

UTILISATION DE MODÈLES INTERMÉDIAIRES POUR LE MAPPING DE PARAMÈTRES DE SYNTHÈSE

RAPPORT FINAL

Master ATIAM - 16 juin 2008

Emilien GHOMI

sous la direction d'Hugues GENEVOIS



INSTITUT DE RECHERCHE ET
COORDINATION ACOUSTIQUE / MUSIQUE

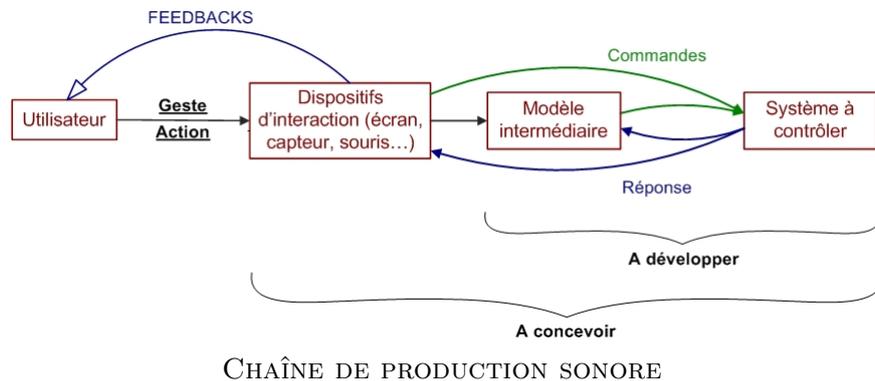


EQUIPE LUTHERIES, ACOUSTIQUE, MUSIQUE
DE L'INSTITUT JEAN LE ROND D'ALEMBERT

Résumé

Cette étude repose sur trois points.

Premièrement, l'établissement d'un état de l'art commun à tous les domaines de recherche impliqués dans la chaîne suivante, qui sert aujourd'hui de colonne vertébrale aux applications musicales interactives :



Deuxièmement, il s'agit de s'attarder sur la stratégie qui consiste à placer un modèle intermédiaire entre l'utilisateur et le dispositif de synthèse sonore, dans le but d'enrichir l'interaction. Et finalement, cette étude doit déboucher sur la définition d'une architecture qui éclaircit cette chaîne de production sonore, et sur l'implémentation d'un modèle intermédiaire pour en étudier le comportement au sein de cette architecture.

Lexique

- **Entrée / sortie** : concept rattaché à une vision en blocs. L'entrée est constituée de l'ensemble des informations qui rentrent dans un bloc, et la sortie des informations qui en sortent.
- **Dispositif d'interaction / Contrôleur / Périphérique de saisie** : différents termes utilisés pour parler du dispositif physique de contrôle utilisé (souris, clavier, capteurs de mouvement, joystick...)
- **Paramètres d'entrée / Paramètres de contrôle / Paramètre des dispositifs d'interaction** : mots pour évoquer ce qui parvient à la partie logicielle de la chaîne d'interaction, c'est-à-dire les gestes discrétisés par les dispositifs d'interaction.
- **Paramètres à contrôler / Variables du système / Paramètres du dispositif sonore** : éléments variables du système dont on peut contrôler l'évolution

Remerciements

Merci à Hugues Genevois de m'avoir accueilli une nouvelle fois dans ce beau lieu de la recherche, merci pour sa confiance et son humour.

Je tiens à remercier toute l'équipe du LAM et la sacrée team de chaleureux stagiaires qui n'oubliait jamais de me visiter dans mon beau bout de bureau bariolé.

Merci à Serge De Laubier pour ses méta-idées et ses méta-phrases, à Nicolas Montgermont d'avoir partagé ses expériences, à Vincent et Guillaume d'avoir été présents au Max

Un grand merci aussi à l'équipe d'ATIAM qui nous a permis de rencontrer des compositeurs de talents (Fabien Lévy, Claudy Malherbe, Jean-Claude Risset, Georges Bloch...), qui sont des exemples de créativité et des sources d'inspiration pour l'innovation dans les techniques musicales.

Merci à Carlos et Fleur pour leur patience!

Salutations basses à l'AMC (ATIAM Moustache Club).

Et finalement, je remercie tous ceux, musiciens ou non, scientifiques ou pas, qui sont venus tâter de la tablette graphique chez moi et qui n'ont jamais hésité à me dire ce qu'ils en pensaient.

*If everybody would agree that their current reality is **A** reality, and that what we essentially share is our capacity for constructing a reality, then perhaps we could all agree on a meta-agreement for computing a reality that would mean survival and dignity for everyone on the planet, rather than each group being sold on a particular way of doing things.*

(Fransisco Varela)

Table des matières

Résumé	i
Lexique	i
Remerciements	ii
Table des figures	v
Presentation du laboratoire	1
Introduction	2
1 Contexte scientifique et état de l'art	5
1.1 Le geste musical	5
1.1.1 Définition	5
1.1.2 Catégorisation	6
1.2 Mapping	7
1.2.1 Fondements	7
1.2.2 Conception	7
1.2.3 Complexité	8
1.3 "Modèles de comportement"	9
1.4 La théorie des Interactions Homme(s)-Machine(s)	10
1.4.1 L'interaction instrumentale	11
1.4.2 L'interaction située	12
1.4.3 Les modèles architecturaux	13
1.5 L'apport de l'enaction	13
1.5.1 Enaction et co-construction	13
1.5.2 L'intérêt pour la conception	14
2 Prise de position et choix de l'expérience	15
2.1 Positionnement et exigences	15
2.2 Choix du modèle intermédiaire pour l'expérience	17
2.3 Fonctionnement	17
2.4 Image et son	22

3	Développement et Résultats	25
3.1	Périphériques	25
3.2	Réalisation informatique	26
3.3	Architecture du modèle	27
3.4	Tests de blocs de synthèse	27
3.4.1	multiPVoc	28
3.4.2	ResonaTord	28
3.4.3	granHola	29
3.4.4	EffectChainEnOr	29
3.4.5	Perspectives	30
	 Conclusions	 31
	 References	 I

Table des figures

1	CHAÎNE DE PRODUCTION SONORE	4
2	MISE EN PLACE D'UN MODÈLE INTERMÉDIAIRE	4
3	REPRÉSENTATION DU MODÈLE TOPOLOGIQUE	18
4	REPRÉSENTATION DU MODÈLE TOPOLOGIQUE : TAILLES DE OBJETS	19
5	REPRÉSENTATION DU MODÈLE TOPOLOGIQUE : COULEURS DE OBJETS	19
6	REPRÉSENTATION DU MODÈLE TOPOLOGIQUE : FORMES DE OBJETS	19
7	INFLUENCE DE LA TAILLE DE L'OBJET SUR LA COURBE DE PROXIMITÉ	20
8	INFLUENCE DE LA COULEUR DE L'OBJET SUR LA COURBE DE PROXIMITÉ	20
9	FONCTIONNEMENT DU MODÈLE TOPOLOGIQUE	22
10	DISPOSITIF D'INTERACTION (CONTRÔLEURS) TESTÉS	25
11	DIAGRAMME DE FONCTIONNEMENT DE MULTIPVOC	28
12	DIAGRAMME DE FONCTIONNEMENT DE RESONATORD	28
13	DIAGRAMME DE FONCTIONNEMENT DE GRANHOLA	29
14	DIAGRAMME DE FONCTIONNEMENT D'EFFECTCHAINENOR	30

Presentation du laboratoire

L'équipe Lutheries-Acoustique-Musique (LAM, anciennement Laboratoire d'Acoustique Musicale) rattachée à l'Institut Jean-le-Rond d'Alembert (IJLRDA) et dirigée par Jean-Dominique Polack (directeur de la structure) et Hugues Genevois (responsable de l'équipe) est un laboratoire public de recherche créé en 1963 dans le département de Mécanique de la Faculté des sciences par Emile Leipp, sous la tutelle de l'Université Pierre et Marie Curie, du CNRS et du ministère de la Culture et de la Communication.

Ce laboratoire a pour but de traiter la musique, la voix, les bruits ou tout autre phénomène sonore du point de vue des sciences pour l'ingénieur (physique, acoustique) et celui des sciences humaines (psychologie cognitive, linguistique). Créé afin de répondre à la demande grandissante d'étude des sons et des bruits, de leurs modes de production, de leurs rayonnements et des divers problèmes de perception qui s'y rattachent, son programme de recherches prend souvent comme point de départ des problèmes concrets qui amènent des interrogations fondamentales.

Il s'articule autour des trois thèmes suivants :

- **Instruments de musique et voix**, qui regroupe les études du fonctionnement des instruments de musique (vocation première du LAM, liée à l'histoire du laboratoire au sein de la mécanique à l'UPMC). Elles ont porté ces dernières années sur la voix, la guitare, les cloches, le piano, l'orgue, les flûtes...
- **Perception et cognition**, qui développe, dans une optique cognitive, des recherches sur la qualité sonore par une approche des phénomènes sémantiques (analyse psycholinguistique)
- **Techniques audio**, qui est structuré autour de la prise et de la restitution du son, du traitement du signal et de la synthèse sonore, et enfin, autour du transfert, du stockage et de la conservation des enregistrements. Aussi par l'avènement des techniques numériques et la chute du clivage son/image, il couvre aujourd'hui tout le domaine audiovisuel

Les 27 personnes (permanents, chercheurs, linguistes, spécialistes du traitement du signal, doctorants) partagent une même passion pour la musique et la dynamique d'interaction de ses trois départements constitue un élément fondamental du fonctionnement de ce laboratoire.

Introduction

Dans le jeu des instruments acoustiques, la chaîne de production sonore telle qu'elle est souvent définie aujourd'hui (contrôle avec contraintes, retour haptique, production sonore par le système, retour acoustique et cognitif) est bien souvent difficile à étudier car les différents étages sont interdépendants et entremêlés.

Avec l'invention de la synthèse est née une possibilité de découplage total entre l'énergie produite par le musicien, et l'énergie acoustique en sortie du dispositif sonore [24]. Le premier exemple de synthétiseur est l'orgue (acoustique), face auquel le musicien déploie de l'énergie pour actionner les touches du clavier, alors que c'est l'air circulant dans les tuyaux et envoyé par des soufflets mécaniques qui produit le son. Nombreuses furent alors les expérimentations visant l'exploitation la brèche ainsi ouverte, en définissant un fonctionnement efficace d'un système électronique ou informatique entre l'utilisateur et le son qu'il produit.

Ce découplage présente principalement deux intérêts : il permet de définir des systèmes qui n'existeraient pas dans la nature, et d'imposer au système une évolution dans le temps qui peut changer radicalement sa réaction à une excitation. Le simple exemple de l'orgue est toujours valable ici : il serait impossible d'avoir un instrument à vent avec des résonateurs de la taille de ceux de l'orgue, et on remarque que le son produit par une même touche peut être tout à fait différent en fonction des "tirettes" actionnées au cours du jeu.

A partir de cette idée, et sans tenir compte pour l'instant des développements déjà réalisés, nous décidons de regrouper ces systèmes complexes servant d'intermédiaires entre le musicien et le son produit sous le terme de "modèles intermédiaires". Un modèle intermédiaire est donc un système régi par un ensemble de lois définissant sa réaction à des excitations, et qui présente des éléments variables et des états accessibles.

L'utilisation d'un tel modèle implique le déploiement de ce que l'on nomme "mapping" et qui n'est autre qu'une mise en relation des gestes effectués par le musicien avec les paramètres du dispositif sonore. Dans notre cas de système "tout numérique", les gestes du musicien sont mesurés et discrétisés par des dispositifs d'interaction comme des capteurs de mouvement par exemple. Plus globalement, nous dirons qu'il s'agit d'une mise en relation entre les paramètres d'un dispositif de contrôle ("d'entrée") et les éléments variables d'un système. Un exemple simple de mapping en musique est celui d'un fader de console numérique avec lequel on veut modifier la distance

perçue d'une source sonore. Pour ce faire, ce simple fader à une dimension doit faire varier non seulement l'amplitude du signal, mais également une réverbération et un délai. En effet, l'onde acoustique venant d'une source plus lointaine met plus de temps à nous parvenir, l'énergie de son champ réverbéré est plus importante et celle de son champ direct est moindre. Il y a donc bien une mise en relation entre la valeur de position du fader et trois variables du système sonore.

Mais cette notion de mapping n'a pas été inventée avec la synthèse sonore, elle est inhérente au comportement de chaque outil. C'est la possibilité de considérer séparément chaque étage de la chaîne de production sonore (notamment grâce au découplage amené par la synthèse) qui a permis de constater l'importance du mapping.

A partir de là, plusieurs approches furent déployées, notamment celle d'incorporer des formules mathématiques dans cette couche de mapping. Par exemple, imaginons un cas où une incrémentation d'une unité de la valeur d'un paramètre d'entrée multiplierait par deux la valeur d'un paramètre du système de synthèse. C'est en ce sens que des chercheurs allèrent jusqu'à inclure des " modèles physiques " dans la couche de mapping, c'est-à-dire une évolution temporelle issue des équations de la physique, qui est le lieu de diverses forces virtuelles, et dans laquelle des objets virtuels évoluent, pouvant même interagir entre eux !

Bien que la création musicale soit un domaine moteur pour ce type de recherches, le problème ne tarda pas à se poser de la catégorisation des opérations mathématiques effectuées entre le paramètre d'entrée et le paramètre du système. S'agit-il d'un modèle à part entière, d'une seule équation faisant intervenir plusieurs paramètres à la fois, d'une mise à l'échelle... ? Effectivement, on remarque dans des articles comme celui de Momeni et Henry [40] que la distinction n'est finalement pas triviale entre mapping, analyse, modification d'échelle, modèle...

L'objet de ce stage est la création d'une architecture de développement s'appuyant sur une véritable séparation des différents étages de la chaîne de production audiovisuelle, dans le but de généraliser, de théoriser et d'étudier la mise en place de "modèles intermédiaires".

Ainsi, la chaîne que nous choisissons de considérer, qui est issue des mondes des IHM et de la musique, est la suivante.

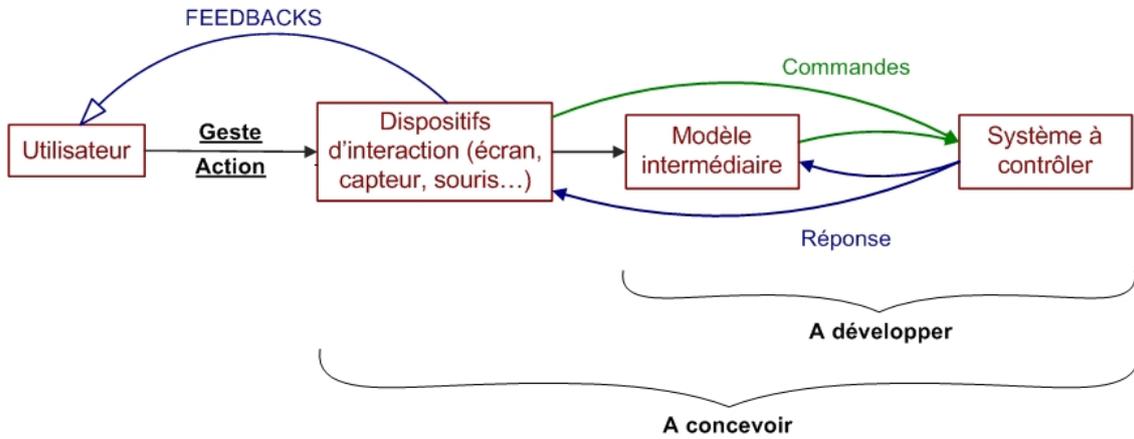


FIG. 1 – CHAÎNE DE PRODUCTION SONORE

Notre étude aura lieu dans la partie de la chaîne qui est en lien direct avec le modèle intermédiaire.

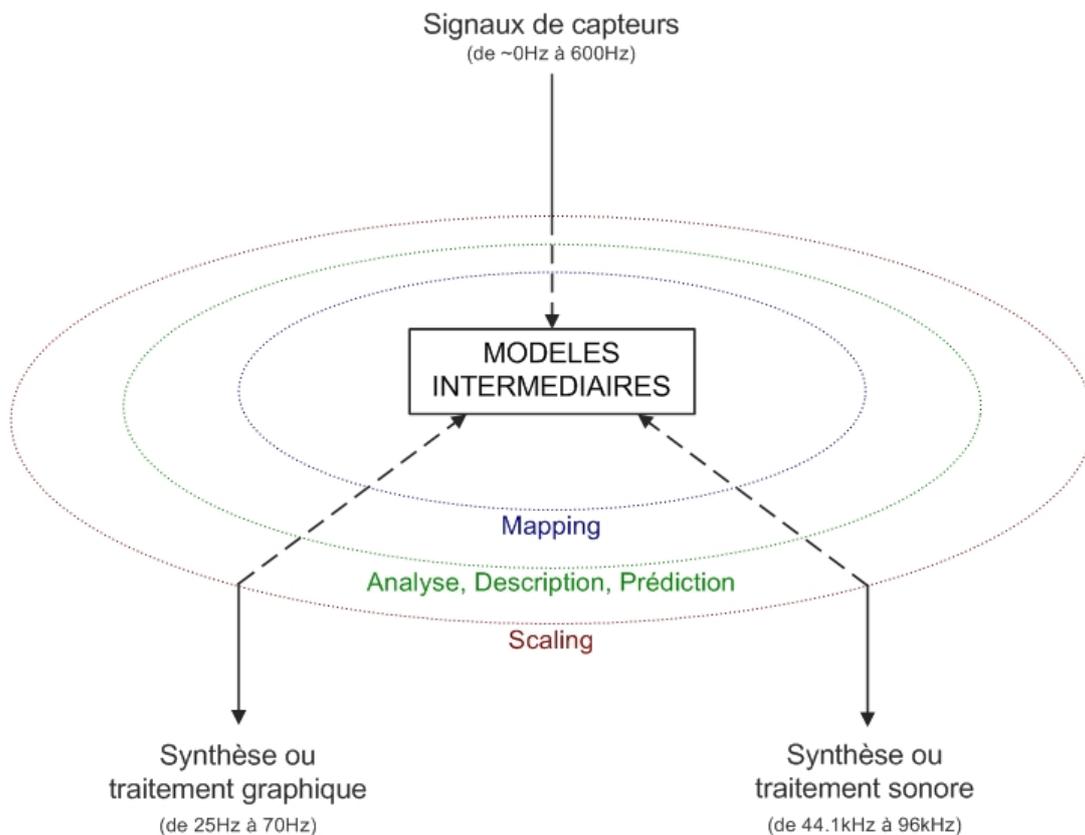


FIG. 2 – MISE EN PLACE D'UN MODÈLE INTERMÉDIAIRE

1 Contexte scientifique et état de l'art

Une telle problématique est inévitablement pluridisciplinaire. Effectivement, permettre la création de sonorités intéressantes, créer une application ergonomique, étudier la perception de l'utilisateur et optimiser l'architecture d'un code informatique sont des problématiques qui demandent des compétences et des approches très différentes et relèvent encore souvent d'équipes de recherches différentes.

C'est pourquoi, au risque d'aborder des thématiques apparemment ancrées dans les préoccupations d'un seul de ces domaines, ou d'évoquer des thèmes qui ne sont pas pleinement familiers aux développeurs d'applications musicales interactives, nous allons essayer d'évoquer ici les problématiques fondamentales de l'ensemble de la chaîne de production sonore présentée dans l'introduction.

1.1 Le geste musical

1.1.1 Définition

Tout d'abord, il est nécessaire d'apporter quelques définitions qui découlent de l'étude du geste de l'instrumentiste et tentent de formaliser au mieux la relation entre le musicien et son instrument (quel qu'il soit).

D'après Cadoz et Wanderley ([11], [15]), on appelle geste "une posture ou un mouvement du corps ou d'une partie du corps". Un geste instrumental a ceci de particulier qu'il n'est pas libre dans l'espace mais s'effectue "en contact avec un outil ou une matière". Cette dernière définition semble vraie pour les instruments de musique acoustique, mais ne l'est déjà plus tout à fait pour un instrument de synthèse tel que le Theremin. En effet, on joue du Theremin en modifiant la position des mains autour des antennes de l'instrument, mais sans aucun contact direct. Cependant, même si nous ne retenons pas cette notion de contact physique, la nécessité d'un outil semble primordiale.

Un instrument de musique est alors, quoi qu'il en soit, un "corps qui effectue la transformation du geste au son, dans le temps même de l'action". La notion de temporalité et de "temps-réel" est avancée à juste titre, et la problématique de transformation du geste vers le son est à la base du clivage entre instrument et contrôleur. Effectivement, peut-on dire qu'un ensemble de capteurs est un **instrument de musique** si le seul lieu de la synthèse sonore est l'ordinateur qu'il contrôle et qui ne fait donc pas partie de la même entité physique? Nous choisirons dans

cet exposé de ne pas nous attarder sur ce clivage, tout simplement car nous étudions la chaîne de production bloc par bloc, et que les dispositifs de captation de mouvements que nous utilisons sont effectivement dissociés de la production acoustique intégralement commandée par ordinateur. Le fait d'appeler " instrument " un ensemble de contrôleurs choisis ou l'ensemble constitué par les contrôleurs et une partie logicielle de l'ordinateur n'influencera pas directement nos expérimentations. En revanche, la notion fondamentale inhérente au fonctionnement des instruments acoustiques dont il faut s'inspirer est que le résultat sonore porte la trace du geste effectué, et ce lien doit être le plus clair possible, au moins pour le musicien. C'est ainsi qu'il pourra reconnaître et prévoir les conséquences musicales de ses gestes et, de ce fait , "apprendre" à jouer.

Enfin, les auteurs nomment geste de l'instrumentiste l'ensemble des actions physiques effectuées en situation de jeu musical, et l'on remarque que cette définition n'est pas limitée aux gestes qui fournissent effectivement de l'énergie pour la production acoustique.

1.1.2 Catégorisation

Depuis de nombreuses années, les gestes instrumentaux ont été catégorisés et différenciés au sein de cette définition : "effecteur, accompagnateur et figuratif" (François Delalande) ou " geste d'excitation, geste de modification et gestes de sélection" (Cadoz, [11]). Nous nous positionnons dans la terminologie définie par Cadoz. Le geste excitateur aura pour résultat une production sonore effective en fournissant de l'énergie à l'instrument musical, le geste de modification permettra d'altérer la nature du son en intervenant sur les paramètres ou la structure de la chaîne instrumentale principale, et le geste de sélection permettra de passer d'un certain comportement global à un autre en alternant entre des sous-ensembles de la chaîne instrumentale.

De ces premières considérations sur le geste musical, il apparaît indispensable que le dispositif puisse capter des gestes de catégories différentes, pour la construction d'une interaction complète propre au jeu instrumental. D'autre part, on remarque dès maintenant que la contrainte de temps-réel sera inévitable.

1.2 Mapping

1.2.1 Fondements

Étudions ensuite les différentes méthodologies de mapping ayant été formalisées. Wanderley insiste sur une répartition en 3, puis 4 types de mapping : one-to-one, one-to-many, many-to-many, few-to-many. Comme leurs noms l'indiquent, le premier type de mapping (one-to-one) met en relation un seul paramètre du dispositif de contrôle avec un seul paramètre du système à contrôler et ainsi de suite. Rovan préfère les appeler "divergent" et "convergent", ce qui regroupe un grand nombre de configurations complexes possibles [35].

Plusieurs théories du mapping présentent des résultats intéressants. Par exemple, Goudeseune [26] définit l'ordre d'un mapping comme étant le nombre d'opérations effectuées entre le paramètre de contrôle et le paramètre à contrôler. D'autre part, il y définit les paramètres "secondaires" qui modifient le rôle des paramètres "primaires". Par exemple le gain de sortie d'une console de mixage serait un paramètre primaire, alors qu'un bouton "mute" serait un paramètre secondaire. Il insiste également sur le fait qu'il ne faut pas mettre en oeuvre la même stratégie de mapping suivant que l'on veuille réaliser une tâche simple ou complexe. Et il justifie la réduction de dimension par le mapping (contrôler m paramètres du système avec n paramètres d'entrée, $n < m$) comme un moyen de réduire le problème aux paramètres qui ont un sens perceptif, les autres ne pouvant de toute façon pas être tous perçus par le musicien dans le temps de son jeu.

1.2.2 Conception

Si l'on aborde cette problématique à la lumière des préoccupations d'un développement informatique, on voit qu'il faudra définir une "couche de mapping" (ou un "bloc de mapping") pour faire le lien entre les paramètres d'entrée et les variables du système. Cette couche doit donc permettre de déployer telle ou telle stratégie de mapping en fonction des paramètres à contrôler et de la tâche à effectuer (qui dépendront évidemment des deux blocs qui communiquent par la liaison considérée). Comme pour Steiner et sa bibliothèque de mapping pour PureData ([37]) il n'est pas question ici de donner accès à des fonctions mathématiques complexes pour définir chaque lien du mapping. Il faut partir de liens élémentaires et faciles à mettre en oeuvre par l'utilisateur. Si l'on en juge la chaîne de production sonore présentée dans l'introduction (figure 2), nous remarquons que certaines opérations mathématiques auront davantage leur place dans la couche de "scaling", et certains comportements gagneront à être considérés comme appartenant au modèle lui-même.

1.2.3 Complexité

Cependant, même si les calculs mathématiques sont simples, on peut constituer un mapping complexe par le type de liaisons que l'on établit (comme évoqué dans l'introduction : les variations d'un seul paramètre d'entrée peuvent modifier les valeurs de plusieurs variables du système).

Hunt ([25], [24]) montre par plusieurs expériences que si les mappings simples peuvent être aussi efficaces que les mappings complexes de prime abord, leurs évolutions au cours du temps sont très différentes. Ses tests attribuent des scores aux utilisateurs suivant leur succès dans la réalisation de tâches diverses. Les mappings complexes donnent (sauf pour les tâches élémentaires), des scores plus élevés que les mappings simples et permettent rapidement une évolution des scores. On parle d'apprentissage en ce sens qu'un mapping complexe aura une première prise en main plus difficile, mais présente un fonctionnement qui pourra être appréhendé au fur et à mesure de l'utilisation pour permettre à l'utilisateur de vivre une véritable évolution du statut de novice à celui d'utilisateur expérimenté.

Nous choisissons ici d'écarter également la notion fréquemment avancée de "mapping dynamique", en la divisant en deux considérations distinctes : la couche de mapping peut être altérée par un geste de modification ou de sélection (par exemple le même paramètre d'entrée contrôle une autre variable du système quand tel ou tel geste est réalisé), et le côté dynamique au sens de l'évolution temporelle sera pour nous partie intégrante du fonctionnement du modèle intermédiaire.

Nous refuserons également de parler de "mapping par modèles physiques" ou "mapping par modèles intermédiaires". Dans le diagramme proposé dans l'introduction, le mapping a bien son rôle propre, et le modèle son fonctionnement propre, il est donc préférable de séparer les deux pour préserver la modularité de notre système.

C'est dans cette préoccupation de réduire la couche de mapping à l'essentiel que nous avons défini la couche appelée "analyse - description - prédiction" dans le diagramme présenté en introduction. En effet, si nous voulons par exemple contrôler une variable du système avec la vitesse d'évolution d'un paramètre d'entrée, c'est dans cette couche que l'on trouvera ce calcul, et non dans la couche de mapping. Il en va de même si nous voulons reconnaître un geste à partir des valeurs envoyées par le dispositif d'interaction. Ces nouveaux éléments qui décrivent l'évolution des paramètres de contrôle (vitesse, accélération, geste, entropie...) deviennent donc de nouveaux

paramètres envoyés à la couche de mapping (où ils seront redirigés vers les variables du modèle ou du système).

1.3 "Modèles de comportement"

Plutôt que d'établir un mapping direct entre les paramètres des dispositifs d'interaction et les variables de contrôle, plusieurs chercheurs proposent de passer par un "modèle" au comportement complexe, lieu d'interconnexions diverses, voire d'une certaine autonomie.

Cette approche permet de tendre vers un véritable instrument numérique, car elle apporte de nombreuses caractéristiques indispensables à un instrument de musique, comme par exemple une certaine perception des réactions du modèle, qui imposent des contraintes pouvant délimiter des modes de jeu et donner naissance à une liberté d'expression.

Les modèles de comportement sont variés, et leur étude n'a pas fait l'objet d'une réelle classification. En consultant diverses publications, on observe :

- les modèles dits "physiques" par abus de langage, qui sont en général basés sur des liens masse-ressort (Mongermont, Henry [41]) issus du modèle masse-interaction qui a été rendu accessible en informatique par la formalisation de Cadoz [38])
- la vraie classe des modèles physiques qui peuvent être basés sur d'autres comportements de la physique [38] (thermodynamique, électromagnétisme, gravitation...)
- les modèles de signaux (faire passer un signal de contrôle par une fonction de transfert par exemple)
- les "modèles" basés sur un mapping dit dynamique (réseaux de neurones [34], apprentissage...)
- les modèles statistiques (stochastiques de Xenakis, Stochos et Cosmos de Sinan Bokesoy, les "vols d'étourneaux", les " bancs de poissons "...)
- les modèles géométriques (Ali Momeni, David Wessel) et ceux que nous appellerons " topologiques " dont les principales variables sont spatiales : Syter (GRM), Hyperprism (Arboretum), Interpolator (Martin Spain), pMix (Oliver Larkin), Metasurface (AudioMulch)

Cadoz définit 10 critères pour évaluer la pertinence d'un modèle physique [38] (basés sur les 10 critères de Jaffe pour évaluer un dispositif de synthèse) axés notamment sur l'aspect "plausible" du son résultant, le retour haptique possible, et surtout la capacité du modèle à s'accorder à la

"réalité" intellectuelle construite par l'utilisateur, ce qui permettra de lui donner à voir de "vrais" objets, et non plus des objets virtuels.

Les points communs de ces différents modèles sont une évolution complexe contrôlée initialement par des paramètres simples, la multiplicité des variables du modèle et la rapidité d'exécution. On peut par exemple concevoir un modèle masses-ressorts grâce auquel on récupère en temps réel les coordonnées de milliers de masses en n'en contrôlant qu'une seule qui est reliée à toutes les autres.

Il est important de noter une distinction entre les modèles physiques dont la vocation est de générer directement le son (le mouvement du modèle est interprété comme une vibration qui génère une onde acoustique : Cadoz, les modèles d'instruments de l'IRCAM...), de ceux qui visent à contrôler les variables d'un système de synthèse (comme avec les bibliothèques PMPD et MSD développées pour Max/MSP et PureData par Cyrille Henry et Nicolas Mongermont).

1.4 La théorie des Interactions Homme(s)-Machine(s)

Le domaine des interactions homme-machine n'a pas encore la chance d'avoir une théorie fondamentale pour décrire chaque phénomène. Il existe cependant deux niveaux d'analyse et de conception des interactions : les **paradigmes d'interaction**, pour la conception de haut-niveau, et les **modèles d'interaction** pour la description opérationnelle [20].

Les trois principaux paradigmes d'interaction sont les suivants : l'ordinateur-outil (comme extension des capacités humaines), l'ordinateur-partenaire (approche anthropomorphique) et l'ordinateur-moyen (communication entre humains). L'objectif pour des chercheurs tels que Michel Beaudouin-Lafon et Wendy Mackay de l'équipe InSitu serait de regrouper ces trois paradigmes en une vision unique.

Quant aux modèles d'interaction, ils doivent fournir un cadre aux concepteurs et aux développeurs pour la création de systèmes interactifs, c'est-à-dire proposer un assemblage des différentes techniques d'interaction pour permettre un fonctionnement efficace. On peut évaluer ces modèles selon trois dimensions :

- **leur pouvoir descriptif**, c'est-à-dire leur capacité à décrire des interfaces existantes et d'autres nouvelles

- **leur pouvoir évaluatif/comparatif** qui propose des méthodes de comparaison de plusieurs techniques d'interaction alternatives ayant été développées, ce qui dispense d'une tentative hasardeuse de décision a priori de ce qui sera efficace ou non
- **leur pouvoir génératif** qui permet de créer de nouvelles techniques d'interaction

1.4.1 L'interaction instrumentale

" L'interaction instrumentale " [22] est un modèle d'interaction qui permet d'utiliser le paradigme de l'ordinateur-outil sur le plan opérationnel. En effet, il est fréquent que nous utilisions des objets secondaires pour étendre nos capacités et appliquer un effet souhaité sur un objet physique. Ce sont ces objets secondaires que nous appelons outils et le principe de l'interaction instrumentale est le même que pour les outils physiques : interagir avec des objets secondaires pour agir sur les objets d'intérêt (que l'on appelle "objets du domaine").

Un instrument est un transducteur entre l'utilisateur et l'objet du domaine qui fonctionne dans les deux sens. Une action physique est effectuée par l'utilisateur, l'instrument réagit et transmet une commande à l'objet du domaine. Ce dernier renvoie une réponse à l'instrument, qui fournit un feedback à l'utilisateur.

D'autre part, un instrument doit être "activé" pour pouvoir être contrôlé par l'utilisateur. Par exemple, une barre de scrolling qui s'active au passage de la souris est le lieu de ce que l'on nomme une "activation spatiale". Par suite, on peut également permettre le passage d'un état global du système à un autre pour que la même action sur le même dispositif d'interaction amène l'activation et l'utilisation de différents instruments.

Le pouvoir évaluatif de ce modèle repose sur trois critères :

- **le degré d'indirection**, lié à la distance spatiale et temporelle entre l'objet manipulé et l'instrument avec lequel on le manipule
- **le degré d'intégration**, calculé comme le rapport des degrés de liberté du dispositif d'entrée sur ceux de l'instrument
- **le degré de conformance**, qui compare la nature de l'interaction physique effectuée sur l'instrument avec celle de la réponse de l'objet, et amène ainsi deux taxonomies (l'une concernant la relation entre l'instrument et l'objet du domaine, et l'autre décrivant comment l'utilisateur agit sur l'instrument)

Pour ce qui est du pouvoir génératif, il a été montré qu'il pouvait être augmenté grâce aux trois principes de conception suivants [21] :

- **la réification**, qui est la construction d'un objet d'interface à partir d'un concept ou d'une commande abstraite; par exemple une scroll-bar est la réification du défilement dans un fichier texte
- **le polymorphisme**, qui traduit l'aptitude de l'instrument à être utilisé dans des contextes et sur des objets divers; comme le " copier-coller " qui est applicable à des chaînes de caractères dans un éditeur texte, des pistes audio dans un séquenceur...
- **la réutilisation**, qui permet à l'instrument développé d'être réutilisé plus facilement avec une entrée qu'on lui a déjà injecté, ou de rendre sa sortie réutilisable par la suite; comme les macros dans Excel

Comme nous parlons d'interaction instrumentale, il est tout à fait concevable de rajouter un étage de réification, et de concevoir ainsi des instruments qui enverraient des commandes à d'autres instruments...sortes de méta-instruments.

1.4.2 L'interaction située

Finalement, l'interaction peut être observée à deux niveaux distincts. Premièrement, Suchman parle "d'interaction située" pour évoquer une démarche où le système doit pouvoir **répondre aux actions** de l'utilisateur. Ces travaux sont continués aujourd'hui par Beaudouin-Lafon et Mackay [20]. Quant au phénomène "capteur-moteur", il impose que le système **corresponde aux capacités de l'utilisateur**. Certaines lois (Fitts, McGuffin...) permettent d'évaluer les techniques d'interaction sur ce dernier plan.

C'est ainsi que l'interaction et l'objet à manipuler sont mis au coeur du débat, et qu'on voit naître les premières applications post-WIMP (Windows, Icon, Menu, Pointer). Ces systèmes ont l'avantage de permettre un couplage fort entre l'action physique de l'utilisateur et la réponse de l'objet contrôlé. D'autre part, il est courant d'utiliser simultanément plusieurs dispositifs d'interaction pour effectuer des interactions plus complexes, ou par exemple permettre l'activation et l'utilisation simultanées de plusieurs instruments ([13], [14]).

Aujourd'hui, l'objectif principal des chercheurs en IHM est donc de créer des boîtes à outils informatiques dans lesquelles l'objet de base est l'interaction, prévue pour certaines tâches, mais réutilisable suivant les besoins de l'utilisateur.

1.4.3 Les modèles architecturaux

On appelle "modèle architectural d'interaction" la description des éléments fonctionnels d'une interface et des relations qui les lient. Ces modèles architecturaux sont au développement ce que le modèle d'interaction est à la conception. Les trois propriétés définies par Beaudouin-Lafon [20] pour évaluer les architectures d'interaction sont appelées :

- **"reinterpretabilité"**, capacité à être utilisé dans un contexte pour lequel le système n'était pas prévu, suivant les désirs de l'utilisateur
- **"resilience"**, résistance aux changements, qui est nécessaire pour que la propriété précédente soit valable
- **"scalability"**, capacité à supporter les effets d'échelle, comme utiliser une interaction prévue pour un fichier sur un groupe de fichiers

Ces propriétés sont indispensables car il est maintenant reconnu que les utilisateurs détournent souvent les techniques d'interaction ou même les instruments développés au cours de leurs interactions.

Il est donc question aujourd'hui de dépasser une démarche de prototypage d'une application interactive de création musicale à partir de ce modèle WIMP, par laquelle on choisirait par exemple le timbre dans un menu, et on "jouerait" les sons à la souris. Il est certes encore trop difficile d'intégrer de telles interactions directement dans les systèmes d'exploitation, mais certains logiciels de prototypage ou de programmation assistée (Max/MSP, PureData, Processing, VVVV...) peuvent tout à fait permettre le développement d'applications contenant des interactions de ce type, sans s'attarder sur de tels développements "à la main" avec les langages de programmation habituels.

1.5 L'apport de l'enaction

1.5.1 Enaction et co-construction

La thèse de l'enaction, proposée par Fransisco Varela en 1991 défend l'idée que la cognition est avant tout incarnée, chaque espèce a sa "réalité" propre, et évolue dans son propre monde avec ses propres règles (que l'on appelle conditions écologiquement situées). Il n'existe donc pas d'objectivité et de représentation universelle du monde pour l'ensemble de l'espèce humaine. Le corps est indissociable des mécanismes cognitifs, et il en est même le contexte. On parle ainsi de cognition située ou d'enaction.

L'idée de la co-construction avancée par Edgar Morin (et Horacio Vaggione pour ce qui est

du domaine musical) précise que la réalité est construite par une collaboration entre le monde extérieur et notre esprit, qui a lieu dans l'interaction. C'est finalement ici que l'on peut parler d'émergence. De cette co-construction située qui a lieu dans et par l'interaction, émerge une réalité propre à l'individu, ensemble de ses perceptions (actives) de l'environnement.

1.5.2 L'intérêt pour la conception

Un instrument ne peut donc pas être objectivement parfait, complet et universel, il faut voir si la " réalité " construite par l'utilisateur pendant l'interaction est efficace et intéressante par rapport au but recherché (dans notre cas, créer de la musique). C'est en ce sens que Castagne et Cadoz [38] parlent d'un " bon modèle mental " en évoquant un modèle qui permet à l'utilisateur d'anticiper le résultat de ses actions et de faciliter l'interaction avec le dispositif de synthèse sonore.

C'est pourquoi une démarche de conception d'un système de création musicale qui prendrait en considération ces nombreuses théories ne peut pas espérer créer une interface universelle et parfaite. La question serait plutôt de définir une utilisation claire avec ses limites en tenant compte de la perception et des réactions des utilisateurs. Il est d'ailleurs de même pour un instrument de musique : ses limites ne sont pas vraiment des contraintes, mais plutôt les bornes d'une utilisation "standard" et efficace, sortes de repères, et l'on peut jouer à les exploiter au maximum, voire à les dépasser.

La conception prend alors une autre couleur, car elle ne s'appuie plus sur le respect de fondements universels, mais sur une attention constante portée sur la cognition de l'utilisateur.

2 Prise de position et choix de l'expérience

2.1 Positionnement et exigences

Pour commencer, il n'était pas innocent de considérer la chaîne de production sonore bloc par bloc, de la perception de l'utilisateur jusqu'à la production acoustique. L'architecture modulaire qui a été choisie fonctionne aussi en ce sens : l'intérêt est de considérer non seulement l'utilisateur en tant que tel (suivant les thèses de Varela et Cadoz), mais aussi le système comme proposant des instruments réutilisables et efficaces pour des tâches prédéfinies (travaux de Beaudouin-Lafon et Mackay), ainsi que la grande expérience intellectualisée ou empirique des concepteurs d'application interactives de création musicale (Wanderley, Van Nort, Kessous, Rovan, Jean-Michel Couturier...).

L'originalité de cette approche est de prendre pleinement en compte chaque bloc du système, en mettant la perception de l'utilisateur au centre. En effet, l'efficacité d'une application, aussi réfléchie soit-elle, reste liée à l'enaction qui découle de son utilisation.

C'est par cette volonté de ramener chaque bloc à son fonctionnement élémentaire que l'application fait preuve de modularité : le modèle intermédiaire doit pouvoir recevoir des informations de multiples dispositifs d'interaction, et contrôler divers systèmes de synthèse graphique et sonore.

Les premiers résultats sonores auraient évidemment pu être obtenus en un temps de développement beaucoup plus court en raccordant directement les dispositifs d'interaction aux variables d'un système de synthèse, mais la modularité permet un grand nombre de tests, et surtout l'étude indépendante des blocs en ne modifiant qu'un point très précis de la chaîne. Cette possibilité est loin d'être négligeable car toutes les recherches visant à la création d'un instrument numérique de musique ou tout simplement à l'étude de modèles de comportement se sont accordés sur le fait que l'efficacité d'un modèle ne dépend pas uniquement des lois qui le régissent, mais également de réglages précis (sensibilité à une excitation, amplitude et vitesse de la réponse...).

Comme nos préoccupations sont avant tout musicales, nous nous efforçons de nous attacher à la notion d'instrument (au sens des IHM et au sens musical). Notre système complet (de la figure 2) doit donc répondre à certaines attentes :

- Il doit imposer des limites et des contraintes choisies qui ne sont pas des limitations à proprement parler, mais servent de repères à l'utilisateur pour une co-construction par l'interaction

- Il faut que les actions donnant une réponse riche ne soient pas toutes issues de pensées réflexives. La réalité perçue par l'utilisateur doit être assez claire et évocatrice pour laisser une place à l'intuition
- De la même manière, le système doit avoir un comportement globalement appréhensible voire prédictible pour que la conséquence d'une interaction soit identifiée, reconnue, mémorisée et qu'elle puisse être obtenue à nouveau (ce qui représente une forme d'apprentissage)
- En revanche, le résultat à une même interaction (au sens de l'humain et avec toutes les variations sensibles que cela implique) ne doit jamais être strictement identique. La captation des paramètres et leur traitement doivent donc être suffisamment fins pour amener des différences subtiles
- Le résultat sonore dépend de la facture instrumentale. Il faut donc que le modèle intermédiaire accepte des réglages et que l'on distingue la phase de " construction " de la phase de " jeu "
- Nous proposons un contrôle bi-manuel (comme c'est souvent le cas pour les instruments de musique) constituant un dispositif asymétrique

Kessous a mené un certain nombre de recherches sur le contrôle bi-manuel [13] en s'appuyant notamment sur le modèle de chaîne cinématique d'Yves Guiard. Ce modèle définit le corps en mouvement comme une succession de moteurs virtuels en série, chacun organisant son mouvement en fonction du mouvement du moteur précédent [14].

Ces théories expliquent qu'on peut considérer une main comme étant la "préférée" et il en découle deux échelles temporelles et spatiales distinctes : la main préférée permet un contrôle plus fin, plus rapide, mais elle ne peut agir que relativement à l'autre et donc son mouvement ne peut avoir lieu qu'après le mouvement de l'autre main (qui sert de "repère").

Un accent particulier est mis sur l'ergonomie du système, tant dans sa construction que dans son utilisation. Nous avons été amenés à échanger des idées et à travailler avec Serge de Laubier de Puce Muse, et nos problématiques sur ce projet sont communes : la création musicale, la performance scénique et la pédagogie.

D'autre part, nous avons décidé de ne pas définir d'apprentissage (par le modèle) des interactions réalisées (même si la couche de description / prédiction peut en présenter) car un tel dispositif favoriserait un contrôle expert par rapport à l'interaction d'un "novice" (Paradiso, [18]). Nous définissons donc un mapping dit "explicite" (Hunt, Wanderley, Kirk, [25]), ce qui ne l'empêche pas de pouvoir être modifié volontairement au cours du jeu par des gestes de modification ou de sélection. A terme, et quels que soient les modèles implémentés dans cette

architecture, l'utilisateur ne devrait avoir qu'à choisir certains blocs de notre chaîne de production sonore, et les autres lui seraient proposés. Il peut par exemple choisir le bloc de synthèse sonore (synthèse granulaire...) ou le modèle intermédiaire (masses-ressorts, stochastiques...) et des contrôleurs pourraient lui être proposés suivant la tâche visée. L'objectif de ces propositions serait de construire une chaîne cohérente, et que la perception du système par l'utilisateur le soit donc aussi.

2.2 Choix du modèle intermédiaire pour l'expérience

Nous choisissons finalement d'implémenter pour l'expérience un modèle topologique en 3 dimensions car il s'inscrit d'emblée dans notre problématique de modularité. En effet, on peut définir un espace, y mettre des "objets" et décider qu'ils correspondront à des échantillons audio à lire, des chaînes d'effets audionumériques vers lesquelles envoyer le flux audio, des notes à jouer... ([29], [30], [31], [32], [33]). Dans tous les cas, les objets situés dans l'espace peuvent, comme dans Syter, être des états de la chaîne de production sonore et le fait de passer de l'un à l'autre constitue bien un geste de sélection qui peut altérer la couche de mapping et le contrôle du bloc de synthèse sonore.

De plus, la problématique d'interpolation est ici primordiale car tout l'intérêt d'un modèle topologique est l'émergence d'un comportement dans tout l'espace à partir de la définition du comportement en un nombre fini de points (Watson, 1992). Effectivement, l'interpolation a lieu dans un espace délimité dont les bornes représentent des états choisis et on peut, en évoluant dans cet espace, atteindre toutes les configurations intermédiaires. On peut même concevoir (et c'est le cas dans notre modèle) une extrapolation lorsque l'on sort de la zone spatiale délimitée par les états choisis.

2.3 Fonctionnement

Nous sommes donc dans un espace à trois dimensions et, pour ce qui est de sa visualisation, nous fournissons trois angles de vue pour que l'utilisateur puisse s'y repérer au mieux : une vue en perspective, une vue du dessus et une vue de côté.

Dans cette espace, on trouve des objets de trois types : le point de position (qui montre à l'utilisateur la position courante pointée par le dispositif d'interaction), le point d'observation (qui peut être déplacé par l'utilisateur) et des objets qu'on ajoute. (Remarquons que tant que le bloc de synthèse et la couche de mapping n'ont pas été choisis et "câblés", ces objets n'ont pas

encore de rôle précis. Ce ne sont que des formes en 3d placées dans le même espace que le point d'observation).

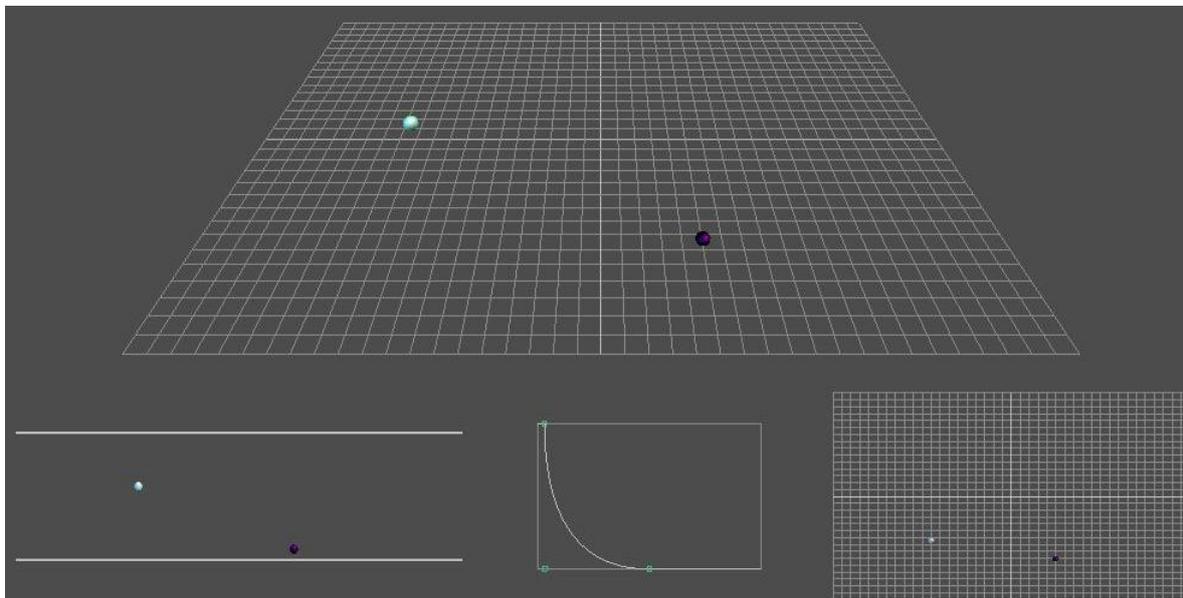


FIG. 3 – REPRÉSENTATION DU MODÈLE TOPOLOGIQUE

Dans la représentation ci-dessus de l'espace topologique, on voit au milieu la vue en perspective, en bas à gauche la vue de côté et en bas à droite la vue du dessus. Le point coloré est le point de position, et le point noir est le point d'observation.

On ajoute ensuite des objets, et on calcule alors en permanence la distance entre le point d'observation et ces objets. Mais il est intéressant que tous les objets ne soient pas équivalents aux yeux de ce calcul de distance, et après ces considérations purement géométriques, nous voulons calculer une autre grandeur plus subjective que nous nommerons "proximité".

Il faut alors que les objets aient certains caractères ou attributs qui influencent cette "proximité" perçue par le point d'observation. C'est pour cette raison (et également par un pur souci d'identification visuelle), que nous avons décidé de donner trois attributs à chaque objet créé : la taille, la couleur et la forme.

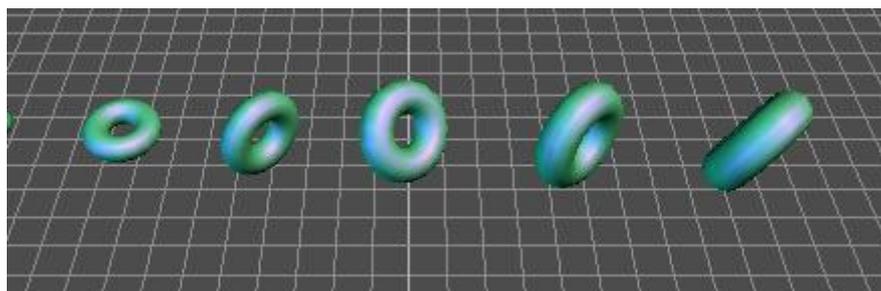


FIG. 4 – REPRÉSENTATION DU MODÈLE TOPOLOGIQUE : TAILLES DE OBJETS

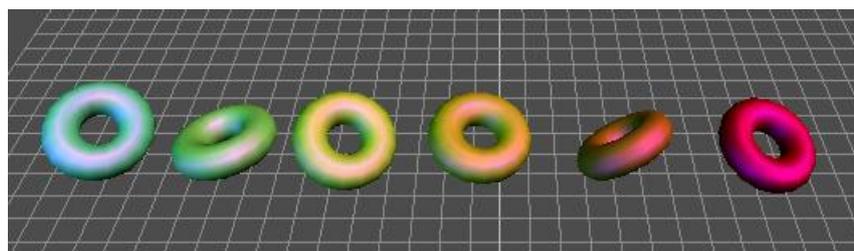


FIG. 5 – REPRÉSENTATION DU MODÈLE TOPOLOGIQUE : COULEURS DE OBJETS

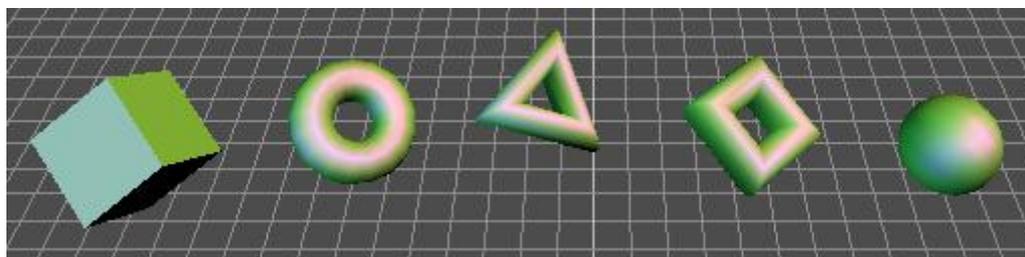


FIG. 6 – REPRÉSENTATION DU MODÈLE TOPOLOGIQUE : FORMES DE OBJETS

Globalement, la taille définit la portée de l'objet (un petit objet sera moins facilement perçu par le point d'observation qu'un objet plus gros qui sera perçu sur toute l'étendue de l'espace). Et la couleur définit le comportement au sein de cette portée : un objet ayant la valeur maximale de couleur a une influence importante sur la quasi-totalité de sa portée, alors qu'un objet ayant la valeur minimale de couleur n'aura une influence importante que si le point d'observation en est très proche, puis son influence décroîtra rapidement au sein même de sa portée. La forme n'a pas encore de rôle précis, mais elle pourrait en avoir un en fonction du bloc de synthèse choisi (par exemple un système de spatialisation où des sphères représentent des fichiers audio lus et des cubes représentent les enceintes)

Pour représenter cette influence de la couleur et de la taille de l'objet sur le calcul de la proximité, nous passons par le tracé d'une "courbe de proximité", qui n'est autre que la courbe de correspondance distance / proximité (on lit la distance en abscisse et la proximité en ordonnée).

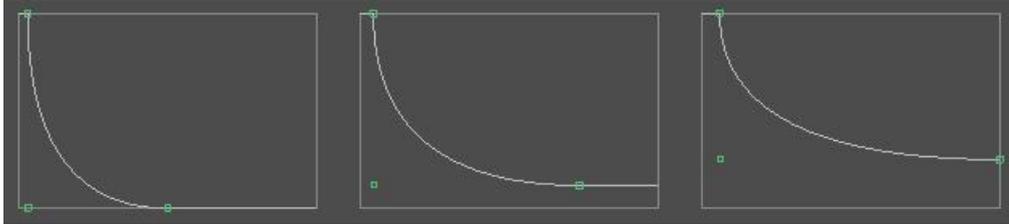


FIG. 7 – INFLUENCE DE LA TAILLE DE L'OBJET SUR LA COURBE DE PROXIMITÉ

Sur cette figure, on voit la modification de portée apportée à la courbe de proximité en fonction de la taille de l'objet. Pour la taille minimale (à gauche), dès que la distance entre l'objet et le point d'observation atteint la moitié de l'espace, la proximité est nulle et l'objet n'est plus perçu, alors que pour la taille maximale, la proximité a toujours une valeur assez importante, même à la distance maximale de l'espace.

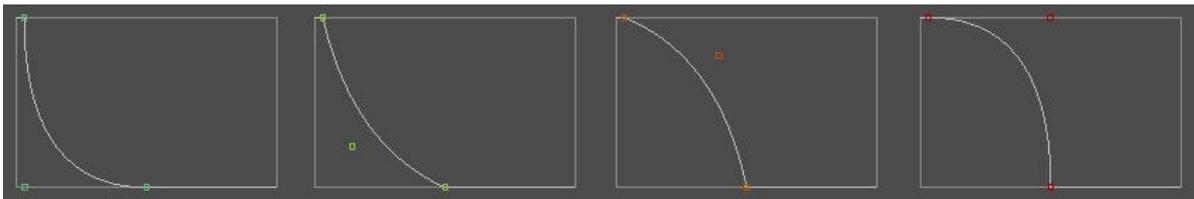


FIG. 8 – INFLUENCE DE LA COULEUR DE L'OBJET SUR LA COURBE DE PROXIMITÉ

Ici, on voit l'influence de la couleur. A gauche on a une valeur de couleur minimale, et à droite une valeur maximale. La portée ne change pas, mais le comportement sur cette portée est grandement modifié et cela se constate sur la forme de la courbe. Comme expliqué plus haut, l'objet de gauche de cette figure n'aura une influence puissante que sur une petite partie de sa portée. Pour visualiser l'évolution de cette proximité, nous avons ajouté un mouvement de rotation et de vibration qui dépendent de ce calcul. Un objet dont la proximité au point d'observation est grande tournera rapidement et vibrera fortement autour de sa position courante, alors qu'un

objet dont la proximité est nulle sera absolument immobile.

Les variables de notre modèle topologique sont donc les suivantes :

- la position du point de position
- la position du point d'observation
- le nombre d'objets
- et pour chaque objet :
 - sa couleur
 - sa taille
 - sa forme
 - sa distance au point d'observation
 - sa proximité par au point d'observation

Evidemment, la couche d'analyse pourra se charger d'étudier l'évolution de ces paramètres. Il semble d'ailleurs intéressant d'y ajouter d'emblée quatre calculs de vitesse : celle du point de position, du point d'observation, et les vitesses d'évolution de la distance et de la proximité de chaque objet.

Finalement, ces variables seront (après avoir franchi les couches de mapping, d'analyse et de mise à l'échelle) les paramètres de contrôle de nos blocs de synthèses sonore et graphique.

Pour revenir sur notre préoccupation quant à l'ergonomie du système, la construction de l'espace 3d a été rendue aisée, et les actions de construction (positionnement des objets) et de jeu (parcours du point d'observation) sont complémentaires mais dissociées (bien que la construction de l'espace puisse être sensiblement modifiée pendant le jeu, car les objets peuvent être déplacés et leurs attributs modifiés).

De par la modularité mise en oeuvre, il est possible d'enrichir notre système de plusieurs blocs de synthèse, de traitement ou même de plusieurs modèles intermédiaires et plusieurs stratégies de mapping fonctionnant en parallèle comme l'ont fait Van Nort et Wanderley [36] (par exemple, on peut imaginer que la prédiction d'un geste influe sur la structure générale du discours musical, pendant que les positions reçues par des capteurs au cours de ce geste déterminent les paramètres du son produit).

Voici un diagramme qui récapitule le fonctionnement de notre modèle topologique.

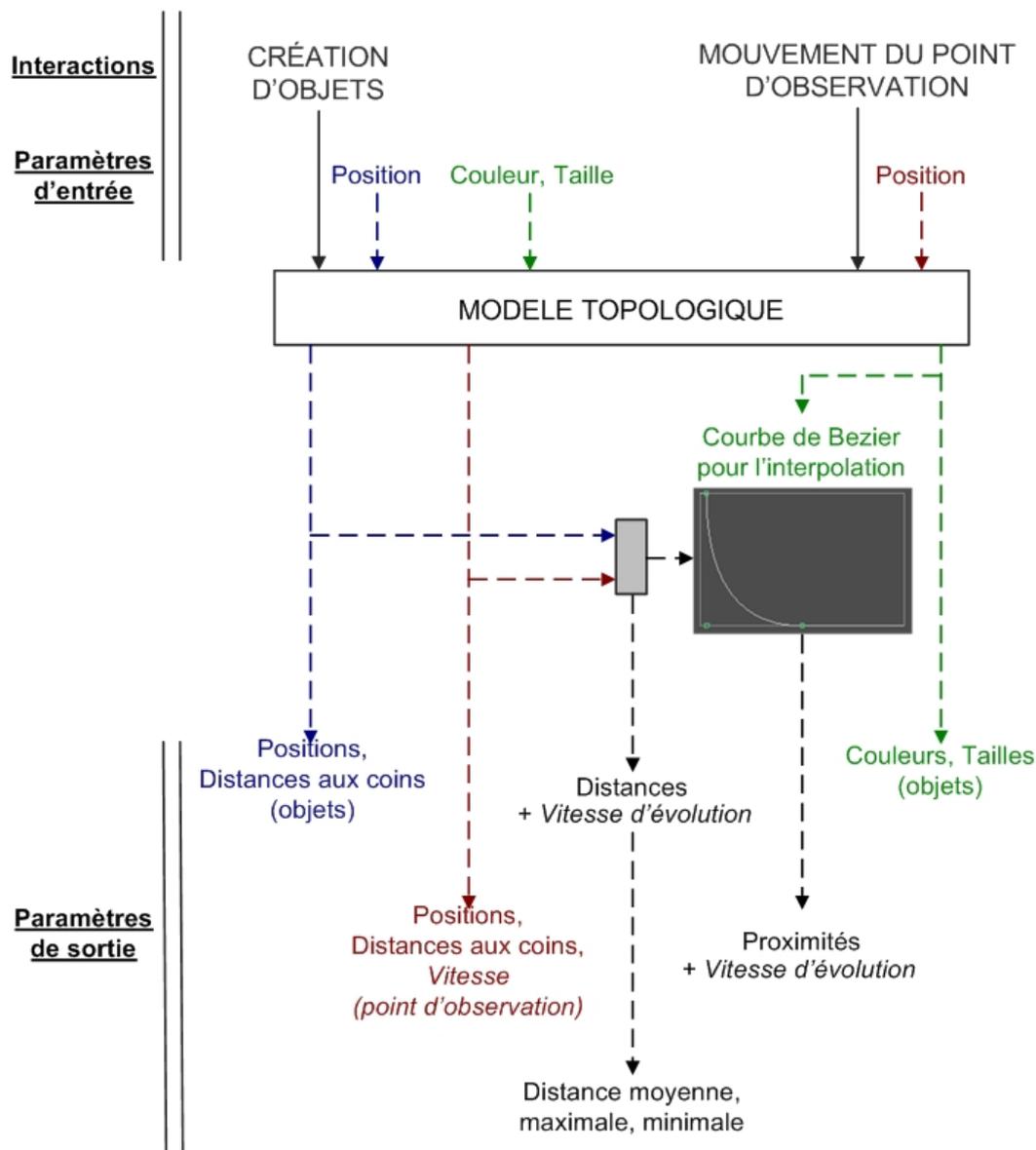


FIG. 9 – FONCTIONNEMENT DU MODÈLE TOPOLOGIQUE

2.4 Image et son

Tout d'abord, notons que nous n'avons pas d'a priori particulier sur la nature des blocs sonores et graphiques. Dans les deux cas, ils peuvent reposer sur de la synthèse (par exemple synthèse additive pour le sonore et placage de couleurs pour le graphique) ou du traitement (chaîne d'effets audio numériques pour le sonore, et opération sur la matrice d'un fichier vidéo pour le graphique).

Avec cette architecture et ces données de contrôle complexes, des blocs de synthèse ou de traitement simples doivent être suffisants pour produire un son riche. On connaît par exemple la richesse et la qualité des sonorités accessibles par la synthèse FM, mais on sait aussi que toute la difficulté consiste à développer un système qui permet de la contrôler efficacement pour profiter de cette richesse.

Une considération qui semble intéressante par rapport à notre problématique de création musicale serait de permettre une exploration intuitive du système. Par exemple, au lieu de choisir a priori le rôle d'un objet que l'on va placer, on pourrait avoir accès à un jeu de paramètres du bloc de synthèse, et chercher "à l'oreille" un état intéressant à fixer (ou réifier) en créant un objet (il n'y aurait pas non plus d'étude mathématique ou d'analyse comme pour construire l'espace des timbres de Wessel, la définition des objets ne reposerait que sur la perception de l'utilisateur).

C'est en ce sens qu'il faut que cette exploration soit accessible, pour permettre à l'utilisateur de définir ses objets en ne se focalisant que sur la modalité concernée. Ce n'est pas le cas par exemple des applications `pmix` et `int.lib` d'Oliver Larkin [30], dans lesquelles il faut en passer par une étape de configuration dans Max/MSP (basée sur des menus, des listes, et la saisie de chaînes de caractères) avant de pouvoir se déplacer dans l'espace d'interpolation ainsi créé.

Nous n'avons pas pour but de redéfinir ou redévelopper des dispositifs de synthèse. Nous sommes partis de dispositifs fréquemment utilisés et largement connus, dont les paramètres d'entrée sont clairs et fiables (synthèse FM, additive, granulaire, lecture d'échantillon...).

L'aspect graphique a été mis en retrait au cours du stage, en faveur de l'aspect sonore. Cependant, quelques expérimentations ont été faites pour représenter simplement l'évolution des paramètres. En tous cas, le modèle topologique sert à la fois de "représentation graphique d'appoint", et de représentation géométrique qui permet de visualiser le modèle sous une forme à laquelle nous sommes habitués, ce qui permet de réduire la charge cognitive (Mulder, [25])

La question du lien entre la vue et l'ouïe est par ailleurs extrêmement complexe. Le problème des simples représentations graphiques du son est qu'elles sont orthothétiques, c'est-à-dire qu'elles représentent exactement le phénomène sonore (selon tel ou tel point de vue) et n'y apportent donc pas grand chose, tant du point de vue de la visualisation, que de la perception ou de l'exploration [48].

De plus, il existe de fortes différences entre le son et l'image : le son se propage en trois dimensions et n'existe pas sans l'écoulement du temps (un "son fixe" n'a pas vraiment de sens), alors que la vidéo est bien souvent projetée sur deux dimensions, et une image fixe (photographie) a bien un sens en soit, sans que le temps ne s'écoule.

Il existe donc deux approches pour les applications audiovisuelles : l'approche séquentielle qui consiste à analyser le son pour contrôler l'image, et l'approche synchrone dans laquelle le son et l'image sont commandées simultanément par les mêmes paramètres de contrôle, ce qui crée un lien intrinsèque entre le sonore et le visuel.

Le bloc vidéo a néanmoins un rôle de taille, car la vue nous dirige énormément dans la construction d'une image mentale et le modèle de perception mis en oeuvre n'est pas le même suivant que le son est seul, accompagné d'une simple visualisation du fonctionnement (affichage de la représentation graphique du modèle topologique) ou mis en parallèle d'une véritable synthèse graphique.

3 Développement et Résultats

3.1 Périphériques

Premièrement, nous avons choisit d'utiliser des contrôleurs grand public permettant différents types de geste, un comportement bi-manuel asymétrique et présentant des contrôles de plusieurs types (relatifs, absolus, bornés ou non, avec ou sans retour visuel direct...[26]). En effet, il est intéressant de partir de contrôleurs dont l'efficacité est avérée car ils ont fait preuve d'un bon comportement lors de la réalisation de tâches élémentaires (Orio, Schnell, Wanderley, [16]). Nous n'avons qu'à les soumettre aux tâches élémentaires de notre système pour choisir le contrôleur adéquat.

Les auteurs ont défini des tâches adaptées au fonctionnement d'un contrôleur musical agissant sur l'espace timbral (ce qui n'est pas notre cas), mais nous prenons quand même en considération leur insistance sur les démarches que l'on doit pouvoir faire avec un contrôleur musical : apprentissage, exploration de l'espace sonore, le contrôle précis de paramètres et le contrôle temporel précis.

Wessel et Wright formulent les questions suivantes pour l'appréciation d'un contrôleur : est-il agréable ? Permet-il de développer des compétences particulières ? Amène-t-il un comportement particulier ? Est-il cohérent avec le modèle cognitif pour le contrôle ? [17]



FIG. 10 – DISPOSITIF D'INTERACTION (CONTRÔLEURS) TESTÉS

Finalement, nous avons choisi une tablette graphique pour la main préférée. Sa position est suivie en permanence par le point de position, et les différents boutons du stylet permettent de

saisir un objet pour le déplacer, en créer un nouveau ou en supprimer un existant.

Pour l'autre main, c'est le contrôleur MIDI BCF2000 qui fut retenu. Les attributs des objets (couleur, taille, forme) sont définis grâce à ses potentiomètres, et ses boutons permettent de choisir l'objet actif (pour le déplacement ou le changement des attributs), d'enregistrer et de charger des configurations spatiales.

3.2 Réalisation informatique

Tout le développement a été fait dans Max/MSP ([1], [2]) car c'est une application idéale pour la réception de données de contrôle et le temps réel. Il est également très avantageux de pouvoir profiter des ressources mises en ligne par la communauté de développeurs.

Néanmoins, le code nécessaire (pour les structures dynamiques) a été écrit en Lua, qui est un langage de script à typage dynamique utilisé notamment en intelligence artificielle et en jeu vidéo, et accessible dans Max/MSP grâce à l'objet `jit.gl.lua` de Wesley Smith ([3], [4]). Cela permet également la création de fonctions avancées qui aurait été très fastidieuse dans Max/MSP.

De plus, l'utilisation de cet objet Max/MSP a permis d'appeler directement des fonctions OpenGL standards dans le code Lua et donc d'avoir facilement la main mise sur l'éclairage, la caméra, les couleurs...([5], [6], [7], [8], [9])

A l'intérieur du patch Max/MSP, du script Max fut aussi utilisé pour gérer les gestes de sélection. Ce genre de script permet par exemple de passer d'un bloc de synthèse à un autre en recâblant l'intégralité des objets présents sur le patch en un seul clic.

Pour ce qui est du code, de nombreux efforts furent mis en oeuvre pour garder un bon niveau de performances. Les deux indicateurs utilisés à ces fins furent l'objet `"cpuclock"` de Max/MSP qui donne le temps processeur à l'instant voulu (et permet donc de comparer les temps d'exécution de deux chaînes), et le niveau de charge du processeur renseigné par Windows. Cela permit de prendre conscience que la recherche d'un point par dichotomie était beaucoup plus efficace, que la multiplication du nombre de facettes des formes OpenGL faisant augmenter considérablement la consommation de ressources, qu'il était bien plus avantageux de ne faire les calculs de distances et de proximités que pour les objets qui sont vraiment concernés...

3.3 Architecture du modèle

Les éléments variables du modèle topologique furent systématiquement stockés dans des variables globales car ils représentent l'état du modèle. Un jeu complexe d'accesseurs fut alors mis en place pour que plusieurs interactions n'accèdent pas simultanément aux mêmes variables globales. C'est en ce sens que fut également implémenté un système "d'exclusions mutuelles" (mutex).

Ensuite, le code fut réparti selon l'architecture propre à une bibliothèque, c'est-à-dire que chaque fichier Lua renferme un ensemble de fonctions qui partagent des natures ou des domaines communs. Voici la liste des fichiers (pour un total d'environ 2300 lignes de code) :

- Camera.lua : Contrôle de la caméra OpenGL
- ColorsShapes.lua : création des tables de couleurs (dégradés) et de formes
- Data.lua : deux fonctions de copie de valeurs (copyVect et copyMatrix) pour éviter d'être piégé par le passage par adresse systématique propre au langage Lua
- Debug.lua : deux fonctions pour afficher les valeurs d'un vecteur et d'une matrice dans la console de Max/MSP
- Distances.lua : gère le calcul des distances, des points de la courbe de proximité si la couleur ou la taille de l'objet ont été modifiées, et des proximités
- Draw.lua : gestion de la lumière et des matériaux OpenGL, et fonction de dessin principale
- glContext.lua : réglages généraux pour le contexte OpenGL de la fenêtre fournie par Max
- glLists.lua : les listes OpenGL qui sont des formes définies une fois pour toutes, stockées en mémoire et donc dessinées beaucoup plus rapidement
- inMAX.lua : les fonctions que Max appelle sans qu'on lui demande (comme une fonction qui est appelée quand le script est chargé)
- Maths.lua : calcule de normes, normalisation, maximum d'un vecteur, maximum d'une matrice, somme des éléments d'un vecteur ou d'une matrice...
- objManagement.lua : création des formes 3d (à partir des prototypes fournis par Jitter dans Max/MSP). Gestion des tableaux d'attributs des objets (à la création ou la suppression d'un objet) et des presets
- UserInterface.lua : accesseurs, modification des attributs des objets, mouvements spatiaux

3.4 Tests de blocs de synthèse

Une fois le modèle suffisamment stable et l'architecture du code suffisamment claire, des blocs de synthèse sonore furent développés pour tester et observer des comportements et utilisations

très différents de notre modèle topologique.

3.4.1 multiPVoc

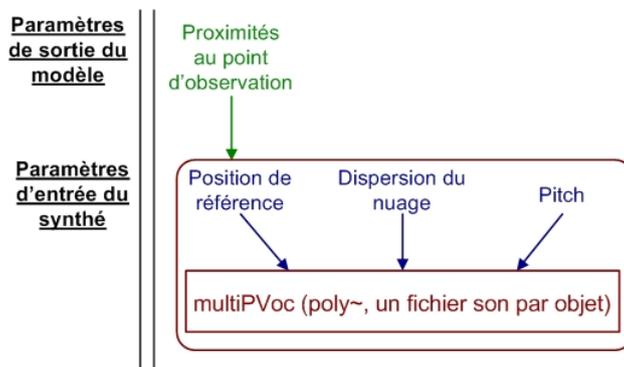


FIG. 11 – DIAGRAMME DE FONCTIONNEMENT DE MULTIPVOC

Ce bloc de synthèse est basé sur un objet de scrubbing développé par Richard Dudas. Son fonctionnement est des plus simples : chaque objet crée correspond à un fichier son, et la proximité de cet objet au point d'observation détermine la position actuelle de sa tête de lecture (si l'on est proche de l'objet, on est au début du fichier qui lui a été associé).

3.4.2 ResonaTord

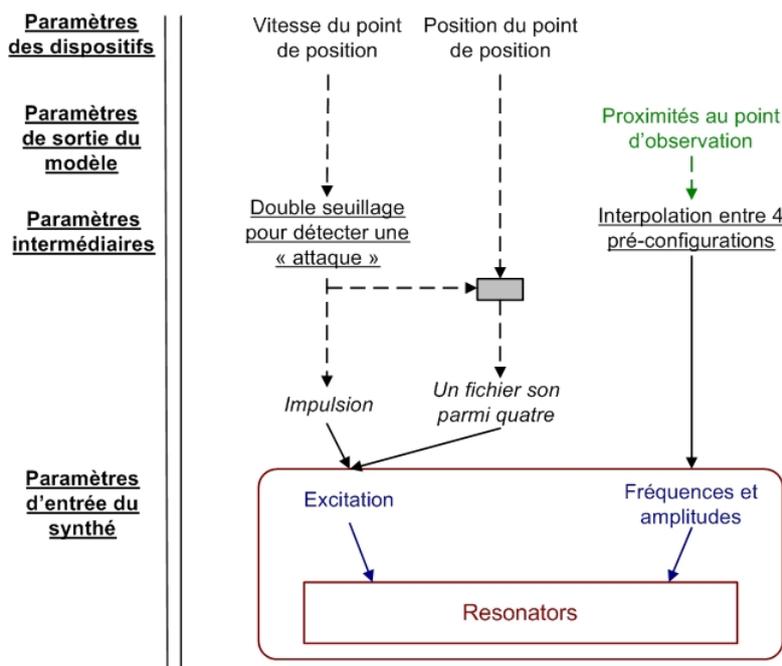


FIG. 12 – DIAGRAMME DE FONCTIONNEMENT DE RESONATORD

Ici le fonctionnement est plus complexe. Ce bloc de synthèse est contrôlé à la fois par des variables du modèle topologique et par une analyse de ces variables. Le son crée vient de l'envoi d'une valeur numérique seule (impulsion) ou d'un fichier son dans un filtre résonnant. D'une part, la vitesse du point de position (qui n'est en fait que la vitesse du stylet sur la tablette graphique), provoque l'envoi d'un signal dans les résonateurs, alors que la liste des proximités normalisées définit l'état actuel des filtres résonnants par interpolation de 4 configurations situées dans l'espace (4 objets).

3.4.3 granHola

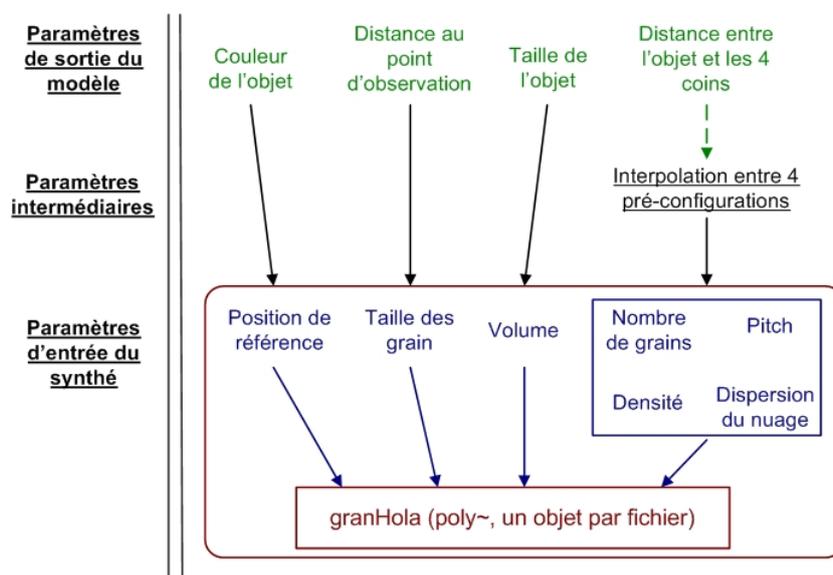


FIG. 13 – DIAGRAMME DE FONCTIONNEMENT DE GRANHOLA

Ce bloc repose sur de la synthèse granulaire. Chaque objet correspond à un fichier son. La couleur de l'objet donne la position de référence dans le fichier son, la taille spécifie le volume, et c'est la distance au point d'observation qui donne la taille du grain que l'on veut. Les états (du granulateur), ne sont pas ici les objets mobiles mais simplement les 4 coins du plan qui se trouve en bas de notre espace 3d. Chaque état présente un jeu de 4 paramètres : le nombre de grains, la densité des grains (s'il y a ou non un silence entre deux grains successifs) et la dispersion du nuage de grains.

3.4.4 EffectChainEnOr

Ici la richesse n'est pas dans l'interpolation elle-même mais dans les sons qui peuvent en découler. Cinq effets audio numériques ont été mis en chaîne, la liste des proximités des objets au point d'observation permet d'interpoler entre 4 pré-configurations de cette chaîne.

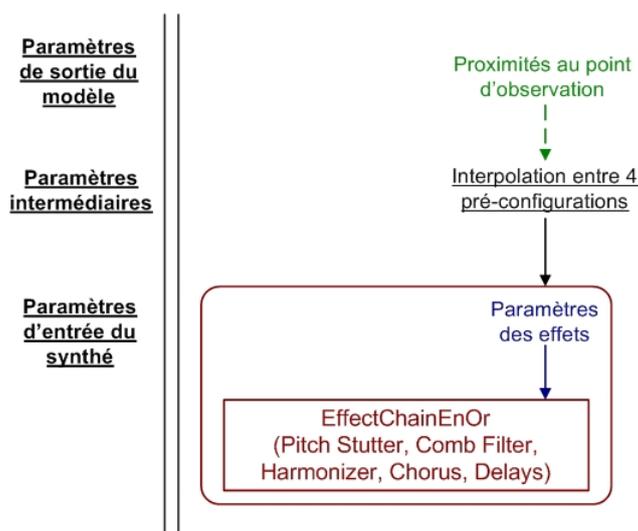


FIG. 14 – DIAGRAMME DE FONCTIONNEMENT D’EFFECTCHAINENOR

3.4.5 Perspectives

La notion d’exploration, plusieurs fois évoquée ici, permettrait une construction de l’espace et un choix des objets très intuitifs. De plus, elle permettrait certainement une liberté telle que de nouvelles sonorités seraient rendues accessibles.

Il serait très intéressant de profiter de la structure du LAM pour mettre en place des tests perceptifs liés à l’utilisation du modèle et des blocs de synthèse développés.

A terme, il serait intéressant de pouvoir gérer plusieurs espaces du modèle topologique en parallèle (par exemple un pour synthétiser des sons et un autre pour les traiter).

Mais l’un des freins majeurs à un développement plus avancé du modèle réside dans le fait que Lua est un langage de script (ici interprété dans Max et sans que Max soit en version de debug) et nous ne pouvons donc pas débogger le code réalisé ici. Quelques erreurs mineures perdues dans le code nuisent pourtant à sa stabilité et nous dissuadent de pousser le dispositif dans ses retranchements. La meilleure solution serait de développer un external pour Max en C++ sitôt les premiers tests perceptifs effectués (et les premiers résultats confirmés).

Conclusions

La décomposition de la chaîne de production sonore et la mise en relation des trois théories (IHM, Enaction et Interfaces Musicales) ouvre des portes vers des applications efficaces et intuitives.

Les premiers tests nous ont montré que l'architecture modulaire était d'une efficacité remarquable (le développement d'un instrument prend une demie journée pour qu'il soit jouable et permette une vraie exploration du matériau sonore).

Pour l'instant, force est de constater que quel que soit le fonctionnement choisi (du moment qu'il est cohérent) on s'habitue à chaque fois au nouveau fonctionnement en s'appuyant sur la représentation graphique de l'espace topologique. C'est pourquoi il faudrait d'abord tester la navigation dans le modèle topologique à l'aveugle, puis avec un dispositif de synthèse vidéo.

Il est aussi fortement envisagé d'essayer tous types de modèles dans cette architecture, et de voir quels sont les particularités et les difficultés des uns et des autres.

Je pense que l'utilisation optimale d'une telle interface consisterait, comme évoqué plus haut, en le choix de quelques blocs de la chaîne de création musicale, et que le système fasse des propositions quant aux blocs qui n'ont pas encore été choisis, en fonction de la tâche à accomplir.

Références

- [1] **CYCLING 74** - *Max 4.5 Reference manual* - (Manual, 2006)
- [2] **CYCLING 74** - *Max 4.5 Tutorials and Topics* - (Manual, 2005)
- [3] **Wesley SMITH, Graham WAKEFIELD** - *Real-time Multimedia Composition using Lua* - (Digital Arts Week Proceedings, 2007)
- [4] **Wesley SMITH, Graham WAKEFIELD** - *Using Lua For Audiovisual Composition* - (ICMC Proceedings, 2007)
- [5] **Antoine BOUTHORS** - *Tutoriaux OpenGL* - (Antoine Bouthors website, IMAG)
- [6] **Nicolas JANEY** - *Tutoriel d'infographie* - (Nicolas Janey website, Université de Besançon)
- [7] **Eric BITTAR** - *Formation à OpenGL* - (Eric Bittar website, Université de Reims)
- [8] **Jérôme JOUVIE** - *OpenGL tutorials* - (Jérôme Jovie website)
- [9] **Didier VERNA** - *Cours OpenGL* - (Didier Verna website, Epitech)

Contrôle, contrôleurs et IHM

- [10] **William BUXTON** - *Human Input to Computer Systems : Theories, Techniques and Technology* - (Website)
- [11] **Claude CADOZ** - *Instrumental gesture and musical composition* - (ICMC Proceedings, 1988)
- [12] **Marc BATTIER, Marcelo WANDERLEY** - *Trends in Gestural Control of Music* - (IRCAM, 2000)
- [13] **Loïc KESSOUS, Daniel ARFIB** - *Bimanuality in Alternate Musical Instruments* - (NIME Proceedings, 2003)
- [14] **Ken HINCKLEY, Randy PAUSCH, Dennis PROFFITT, James PATTEN, Neal KASSELL** - *Cooperative Bimanual Action* - (CHI, 1997)
- [15] **Marcelo WANDERLEY** - *Gestural Control of Music* - (IRCAM)
- [16] **Nicola ORIO, Norbert SCHNELL, Marcelo WANDERLEY** - *Input Devices for Musical Expression : Borrowing Tools from HCI* - (NIME Proceedings, 2001)
- [17] **David WESSEL, Matthew WRIGHT** - *Problems and Prospects for Intimate Musical Control of Computers* - (NIME Proceedings, 2001)

- [18] **Joseph PARADISO, Sile O'MODHRAIN** - *Current Trends in Electronic Music Interfaces* - (Journal of New Music Research, 2003)
- [19] **Jean-Michel COUTURIER** - *Utilisation avancée d'interfaces graphiques dans le contrôle gestuel de processus sonores* - (Thèse LMA, 2004)
- [20] **Michel BEAUDOUIN-LAFON** - *Designing Interaction, not Interfaces* - (AVI Proceedings, 2004)
- [21] **Michel BEAUDOUIN-LAFON, Wendy E. MACKAY** - *Reification, Polymorphism and Reuse : Three principles for Designing Visual Interfaces* - (AVI Proceedings, 2000)
- [22] **Michel BEAUDOUIN-LAFON** - *Instrumental Interaction : An Interaction Model for Designing Post-WIMP User Interfaces* - (CHI Proceedings, 2000)

Mapping et modèles de comportement

- [23] **Todd WINKLER** - *Making Motion Musical : Gesture Mapping Strategies for Interactive Computer Music* - (ICMC Proceedings, 1995)
- [24] **Andy HUNT, Marcelo WANDERLEY, Matthew PARADIS** - *The importance of parameter mapping in electronic instrument design* - (NIME Proceedings, 2002)
- [25] **Andy HUNT, Marcelo WANDERLEY, Ross KIRK** - *Towards a Model for Instrumental Mapping in Expert Musical Interaction* - (ICMC Proceedings, 2000)
- [26] **Camille GOUDESEUNE** - *Interpolated Mappings for Musical Instruments* - (Organised Sound, 2002)
- [27] **Joel CHADABE** - *The Limitations of Mapping as a Structural Descriptive in Electronic Instruments* - (NIME Proceedings, 2002)
- [28] **Doug VAN NORT, Marcelo WANDERLEY, Philippe DEPALLE** - *On the Choice of Mappings Based on Geometric Properties* - (NIME Proceedings, 2004)
- [29] **Jean-François ALLOUIS, Jean-Yves BERNIER** - *The SYTER project : Sound processor design and software overview* - (ICMC Proceedings, 1982)
- [30] **Oliver LARKIN** - *int.lib - A Graphical Preset Interpolator For Max MSP* - (ICMC Proceedings, 2007)
- [31] **Ross BENCINA** - *The Metasurface : Applying Natural Neighbour Interpolation to Two to Many Mapping* - (NIME Proceedings, 2005)

- [32] **Martin SPAIN, Richard POLFREMAN** - *Interpolator : A Two-dimensional Graphical Interpolation System for the Simultaneous Control of Digital Signal Processing Parameters* - (Music without Walls? Music without Instruments?, 2001)
- [33] **Ali MOMENI, David WESSEL** - *Characterizing and Controlling Musical Material Intuitively with Geometric Models* - (NIME Proceedings, 2003)
- [34] **Arshia CONT, Thierry CODUYS, Cyrille HENRI** - *Real-time Gesture Mapping in Pd Environment using Neural Networks* - (NIME Proceedings, 2004)
- [35] **Joseph Butch ROVAN, Marcelo WANDERLEY, Shlomo DUBNOV, Philippe DEPALLE** - *Instrumental gestural mapping strategies as Expressivity Determinants in Computer Music Performance* - (IRCAM, 1997)
- [36] **Doug VAN NORT, Marcelo WANDERLEY** - *Control Strategies for Navigation of Complex Sonic Spaces* - (NIME Proceedings, 2007)
- [37] **Hans-Christoph STEINER** - *Towards a catalog and software library of mapping methods* - (NIME Proceedings, 2006)
- [38] **Nicolas CASTAGNE, Claude CADOZ** - *10 Criteria For Evaluating Physical Modelling For Music Creation* - (DAFX Proceedings, 2003)
- [39] **Nicolas CASTAGNE, Claude CADOZ** - *Creating Music By Means Of "Physical Thinking" - The Musician Oriented Genesis Environment* - (DAFX Proceedings, 2002)
- [40] **Ali MOMENI, Cyrille Henry** - *Dynamic Independent Mapping Layers for Concurrent Control of Audio and Video Synthesis* - (Computer Music Journal, Spring 2006, drpichon.free.fr/ch/)
- [41] **Nicolas MONTGERMONT** - *Modèles Physiques Particulaires En Environnement Temps-Réel : Application au contrôle des paramètres de synthèse* - (Mémoire Master ATIAM, 2005)
- [42] **Ali MOMENI** - *Composing Instruments : Inventing and Performing with Generative Computer-based Instruments* - (Thèse Berkeley, 2005)

Synthèses

- [43] **Pierre SCHAEFFER** - *Solfège de l'objet sonore* - (GRM, 1966)
- [44] **Rudolf RABENSTEIN, Lutz TRAUTMANN** - *Digital Sound Synthesis by Physical Modelling* - (ISPA Proceedings, 2001)
- [45] **Julius O. SMITH** - *Physical Audio Signal Processing* - (CCRMA Website)

- [46] **Julius O. SMITH** - *Spectral Audio Signal Processing* - (CCRMA Website)
- [47] **Adrian FREED** - *Clear, Efficient Audio Signal Processing in ANSI C* - (CNMAT Website)
- [48] **Benoît COURRIBET** - *Réflexions sur les relations musique/vidéo et stratégies de mapping pour MaxMSP-Jitter* - (JIM Proceedings, 2005)

Perception

- [49] **Christian JACQUEMIN, Serge DE LAUBIER** - *Transmodal Feedback as a New Perspective for Audio-visual Effects* - (NIME Proceedings, 2006)
- [50] **Arnaud SICARD** - *Etude perceptive des inter-relations morphodynamiques gestes-images-sons* - (Stage ATIAM au LAM, 2006)
- [51] **Garth PAINE** - *Gesture and Musical Interaction : Interactive Engagement Through Dynamic Morphology* - (NIME Proceedings, 2004)