

CBOX : ENTRE TEMPS REEL ET TEMPS DIFFERE

Karim Barkati et Mario Lorenzo

CICM - Université Paris VIII
MSH Paris Nord

karim.barkati@free.fr

mariolorenzo@fr.st

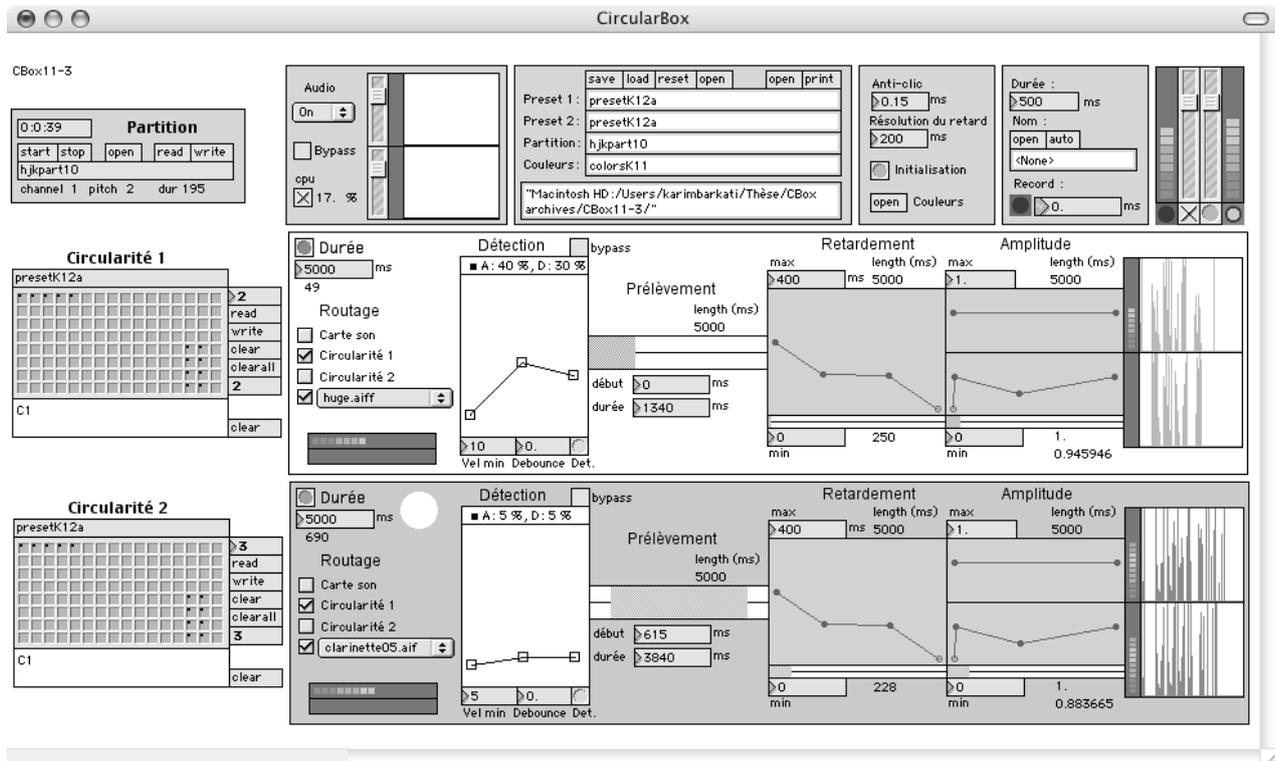


Figure 1. Interface graphique de la CircularBox11-3 jouant une partition programmée.

RÉSUMÉ

Nous souhaitons présenter ici une partie de nos recherches universitaires dans leur état actuel de « work in progress », non pour démontrer un résultat définitif, mais pour ce qu'elles mettent en relief des questions qui nous intéressent : autour du temps réel et du temps différé, de la circularité, de la réactivité logicielle, et du rapport entre micro temps et temps réel.

Ainsi, nous étudierons d'abord la mixité logicielle entre temps réel et temps différé à travers l'exemple de la CBox, un programme que nous avons développé dans l'environnement de programmation Max/MSP. Ensuite, nous présenterons la circularité d'un point de vue conceptuel, puis d'un point de vue plus pratique, telle qu'elle est implémentée dans ce logiciel, pour distinguer deux types structurels de bouclages dynamiques et réactifs. Enfin, nous reposerons la question du rapport entre le micro temps et l'informatique temps réel en apportant des éléments de réponse guidés par un souci musical.

1. INTRODUCTION

L'idée de *circularité* proposée par Francisco Varela [19] pour caractériser les systèmes autonomes a été le point de départ d'une réflexion sur quelques problématiques contemporaines et du développement de notre logiciel, dont la version actuelle se nomme CBox11-3 (pour CircularBox).

Bien que nous nous soyons inspirés du paradigme connexionniste et en particulier de l'étude des systèmes auto-organisés de Varela et Maturana [8], il est important de préciser que notre approche ne vise pas à établir des modèles¹, ni à s'opposer aux applications

¹ Nous pensons par exemple aux premières expériences de créativité artificielle (Hiller, Barbaud, Xenakis), plus récemment aux automates cellulaires de Miranda [9], et à l'implémentation d'agents autonomes en Max/MSP de Malt [7]

d'orientation plus déterministes. En effet, en tant que musiciens, nos travaux relèvent essentiellement de la recherche en composition assistée par ordinateur (CAO), à la croisée de l'algorithmique, de l'interaction et de l'interprétation : au sein des techniques de synthèse et de transformation du son, il s'agit de trouver des alternatives aux approches déterministes, en incorporant, dans une certaine mesure, des idées cognitives susceptibles d'apporter de nouvelles réponses à nos problématiques musicales.

CBox est un logiciel programmé dans l'environnement Max/MSP, à la fois temps réel (TR) et temps différé (TD). Essentiellement, il s'agit d'un circuit composé de deux *processus circulaires*, ayant tous les deux exactement la même architecture et pouvant être connectés entre eux ou simplement à eux-mêmes (feedback), ou dans d'autres combinaisons possibles selon les entrées utilisables (Carte son, C1, C2, Fichiers son).

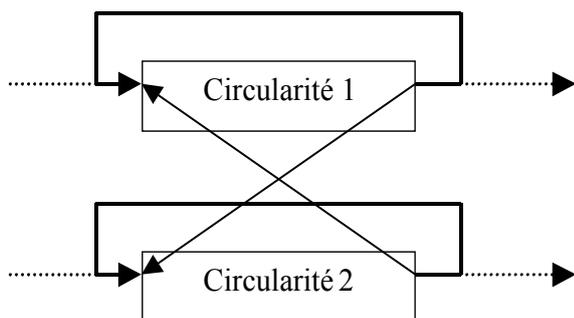


Figure 2 : schéma des branchements circulaires.

Du point de vue de la lutherie informatique, le logiciel CBox pourrait être considéré comme un instrument à la fois composable et interactif. Il fonctionne en temps réel et reçoit le signal sonore pour le transformer dynamiquement (cf. Fig.3) par des opérations de :

- routage (avec l'objet *gate~*)
- détection d'attaques (avec l'objet *bonk~*)
- prélèvement (avec *line~* et *delay~*)
- retardement (avec *tapin~* et *tapout~*)
- réinjection. (avec *send~* feedback)

Ces opérations sont doublées en deux modules identiques, les circularités C1 et C2. Les paramètres sont enregistrés puis rappelés depuis l'objet graphique *preset*. Finalement, toutes les configurations stockées dans les deux *presets* peuvent être déclenchées par une partition MIDI à deux voix (grâce à l'objet *detonate*), ou en temps réel par un instrumentiste à l'aide du clavier alphanumérique (objet *key*).

2. MIXITE TEMPS REEL ET TEMPS DIFFERE

« Temps réel » et « temps différé » sont deux expressions souvent opposées dans le champs de l'informatique musicale pour distinguer des applications selon le contexte des pratiques musicales, sonores et compositionnelles. Le dictionnaire des arts médiatiques¹ donne la définition suivante du temps réel :

« Modalité temporelle des systèmes de traitement de l'information dans lesquels il n'y a pas de délai entre la sortie d'informations et l'entrée de données, ou, si l'on veut, dans lesquels l'output suit immédiatement l'input. Le temps réel est caractéristique du mode interactif. (...) Certains traitements ou synthèses sonores informatiques exigent trop de calculs pour être effectués en temps réel. On obtient donc le résultat avec un délai qui peut parfois être très long : c'est le « temps différé ». Mais la puissance actuelle des ordinateurs est telle que de plus en plus de systèmes de synthèse réagissent en « temps réel », ce qui permet au compositeur d'avoir un contrôle perceptif immédiat et continu de son travail. »

De notre point de vue de musiciens, la pratique habituelle amène à considérer globalement que le « temps réel » désigne les situations de concert ou de sonorisation où l'informatique intervient directement et activement, avec une instantanéité apparente, tandis que le « temps différé » désigne davantage les situations de studio associées au temps de l'acte compositionnel et de la réflexion. Nous étudierons à travers le cas de la CBox une approche mixte, où TR et TD coexistent de façon nécessaire.

2.1. Temps différé

« (...) les propositions du temps réel manquent de variété, et elle incitent à des réactions réflexes et donc irréfléchies, donnant souvent lieu à des stéréotypes vite usés ou périmés. » [13]

Historiquement, les applications informatiques étant limitées par la faible puissance de calcul des premiers ordinateurs, les premiers logiciels ne pouvaient calculer du signal sonore qu'en temps différé, c'est-à-dire qu'il fallait d'abord saisir les données, puis lancer l'opération de calcul du son, et enfin attendre jusqu'à des heures entières avant de pouvoir l'écouter, et éventuellement recommencer en modifiant quelques paramètres si le résultat n'était pas satisfaisant... Citons Music V et Csound parmi les logiciels historiquement marquants.

En dépit de cet inconvénient de l'attente, aujourd'hui largement réduite, le « temps différé » s'avère incontournable à l'heure de composer. En effet, le retour sur des choix compositionnels, les connexions multiples entre différentes échelles temporelles, l'inclusion de singularités, le travail détaillé au niveau du micro temps, sont parmi les « calculs » dont le temps réel ne fournit que des « résultats » partiels. Car la composition musicale, telle que nous l'envisageons, est intimement liée aux aspects cognitifs, où l'interaction a

¹ Dictionnaire des arts médiatiques © 1996, Groupe de recherche en arts médiatiques - UQAM

un rôle central. Dans ce sens, la possibilité d'inclure l'improvisation à la composition grâce au temps réel, certes peut apporter des réponses mais l'encadrement doit être précis. Le risque de tomber dans des gestes stéréotypés, non réfléchis, est grand. Nous reviendrons plus tard sur cette position.

Il s'ajoute une autre contrainte, cette fois-ci liée à la problématique du marché : le phénomène d'obsolescence propre à l'informatique, qui risque de rendre les œuvres « périssables » pour reprendre l'expression de Jean-Claude Risset¹.

L'un des enjeux de ce projet réside dans la possibilité de composer entièrement une pièce avec la CBox, c'est-à-dire de disposer du maximum de fonctions, de commodités d'écriture et de mémorisation des paramètres, d'espaces de structurations multi-échelles, pour en faire un outil de composition à part entière, complètement envisageable comme un outil « temps différé ». Plusieurs mécanismes ont été implémentés dans ce but :

- la partition MIDI, avec l'objet *detonate*
- l'interface graphique de saisie des paramètres, avec des enveloppes, des espaces pour les nombres et pour du texte (commentaires)
- la mémorisation et la sauvegarde des paramètres, avec les objets *preset*
- la mémorisation et la sauvegarde des différents fichiers utilisés, dans un unique fichier de démarrage
- l'enregistrement du résultat sonore sur disque dur

2.2. Temps réel

Un deuxième enjeu important de ce logiciel réside dans la possibilité de l'utiliser en situation de concert, qu'il s'agisse d'improvisation ou bien d'interprétation d'une pièce écrite avec une partition « mixte » pour le programme et le ou les interprètes. On peut imaginer d'autres applications, comme pour des installations sonores, puisque la CBox est capable de réagir à l'environnement sonore en détectant des attaques dans le signal d'entrée (cf. ci-dessous 3.1 « Écoute » et réactivité logicielles).

Le temps réel possède des intérêts non négligeables au regard de nos préoccupations musicales. Depuis que la puissance de calcul des ordinateurs le permet, beaucoup d'applications temps réel sont apparues - traitements, effets, plug-ins, interfaces interactives - ainsi que plusieurs environnements de programmation dédiés au temps réel comme Max/MSP, SuperCollider,

¹ « Le véritable problème est celui de l'évolution très rapide des machines commerciales assez puissantes pour traiter le son en temps réel. En effet, comme le remarque Risset, « l'obsolescence technologique » risque ici tout particulièrement de « rendre l'œuvre périssable », puisque celle-ci se refuse à être fixée sur un support (comme dans le cas des musiques sur bande) et ne fait que coder des protocoles d'interaction avec un ordinateur... soumis aux lois du marché. » [16]

Pure Data, jmax. Ces programmes sont susceptibles d'interagir pendant une performance musicale de façons très diverses : en captant les mouvements ou déplacement d'un danseur, en projetant des séquences vidéo en fonction d'événements sonores, en suivant une interprétation musicale par rapport à la partition pour déclencher précisément des fichiers sons ou des traitements sonores, ou de toute autre manière calculable en temps réel par un système informatique, c'est-à-dire sans délai trop perceptible. Ainsi la plupart des programmes temps réel donnent accès à une interactivité séduisante et parfois spectaculaire, ils dotent d'une dimension presque « vivante » les performances interactives, et peuvent ouvrir un dialogue entre l'activité humaine et l'activité de la machine, tant qu'ils préservent une cohérence causale perceptible. L'informatique peut alors accompagner, prolonger, voire découpler le geste humain mis en scène, ou bien, sur un autre plan, ouvrir un dialogue homme-machine².

En ce qui concerne notre démarche par rapport au temps réel, c'est cette possibilité de dialogue qui nous intéresse, soit dans le travail de studio, soit pendant la performance ou le concert. Avec la CBox, ce dialogue a lieu selon des modalités particulières dont une partie est à préciser lors de la composition de la partition du logiciel, et dont les autres sont inhérentes aux spécificités de l'« écoute informatique » telle qu'elle a été programmée dans ce logiciel, ici à partir de l'objet *bonk~*.

Comme nous le verrons par la suite, cette « écoute » du logiciel introduit une part d'imprévisibilité dont les conséquences posent des questions différentes, selon qu'il s'agit de composition ou d'interprétation.

2.3. La mixité comme région fractionnaire

La frontière entre temps réel (TR) et temps différé (TD) se fait de moins en moins nette, soit que les logiciels considérés comme TD se rapprochent du TR grâce au progrès technologique, soit que les logiciels TR incorporent de plus en plus d'éléments TD. Plus les logiciels incorporent une mixité entre TR et TD, plus cette frontière devient ténue, et plus nous pouvons concevoir des régions fractionnaires entre ces deux pôles.

La CBox autorise une exploration particulière de cette nouvelle région, car nous pouvons envisager ce logiciel de deux manières différentes et complémentaires en considérant qu'il permet de faire : soit de « l'improvisation pré-composée », soit de la « composition réactive ». Quel que soit l'angle avec lequel on aborde la CBox, on ne peut pas ne pas passer par les deux temps. En effet, une pratique improvisatrice, considérée comme TR, ne pourra pas ici faire l'économie d'une phase de composition des paramètres du logiciel, phase considérée comme TD. Réciproquement, une pratique compositionnelle ne pourra pas faire l'économie d'une phase d'improvisation, de tâtonnement interactif et dynamique avec ce logiciel. Il ne s'agit pas d'une conséquence fortuite ou d'un constat postérieur, mais

² Pour une approche du paradigme interactif, cf. Wegner [21].

bien d'une volonté double dès le début de la conception ; il se trouve par ailleurs que les tendances de chacun des deux concepteurs par rapport à la priorité TR ou TD s'opposent, de sorte qu'à chaque étape, lorsque que la programmation du logiciel avançait dans une direction, l'étape suivante compensait dans l'autre direction.

Cette « double compétence » du logiciel, outre de fournir un cas d'étude exemplaire, présente quelques avantages et quelques inconvénients. D'abord, ne pas choisir l'un des deux pôles comme cible principale alourdit le programme, puisqu'il faut implémenter les

couches spécifiques à la fois du TR et du TD. Ensuite, passé ce problème technique, l'improvisation intuitive autorisée par les aptitudes TR de la CBox stimule la composition : en jouant avec le programme interactif on peut se familiariser rapidement. Enfin, la nécessité de « pré-composer » l'improvisation à diverses échelles temporelles simultanément permet potentiellement d'échapper au risque du manque de réflexion, tout en garantissant pour la performance un dialogue homme-machine.

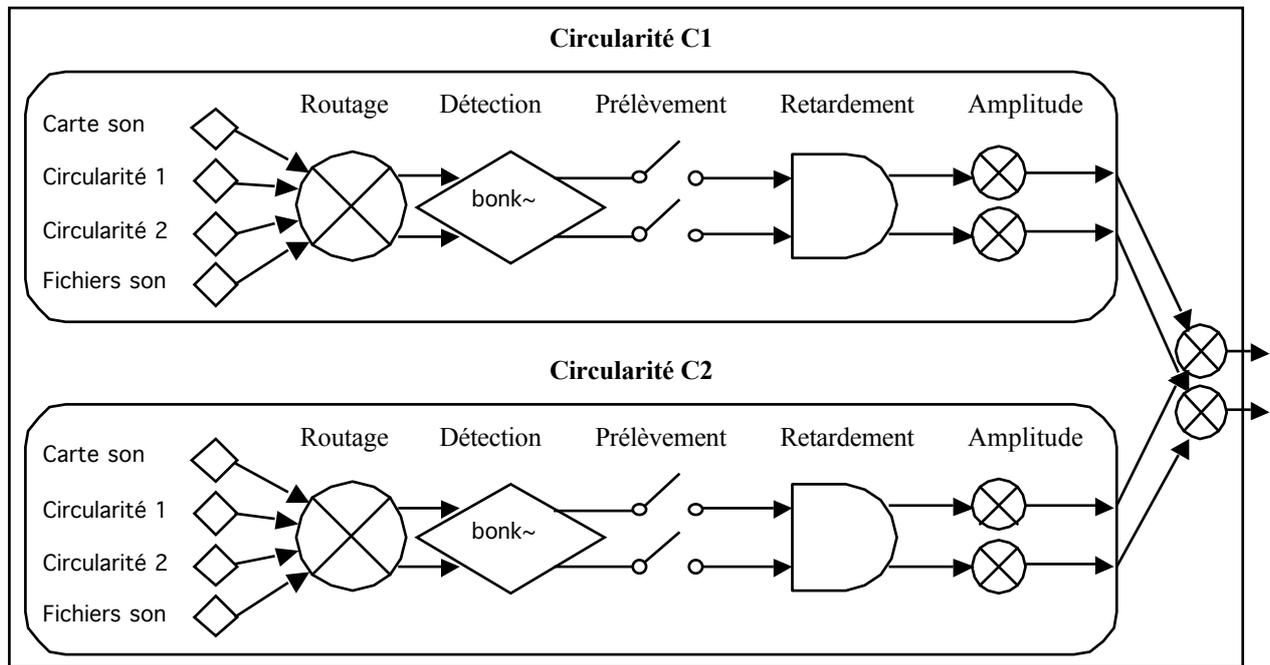


Figure 3 : schéma structurel de la chaîne de traitement du signal.

3. CIRCULARITE

Le principe fondateur de la CBox repose sur l'idée de *circularité* à la fois comme simple mécanisme de réinjection du signal de sortie à l'entrée de courtes chaînes opératoires et comme possibilité de connexion réciproque de ces deux processus. Nous distinguons ainsi des circularités de « type 1 », qui bouclent seulement sur elles-mêmes, et des circularités de « type 2 », qui connectent deux processus entre eux de façon à ce qu'ils se définissent mutuellement¹.

¹ Nous nous sommes inspirés des écrits de Francisco Varela [20] mais les idées de feedback et de circularité apparaissent dès l'origine des sciences cognitives. Dans la première cybernétique déjà, Wiener introduit en 1946 le concept de boucle de rétroaction (feedback), inhérent à toute régulation d'un système, sur la base de l'écart observé entre son action effective (output) et le résultat projeté (goal) [3]. Quant à l'idée de circularité en tant qu' « auto-organisation », on peut en trouver des traces chez Paul Weiss. Ainsi, d'après Dupuy :

« ...Weiss énonce un principe de causalité circulaire entre niveaux d'intégration emboîtés. (...) Dans un « système »,

3.1. « Ecoute » et réactivité logicielles

Ces processus circulaires incluent chacun une « écoute » informatique, ce qui nous permet d'introduire l'idée de *réactivité logicielle*. Pour l'instant, cette « écoute » reste assez élémentaire : elle repose essentiellement sur la détection d'attaques opérée par l'objet *bonk~*². *Bonk*, programmé par Miller Puckette [11], détecte les attaques définies par un changement de forme dans l'enveloppe spectrale. Ainsi, la CBox reçoit un signal audionumérique et le convertit en temps réel en une suite de valeurs résultant de l'analyse de ce signal. 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.

les lois de la physique laissent aux éléments individuels de nombreux degrés de liberté. Cette indétermination à la base va être réduite par les contraintes exercées par le tout, lesquelles résultent elles-mêmes de la composition des activités élémentaires. Le tout et les éléments se déterminent mutuellement. » [3]

² Nous avons déjà utilisé cet objet pour du déclenchement de sons dans une autre application, K-Box [1].

Cette sensibilité d'écoute nous permet théoriquement de travailler dans le domaine du micro temps, essentiel à la composition des sons¹. Les limites de l'usage pratique de l'objet bonk nous a cependant conduits à nous reposer la question de l'écoute informatique, en particulier le souhait d'accéder à d'autres axes « perceptifs » que la seule détection d'attaques. Nous avons donc envisagé la substitution de bonk par des analyses plus complètes grâce à l'exploitation des données spectrales, par exemple avec l'objet **analyze** de Tristan Gehan² à partir duquel nous avons développé un programme Visualyzer pour expérimenter les possibilités de ces analyses. Cet objet révèle plusieurs paramètres grâce à l'analyse en TR des données spectrales et énergétiques : l'attaque, l'amplitude instantanée, la hauteur (pitch), la brillance (brightness), le degré d'harmonicité (noisiness). Plusieurs inconvénients sont apparus : d'abord, la fenêtre d'analyse d'analyze demande 23ms contre 6ms pour celle de bonk, ensuite, « ...une représentation temps-fréquence n'est qu'un point de départ, mais nullement une photographie de notre image mentale du son évolutif. » [13]. Nous avons donc songé à élaborer quelque chose de plus proche de notre propre écoute, un parcours d'écoute composé, c'est à dire une bibliothèque de formes dynamiques subjectives à partir de parcours paramétriques dans un temps donné. Cette bibliothèque d'abord personnelle et subjective pourrait devenir plus générale en augmentant de façon critique sa taille et la variété des formes enregistrées. Nous avons renoncé devant la lourdeur de la tâche et l'ouverture de la question de « l'écoute informatisée », qui pourrait peut-être d'ailleurs trouver quelques réponses avec le connexionnisme³.

Bien que bonk~ ne permette qu'une écoute limitée à la détection d'attaques dans le signal, la CBox, grâce à une interface de détection, exploite les possibilités de paramétrage de cet objet: *high threshold*, *low threshold*, *minvel*, et *debounce*. Ainsi, une enveloppe graphique à deux pentes définit la forme de la croissance du signal, qui doit dépasser le seuil supérieur (*high threshold*) puis passer sous le seuil inférieur (*low threshold*) pour déterminer une attaque. Par ailleurs, deux boîtes numériques définissent deux filtres: *minvel* pour filtrer les attaques d'amplitude inférieure à ce seuil, et *debounce* pour imposer un temps minimum entre deux attaques.

Il faudrait aussi insister sur le fait que cette réactivité est accompagnée par des choix compositionnels (les prélèvements) et leur mise en relation (routage) ce qui le distingue d'une approche purement aléatoire. C'est de cette façon que nous établissons notre dialogue avec

¹ Cela dit, en termes fréquentiels nous sommes évidemment limités. À cet égard, nous travaillons actuellement sur d'autres mécanismes compensatoires, dont un moteur de répétition micro temporelle séparé.

² <http://web.media.mit.edu/~tristan/maxmsp.html>

³ «... l'action cognitive ou perceptive, appréhendée comme ontogénèse, en tant qu'elle fait naître, suivant des processus temporels propres, des sensations à partir d'un environnement, ainsi que des associations sur des sensations, ne peut se décrire en termes de représentations... [5]

l'outil, entre temps réel et temps différé, entre déterminisme et indéterminisme. Une voix moyenne.

3.2. Degré d'imprévisibilité

Cette voix moyenne, entre déterminisme et indéterminisme, résulte en grande partie du mécanisme d'écoute logicielle et apporte une imprévisibilité encadrée, un côté peut-être plus « vivant » à l'outil. Il s'agit de préciser les modalités de la réaction du logiciel, sans que l'on puisse être certain du résultat qui sera effectivement produit. L'imprévisibilité pourrait être intégrée musicalement à plusieurs niveaux, mais si possible seulement là où on le souhaite, car un dialogue sans évidence causale risque d'engendrer une frustration musicale rédhitoire. Imprévisible ne signifie pas aléatoire. Nous distinguons au moins deux niveaux d'imprévisibilité selon qu'il s'agit du temps de la composition ou bien du temps de la performance.

Comme nous l'avons vu précédemment, la réactivité du logiciel permet d'obtenir des résultats imprévus. Ces imprévisibilités, loin d'être des « erreurs », sont du point de vue de la composition riches en conséquences, porteuses de formes. Elles permettent d'emprunter des chemins qui n'auraient pas pu être envisageables d'avance (par une formule quelconque) et ainsi de composer des singularités, qui seront ensuite réintroduites dans le processus circulaire.

« Aujourd'hui nous sommes dans une situation où le compositeur ne se limite plus à planifier un processus pour le regarder marcher tout seul, en attendant qu'il lui donne quelque chose : il interagit à tout moment avec lui, pour produire du formel. Même si c'est le compositeur lui-même qui a construit le système, il peut s'investir à tout moment dans une « performance » avec les données fournies par son système ; les sorties ne sont donc pas automatiques, mais du point de vue du système, imprévisibles. On peut dire donc qu'il produit ainsi des singularités. » [19]

Mais, si pour la composition ces imprévisibilités ont un rôle formel (porteuses de formes), il en est autrement lors d'une performance, en particulier s'il s'agit d'une interprétation. En effet, l'intérêt d'une réactivité logicielle au niveau micro temporel réside dans la possibilité de donner à l'interprète une sensibilité de jeu (propre à toute interprétation) sans pour autant perdre la cohérence compositionnelle. Nous souhaitons que les fluctuations micro temporelles de l'interprétation engendrent des résultats différents, sans que ces résultats n'aient rien à voir entre eux ; c'est-à-dire formaliser des abstractions qui définissent des classes de résultats en intégrant à la fois cohérence et imprévisibilité. Pour cela, d'une part, l'instrumentiste doit se familiariser avec l'œuvre (en tant que composition réactive) et, d'autre part, le travail de composition doit prendre en compte le degré d'imprévisibilité du geste humain.

3.3. Un facteur de complexité

Le mécanisme global de la CBox autorise le couplage des deux processus circulaires (C1 et C2) du logiciel, ce qui engendre rapidement une certaine complexité en établissant une circularité de « type 2 ». De fait, la

composition musicale est porteuse de complexité lorsque les échelles temporelles qu'elle véhicule se co-déterminent par une même fonction, lorsqu'elles interagissent dans des processus circulaires de « type n », avec n supérieur à notre niveau 2 élémentaire. D'après Lévy-Leblond, la complexité, « c'est la conjugaison à la fois d'une hétérogénéité structurelle et d'une réciprocité fonctionnelle. » [19]

Certes, le recours à l'aléatoire ou à l'improvisation partielle, pour palier à la difficulté de déterminer à la fois l'ensemble et les détails, peut être une source d'inattendus, mais ils restent souvent d'une complexité faible, car dépourvus des fonctions réciproques où les niveaux temporels restent séparés. L'improvisation, par son immédiateté, empêche la possibilité systématique de travailler sur plusieurs niveaux, comme dirait Dahlhaus, elle est « monolithique », elle se concentre presque toujours sur un seul élément. [2] Ces limitations nous ont conduits à implémenter une circularité de « type 2 », par le couplage de deux niveaux temporels (qu'on pourrait associer d'une manière schématique au local et au global) et cela nous permet déjà d'aborder empiriquement la complexité. Dans le sens post-biologique, il faudrait beaucoup de circularités pour s'approcher de la complexité des simulations de vie artificielle, et surtout beaucoup plus de puissance de calcul. Il faudrait aussi imaginer une autre façon de composer, car écrire pour 2000 processus circulaires au lieu de 2 implique nécessairement des changements radicaux.

Même avec notre puissance de calcul limitée, avec beaucoup de compromis, la CBox pourrait finir par devenir un logiciel normal plutôt qu'expérimental, mais pour l'instant elle nous a déjà permis de formuler des questions qui nous intéressent. En tout état de cause, le travail soigné des prélèvements à différentes échelles - des singularités - et de leur mise en relation grâce au couplage de deux circularités permet d'établir cette fonction commune qui caractérise la complexité.

4. QUELLE NECESSITE MICRO TEMPORELLE ?

Le niveau micro temporel est devenu aujourd'hui un domaine incontournable dans la composition musicale assistée par ordinateur. Les célèbres travaux d'analyse par synthèse de Risset [12] ont montré l'importance de l'articulation au domaine du micro temps et, de fait, le micro temps est désormais, non sans difficultés, un champs supplémentaire à composer.

4.1. Entre déterminisme et aléatoire

Depuis les premières recherches du physicien Denis Gabor dès 1946 sur les « quantas sonores », de nombreuses recherches et applications ont été réalisées autour de ce qu'on appelle aujourd'hui le micro son [15], dont la synthèse granulaire [14], [17], l'une des techniques les plus développées¹. Mais pouvoir

¹ Pour une autre approche, voir la transformée en ondelettes cf. Kronland-Marinet [6]

descendre jusqu'aux échelles les plus petites n'est pas toujours synonyme de pouvoir composer le micro temps. Si paradoxal que cela puisse paraître, la synthèse granulaire ne garantit pas une véritable incorporation du micro temps à l'intérieur d'une composition.

En effet, en raison de la quantité massive de données nécessaires pour piloter cette technique, la synthèse granulaire fait appel à des unités d'organisation de haut niveau [15]. Ianis Xenakis, un des premiers à concevoir une utilisation musicale des grains sonores et de leur distribution dans le temps, a utilisé les théories mathématiques des probabilités [22]. Etant donné la qualité de ses œuvres, il est certain que ses recherches ont abouti à des résultats très innovants et lui ont permis de trouver une alternative à l'approche sérielle. Cependant, si les probabilités apportent des résultats inattendus, source de nouveaux matériaux sonores, elles envisagent finalement souvent du macro timbre ou des macro sons continus ou discontinus [10], ce qui devient paradoxal pour une approche micro temporelle au départ. Par ailleurs, comme écrit Roads :

« Un caractère purement aléatoire est un idéal, et fait référence à une absence complète de biais dans le choix des valeurs. Des algorithmes logiciels pour l'aléatoire ont un caractère pseudo-aléatoire (déterministe). Les valeurs qu'ils produisent tendent à être uniformes et adirectionnelles, ne tendant que vers la moyenne. » [15]

4.2. Inter-relation des échelles temporelles

« L'idée d'une musique qui procède totalement d'un seul principe, où le tout et les détails sont déterminés par la série, cette idée s'est avérée utopique. » [2]

Ainsi, avec la synthèse granulaire, on obtient des flux ou des nuages de grains qui, certes, sont porteurs de « couleurs », mais dont le niveau de complexité reste souvent faible. C'est la raison pour laquelle lorsque le compositeur fait appel à ces techniques, il doit toujours par la suite « adapter » le nouveau matériau, par des actions manuelles², parfois très nombreuses, au tissu complexe de l'œuvre. En tout état de cause, nous soutenons que des techniques qui sont uniquement organisées par des unités temporelles d'ordre supérieur, et qui par conséquent négligent l'inter-relation entre les différents niveaux temporels, manquent de pertinence musicale³. Notre démarche se révèle à nouveau intermédiaire, entre la génération et la composition,

² Même Xenakis, qui a utilisé avec rigueur les lois des probabilités, à l'heure de composer n'hésite pas à laisser la place à des actions directes, plus intuitives, au profit de l'œuvre. [4]

³ L'ordinateur se montre aujourd'hui particulièrement performant pour générer du matériau, avec toutefois quelques dérives possibles. Comme le signale Vaggione : «... de notre point de vue concernant la composition musicale, on ne saurait échapper au niveau du matériau, bien entendu ; mais on est en droit d'exercer une pensée musicale concernant ce niveau – car le « matériau » peut être aussi illusoire qu'aliénant, du moment qu'on ne le pense pas musicalement, c'est-à-dire qu'on ne le compose pas. » [19]

puisqu'on peut interagir à tout moment avec ce qu'on est en train de produire.

Si les travaux fondateurs de Risset sur la brillance sont d'une grande importance pour l'aspect purement micro temporel, ils le sont encore davantage pour l'inter-relation des échelles qui sont tissées à l'intérieur même du micro temps. Le micro temps, est essentiel pour la composition de sons, mais, outre le rapport à la perception, il n'implique pas de nouveaux problèmes (si problèmes il y a) compositionnels. C'est seulement un niveau de plus à composer, à intégrer.

4.3. Micro temps et temps réel

« tout système numérique ne peut atteindre en temps réel qu'une certaine limite de complexité... L'oreille adaptée aux finesses acoustiques, est particulièrement exigeante. Et il restera toujours le principe d'une limite. » [13]

Actuellement, le seuil de prédictibilité du logiciel avoisine les 40 à 60 ms dans de bonnes conditions selon la machine, ce qui représente justement une zone critique entre micro et macro temps. Différentes causes concourent à cet état de fait, à plusieurs niveaux des systèmes TR. A moins d'opérer sur des heuristiques, l'immense majorité des logiciels TR calculent le signal par rapport au passé et pâtissent donc fatalement d'un temps de latence intrinsèque à l'opération à effectuer, proportionnellement à la taille de sa fenêtre temporelle et sa vitesse de glissement. En outre, les interfaces de contrôle et leurs pilotes logiciels sont eux-mêmes cadencés à des vitesses qui interdisent un accès micro temporel précis. Enfin, la machine, la carte son, et le système d'exploitation possèdent leur propre temps de latence qui peut se cumuler aux autres.

Paradoxalement, le temps réel lui-même finit par devenir porteur de macro temps d'un point de vue informatique, une fois que le micro temps dont il pouvait être porteur a été filtré par les différents temps de latence rencontrés lors de la traversée de la chaîne informatique.

Nous étudions aujourd'hui la possibilité de lever l'impasse du micro temps dans la CBox grâce à une interface de prélèvement à plusieurs lignes temporelles qui permettraient de mieux opérer et de formaliser un travail micro temporel. Nous pourrions ainsi paramétrer une forme précise pour plusieurs prélèvements micro temporels en séquence, dans une échelle temporelle supérieure qui resterait confortable pour la machine et le logiciel. D'autre part, nous envisageons un moteur fréquentiel local de répétitions micro temporelles qui pourrait s'avérer complémentaire.

5. CONCLUSION

La frontière entre temps réel et temps différé se fait de moins en moins nette, maintenant que les logiciels dédiés au studio produisent le résultat de leurs calculs dans un temps proche de l'immédiateté, et réciproquement que les logiciels dédiés aux performances demandent parfois davantage de travail de

composition en amont. La CBox autorise une exploration particulière *entre* ces deux pôles qui sont réellement indispensables ici ; on ne peut pas ne pas expérimenter son interactivité, ni s'abstenir de faire des choix compositionnels. Nous avons donc défini la notion de région fractionnaire pour qualifier les applications simultanément TR et TD, lorsque les deux temps sont obligatoirement expérimentés quelle que soit l'orientation préférentielle de la pratique effective.

La pratique et la réflexion autour de ce logiciel nous a aussi permis de clarifier le concept de circularité, en distinguant deux types, correspondants au nombre de processus circulaires concernés par le bouclage dynamique, du simple feedback à des inter-relations de niveau *n*.

Ensuite, la CBox demande de reformuler la question de l'articulation musicale du micro temps et du temps réel, en apportant des éléments de réponses techniques et structurels, grâce à des mécanismes compensatoires des temps d'analyse et de latence comme l'écriture explicite de prélèvements en séquence dans le signal et bientôt un moteur fréquentiel local de répétitions micro temporelles.

Enfin, pour ne pas finir, puisque nous préparons un concert pour clarinette et CBox, il y aura encore de nombreux allers et retours entre la programmation de ce curieux logiciel et nos expérimentations musicales, compositionnelles et scéniques, comme une conclusion circulaire...

6. REFERENCES

- [1] Barkati, K. « K-Box : Programmation d'une extension logicielle pour instrument percussif. » *Textes de communications JIM 2001*, pp.27-37, Bourges, 2001.
- [2] Dahlhaus, C. *Essais sur la Nouvelle Musique*, Editions Contrechamps, Paris, 2004.
- [3] Dupuy, J.P. *Aux origines des sciences cognitives*, Editions La Découverte, Paris, 1994.
- [4] Gibson, B. « Théorie et pratique dans la musique de Iannis Xenakis : à propos du montage », *Thèse de doctorat*, Ecole des Hautes Etudes en Sciences Sociales, Paris, 2003.
- [5] Kiss, J. *Composition musicale et sciences cognitives*, Editions L'Harmattan, Paris, 2004.
- [6] Kronland-Martinet, R. « The use of the wavelet transform for the analysis, synthesis and processing of speech and music sounds », *Computer Music Journal* 12 n°4, pp.11-20, MIT Press, Cambridge, Massachusetts, 1988.
- [7] Malt, M. « Les mathématiques et la composition assistée par ordinateur, concepts outils et modèles », *Thèse de doctorat de Musique et musicologie du XXème siècle*, Ecole des hautes Etudes en Sciences Sociales, directeur de thèse Marc Battier, Paris, 2000.
- [8] Maturana, H. et Varela, F. *L'arbre de la connaissance*, Addison-Wesley France, Paris, 1994.
- [9] Miranda, E. R. « Music composition using cellular automata », *Languages of Design Vol.2*, pp. 105-117, USA, 1994.

- [10] Pape, G. et Bokesoy, S. « Présentation de Stochos », *Actes des JIM 2003*, Montbéliard, 2003.
- [11] Puckette, M. *Real-time audio analysis tools for Pd and MSP*, <http://www.crcs.ucsd.edu/~msp>, 1998.
- [12] Risset, J.C. et Wessel, D. « Exploration du timbre par analyse et synthèse », *Le timbre, métaphore pour la composition*, Editions Christian Bourgois, pp.102-131, Paris, 1991.
- [13] Risset, J.C. « Temps et musique numérique », *Le temps en musique électroacoustique, Actes V*, pp.141-145, Editions Mnemosyne, Académie Bourges, 1999/2000.
- [14] Roads, C. "Automated granular synthesis of sound". *Computer Music Journal*, 2 (2), pp. 61-62, MIT Press, Cambridge, Massachusetts, 1978.
- [15] Roads, C. "Microsound", MIT Press, Cambridge, Massachusetts, 2001.
- [16] Szendy, P. « Musique, temps réel ». *Résonance n°14*, Ircam - Centre Georges- Pompidou, Paris, octobre 1998.
- [17] Truax, B. "Real-time granular synthesis with the DMX-1000." *Proceedings of the 1986 International Computer Music Conference*, pp.231-235, San Francisco, 1986.
- [18] Vaggione, H. « Sons, temps, objet, syntaxe. Vers une approche multi-échelle dans la composition assistée par ordinateur. » pp.169-202, *Musique, rationalité, langage*, L'Harmattan, Paris, 1998.
- [19] Vaggione, H. « De l'opérateur », *Formel/Informel*, L'Harmattan, pp.221-235, Paris, 2003.
- [20] Varela, F. « Autonomie et connaissance. », *Essai sur le vivant*, Edition du Seuil, Paris, 1989.
- [21] Wegner, P. "Why interaction is more powerful than algorithms", *Communications de l'ACM*, New York, 1997.
- [22] Xenakis, I. « Musiques Formelles », *La Revue Musicale n°253-254*, Paris, 1963.