QUELQUES RÉFLEXIONS SUR LA LOGIQUE D'INTERFACE POUR LA CRÉATION MUSICALE ASSISTÉE PAR ORDINATEUR

Kevin Dahan
CICM - Université Paris VIII
MSH Paris Nord
kevin.dahan@wanadoo.fr

ABSTRACT

Alors que de nombreuses avancées ont été faites ces dernier temps pour améliorer les capacités graphiques des ordinateurs, il a été fait peu de cas de ces progrès dans les applications pour la création musicale, en particulier dans le domaine de la composition. Le manque de nouvelles idées graphiques n'est pas uniquement dommageable à l'évolution de bons outils compositionnels, mais tend également à limiter les utilisateurs dans leur champ d'expression. Nous nous baserons sur les notions modernes d'*objet compositionnel*, en tentant de déterminer quels types de représentations et d'interfaces elles demandent. Bien qu'évoquant de nouvelles conceptions d'interfaces utilisateurs, nous tenterons de garder à l'esprit le principal intérêt de ce travail : fournir de nouveaux outils pour un meilleur contrôle sur le processus compositionnel.

1. INTRODUCTION

Cet article traite d'un aspect relativement peu développé dans le domaine de la création musicale assistée par ordinateur, celui de l'interface utilisateur, et plus précisément des interfaces *graphiques* - *GUI*.

En effet, il existe actuellement un problème dans le développement d'interfaces graphiques pour les applications musicales, dans le sens où les applications en question semblent plus se soucier des aspects techniques, et négligent les facettes musicales et artistiques. Le développement de la littérature consacrée au domaine de la représentation des signaux audio, ainsi que les approches conjointes consacrées au traitement du signal et à leur implémentation mettent en lumière ce chiasme ; on se référera notamment à [11] ou [13]). Pourtant, pratiquement aucune approche n'a été initiée pour prendre en compte le domaine symbolique et complexe des structures sonores et musicales.

Il est possible de formuler la même critique pour les applications spécifiquement dédiées à l'audio, et le manque d'innovation graphique est flagrant. Mis à part quelques efforts intéressants - particulièrement dans le domaine de la spatialisation, domaine pour lequel les interfaces tridimensionnelles ont été employées (notamment par [10] ou [16]) ou encore le projet ambitieux *OSE* - décrit dans [3], dans lequel l'idée d'"environnement virtuel" est employé [1] . Pourtant, pratiquement aucune avancée n'a été faite pour permettre un support graphique plus cohérent face à des structures musicales plus complexes - comme notamment les objets compositionnels. En fait, la plupart des logiciels disponibles pour la composition (*OpenMusic*, *PWGL*) ou la gestion des systèmes temps-réel (comme par exemple *MAX/MSP* et a fortiori *PureData* suivent le paradigme de "patches", qui découle du principe analogique de "boîtes noire" interconnectées [2] . En outre, cette approche n'est finalement qu'une reproduction mimétique des procédures employées en studio "physique".

Nous pensons que cette situation inhibe l'efficacité des logiciels pour la composition musicale. Alors que de nombreuses avancées sont faites du côté des technologies et des techniques (par exemple, la qualité des nouveaux algorithmes pour le traitement des signaux audio, nouveaux modèles de synthèse, nouveaux systèmes de détection du mouvement pour les environnements interactifs), très peu d'efforts - voire aucun - n'ont été fait pour permettre une intégration fine de ces avancées dans des environnements de production qui permettraient au compositeur de les utiliser avec peu d'habileté technique.

Cet article est découpé en trois parties : dans la première, nous évoquerons brièvement les notions d'objets dans le contexte de création musicale assistée par ordinateur. Puis, nous évoquerons les méthodes pour représenter ces objets. Finalement, nous décrirons des prototypes d'interfaces que nous avons développé.

2. DÉFINIR DE NOUVELLES STRUCTURES MUSICALES

Comment pensent les compositeurs lors de la création de la musique électroacoustique ? Comment appréhende-t-il l'utilisation de structures musicales complexes, en particulier à l'intérieur d'un environnement informatique d'aide à la création musicale ?

[1] Avec l'introduction de machines puissantes (notamment graphiquement) et relativement abordables, les années 90 ont vu un champ nouveau consacré à la réflexion sur la réalité virtuelle et donc sur les environnements virtuels.

[2] Un contre-exemple intéressant est le logiciel *FMOL*, un outil pour l'improvisation collective, qui introduit différents paradigmes graphiques [8]

2.1. Les structures musicales comme objets polymorphiques

Bien évidemment, la définition strict de ce qu'est une structure musicale est dépendante du contexte. D'une manière générale, le terme est utilisé pour décrire l'organisation globale d'un *item* musical, comme par exemple la cadence dans un discours classique. Pourtant, il est nécessaire de parvenir à une définition spécifique suffisamment large pour embrasser le champ de connaissance particulier qu'est le domaine électroacoustique. L'absence de *style* prédéfini [3] rend la tâche du programmeur délicate pour parvenir à anticiper clairement *quoi* et encore plus *pour quoi* son logiciel sera utilisé [4] . Dans cette absence de compréhension, les spécialisations successives des logiciels d'aide à la composition ont imposé une hiérarchie fonctionnelle de la musique électroacoustique. Par exemple, la synthèse sonore, la composition algorithmique, l'exécution temps-réel, et les logiciels de montage, sont d'une certaine manière imperméables et limitent les possibilités de "transgression" d'une hiérarchie structurelle particulière, depuis le niveau du (micro)son à celui de la macrostructure correspondant à la pièce entière. D'un autre côté, le volume et la diversité des logiciels pour la composition qui ont émergé grâce à la nature expérimentale de la musique informatique est l'une des raisons qui expliquent le succès de la rencontre entre art et science. Il semble par conséquent évident qu'un concept suffisamment puissant est nécessaire pour encapsuler d'une manière cohérente ces deux facettes (hyperspécialisation, et facilité de navigation entre niveaux d'abstractions éloignés).

La notion d'*objet*, employée par Horacio Vaggione[14], nous paraît particulièrement adaptée et valide dans le champ d'application qui nous occupe. Celle-ci peut se comprendre comme étant la définition d'un agrégat multiscalaire, polymorphe et multifonctionnel, dont les caractéristiques ne sont pas uniquement considérable sur le plan esthétique, mais surtout sur le plan technique. L'adaptation à des niveaux d'organisation très différents implique une morphologie particulière des objets compositionnels - une morphologie *adaptive*. Comme l'objet compositionnel est au moins définissable par une appartenance au minimum à deux échelles [5] - phénotype et génotype - nous allons explorer les manifestations morphologiques qui s'expriment dans ces deux domaines. Première conséquence de la multiscalarité de l'objet compositionnel, le *polymorphisme* induit la possibilité, pour un même objet *compositionnel* de pouvoir exister sous plusieurs formes. Bien entendu la multiscalarité entraîne la manifestation d'*avatars* différent, selon le contexte dans lequel se trouve la pièce ; le contexte influe sur la représentation morphologique d'un objet. En outre, il est évident qu'un objet compositionnel se présente donc sous une multitude de formes (morphologies) : structure sonore, algorithme, *gesture* [6] .

L'utilisation de ce concept d'*objet polymorphique*, permettant de représenter des informations *selon* le niveau d'abstraction, est d'un intérêt bien compréhensible pour les éléments qui forment une composition électroacoustique.

2.2. Par-delà la Cybernétique du 1er ordre

Or l'utilisation d'objets adaptifs, complexes, et par dérivation définissant des interactions entre eux, bouleversent la vision traditionnelle de l'informatique musicale, les logiciels courants utilisant des métaphores de briques de constructions. A contrario, l'adoption du principe d'objet compositionnel implique une vision non linéaire et non directionnelle du travail de composition.

Ce paradigme est issu de la première cybernétique : les concepts de réseaux de *boîtes noires* interconnectées doivent être remplacés par une vision plus moderne, celle de *système dynamique* [7] . La conception du système compositionnel passe donc d'une image de construction à celle d'une *émergence* de structures musicales.

3. INTERFACES

Par conséquent, ce changement de paradigme implique une relecture complète des parti-pris pour l'interface envisagés jusqu'à maintenant.

3.1. Paradigmes utilisés

Plusieurs études ([5] et [9]) ont montré que les aspects graphiques des logiciels utilisés habituellement pour la création musicale ne sont pas satisfaisants, et peuvent même s'avérer frustrantes pour les compositeurs. En outre, les interfaces employées semblent être particulièrement inefficaces lors de leur emploi par le compositeur pour définir des structures musicales complexes, basées sur des objets compositionnels, ou tout au moins génèrent une résistance à l'utilisateur expert. L'origine de ce problème va apparaître clairement après avoir passé en revue les paradigmes d'interfaces principaux employés dans les logiciels de composition.

3.1.1. Interfaces orientées macro-structures

Le style d'interface prédominant pour la phase macroscopique de composition (combinaison et arrangement des artefacts musicaux) est la notion de *séquencage*, c'est-à-dire d'*arrangement par pistes*. Les séquenceurs sont largement employés par les compositeurs pour procéder à la mise en place et au rendu final des pièces. Un exemple typique de l'interface graphique obtenue est montrée à la figure 1.

[3] Selon la notion d'"école".

[4] L'étude de [9] semble montrer l'absence d'intérêt du compositeur pour utiliser ce *à quoi* le logiciel a été prévu…

[5] ou système représentationnel

[6] La notion de "geste" nous para î t trop impliquer une intentionnalité défi nie, ainsi qu'une contingence "mécanique" - le terme anglo-saxon de *gesture* contient au contraire la notion de fortuit, d'induction approximative.

[7] Généré par l'utilisation, en lieu et place de *processus*, d'une vision centrée sur l'*interaction* entre objets compositionnels polymorphiques.

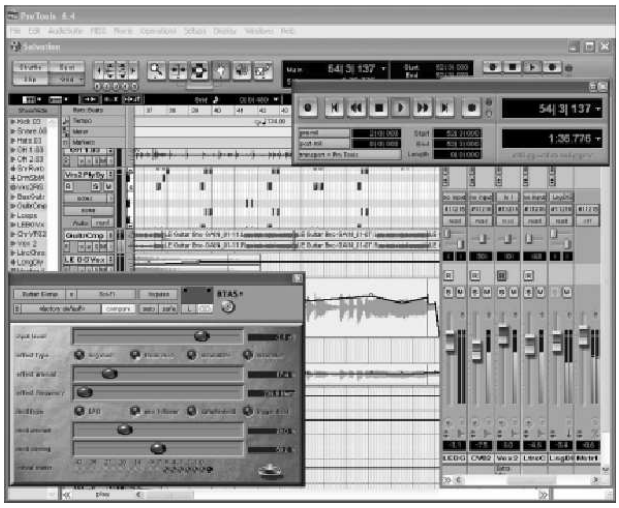

Figure 1. Une situation typique de *Protools*

Nous pensons que ce paradigme d'interface donne un exemple typique de la nature inhibitrice des solutions pour la composition disponibles actuellement.

De manière plus spécifique, ces logiciels sur-simplifient la macrostructure d'une composition, étant donné qu'ils n'opèrent qu'à un niveau d'abstraction unique et uniforme. La manipulation des paramètres sonores est par conséquent rendue anti intuitive, et potentiellement destructrice. Il est fait référence à ce problème dans [9] :

> [. . .] principles important to the composer may be hidden by GUIs that operate at too high a level of abstraction. Availability of easy-to-use, heavily destructive (real-time) processing tools as well as the easiness to assemble a huge number of sounds over a short period of time seems to create a situation where composers are no longer aware of the processes involved in their sound manipulation. This can lead to over-processing and over clustering of sounds, resulting in music which does not refer to any common, shared experience but the experience within the composing community. Thus the accessibility of electroacoustic music to a non-expert audience is effectively denied.

Par contrecoup, et pour conclure de manière claire, le fait de restreindre la manipulation des objets compositionnels à un seul niveau d'abstraction - l'approche par piste - rendent difficile l'implémentation et la réalisation de structures musicales complexes - des objets compositionnels, ce qui implique un encombrement de l'espace sonore (i.e. *bruit*), rendant l'accès aux musiques électroacoustiques difficile.

3.1.2. Interfaces orientées micro-structures

A l'opposé, c'est-à-dire dans le domaine des microstructures, la notion dominante pour la construction d'interface est celle de la *spécification* des structures sonores par la définition d'un ensemble analogique simulé dans le style d'un patch de synthétiseur - la notion de *patchcord* employé par *Max/MSP*. La figure 2 montre un exemple caractéristique de la situation obtenue avec un logiciel qui utilise cette approche.

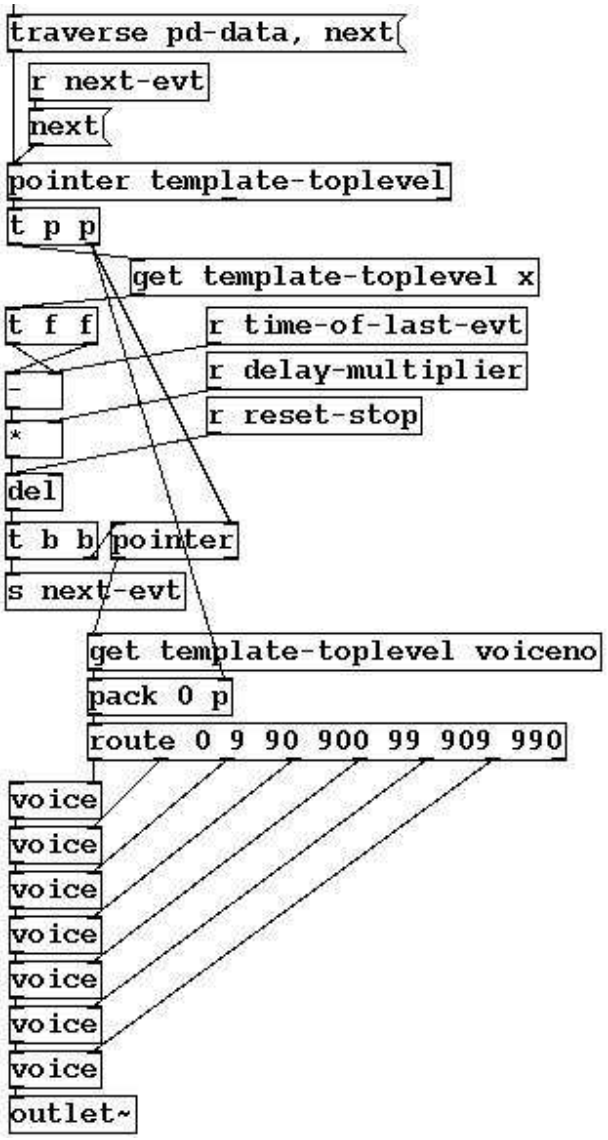

Figure 2. Un patch dans *PureData*

Nous pensons que cette approche est trop largement utilisée. Bien que l'approche autorise effectivement le travail à des niveaux d'abstraction multiples, il impose la restriction d'un travail *uniforme*, dans un environnement analogique simulé [8] . En pratique (et compte tenu de cet aspect numérique de la technologie proposée), ces interfaces fournissent bien évidemment un modèle plus complexe qu'une transposition littérale d'un studio "réel", dans lequel les musiciens câblent des modules dans le but de mo-

[8] On arguera que les techniques disponibles sont fortement numériques. Pourtant, malgré le typage numérique fort, la représentation qui en est fait fonctionne sur une analogie complète des techniques de synthèse analogique, ce qui génère une sorte de dichotomie complexe entre deux conceptions radicalement opposées.

difier un signal musical. Il faut se rappeler bien entendu que la fonction principal des logiciels *à la* Max/MSP est de fournir un environnement pour les situations de temps-réel, i.e. de concert et d'exécution en direct. Cependant, ce type de logiciel est largement employé dans un contexte compositionnel en temps différé, et son emploi dans ce cadre montre clairement le changement de paradigme qui s'est opéré entre "logiciel support de l'exécution" et "logiciel comme base principale de la composition". Ce changement de comportement de la part des compositeurs est aussi du à l'explosion de la puissance de traitement des ordinateurs (notamment personnels) qui permettent d'effectuer en temps-réel des traitements complexes impossible à imaginer auparavant, et donc de faciliter leur inclusion dans le processus compositionnel - il s'agit d'un courant maintenant majeur dans la composition assistée par ordinateur [7].

Bien que le succès de ces outils de composition/exécution ne se soient jamais démenti tout au long de la dernière décennie, nous voyons l'absence de représentation adaptés à des objets complexes comme étant une limitation importante, et surtout inhérente au modèle choisi. En effet, la plupart des logiciels suivent le paradigme de "boîte noire" - lorsqu'un patch devient trop complexe, il est possible de le cacher dans un patch de plus haut niveau, devenant lui même un module de traitement à part entière. Récemment, Miller Puckette (architecte original de Max et à l'origine de PureData) s'est rendu compte de ce problème et a remis en question dans [12] l'utilisation des "listes de notes" - corollaire des représentations en patch - en affirmant qu'elles contribuaient à couper le compositeur des niveaux d'abstraction plus élevés (notamment le niveau symbolique contenu dans l'idée de *score*). Pour surmonter ce problème, PureData implémente une fonction qui permet aux compositeurs de travailler à partir de formes géométriques pour définir un score, comme le montre la figure 3.

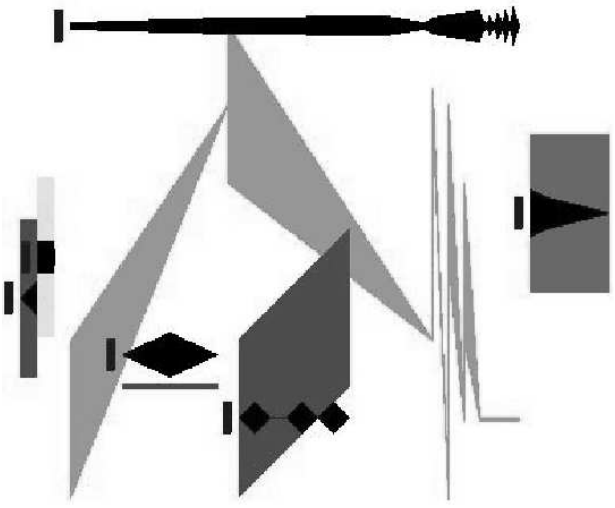

Figure 3. Représentation en score dans *PureData*

Cet article tend à confirmer que la notion de "patch" devenait de plus en plus encombrante dans une logique compositionnelle et que cette fonction de représentation symbolique n'était qu'un palliatif, rarement employée à cause d'une certaine "lourdeur" de mise en oeuvre (les objets de scores ne sont que des objets graphiques attachés aux structures de données définies dans les patches). Finalement, l'utilisation de structures graphiques peuvent permettre aux compositeurs de se concentrer sur les problèmes de haut niveau de structuration d'une pièce, au lieu de travailler sur des problématiques bas niveau qui surviennent couramment en programmation. Pourtant, la difficulté de mélanger des représentation symboliques avec les niveaux plus bas de traitement et de programmation est toujours présente.

3.1.3. Interfaces temporelles

OpenMusic représente, à notre sens, l'un des logiciels de composition à l'interface la plus aboutie pour le moment. L'utilisateur peut non seulement aisément naviguer entre différents niveaux d'abstractions, mais également entre différents types de données (morceaux de code, signaux audio) et de représentations de données (éléments programmés, notation traditionnelle), permettant ainsi de créer certains objets compositionnels complexes, comme le montre la figure 4.

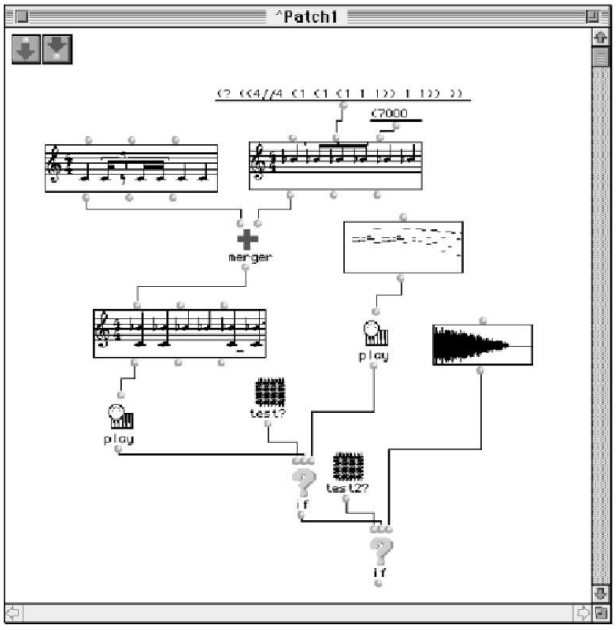

Figure 4. *OpenMusic* : représentations multiples

Les notions de structure musicale ne sont malheureusement que partiellement implémentées. De manière spécifique, nous noterons l'omniprésence du concept de patch et de patchcord, minant l'efficacité général du système de représentation. Un problème plus avancé est celui de la représentation temporelle, qui devient récurrent dès l'émergence de macro-structures. Le concept de "maquette" a été introduit pour dévier le problème inhérent que nous avons détaillé à propos de la vision "analogique" du patch. Comme les auteurs la décrivent, "une maquette peut être

pensée comme un patch spécial dans lequel une dimension temporelle est présente". Effectivement, comme le montre la figure 5, le temps est représenté par une échelle temporelle en abscisse, et, mis à part cette contrainte, le comportement est identique à celui d'un patch "habituel".

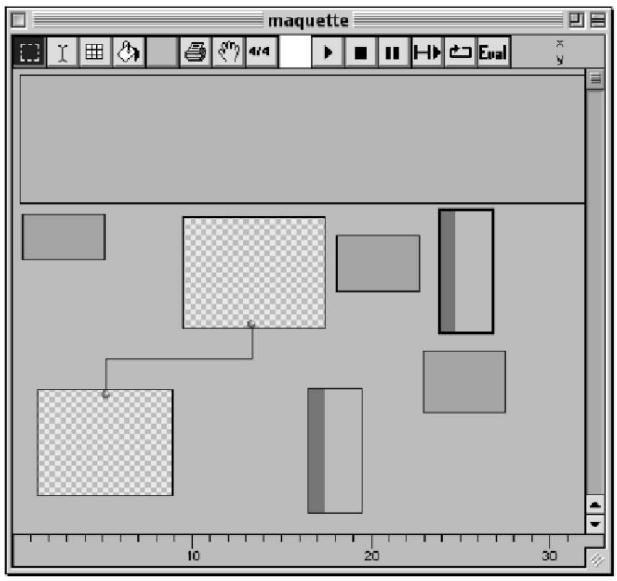

Figure 5. *OpenMusic* : maquette

La réintroduction du temps d'un point de vue représentationnel est évidemment très importante : elle permet de resituer la musique dans sa dimension canonique, celle d'occupation du temps ; cependant, à cause de la nature et du but de la maquette (réorganisation de matériaux préassemblés), elle n'est utilisable que pour le niveau macrostructurel - local ou global. Arriver à se détacher de cette notion macrostructurelle paraît cependant être très important dans le contexte actuel de contrôle fin sur le timbre demandé par de nombreux compositeurs. Rappelons qu'en 1996 Adrian Freed soulignait déjà [6] :

> The multi-dimensionalty of musical data begs for higher dimensions of control and representation impossible with the current "paper on a desktop" metaphor.

En allant plus loin, il nous faut remarquer que cette approche se conforme encore aux paradigmes de computation symbolique et aux conceptions issues de la cybernétique de premier ordre [1], qui sont malheureusement encore largement employées dans une majorité d'applications pour la composition musicale assistée par ordinateur. D'une certaine manière, l'emphase est encore mise sur les *processus*, plutôt que sur la *structuration* et l'*interaction*, que nous avons signalé comme étant les bases du travail compositionnel apportées par la notion d'objet.

3.2. Représentation des objets compositionnels

Trouver de nouvelles manières pour représenter la complexité des objets compositionnels est un champ extrêmement vaste. L'approche radicale que nous avons choisi d'explorer est la suivante : l'interface initialement vide de toute sémantique formelle.

Nous envisageons l'interface sans contenu, contrainte ou sémantique prédéfinis, car le potentiel du compositeur n'est pas guidé de quelque manière que ce soit. L'utilisateur peut donc alors créer ses propres modèles pour décrire les structures et la sémantique qu'il compte développer pour une oeuvre précise. Par conséquent, le paradigme du patch n'est plus intéressant : si nous voulons permettre à l'utilisateur de travailler sur des structures basées sur des *objets compositionnels*, nous n'avons aucun besoin d'introduire les notions de processus et de traitement - l'objet compositionnel, rappelons-le, comporte en lui même une morphologie mais aussi un *comportement*.

3.2.1. Espace de travail non typé

Etant donné que les compositeurs utiliseront *de toute façon* les logiciels à leur service de manière hétérogène et non prévue, il serait inapproprié de leur fournir une interface "orientée". Plusieurs fonctionnalités sont dès lors impliquées : au lieu de proposer une interface sémantiquement riche comme les séquenceurs, dans lesquels l'espace de travail contient déjà un "sens" (orientation temporelle, espaces prédéfinis pour positionner les structures, etc...), il est plus intéressant de provoquer une réaction inverse et de faire en sorte que l'interface réclame qu'on lui donne un sens - celui de la pièce en cours, ou des méthodes de travail spécifiques. Ce type d'interface a été exploré en partie par le logiciel *Boxes* [9] , mais en gardant une logique d'orientation temporelle, ainsi qu'une logique de manipulation par le biais de l'interface (en agrandissant horizontalement l'objet, le son sera dilaté, etc.). Une autre conséquence du refus du typage de l'interface est la possibilité d'importer tout type de données (fichiers audio, texte, MIDI, etc.) dans le système [10] comme objets non typés sémantiquement (c'est à dire des objets sans spécification particulière pour le système. Dans le sens inverse, les objets typés pour le système - interprétables ou interprétés - seront transparents pour l'utilisateur : aucun typage morphologique pour la représentation ne sera imposé, et il sera employé une représentation *neutre* servant uniquement à signaler graphiquement la présence d'un objet.

3.2.2. Représenter la macrostructuration

Il est classique de définir la macrostructuration d'une pièce par un travail se faisant sur trois niveaux : un agencement temporel, un positionnement spatial et un ordonnancement dynamique. L'agencement traditionnel des structures en macrostructure est classiquement fait par les séquenceurs, ce qui impose une logique de travail particulière, par piste.

En suivant l'approche proposée par [4], il est possible d'utiliser le paradigme des *contraintes*, largement employée

[9] Voir [2]

[10] Virtuellement s'entend...

par ailleurs en représentation des musiques ”classiques”, pour systématiser les règles harmoniques classiques traditionnelles par exemple. Le principe est de définir les éléments non plus comme étant indépendants, mais de considérer les structures musicales comme tissant un réseau de relations entre elles. Ces relations sont dénommées contraintes car elles définissent un rapport déterministe entre les éléments.

4. PROTOTYPES

Pour tester la validité de nos hypothèses sur de nouveaux modes de représentation macrostructurelle, nous avons développé un prototype basique [11] . Ce prototype permet de charger des fichiers sons, représentés par des rectangles de taille uniforme. L'utilisateur peut ensuite les arranger sur un espace de travail non typé, et peut exprimer des relations temporelles de deux types entre les différents éléments : concurrentielle ou relative. Ces contraintes temporelles n'influent pas sur le placement des objets, qui reste totalement libre. Ce choix permet de travailler au même niveau d'abstraction (macrostructurel, en l'occurence) sur des structures extrêmement diverses du point de vue temporel, d'autant plus que leur représentation n'est pas guidée par leur durée.

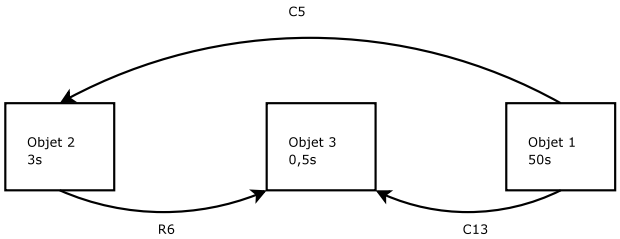

Figure 6. Représentation schématique d'une situation temporelle dans *mondrian*

Par exemple, la situation schématisée dans la figure 6 est la suivante : un objet d'une durée de 50 secondes situé à la droite de l'espace de travail pourra être relié à un objet de 3 secondes positionné à l'extrême gauche, avec une contrainte temporelle concurrentielle de 5 secondes. Ce même objet pourra être connecté à un autre objet situé au centre de l'espace de travail d'une durée de 0,5 seconde grâce à une contrainte temporelle relative de 6 secondes, mais également au premier avec une contrainte concurrentielle de 13 secondes. La figure 7 présente la résolution “classique” de cette situation. Chaque objet peut ainsi être lié un nombre indéfini de fois, ce qui permet d'introduire une notion plus propre de répétition littérale, plutôt que d'utiliser une fonctionnalité destructurante du type *copier/coller*.

La fonctionnalité de *résolution des contraintes temporelles* permet au compositeur d'obtenir une ”partition” linéaire, ce qui est bien entendu utile lors de la phase finale de composition.

[11] L'interface se nomme *mondrian*, est codée en C en utilisant la librairie graphique *evas*

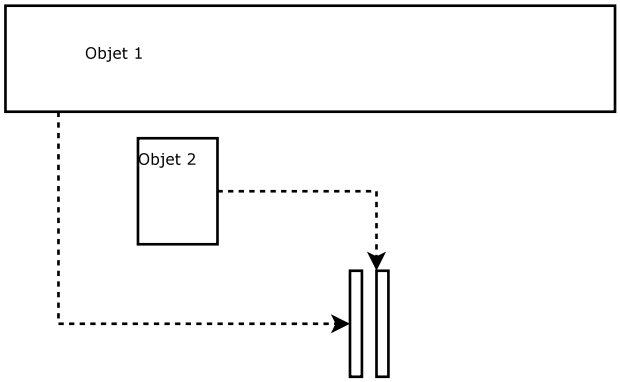

Figure 7. Réalisation de la situation

Le contrôle des paramètres de volume s'obtient par l'utilisation d'une propriété des outils graphiques actuelles appelée *alpha blending*, qui permet de définir et contrôler un degré de transparence pour les objets graphiques (voir figure 8).

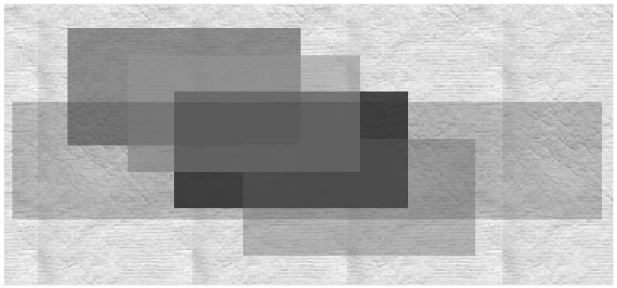

Figure 8. Superposition de structures dans mondrian

Plusieurs objets peuvent ainsi être superposés avec différents degrés de transparence, ce qui permet au compositeur de visualiser directement les structures musicales situées aux différents plans dynamiques de sa composition.

Différentes pistes ont été évoquées, notamment l'emploi de graphismes tridimensionnels, tout en gardant la notion d'espace de travail non typé : ces structures graphiques pourraient par exemple servir à donner différentes représentations aux objets compositionnels (sonagrammes, représentation iconiques, etc. . .). Un écran d'une situation typique de mondrian se trouve dans la figure 9.

Plusieurs approches sont envisagées pour améliorer la lisibilité de l'ensemble, notamment une notion de zoom, qui permettrait de travailler sur un espace beaucoup plus large, l'écran ayant vite tendance à se remplir.

5. CONCLUSION

Dans cet article, les paradigmes actuels des interfaces graphiques pour la composition assistée par ordinateur ont été passés en revue, et il a été déterminé qu'une approche nouvelle s'avérait nécessaire pour permettre d'exploiter les outils conceptuels modernes d'objet compositionnel.

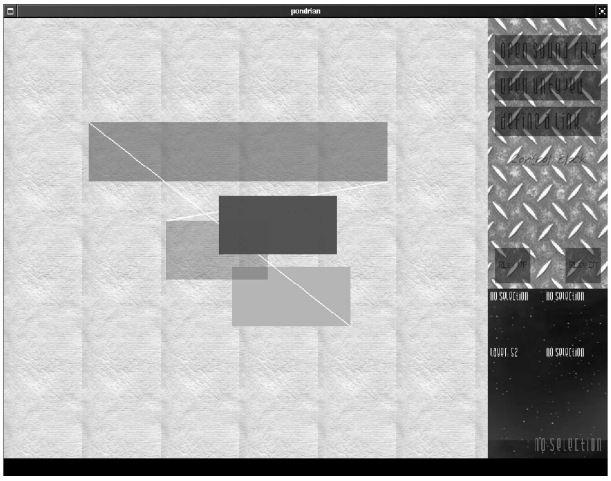

Figure 9. Un espace de travail dans mondrian

De manière spécifique, il a été montré que les notions d'*interface* et de *représentation* graphiques sont d'une importance capitale pour envisager le processus de composition électroacoustique. L'utilisation d'espace sémantique vierge, la volonté de ne pas imprimer de direction dans l'espace de travail, et d'évacuer la notion de processus et de flux de contrôle impliquent de se baser sur des notions d'émergence et d'interactions qui figurent parmi à la fois parmi les théories les plus modernes de la cognition [15] et de l'informatique [17].

6. REFERENCES

[1] Ashby, W. R. *Introduction to Cybernetics*, Chapman & Hall, London, 1956.

[2] Beurivé, A. "Un logiciel de composition musicale combinant un modèle spectral, des structures hiérarchiques et des contraintes", in *Actes des Journées d'Informatique Musicale 2000*, JIM, 2000.

[3] Chaudhary, A. et Freed, O. "Visualization, Editing and Spatialization of Sound Representation using the OSE framework", in *Proceedings of the Audio Engineering Society 107th Convention*, AES, 1999.

[4] Desainte-Catherine, M. et Beurivé, A. "Time Modeling for Musical Composition", in *Proceedings of the 1st International Conference on Fuzzy Systems and Knowledge Discovery : Computational Intelligence for the E-Age*, FSKD, 2002.

[5] Eaglestone, B., et Ford, N. "Computer support for creativity : Help or Hindrance", *ARiADA Texts*, University of East Anglia, 2002.

[6] Freed, A. *Improving Graphical User Interfaces for Computer Music Application*, CNMAT Website, 1996.

[7] Hanappe, P. et Assayag, G. "Composition on top of real-time sound synthesis", *Actes des Journées d'Informatique Musicale*, LMA, Marseille, 1998.

[8] Jordà, S. "New Musical Interfaces and New Music Making Paradigms", in *Proceedings of the NIME Workshop*, Seattle, 2001.

[9] Nuhn, R., Eaglestone, B., Ford, N., Moore, A. J., et Brown, G. "A qualitative survey of composers at work", in *Proceedings of the International Computer Music Conference 2002*, ICMA, Göteborg, 2002.

[10] Pachet, F., et Delerue, O. "A Mixed 2D/3D Interface for Music Spatialization", in *First International Conference on Virtual Worlds*, LNCS, Springer, 1998.

[11] De Poli, G., Piccialli A., et Roads, C. *Representations of Musical Signals*, MIT Press, Cambridge, 1991.

[12] Puckette, M. "Using Pd as a score language", in *Proceedings of the International Computer Music Conference 2002*, ICMA, Göteborg, 2002.

[13] Roads, C., Picciali A., et De Poli, G. *Musical Signal Processing*, Swets & Zeitlinger, Heereweg, 1997.

[14] Vaggione, H. "Some Ontological Remarks about Music Composition Processes", in *Computer Music Journal, 25 :1*, MIT Press, Cambridge, 2001.

[15] Varela, F. *Introduction aux Sciences Cognitives*, Seuil, Paris, 1996.

[16] Vertegaal, R. et Eaglestone, B. "Looking for Sound ? Selling Perceptual Space in Hierarchically Nested Boxes", in *Summary of the ACM CHI'98 Conference on Human Factors in Computing Systems*, ACM SIGCHI, Los Angeles, 1998.

[17] Wegner, P. et Goldin, D. "Computation Beyond Turing Machines", in *Communications of the ACM*, ACM, 2003.