sont convoqués pour la construction du personnage principal. Celui-ci, un baryton, évolue en contrepoint avec son interprète d'espagnol, un rôle féminin, Balseiro, un ténor, ainsi qu'avec douze chanteurs qui tantôt agissent comme un chœur, tantôt comme des personnages secondaires. La partition compte en outre deux pianos, un dispositif de percussion et, bien entendu, des moyens électroacoustiques. L'œuvre dure une heure et quart environ, et se déroule dans un seul espace scénique.

2. Un programme temps-réel basé sur la détection d'attaques percussives

2.1. Architecture du logiciel

2.1.1. Présentation générale

K-Box est un logiciel d'informatique musicale. Plus précisément, il s'agit d'un déclencheur de sons échantillonnés [5], basé sur un système de détection et d'analyse en temps réel d'attaques de type percussives, et doté d'une polyphonie à cinq voies. Il « écoute » le jeu d'un instrument de percussion, et, en fonction de l'intensité de chaque attaque, il déclenche différents sons échantillonnés chargés en mémoire vive. Dans le contexte de l'opéra *Richter, un opéra documentaire de chambre*, on peut se représenter le logiciel K-Box et son attirail électronique comme une excroissance difforme de la grosse caisse, une sorte de greffon artificiel qui a besoin de la grosse-caisse pour s'exprimer, mais qui parle avec d'autres mots. Ce « symbiote » serait un double sensible et multiface, possédant autant de langues que le compositeur-créateur lui en donne. La version 1.61 du logiciel K-Box était suffisamment



stable pour faire nos premières répétitions avec une percussionniste professionnelle : Françoise RIVALLAND a apporté une grosse caisse de taille importante, et nous avons travaillé à définir les limites du système. au niveau des modes de jeu, des baguettes, des vitesses d'exécution, de la position du microphone, et des temps de réponse. Je tiens à la remercier, ainsi que Mario LORENZO pour son professionnalisme et son amitié.

Le logiciel se situe dans une chaîne matérielle précise, dont la grosse caisse et les haut-parleurs constituent les extrémités. Cet ensemble forme ainsi un nouveau corps vibratoire complexe, un nouvel instrument. Différents mécanismes de transduction [3] et de traduction sont mis en jeu à cet effet. La configuration matérielle se compose au minimum des éléments suivants : un microphone, un convertisseur analogique - numérique (CAN), un ordinateur Macintosh (à partir des modèles G3), un convertisseur numérique - analogique (CNA), et deux haut-parleurs, ou un système de diffusion plus complet. Le son circule à travers ces éléments sous trois formes différentes : acoustique, électroacoustique, et audionumérique. Il passe d'un état à un autre grâce aux différents transducteurs et convertisseurs impliqués dans cette chaîne. La grosse caisse n'est pas censée être amplifiée, car l'auditeur doit percevoir le son naturel de l'instrument directement, et n'entendre à travers les haut-parleurs que le son du « méta-instrument ».

Voici la fenêtre principale, qui permet essentiellement d'avoir un accès direct aux paramètres critiques, afin de pouvoir intervenir rapidement en situation de concert :

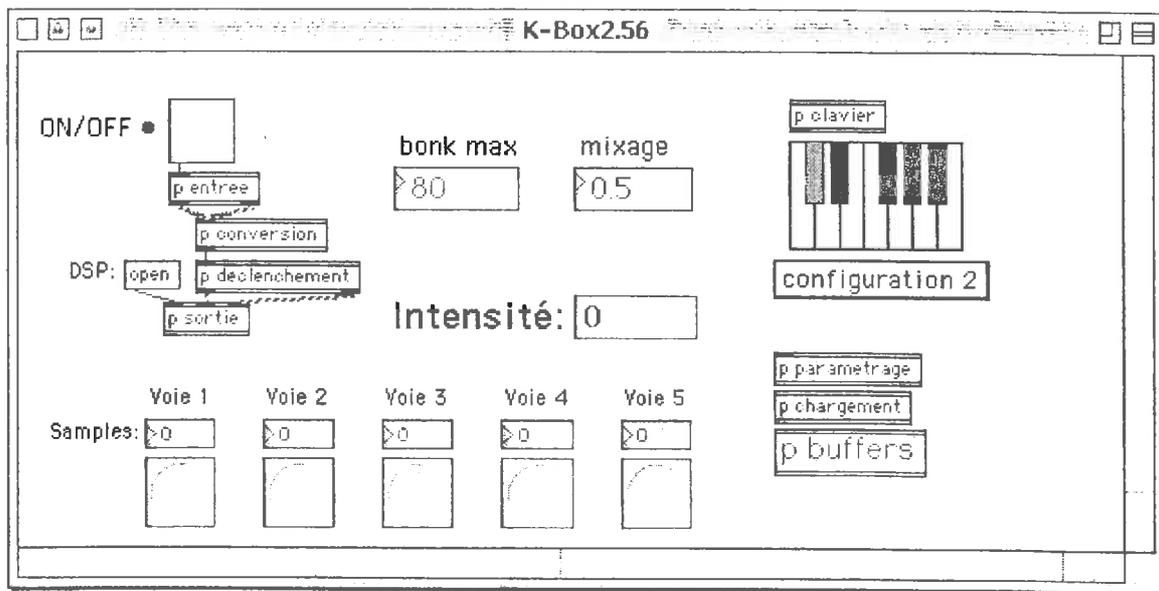


Fig. 1 : Fenêtre principale pour le contrôle en situation de concert

Le schéma suivant reprend le circuit placé en haut et à gauche de la fenêtre principale en indiquant la hiérarchie des patches, et en précisant les objets qui ont un rôle clé au sein de la chaîne de traitement. Sa simplicité et sa totale séquentialité garantissent la reproductibilité du programme.

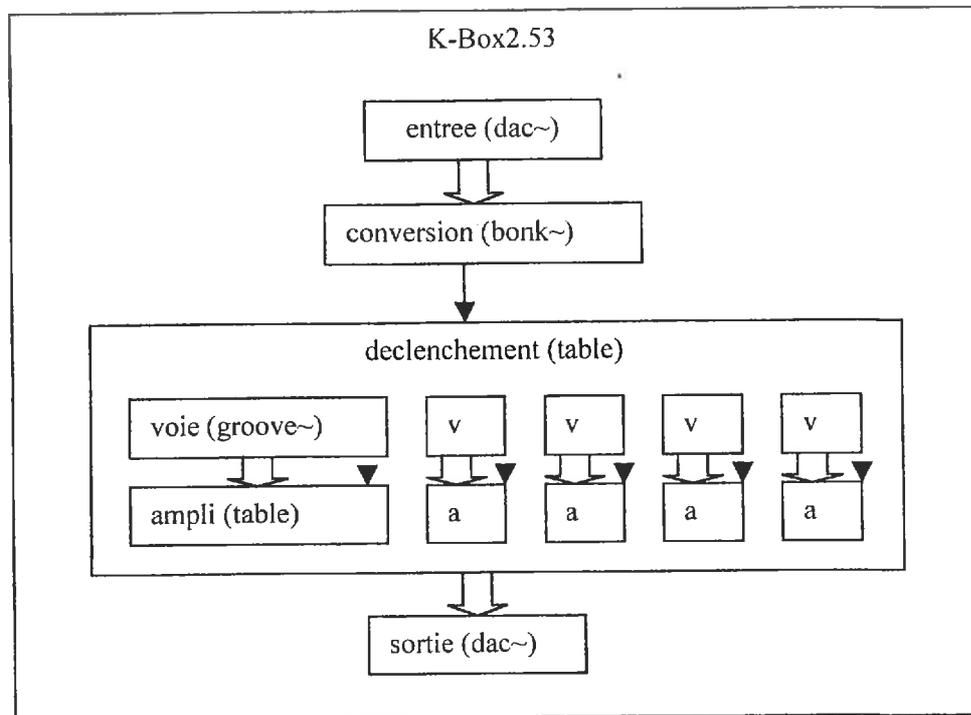


Fig. 2 : Organisation hiérarchique des modules

Les flèches épaisses indiquent la circulation de signal audio numérique, avec un débit constant de 44100 informations par seconde, alors que la flèche noire indique la transmission d'une information non audio, à un débit beaucoup plus lent et non systématique. Le parcours logique du signal s'effectue de manière totalement déterministe et séquentielle.

2.1.2. Description des algorithmes

Le logiciel K-Box s'articule autour des notions d'attaque et d'intensité d'attaque. Chaque attaque détectée dans le signal audio numérique est mesurée dans son amplitude, puis codée parmi 128 valeurs possibles. C'est cette information que nous appelons ici « intensité ». Dans cette version du logiciel, cette information contrôle deux traitements audio numériques : la lecture d'un sample prédéfini, et l'amplification de celui-ci. Nous décrivons dans cette partie à la fois les algorithmes propres au fonctionnement du circuit dépendant de l'intensité, et ceux qui relèvent des autres aspects contextuels à cette chaîne essentielle.

2.1.2.1. La détection et la conversion des attaques

Le patch conversion reçoit un signal audio numérique quelconque, et le convertit en temps réel en une suite discontinue de valeurs résultant de l'analyse de ce signal. Le seul objet `bonk~` suffit à détecter les attaques et à mesurer l'intensité de celles-ci le cas échéant. Le reste du patch sert à paramétrer cet objet, afficher différentes valeurs en situation d'ajustement, et à niveler les mesures d'intensité entre 0 et 127.



L'objet *bonk~* a été programmé par Miller Puckette, puis porté sur MSP par Ted Apel ; la version 1.1 que nous utilisons date de décembre 1999. Cet objet détecte les attaques des instruments percussifs ; elles sont définies par un changement de forme dans l'enveloppe spectrale. Optionnellement, il peut comparer une attaque avec quelques modèles stockés, pour essayer de deviner de quel instrument provient l'attaque. Cependant, les résultats de cette technique ne sont pas fiables. La description théorique de son fonctionnement est disponible à l'adresse <http://www.crea.ucsd.edu/~msp>. Les deux sorties de *bonk~* sont d'une part le spectre de l'attaque, sous la forme d'une liste de onze nombres correspondant à l'énergie du signal dans les onze bandes de fréquence utilisées, et d'autre part une seconde liste précisant le numéro de l'instrument s'il a été identifié, et surtout la « vitesse ». C'est cette unique valeur qui nous intéresse. Elle s'appelle ainsi car l'intensité de l'attaque, pour les percussions, est directement proportionnelle à la vitesse de la frappe. L'analyse de *bonk~* est effectuée sur une fenêtre de 256 points, soit 6 millisecondes pour une fréquence d'échantillonnage de 44100 Hertz, et répétée tous les 128 échantillons, soit toutes les 3 millisecondes.

2.1.2.2. *Le déclenchement des samples*

Lorsque l'information d'intensité a été codée, tout type de traitement est envisageable, en établissant une correspondance entre les valeurs d'intensité possibles et le domaine de définition du traitement à effectuer. La seule limitation réside dans les 128 valeurs de départ. Le premier traitement consiste ainsi à déclencher des samples préchargés en mémoire, dans un ordre préétabli par le compositeur dans une table de correspondance. La fenêtre du patch déclenchement peut être utilisée comme aide à l'apprentissage de l'instrument chargé dans le K-Box. La table de correspondance est dessinée par le compositeur et ajustée en répétition. Elle attribue une unique valeur, entre 1 et 99, à chacune des 128 intensités possibles. Cette valeur correspond au numéro du sample que le compositeur souhaite voir déclenché pour une intensité précise.

Un percussionniste ne possède généralement pas une précision suffisante pour maîtriser 128 nuances différentes. Il suffit pour s'en convaincre d'essayer d'obtenir consécutivement plusieurs fois la même valeur, puis sa voisine immédiate. D'autre part, le système dépend fortement des conditions de captation de l'instrument acoustique : qualité et placement du microphone, réglages de la table de mixage, bruit dans la salle, etc. Toutes ces imprécisions sont minimisées par les diverses possibilités d'ajustements mémorisables du logiciel, mais pas suffisamment pour supprimer un phénomène d'incertitude non négligeable. Cette incertitude conduit à la notion de zones perceptives d'intensité. Nous avons choisi d'en concevoir cinq, par exemple *ppp*, *p*, *m*, *f*, et *fff*, qui forment un compromis entre la précision maximum de l'instrumentiste après apprentissage, et la précision minimum de l'ajustement des paramètres. Il y a donc cinq voies qui correspondent à ces cinq zones, et chaque voie peut accéder à vingt samples : de 1 à 19 pour la première, de 20 à 39 pour la deuxième, de 40 à 59 pour la troisième, de 60 à 79 pour la quatrième, et de 80 à 99 pour la cinquième (fig. 4). Des développements futurs pourraient porter sur des modifications de ces paramétrages, comme le changement de vitesse ou le bouclage. Un objet *meter~* (VU-mètre) permet de contrôler visuellement la bonne marche de la lecture.



2.1.2.3. L'amplification des samples

L'amplification donne une dimension plus réaliste à la perception. C'est aussi, dans le cadre de cette recherche, un exemple extrapolable de traitement audionumérique basé sur l'unique information d'intensité. L'activation du traitement d'amplification repose sur le même principe que le déclenchement des samples : l'un des cinq patches ampli est choisi d'après le numéro du sample correspondant à l'intensité reçue, puis ce patch traite directement l'information d'intensité. L'amplification se fait alors simplement par multiplication du sample tel qu'il est enregistré en mémoire par le coefficient déduit de la table (fig. 5 et 6). Un objet `r antiClic` paramètre à distance la durée d'amortissement lors d'un changement brusque de coefficient, grâce à l'objet `line~`, afin d'éviter de produire des « clics » numériques.

2.1.2.4. La gestion des configurations

Le caractère multiface du logiciel K-Box provient de la possibilité de charger de nouvelles données en situation de concert. Chaque configuration contient à la fois les paramétrages et les nouveaux sons. Il est prévu que l'instrumentiste puisse changer de configuration à distance, à partir d'un contrôleur MIDI. En haut et à droite de la fenêtre principale apparaissent un clavier pour visualiser la note envoyée par le contrôleur, et un menu déroulant pour indiquer le nom de la configuration qui correspond à cette note. Le changement de configuration repose sur des systèmes de chargement de fichiers. Ces fichiers sont de quatre types :

- les configurations
- les tables
- les instruments
- les fichiers audionumériques

Le logiciel K-Box a été pensé comme une boîte à instruments, où un seul instrument peut « chanter » à la fois. On peut penser à un comédien qui change de personnage ou de masque au cours de la pièce, mais ne joue qu'un seul personnage à un instant donné. L'identité de l'instrument dépend beaucoup du potentiel de singularisation des samples qui le composent. De plus, les auditeurs identifient auditivement la « boîte » elle-même, en tant que processus réactif aux attaques percussives. Un changement de configuration entraîne un changement d'instrument, ce qui constitue sans doute la partie la plus perceptible, mais aussi un changement du paramétrage du logiciel et des deux tables d'association.

Le K-Box peut être défini comme un « méta-instrument », afin de rendre compte qu'il s'agit d'un instrument spécial [6]. Après quelques minutes de pratique se développe une conscience musicale du contrôle de cet instrument informatique, malgré le fait que ce contrôle soit indirect. Ce dernier point justifie l'emploi du préfixe « méta ». Quelques autres aspects renforcent cette idée d'abstraction, comme la vacuité latente du logiciel tant qu'une configuration n'est pas chargée, ou la possibilité d'une notation spécifique exhaustive. Enfin, le sens étymologique « après » correspond tout à fait au mécanisme caractéristique du K-Box, qui réagit *d'après* un signal émis, et *après* lui dans le temps, fût-ce de quelques millisecondes, à cause du temps nécessaire à l'écoute, aux conversions, et à l'analyse de ce signal externe.



2.2. La notation

« Il est absolument nécessaire que les compositeurs soient associés à tous les travaux de recherche, de façon à toujours donner le point de vue du musicien et du créateur, ce qui n'a pas toujours été le cas » affirme Tristan Murail dans un entretien avec Danielle Cohen-Lévinas [2]. En effet, cette collaboration est indispensable à plus d'un titre, en particulier en informatique musicale. Les connaissances musicales d'un programmeur, fût-il lui-même compositeur, ne peuvent prétendre se substituer à la richesse d'une collaboration. Le développement du logiciel K-Box nous a confronté dès l'analyse à des problèmes de notation et de représentation, pour pouvoir communiquer efficacement et précisément. Sa nature de boîte, même virtuelle, confère à ce logiciel une certaine gourmandise innée pour les informations à fournir en entrée. Il a donc fallu concevoir un système de notation pour structurer ces données, tel que cette notation soit la plus intuitive possible, et porteuse de sens. Ce système s'articule autour de plusieurs fichiers textuels et graphiques : la configuration, l'instrument, la table de correspondance, et la table d'amplitude. Pour le concevoir, nous avons tenu compte de la relation de ce système avec la partition, et de la propension à l'abstraction qu'il était capable de générer.

2.2.1. Le fichier de configuration

Le fichier de configuration est un fichier textuel précisant les valeurs d'une dizaine de paramètres, et les noms des autres fichiers à charger. Ce fichier centralise toute l'information, permettant ainsi de se représenter totalement un élément de la boîte avec un seul objet conceptuel. Ainsi, lorsque qu'une pièce fait appel à plusieurs configurations, le nom de la configuration à charger peut apparaître directement sur la partition.

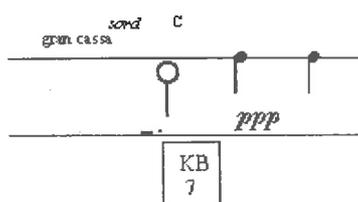


Fig. 3 : Notation d'un changement de configuration sur une partition de grosse caisse

Cette notion de configuration et de changement de configuration fait sens pour le compositeur, car elle correspond à une tradition d'écriture orchestrale de changement d'instrument dans une partition parmi une panoplie de départ, comme un changement de clarinette d'un clarinetiste ou un changement de percussion d'un percussionniste. Plus récemment, cette notion peut faire référence à l'instruction *program change* en MIDI.

2.2.2. Le fichier instrument

Le fichier instrument est aussi un fichier textuel. Il dresse la liste des noms des fichiers audionumériques à charger, ainsi que les numéros des buffers dans lesquels ils doivent être chargés. Cette représentation a l'avantage d'être exhaustive, puisqu'elle précise le nom de tous les samples qui seront chargés pour un instrument. Tout compositeur familiarisé avec les échantillonneurs assimile intuitivement cette représentation à la notion de *banque*. Le fichier



instrument diffère cependant des banques des échantillonneurs au moins par deux aspects. D'abord, les samples ne sont pas affectés à des intervalles de hauteurs (*keygroup*), ni ne répondent à des messages MIDI du type *note-on*. Ensuite, les numéros des buffers sous-tendent la notion de zone perceptive de nuance, qui sont au nombre de cinq dans ce programme. Comme le nombre de samples par zone est variable et que le maximum global est 99 samples, le système de notation permet de repérer les zones d'après le numéro des buffers, en sachant que la première commence à 1, la deuxième à 20, la troisième à 40, la quatrième à 60, et la cinquième à 80.

2.2.3. Les fichiers de tables

Les tables sont des fichiers graphiques, car Max propose des outils d'édition graphique de ces fichiers. Du point de vue strictement informatique, ces tables ne sont qu'une série de couples de valeurs associées, d'où leur nom technique de « tables d'association », mais ne possèdent pas de caractère graphique intrinsèque. L'aspect graphique recèle pour nous un trésor : le compositeur retrouve un outil calligraphique qui lui permet de penser des formes, et donc de retrouver l'abstraction si chère à son art. Une table possède deux dimensions : les valeurs d'entrée en abscisse, et les valeurs de sortie en ordonnée. On peut faire correspondre ainsi pour tout traitement un espace de réaction à un espace de perception. Le logiciel K-Box ne possède qu'un seul espace de perception et deux espaces de traitement, respectivement l'intensité des attaques percussives, codée sur 128 valeurs, le déclenchement d'un sample parmi 99 possibles, et l'amplification d'une zone, codée sur 128 valeurs. Les représentations graphiques des deux tables de correspondance et d'amplification forment des courbes prégnantes pour le compositeur, et superposables verticalement puisqu'elles sont définies à l'identique pour l'intensité en abscisse.

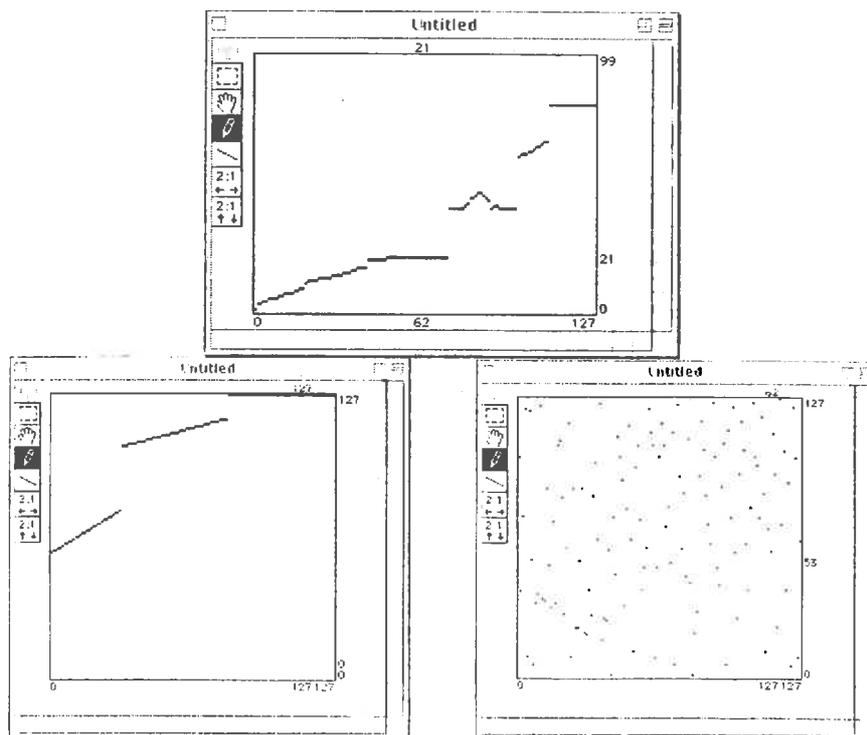


Fig. 4, 5, 6 : Tables de correspondance entre l'intensité des attaques et la liste des samples (fig. 4) ou l'amplification des samples (fig. 5 et 6)



Ces courbes sont directement liées aux samples chargés, et définissent la réponse acoustique du programme, la façon dont il va réagir à ce qu'il « entend ». La notion de zones perceptives exploitée dans le fichier instrument peut être renforcée ou détruite par une courbe. Le compositeur reste maître de ses choix sur la forme de la réaction du programme, et décide du degré de causalité perceptive dans sa notation. La table ci-dessus à droite illustre une volonté évidente de décorrélation.

2.2.4. La partition

La notation traditionnelle convient à cet instrument informatique qu'est le K-Box. Étant donné que le programme ne perçoit que la seule dimension musicale des nuances, la notation rythmique habituelle fait sens, ce qui n'est pas forcément le cas des systèmes plus répandus de déclenchement de samples basés sur les numéros de notes du contrôleur sans corrélation perceptive avec la hauteur du sample. La partition rythmique, sur une portée d'une ligne, n'est en rien mensongère puisqu'elle ne précise pas les hauteurs, et, par contre, elle renseigne sur le type de sample qui va être déclenché si le compositeur a établi un lien entre les nuances et l'identité musicale des samples.

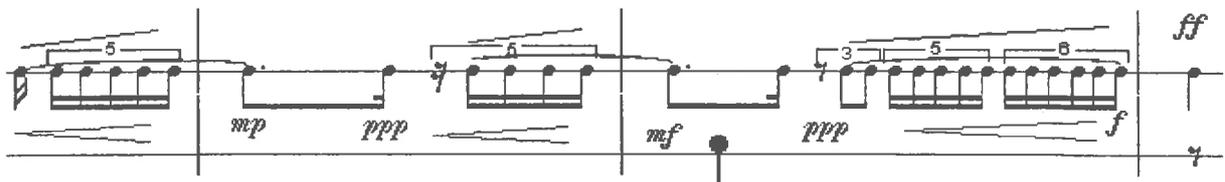


Fig. 7 : Extrait de la partition pour le « Soleil Chantant »

Il n'y a qu'une seule partition pour deux instruments, mais quelques astuces sont envisageables pour modifier l'équilibre, ou le rapport de force, entre le son direct de la grosse caisse et le son diffusé de son double informatique. En effet, on peut paramétrer une configuration pour rendre le programme insensible à certaines baguettes, ou encore modifier la courbe d'amplitude de façon à obtenir une réponse disproportionnée du programme à un jeu percussif pianissimo.

3. Vers la pérennité

Au dire des compositeurs, la pérennité des œuvres qui utilisent du matériel électroacoustique est menacée, voire impossible, notamment à cause de l'obsolescence rapide de ce même matériel. Certains compositeurs se sont même résolus à ce que leurs pièces ne soient jouées qu'une seule fois. Les bandes magnétiques se désagrègent, les ordinateurs sont obsolètes dès leur sortie, les programmes ne sont plus compatibles avec les nouveaux systèmes d'exploitation, et les systèmes dédiés disparaissent, en laissant derrière eux des pièces dont la maintenance coûterait un prix exorbitant. La lutherie électroacoustique [4] rencontre des difficultés supplémentaires dues au temps parfois considérable que représente l'apprentissage d'un nouvel instrument, de surcroît pour jouer une littérature contemporaine souvent difficile techniquement. Dans ce contexte, l'espoir d'une pérennité des œuvres contemporaines utilisant des instruments électroacoustiques semble vain.



Pourtant, nous avons trouvé des formes de solution au travers de cette expérience. Tout d'abord, le contrôle gestuel du K-Box ne nécessite pas d'apprentissage de gestes nouveaux de la part de l'instrumentiste, sinon d'une sensibilité nouvelle. Ensuite, le système de notation élaboré fournit une description exhaustive et relativement intuitive des informations spécifiques à chaque configuration, et à la façon de jouer cet instrument. Les fichiers textuels résistent parfaitement au temps, ainsi que les fichiers audionumériques stockés sur CD-ROM, qui trouveront toujours un logiciel de conversion en cas d'obsolescence du format. Enfin, le logiciel est reprogrammable dans n'importe quel environnement informatique apte à traiter du signal audionumérique, grâce à la description totale de son architecture par le schéma séquentiel suivant :

1. analyse de l'intensité des attaques percussives,
2. codage sur 128 valeurs,
3. déclenchement de la lecture des samples d'après la table de correspondance sur une polyphonie de cinq voies,
4. amplification par voie d'après la table d'amplification,
5. mixage des cinq voies.

La numérisation et le principe informatique de compatibilité ascendante permettent déjà de penser que les instruments informatiques peuvent durer plus longtemps que leurs parents électroacoustiques [7]. Quelques stratégies supplémentaires sont susceptibles d'augmenter leur espérance de vie, en portant la possibilité de dématérialisation à son maximum. Si la partition a été jusqu'à présent le seul lien avec les œuvres du passé, nous proposons de joindre à la partition d'aujourd'hui les informations qui décrivent totalement nos traitements informatiques : à la fois les données, sous forme textuelle, graphique, et audionumérique, et les algorithmes, par un schéma complet. Il devrait être ainsi possible de recréer ces pièces, dans un futur proche ou lointain. En d'autres termes, tout programme fonctionnant à une date donnée est reprogrammable ultérieurement, sous une forme éventuellement différente, et dont on a la certitude qu'il fonctionnera aussi. Il suffit alors que les instruments électroacoustiques se résument à de tels programmes connectés à des appareils triviaux, tels un microphone ou des haut-parleurs, pour qu'il devienne envisageable de recréer les pièces qui utilisent ces instruments, dans un futur proche ou lointain.

Références :

- [1] CADOZ Claude, *Le timbre, métaphore pour la composition*, éditions Christian Bourgois, Ircam, 1991
- [2] CAHIER DE L'IRCAM N°1, *Composition et environnements informatiques*, éditions Ircam - Centre Georges-Pompidou, 1992
- [3] JACOBSON Ricardo E., *Méthode pour l'enseignement des techniques du son*, Université Paris VIII, Département de Musicologie
- [4] LALIBERTÉ Martin, *Un principe de la musique électroacoustique et informatique et son incidence sur la composition musicale* École des Hautes Études en Sciences Sociales, 1984
- [5] LESBROS Vincent, *Atelier Incrémentiel pour la Musique Expérimentale*, Université Paris VIII, Département d'Informatique, 1992
- [6] MACHOVER Tod, « Classic » *Hyperinstruments*, 1992 DOBRIAN Christopher, *The MSP documentation*, University of California, Irvine, 1998



[7] ROADS Curtis, *L'Audionumérique*, version française : Jean de Reydellet, éditions Dunod, Paris 1998

Max et MSP :

<http://www.cycling74.com/>

Centre de Recherche Informatique et Création Musicale :

<http://www.ai.univ-paris8.fr/~vi/cicm/>

Institut International de Musique Electroacoustique de Bourges :

<http://www.gmeb.fr/>

Groupe de Recherches Musicales :

<http://www.ina.fr/GRM/>

Institut de Recherche et Coordination Acoustique - Musique :

<http://www.ircam.fr/>

Société Française d'Informatique musicale :

<http://www.sfim.org/>

Association pour la Création et la Recherche sur les Outils d'Expression :

<http://www-acroe.imag.fr/>





L'Interpolateur, une interface de contrôle multi-paramétrique

Daniel Teruggi

Groupe de Recherches Musicales, Institut National de l'Audiovisuel

116 Av du Prés. Kennedy 75220 Paris Cedex 16

dteruggi@ina.fr

Martin Spain

Faculty of Engineering and Information Sciences

Université de Hertfordshire

College Lane, Hatfield, AL10 91B

m.spain@herts.ac.uk

Résumé : Les interfaces de contrôle ont souvent imité des modèles inspirés des appareils analogiques du passé. Il est difficile dans ce contexte de contrôler plusieurs paramètres simultanément avec l'accès bi-dimensionnel de la souris. L'interpolateur propose une surface de contrôle dans laquelle des ensembles de données sont représentés par des cercles, permettant des interpolations complexes, basées sur le positionnement géographique des objets avec des champs d'action variables et une automatisation des déplacements.

Mots clés : Interpolation, interfaces, contrôle, multi-paramétrique, traitement du son, systèmes de contrôle graphique, GRM Tools.

Interfaces de contrôle

Dans le domaine de la manipulation virtuelle de sons, les interfaces jouent un rôle essentiel par leur implication dans le comportement des utilisateurs. Tout logiciel destiné à une utilisation technique ou de consultation, doit répondre à des contraintes bien précises, mais doit surtout proposer une interface d'utilisation adaptée à la fonction et simplifiant les tâches les plus usuelles.

Il est habituel dans ce domaine de s'inspirer d'analogies gestuelles issues d'expériences quotidiennes ou dictées par des pratiques manuelles. La virtualité des interfaces a été développée à partir de reconstructions ou simulations des modes de contrôle habituellement utilisés pour contrôler des systèmes physiques. Ainsi, dans le domaine du traitement du son, les interfaces graphiques ont simulé les boutons et potentiomètres linéaires des synthétiseurs et boîtes de traitement du passé, cela leur apportait une immédiateté opérationnelle et une clarté de compréhension.

Cette approche de la représentation a imprégné le développement des interfaces de contrôle, surtout ces dernières années, étant donné la facilité de programmation d'interfaces graphiques et l'argument indispensable qu'elles constituent pour tout produit destiné à une commercialisation. La reconstruction virtuelle d'interfaces physiques présente un problème majeur et c'est celui de l'accès de contrôle. Dans un système physique, plusieurs boutons ou réglettes peuvent être contrôlés par les différents doigts des deux mains ; dans une interface informatique l'accès privilégié reste la souris, ce qui limite le nombre de paramètres contrôlables au même temps.



Les systèmes inspirés de la réalité se révèlent alors efficaces si le nombre de paramètres à contrôler reste peu nombreux ; dès qu'une vingtaine de paramètres sont présents dans un système, leur contrôle par un seul paramètre physique devient complexe et décourageant.

Par ailleurs l'approche multi-paramétrique pour contrôler un système complexe n'est pas toujours la meilleure pour contrôler un système. Les algorithmes de traitement et de contrôle présentent toujours un nombre, plus ou moins important, de variables à contrôler (variables qui vont devenir des paramètres dans le contexte de l'interface). Le résultat de l'algorithme peut ne pas être de type paramétrique ; si l'algorithme propose la gestion d'un nuage d'évènements sonores, contrôler chaque événement individuellement peut s'avérer impossible à partir de paramètres simples. Il faut alors des paramètres complexes ou bien des macro-paramètres permettant un contrôle global du système.

Les deux problématiques principales liées au développement des interfaces sont : comment créer des interfaces inspirées des particularités de l'accès de contrôle que représente la souris, et comment créer des systèmes de contrôle global en présence d'une multitude de variables à contrôler.

Le modèle GRM Tools

Le développement des algorithmes a rendu les paramètres de contrôle de plus en plus nombreux et il a fallu trouver des solutions pour les contrôler à partir de l'accès bidimensionnel de la souris. Parmi les solutions trouvées, outre celle d'éliminer les paramètres pouvant être considérés comme superflus, était celle de grouper de nombreux paramètres identiques en un ou deux paramètres généraux qui permettront soit de les contrôler tous en parallèle, soit d'introduire des facteurs de séparation entre les paramètres de manière à introduire une différenciation progressive.

Cette approche globale du contrôle paramétrique, tout en simplifiant le contrôle, introduit un mode opératoire général. Au lieu de déterminer chaque paramètre individuellement, une fonction conditionne le fonctionnement du système, fonctionnement dont le détail reste invisible à l'utilisateur, et le résultat est une action multi-paramétrique sur le son qui altère globalement sa constitution. Pour un ensemble d'oscillateurs, par exemple, on déterminera l'écart entre chaque fréquence et ainsi on contrôlera toutes les fréquences. En mode opératoire, pour mieux saisir le fonctionnement du système, une représentation graphique de l'action permet de comprendre l'implication de la variation du paramètre.

Les GRM Tools¹ ont été construits autour de cette approche qui cherche à réduire le nombre de paramètres tout en les rendant très puissants en termes de conséquences opérationnelles et expliqués à travers des fenêtres dynamiques qui rendent compte des changements principaux. Le nombre de paramètres par algorithme reste relativement réduit (moins de 20) et la combinatoire riche et extrêmement variée.

Un deuxième aspect concernant les algorithmes réside dans la possibilité de capturer l'état de l'ensemble de paramètres à un instant donné, pouvant être mémorisés dans un « setting » ou mémoire d'état. Plusieurs settings peuvent être créés sur chaque algorithme et rappelés de manière instantanée. Ainsi, une configuration du système est réalisée pour l'adapter aux contraintes de l'utilisateur. Un passage progressif d'un setting à un autre peut être réalisé, à

¹ GRM Tools©, logiciel de traitement développé par l'Ina-GRM destiné aux environnements TDM et RTAS de Digidesign, et à l'environnement VST de Steinberg.

vitesse de changement constante ou contrôlée manuellement ; permettant ainsi de trouver des états intermédiaires entre deux états prédéterminés.

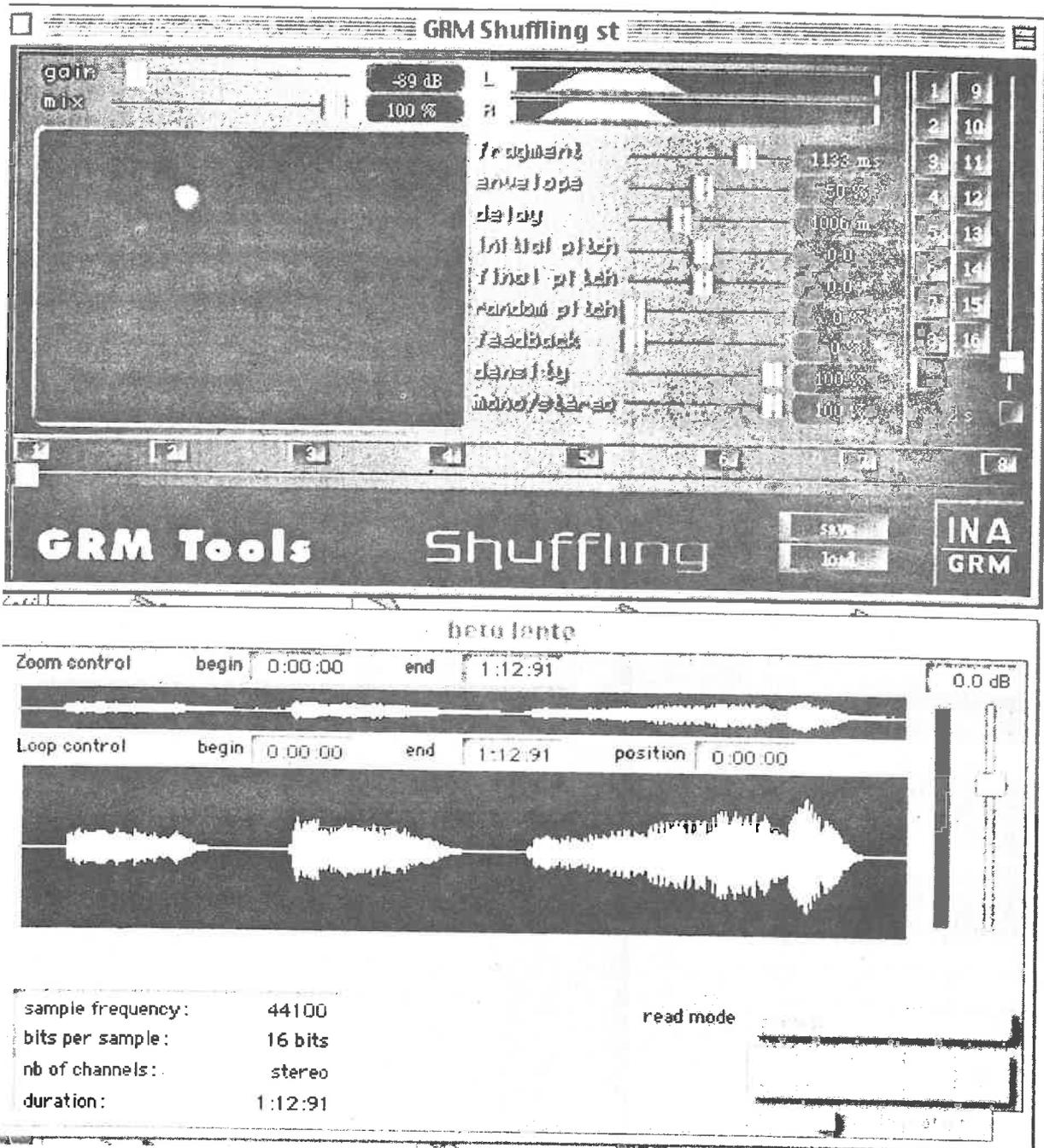


Fig. 1 Environnement de travail GRM Tools, en haut est présenté le plug-in de brassage *Shuffling* avec l'interface bidimensionnelle de contrôle permettant le contrôle simultané de la durée des fragments et de leur position temporelle (paramètres fragment et delay), les 16 cases à droite représentent les mémoires d'état ou « settings ». Plus bas se trouve le player permettant la lecture et enregistrement des sons.

L'Interpolateur

Un autre niveau de travail consiste à faire fonctionner plusieurs algorithmes simultanément sur un même son. Dans ces conditions, le nombre de paramètres devient rapidement très



important et, chaque algorithme devant être modifié à tour de rôle, le contrôle général devient impossible.

Un projet de recherche a été mis en place avec l'Université d'Hertfordshire² à Hatfield en Angleterre, pour développer un outil d'interpolation général pour l'ensemble des algorithmes de GRM Tools. Cet outil permet de mémoriser l'état des paramètres de l'ensemble des algorithmes présents sur GRM Tools et de les placer dans un espace bi-dimensionnel dans lequel des relations de type topographique vont se développer entre ses mémoires, cet espace est appelé espace de contrôle. Un champ d'action est établi autour de chaque mémoire et des interpolations très fines peuvent être réalisées entre plusieurs ensembles de valeurs. Deux types de champ d'action peuvent être établis, les champs circulaires, qui réalisent une interpolation dans toutes les directions à partir du point central, et les champs limités, appelés « projecteurs » qui orientent l'action dans une direction avec un degré d'ouverture et un rayon d'action. Des constructions topographiques très complexes peuvent être construites, les différentes couleurs indiquent des ensembles de paramètres différents associés à cette couleur.

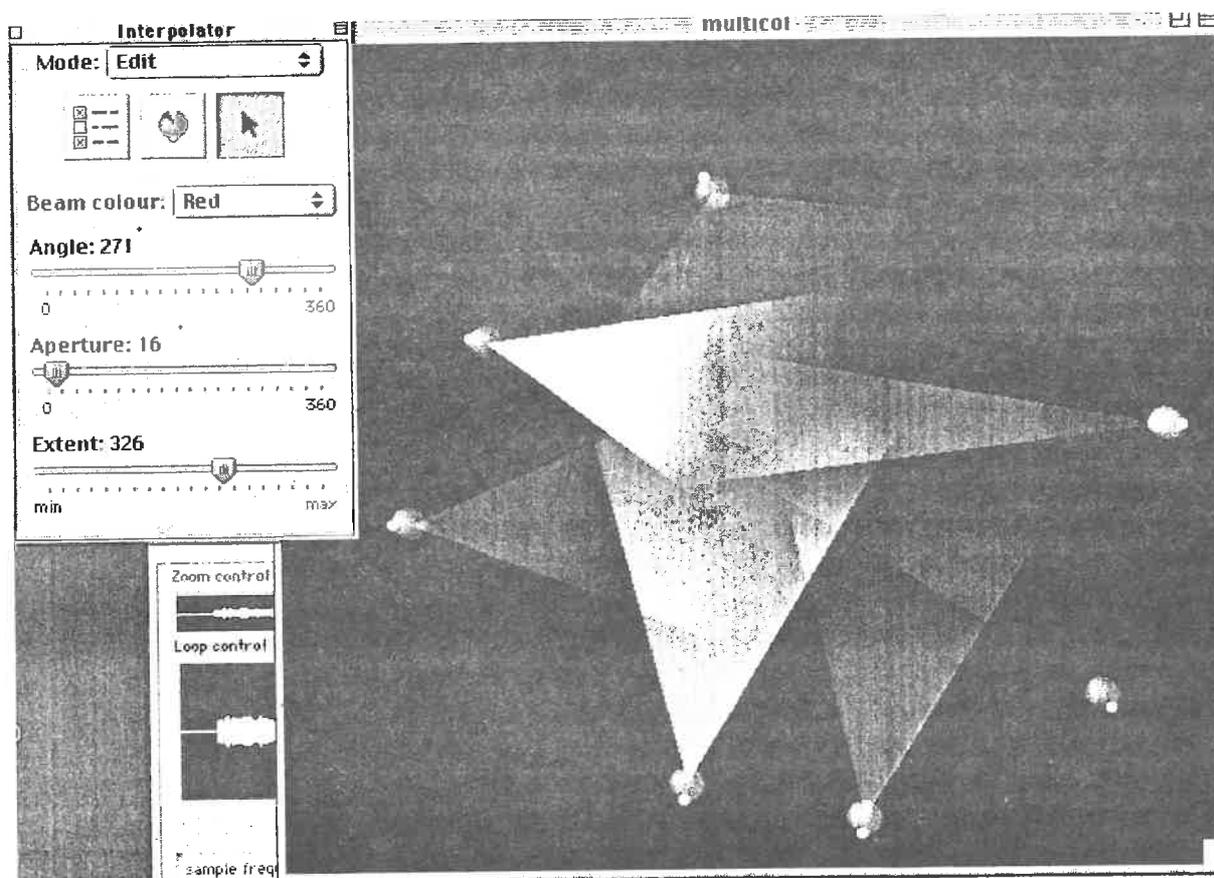


Fig. 2 Interface de contrôle de l'Interpolateur, en haut à gauche se trouve la fenêtre de configuration des paramètres. L'espace central noir où sont placés les projecteurs de différentes couleurs est l'espace de contrôle à l'intérieur duquel se réalisent les interpolations.

Sur l'interface, des ensembles de valeurs des paramètres en provenance d'algorithmes différents de GRM Tools (jusqu'à 4 plug-ins) représentent un état particulier de configuration. Cet état, représenté par un objet graphique constitué d'une sphère grise à partir de laquelle part un rayon de couleur, est placé dans l'espace bi ou tridimensionnel de l'interpolateur (Fig.

² L'université d'Hertfordshire est le seul centre en Grande-Bretagne réunissant les disciplines de : Informatique, Musique, Mathématique et Ingénierie électrique, orientées vers la recherche en Science et Technologie.



2). Des parcours contrôlés manuellement par la souris ou automatisés avec un séquenceur, sont alors possibles entre les différents états. Ces parcours vont provoquer des interpolations entre les différents états selon de modes de comportement différents. Si les états sont représentés par des champs circulaires (Fig. 3), alors l'interpolation est de type gravitationnel, et toutes les mémoires interagissent entre elles si leur rayon d'action se croise. Si les états sont représentés par des champs limités ou « projecteurs (Fig.2), alors l'interpolation se réalise à l'intérieur du rayon dégradé. En quittant le rayon du projecteur un changement abrupt aura lieu.

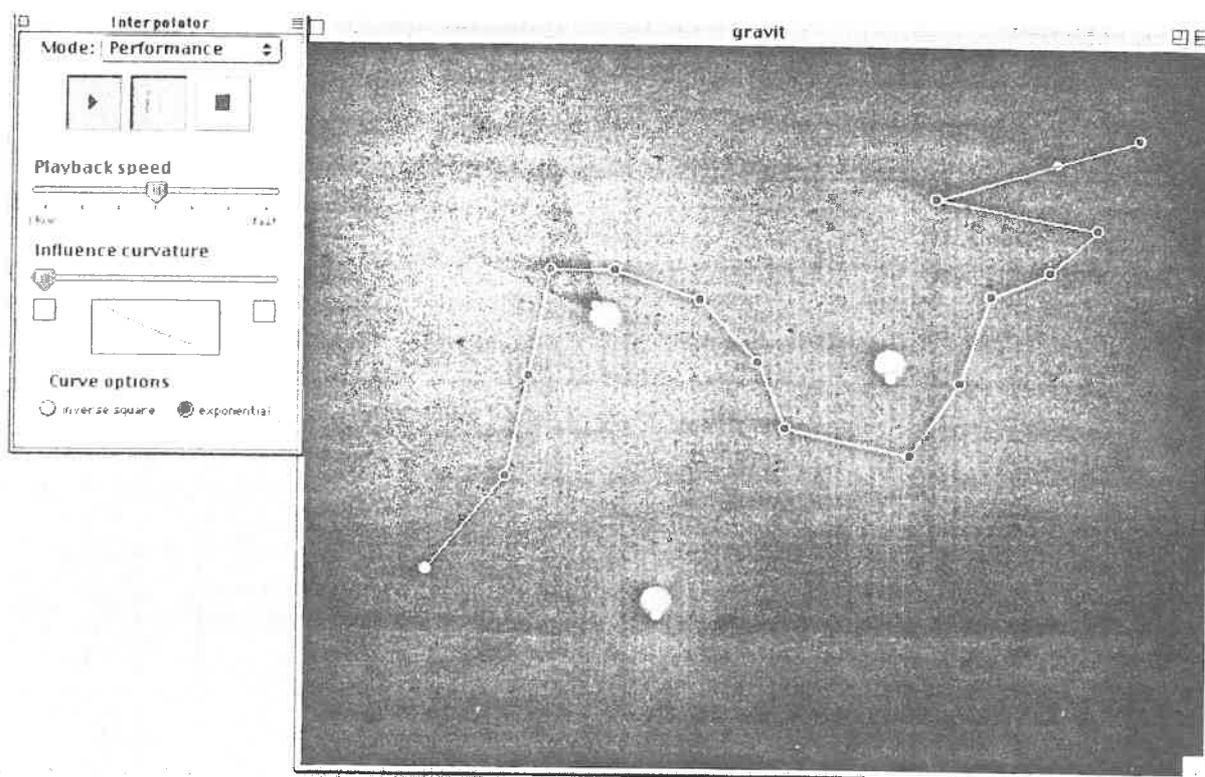


Fig. 3 Autre configuration de l'espace de contrôle avec des champs circulaires et un parcours séquencé qui peut être effectué à différentes vitesses.

L'espace de contrôle (carré noir), de taille variable, est un espace ouvert dans lequel sont incorporées les mémoires représentant les paramètres de différents plug-ins de GRM Tools. Une valeur par défaut lui est attribuée avant de placer les différents états ou mémoires (cette valeur par défaut peut être par exemple un état neutre où aucun traitement est réalisé sur le son). Les plug-ins de GRM Tools sont alors configurés selon les critères de l'utilisateur et un état ou mémoire est créée sur l'espace de contrôle ; cet état reprend les valeurs des paramètres des différents plug-ins. Une sélection peut être réalisé de telle sorte à ne capturer que certains paramètres ou bien en ne mémorisant que certains plug-ins. À ce moment-là des mémoires de couleur différentes sont créés (au choix de l'utilisateur) pour ainsi différencier des états qui contiendraient certains plug-ins et pas d'autres. Finalement la largeur du rayon est fixée ainsi que la longueur de son rayon d'action. Une fois les mémoires créées elles sont placées à différents endroits de l'espace de contrôle et l'on peut ; soit les rappeler, soit explorer l'espace existant entre elles. Si des rayons se croisent, une interpolation entre 2 ou plusieurs états se réalisera ; si une mémoire est isolée, l'interpolation se fera entre ses valeurs et les valeurs du fond données au départ.

Pendant l'interpolation, à tout moment, un nouvel état peut être créée. Ce nouvel état va modifier l'équilibre général du système. Un changement de position géographique d'un état apportera aussi un changement dans l'équilibre du système. Le système peut s'utiliser pour réaliser des interpolations entre deux états (passage progressif d'un ensemble de valeurs à un



autre) ou bien comme un espace d'exploration dès que plus de deux états interfèrent entre eux. Dans ce cas, il est possible d'obtenir des résultats tout à fait inattendus, nés de l'interaction entre les différentes valeurs.

D'un point de vue musical, un ensemble mémorisé dans l'interpolateur représente une décision à caractère musical. Une décision est une configuration intelligente d'un système selon des critères d'organisation propres à l'utilisateur ; celui-ci fixe des états des plug-ins et personnalise ainsi le système en fonction de ses objectifs ou désirs musicaux. Plusieurs personnalisations ou décisions sont introduites dans l'interpolateur et un mode de circulation est établi entre les décisions. Le résultat d'une telle action peut revêtir déjà un caractère musical abouti bien que très souvent le résultat soit une pré-organisation de type musical, c'est-à-dire un enchaînement d'événements établis selon les critères de l'auteur. La musique pouvant représenter un état encore supérieur de combinaison d'organisations sonores intentionnelles de ce type.

Concernant le passage d'un ensemble de données à un autre, les interpolations sont relativement faciles à percevoir dès qu'il s'agit de variations de hauteur ou d'intensité, mais elles deviennent difficiles à prévoir dès que les interpolations concernent d'autres types de paramètres dont le comportement est moins évident pour notre perception (des paramètres du type *random*, des paramètres d'écartement de fréquences ou des réinjections ont un comportement souvent difficile à suivre de manière linéaire, surtout s'ils sont combinés). La particularité du travail avec le son, dans lequel la perception d'une évolution ou d'une rupture dépend des phénomènes de perception, font que la linéarité d'une interpolation n'est pas forcément le meilleur modèle de description d'une variation. Très souvent une variation continue apportera un comportement de type "catastrophique" pour notre perception, de ce point de vue, l'interpolateur est un superbe outil pour faire émerger des comportements imprévus, toujours construits autour de la configuration du système réalisée par l'utilisateur en vue de l'obtention d'un résultat à caractère musical.

Une première version du logiciel a été développée par Martin Spain, étudiant en licence à l'Université de Hertfordshire, sous la direction de Richard Polfreman. Pour cela une version spéciale des GRM Tools a été développée par Emmanuel Favreau, pour pouvoir permettre le contrôle des paramètres à des ressources extérieures au logiciel. Cette première version a été finie en novembre 2000.

Dans cette première version, il est possible de capturer l'ensemble des paramètres et les organiser dans l'espace bi-dimensionnel de la surface de contrôle. Des interpolations peuvent ensuite se réaliser manuellement entre chaque mémoire et automatiquement à l'aide d'un séquenceur. Pour cette dernière opération, plusieurs points reliés par des traits sont placés dans l'espace et cette trajectoire est parcourue à des vitesses variables permettant de reproduire un parcours et de le modifier progressivement (voir Fig. 3).

Une phase de test est réalisée actuellement auprès des utilisateurs, ce qui permettra de vérifier son efficacité en termes de production et déterminer les améliorations à effectuer. Il faudra par la suite envisager une version intégrée à GRM Tools et étudier les possibilités d'utilisation de l'interface de contrôle sur d'autres applications y compris des applications non liées au contrôle de paramètres sonores.



Bibliographie :

E. Favreau, G. Racot, D. Teruggi, Evolution des outils, évolution des idées, dans Interfaces homme-machine et création musicale, Hermès science, Paris, 1999.

D. Teruggi : Le système Syter, Thèse de Doctorat, Université de Paris VIII, Saint-Denis, 1998.

D. Teruggi, L'interactivité dans les processus de création sonore, dans Interfaces homme-machine et création musicale, Hermès science, Paris, 1999.

