

GENESIS³ – PLATE-FORME POUR LA CREATION MUSICALE A L'AIDE DES MODELES PHYSIQUES CORDIS-ANIMA

Nicolas Castagné
ACROE
castagne@imag.fr

Claude Cadoz
ACROE & Laboratoire ICA, INPG
cadoz@imag.fr

RÉSUMÉ

GENESIS³ est une toute nouvelle version de l'environnement GENESIS pour la création musicale à l'aide des modèles physiques masse-interaction CORDIS-ANIMA. Elle a été conçue avec un recul de plus de dix années sur l'ancienne version et ses utilisations. Nous profitons de la sortie de GENESIS³ pour présenter une analyse du « paradigme logiciel » porté par GENESIS, en particulier en ce qui concerne les fonctionnalités et l'ergonomie offertes, les processus de création que l'environnement permet de mettre en œuvre et son positionnement dans l'état de l'art des logiciels musicaux utilisant les modèles physiques.

1. INTRODUCTION

Les « modèles physiques » ont une relativement longue histoire dans le domaine de l'Informatique Musicale. Depuis les premiers travaux dans les années 70 de Hiller et Ruiz, puis de Cadoz, de nombreuses techniques de modélisation et de nombreux modèles, algorithmes, et logiciels ont été proposés. Le travail de recherche et de développement reste intense, et l'engouement suscité par les « modèles physiques » dès les années 80 reste fort. Pourtant, la modélisation physique peine à s'installer dans les pratiques musicales.

1.1. Outils

Une rapide revue des logiciels qui ont recours au « modèle physique » montre que la majorité d'entre eux sont construits autour d'un algorithme modélisant une catégorie de corps sonore. Ils se présentent alors comme des systèmes de synthèse « haut de gamme » dédiés à la synthèse d'une catégorie de sons et viennent s'insérer dans les outils courants de l'informatique musicale – soit sous forme de « *plug-ins* » de synthèse, soit sous forme de synthétiseur hardware. Mais ils ne proposent pas, ou très peu, de moyens de modélisation. A l'inverse, il existe encore peu de logiciels ou systèmes qui visent à rendre l'utilisateur - musicien responsable de la conception des modèles. La plupart des environnements de modélisation disponibles (bibliothèque C/C++, langages dédiés...) sont de fait plus destinés à un usage scientifique que musical. Et, dans les quelques modèles proposés au musicien [8, 9, 10, 13], les briques de modélisation sont le plus souvent des

modèles de structure vibrante déjà construits, qu'il s'agit essentiellement de paramétrer et assembler entre eux.

Ainsi, si l'intérêt du modèle physique en ce qui concerne la qualité des sons générés tend à se répandre, peu de systèmes font le double pari que « l'utilisateur final » (musicien, compositeur...) est à même de mettre en œuvre un processus de modélisation physique complexe et qu'un tel processus a un intérêt réel dans le cadre d'une activité de création musicale.

Quelques raisons peuvent être avancées.

La première a trait aux réalités du marché de la synthèse sonore, dont l'un des moteurs essentiels est la recherche d'une plus grande qualité des sons de synthèse. Dans ce contexte, les modèles physiques offrent par nature un intérêt évident. Il n'est donc guère étonnant qu'ils se soient d'abord répandus en tant qu'algorithme de synthèse ou « *qu'expandeur* » : ils rencontrent ainsi une demande d'un « immédiatement utilisable ».

Une seconde raison envisageable est que le « portage aux usages » du modèle physique est pensé dans une situation de forte prédominance des systèmes de traitement du signal (qu'il soit sonore ou événementiel/MIDI), assise sur plus de cinquante années d'Informatique Musicale. La modélisation physique a ainsi tendance à être envisagée dans le contexte des techniques, systèmes, pratiques et savoir faire relevant du paradigme du signal, via l'incorporation de modèles dans des environnements logiciels « signal » répandus et pris en main par le plus grand nombre – exemples [8, 13].

Enfin, une troisième raison, plus délicate à analyser, relève à la fois de présupposés quant aux compétences nécessaires à la mise au point d'un modèle physique et de nécessités quant aux formalismes de modélisation susceptibles d'être pris en main par un musicien. Pour beaucoup, concevoir un modèle physique demanderait obligatoirement de manipuler des équations complexes et des moyens d'analyse numérique délicats, fort éloignés de tout discours et de tout savoir-faire musical. Ce n'est, pourtant, qu'un à-priori. De la même manière que les environnements de traitement du signal modulaire permettent de réaliser des systèmes de traitement à très haut niveau sans que, par exemple, une connaissance de la « transformée en Z » ne soit indispensable, il est possible d'envisager des systèmes permettant à un utilisateur final de concevoir des modèles physiques très complexes sans qu'il ne soit, par

exemple, nécessaire de manipuler frontalement des équations différentielles non-linéaires. La responsabilité des concepteurs de systèmes est, ici, engagée.

1.2. Création

Si l'on se tourne maintenant vers les usages du « modèle physique » en situation de création musicale, il apparaît que la modélisation physique est encore peu courante. Plus précisément, si des modèles préconçus sont bien utilisés en tant qu'algorithme de synthèse, parfois d'ailleurs sans que leur statut de « modèle physique » ne soit connu de l'utilisateur, les compositeurs ou musiciens sont encore rarement confrontés à l'activité de modélisation physique elle-même. Il est ainsi remarquable, par exemple, que des analyses des rapports entre création et « modèle physique », telle que celle proposée dans [5], se penchent essentiellement sur l'utilisation de modèles physiques par des compositeurs (modèles qui sont incidemment en général conçus préalablement par des scientifiques), et non pas sur l'impact du recours à la modélisation physique sur le travail compositionnel.

En d'autres termes, si le modèle physique est parfois utilisé par des musiciens, il n'est toujours pas « parlé ». Et l'impact potentiel du paradigme de la modélisation physique sur les processus de la création n'est pas encore pleinement mesuré.

1.3. Parallèle

Un parallèle avec le développement de la « synthèse du signal » fait apparaître un certain nombre de points marquants.

Tout d'abord, alors que l'univers du traitement du signal dans le contexte de l'Informatique Musicale s'est historiquement fondé sur une triple recherche technologique, scientifique et musicale (à commencer par les travaux de Mathews, Risset et leurs équipes), il semble que le modèle physique tende à être envisagé de façon beaucoup plus séparée : aux scientifiques la responsabilité et la prérogative de faire des modèles, aux musiciens celle de les utiliser.

Ensuite, alors que les outils « signal » étaient dès l'origine conçus de façon modulaire et ouverte, permettant à l'utilisateur de tester ses modèles, les « modèles physiques » sont quant à eux le plus souvent présentés comme des « boîtes noires » de synthèse dotées de quelques paramètres de réglage.

Enfin, alors qu'il existe aujourd'hui de nombreux outils modulaires permettant à un musicien de pratiquer le traitement du signal à très haut niveau, sans pour autant qu'il ne lui soit nécessaire de manipuler frontalement la théorie du signal, très peu de projets logiciels s'appuient sur un pari comparable en matière de modélisation physique.

L'évolution des outils « signal » a eu tendance à s'accompagner du développement progressif d'un véritable savoir faire en matière de compréhension,

synthèse et manipulation du signal sonore musicale auprès des musiciens (à minima des musiciens intéressés), allant jusqu'à ce qu'informatique musicale et traitement du signal sonore soient pour beaucoup presque synonymes. De façon notable, le développement des outils « physique » tend à sauter cette étape. Il y a là un risque de passer à côté des réels apports du paradigme de la modélisation physique en tant qu'outil pour la création.

1.4. Et pourtant...

Pourtant, la modélisation physique peut bien être envisagée comme une activité musicale.

Un musicien (qu'il soit compositeur ou instrumentiste) acquiert au cours de sa formation une connaissance à minima intuitive de la physique des corps sonores et une sensibilité forte à la matérialité des objets musicaux. Pour ne prendre que quelques exemples, les notions d'énergie, d'oscillation, d'interaction entre objets, d'inertie, de tension, d'amortissement, d'excitation, de percussion font partie du langage de la musique et du savoir intime du musicien, qu'il soit instrumentiste ou compositeur. Or, ces notions sont bien mieux appréhendables par le paradigme de la modélisation physique que par celui du signal.

De là, il est permis de supposer que la « modélisation physique » est à même d'être embrassée par un musicien, et ce de façon tout aussi pertinente, et peut être plus pertinente, que lorsqu'il s'agit d'avoir recours à la « modélisation du signal », qu'il soit sonore ou événementiel. Encore faut-il, bien sûr, d'une part que les techniques de modélisation et les environnements de modélisation adéquats soient conçus, d'autre part que l'utilisateur - musicien développe un savoir faire pratique, théorique et artistique avec ces outils.

1.5. GENESIS

Le projet GENESIS s'inscrit dans ce contexte. Considéré dans son ensemble, le projet a pour objet de faire figurer la modélisation physique au coeur des outils de l'utilisateur-créateur, ou encore de faire de la modélisation physique une activité de création.

GENESIS est un modéleur - simulateur dédié à la création musicale à l'aide des modèles physiques masse-interaction ou « particuliers ». Avec cet environnement, l'utilisateur est totalement maître de l'activité abstraite de construction du modèle.

Le projet s'appuie sur une quadruple recherche : recherche *technologique*, en ce qu'il s'agit d'identifier les principes, fonctionnalités et ergonomies originales d'un nouveau « paradigme logiciel » pour la musique ; recherche *scientifique*, en ce qu'il convient de porter un regard scientifique sur les possibilités en matière de modélisation dans l'environnement [11] : recherche *pédagogique*, afin d'identifier les moyens permettant de développer progressivement un savoir faire pratique et théorique auprès des utilisateurs [12] ; recherche

musicale enfin, relativement aux nouveaux processus de création qu'il convient d'inventer et aux nouvelles musicologies portées en gestation par le projet.

A l'occasion de la sortie de GENESIS³ (G³), la toute nouvelle version de GENESIS, cet article propose une analyse du « paradigme logiciel » propre à l'environnement, en particulier en portant un regard sur les fonctionnalités et l'ergonomie du logiciel et sur les processus de création qui sont mis en œuvre en l'utilisant.

La section 2 présente les objectifs qui ont gouverné la maturation de GENESIS³. Les sections 3 à 9 discutent les fondamentaux de l'environnement : formalisme de modélisation, fonctionnalités majeures, principes ergonomiques. Enfin, la section 10 propose une analyse des processus de création avec le logiciel.

2. DE GENESIS-1 A GENESIS³

Le projet GENESIS est fondée sur les recherches menées au groupe ACROE-ICA depuis 1976. Il a débuté en 1990 par la réalisation d'un premier prototype d'interface graphique et s'est poursuivi en 1995 par un prototype de GENESIS-1. De 1996 à 2005, la série GENESIS-1 [3] a été conçue et développée dans un processus en boucle. En parallèle, un réseau international d'utilisateurs a été progressivement constitué. GENESIS-1 est aujourd'hui diffusé dans plusieurs centres de Recherche et de Création. Il est utilisé régulièrement par un nombre croissant de compositeurs, pédagogues et étudiants. Il a été impliqué dans la création d'œuvres musicales régulièrement primées.

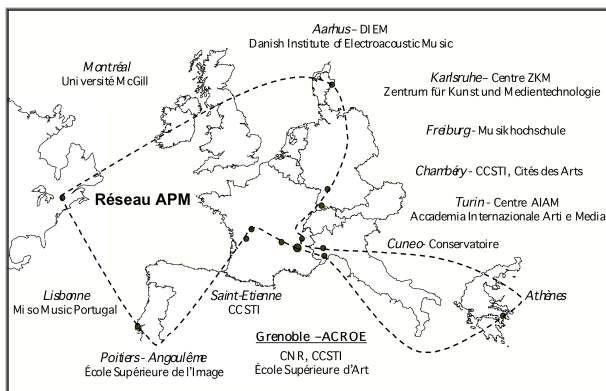


Figure 1 : diffusion de GENESIS

La conception d'un environnement nouveau et destiné au créateur tel que GENESIS *ne peut* se ramener aux méthodologies usuelles de l'Interaction Homme Machine. La collaboration avec les utilisateurs s'est déroulée comme une participation active à la maturation du logiciel, pendant laquelle la « raison d'être » de l'outil elle-même émergeait au fur et à mesure de sa conception. Autour de quelques objectifs communs, il s'agissait d'inventer des processus de création, dans une interaction mêlant les trois types de points de vue : scientifique, technologique et artistique. L'outil apparaît

alors comme le moyen qui permet de mettre en œuvre ces processus, mais aussi d'une certaine façon comme leur représentation. Cette démarche de « création d'un outil pour la création » nécessite du temps. Plus de dix années après la première version de GENESIS-1, cette démarche est plus que jamais active.

En 2005, nous avons entrepris une refonte profonde de l'environnement. Ce travail a abouti en janvier 2009 à la version *alpha* de GENESIS³ (G³), actuellement en cours de test. Les objectifs poursuivis dans la conception de cette nouvelle version sont multiples.

Il s'agissait d'abord de disposer d'une version s'insérant dans les plateformes informatiques des utilisateurs. En conséquence, G³ fonctionne sur Macintosh, sur Windows et UNIX/Linux, et dispose de nouvelles fonctionnalités permettant une insertion aisée de l'environnement dans les processus de travail des utilisateurs (import/export, copier/coller, lien avec d'autres environnements, etc.).

Plus fondamentalement, il s'agissait de stabiliser l'ensemble des éléments qui font la signature de GENESIS. G³ se présente donc comme une maturation de GENESIS-1, forte des nombreux tests et prototypes réalisés avec l'ancienne version et plus encore des connaissances accumulées lors de ses utilisations en situation de création.

Enfin, un troisième objectif consistait à poser les bases conceptuelles (fonctionnalités fondamentale, ergonomie générale, éléments génériques d'interface, langage de modélisation...) et matérielles (structures de données optimisées, codes réutilisables, formats de fichiers...) des futurs environnements de modélisation - simulation qui seront réalisés dans le laboratoire.

Pour ce faire, la conception de G³ a mobilisé l'ensemble des équipes du laboratoire et a nécessité une analyse approfondie de chacun des logiciels de modélisation - simulation réalisés à ce jour (en particulier GENESIS-1 pour la création musicale et MIMESIS [6] pour la création de mouvement et d'images animées) ainsi qu'une étude transversale des besoins. Le cœur de G³ est le résultat de ce processus de convergence. Aussi, plus qu'un simple « portage » d'un GENESIS-1 amélioré, G³ est la première brique d'une suite logicielle complète de modélisation - simulation physique pour la création. Derrière G³ se profile une nouvelle version de l'environnement MIMESIS ainsi qu'un nouvel environnement pour la modélisation et la création avec les systèmes de simulation temps réels multisensorielle ERGON [7] dotés de périphériques à retour d'effort. Ce dernier environnement sera, alors, un modeleur pour la modélisation d'objets dynamiques multisensoriels visibles, sonores, tactiles et manipulables dans une perspective de création, dans lequel cohabiteront des « objets » ou modèles multi-échelle, multi-fréquence et multi-dimensionnel (1D, 2D ou 3D). Ainsi, la mise en œuvre de G³ préfigure la découverte et l'expérimentation de nouvelles pratiques artistiques – et, pourquoi pas, d'un nouvel « Art des objets virtuels ».

3. FORMALISME

GENESIS est construit sur, et autour, d'une version du formalisme CORDIS-ANIMA [1], qui propose une formalisation précise du principe de modélisation et de simulation physique masse-interaction.

CORDIS-ANIMA est un formalisme extrêmement modulaire, ce qui présente un avantage évident lorsqu'il s'agit de laisser un utilisateur - créateur concevoir ses propres modèles. Un modèle CORDIS-ANIMA est un réseau de modules physiques élémentaires. Chaque module est un modèle d'un comportement physique élémentaire ; il est muni d'un algorithme paramétré encodant ce comportement. On distingue deux catégories essentielles de modules : les <MAT>, éléments matériels dotés d'une position dans l'espace de simulation, dont l'algorithme calcule la nouvelle position sur la base des forces reçues et les <LIA>, éléments de liaison connectés à deux <MAT>, dont l'algorithme calcule une force en fonction de la position et de la vitesse des <MAT> connectés. CORDIS-ANIMA définit également de nombreux *types de modules* <MAT> et <LIA> en fonction du comportement élémentaire modélisé (pour les <MAT> : point fixe, inertie, entrée ou sortie... Pour les <LIA> : élasticité, viscosité, relation de butée percussive, nombreux types d'interactions non linéaires...). Un modèle consiste alors en l'assemblage d'instances de ces types de modules (ou plus simplement «de modules») au sein d'un réseau physique.

Au delà du cadre formel fort et cohérent que propose CORDIS-ANIMA, des libertés sont laissées dans l'implantation qui en est faite. Nous précisons dans la suite la version de CORDIS-ANIMA qui est utilisée dans G³.

3.1. Uni-dimensionnalité

Le formalisme ne présuppose pas la dimensionnalité de l'espace de simulation (c'est-à-dire la dimension des vecteurs encodant les positions et les forces). Cet espace peut, en particulier, être uni-dimensionnel, plan ou tridimensionnel.

Dans GENESIS, l'espace de simulation est *unidimensionnel* : toute variable physique (force, position) n'est définie que sur un axe unique (« l'axe des mouvements ») et n'est calculée que suivant cet axe. De plus, les algorithmes des <LIA> ne considèrent pas la *distance* entre les <MAT> connectés, mais la *différence de positions* sur cet axe (qui peut être négative).

Le choix de la version unidimensionnelle peut se justifier par le fait que les vibrations acoustiques dans les objets réels sont en première approche le plus souvent mono dimensionnelles. Aussi, les modèles unidimensionnels permettent d'obtenir une très large variabilité dans les sons de synthèse, avec une bonne pertinence sonore. Qui plus est, n'avoir qu'un unique axe de mouvement permet de considérablement faciliter le processus de modélisation, en évitant la prise en compte de nombre des difficultés inhérentes à la

spatialité. Enfin, tel que discuté dans la section 5, cela autorise d'utiliser les deux dimensions de l'établi de modélisation comme un espace de liberté dans l'organisation des modèles.

3.2. Un très petit nombre de type de modules...

Nous avons veillé à limiter au maximum les types de modules disponibles dans GENESIS. Des tests ont été effectués au fur et à mesure des versions afin d'estimer la pertinence des modules envisagés. Dans G³, 12 modules, et seulement 12, ont été retenus et sont disponibles pour concevoir des modèles. De même, seulement 6 types de paramètres physiques et 2 types de conditions initiales existent. L'ensemble est résumé dans le tableau de la figure 2. Il convient de noter, qui plus est, que 4 de ces types de modules remplissent la fonction d'entrée/sortie et que les REF et CEL sont en fait des combinaisons d'autres types de modules. La base d'algorithmes physiques est donc réduite à 6.

<i>Modules <MAT> physiques</i>			
MAS	Inertie	$X0, V0$	Masse ponctuelle
SOL	Aucun	$X0$	Point fixe
CEL	Inertie Raideur Amortissement	$X0, V0$	Oscillateur physique élémentaire
<i>Modules <LIA> physiques</i>			
RES	Raideur		Elasticité
FRO	Amortissement		Viscosité
REF	Raideur Amortissement		Visco-élasticité
BUT	Raideur, Amortissement		Butée (liaison visco-élastique conditionnée à la différence de position)
LNL	Courbe $f(\Delta\text{Position})$ Courbe $f(\Delta\text{Vitesse})$		Liaison non-linéaire généralisée
<i><MAT> d'entrée/sortie</i>			
SOF	Aucun	$X0$	Sortie. Point fixe. Envoie la force reçue dans un fichier son ou sur haut parleur
ENX	Aucun	Aucun	Entrée. Lit sa position dans un fichier geste.
<i>Demi-<LIA> d'entrée/sortie (connectés à un <MAT> unique)</i>			
SOX	Aucun		Sortie. Point fixe. Envoie la position du <MAT> connecté dans un fichier son ou sur haut parleur.
ENF	Aucun		Entrée. Lit la force à appliquer au <MAT> connecté dans un fichier geste

Figure 2. les 12 types de modules. En deuxième colonne : les paramètres physiques (*et conditions initiales* pour les <MAT>)

Cette extrême limitation du nombre et de la complexité des briques élémentaires supportant l'activité de modélisation fait qu'il est relativement aisé pour l'utilisateur de s'imprégner en profondeur de leurs caractéristiques. C'est également une signature forte de

GENESIS, qui le distingue notablement de ce qui est proposé dans les autres environnements modulaires, en particulier pour la création musicale, dans lesquels les types de modules élémentaires sont bien plus nombreux et souvent plus complexes.

3.3. ... mais un très grand nombre d'instances

A l'inverse, conformément aux besoins des utilisations exigeantes du logiciel, G³ a été conçu pour permettre la mise en œuvre d'un nombre très important de modules (ie : d'instances des types de modules). La version actuelle accepte sans difficulté 100000 modules et nous travaillons pour découpler ce nombre.

On peut remarquer que les modèles réalisés dans d'autres environnements modulaires en Informatique Musicale comportent typiquement de l'ordre de 100 à 1000 fois moins d'éléments. Incidemment, en Informatique Graphique, si bien évidemment les logiciels de modélisation 3D supportent de très nombreux *vertex*, ceux-ci sont très rarement considérés individuellement mais au contraire manipulés en groupe *via* les formes qu'ils constituent.

Ainsi, le support d'un très grand nombre de modules est à nouveau une spécificité de GENESIS. Il a d'importantes conséquences sur l'ensemble du logiciel, tel que discuté dans la suite de cet article.

4. PRINCIPES

Une « philosophie » centrale dans GENESIS est de laisser l'utilisateur travailler au plus près du formalisme. GENESIS pourrait donc se définir comme un environnement permettant de pratiquer CORDIS-ANIMA. Cela est rendu possible car CORDIS-ANIMA présente *par nature* une forte « utilisabilité » (voir [4]). Cela est également souhaitable afin d'éviter de cacher l'activité de modélisation physique, sur laquelle nous voulons focaliser l'utilisateur, par un recours systématique à des fonctions de plus haut niveau. Ainsi, par exemple, si l'utilisateur souhaite un modèle de corde, il lui est d'abord proposé dans G³ de *construire* cette corde en assemblant et paramétrant des modules de façon adéquate. Cela, toutefois, n'interdit pas d'ajouter des fonctionnalités non immédiatement déduite du formalisme, qui prennent en compte le contexte musical de l'utilisation de GENESIS en permettant par exemple de modifier le modèle en cours d'édition en fonction de critères plus usuels dans les contextes musicaux. G³ propose bien sûr de telles fonctions, présentées dans cet article, mais nous avons systématiquement pris garde qu'elles ne fassent pas mystère de leur action sur le modèle et qu'il soit toujours possible d'en éditer ultérieurement le résultat en revenant à un niveau plus élémentaire.

Une seconde volonté importante, d'ailleurs corollaire de ce qui précède, est de toujours permettre à l'utilisateur d'accéder (de paramétrer, de reconnecter, de supprimer...) chaque module de façon indépendante. C'est la une chose nécessaire aussi bien pour la

topologie du réseau <MAT> / <LIA> (les grands réseaux à topologie régulière sont rares et, au contraire, les réseaux très hétérogènes nécessitant une action locale de l'utilisateur sont courants) que pour les paramètres (des parties à paramètres homogènes peuvent exister, mais ne constituent pas une règle absolue). Cette seconde volonté, alliée au fait qu'un modèle peut avoir un très grand nombre de modules, a là encore un impact sur l'ensemble du logiciel.

Enfin, un troisième choix fondamental est que l'environnement n'est pas orienté vers une utilisation temps réel. Il convient toutefois de noter qu'il n'y a aucune d'opposition *fondamentale* entre GENESIS et le temps réel ; CORDIS-ANIMA d'ailleurs été introduit précisément dans le contexte de la simulation multisensorielle interactive temps réel, qui reste une préoccupation majeure au laboratoire. Par contre, dans la mesure où GENESIS se veut un environnement de création complet, prenant en compte le niveau compositionnel, nous avons préféré mettre l'accent sur la possibilité d'éditer des modèles complexes et de grande taille, donc par nature « temps différé », plutôt que sur un support immédiat du temps réel. Le moteur de simulation embarqué dans GENESIS est donc un moteur « temps différé » ou dans le meilleur des cas « temps réel mou » qui ne prend pas de flux en entrée – à part, le cas échéant un fichier gestuel. La connexion de GENESIS au moteur temps réel dur synchrone développé au laboratoire est, elle, envisagée dans un proche avenir.

5. ETABLI

Tout comme dans G1, G³ est organisé autour d'une représentation graphique des modèles sur un « établi de lutherie » (figure 3). Une palette comprend un réservoir (inépuisable !?) de modules ainsi qu'un certain nombre d'outils de manipulation. Le modèle est éditable sur l'établi à la souris (ajout de module, connexions <LIA> / <MAT>, déplacement et réorganisation, suppression...) suivant les principes de la *manipulation directe*. La manipulation repose largement sur le principe de la sélection. La représentation des modèles et les actions de manipulation sont inspirées de celles de G1 – le lecteur peut se référer à [3] pour plus de détail.

L'unique axe des mouvements (ou « axe physique ») est l'axe perpendiculaire à l'établi. En conséquence, les deux dimensions de l'établi sont vierges de toute signification physique. Déplacer un module suivant sur l'établi n'a *aucun* effet sur les sons générés. L'utilisateur peut utiliser librement la disposition des modules sur l'établi pour représenter tout type d'information de nature symbolique. Les deux dimensions permettent par exemple une approche graphique (ou picturale) du processus de modélisation. Elles sont parfois aussi utilisées pour souligner la topologie du réseau <MAT> / <LIA> d'une partie de l'objet, pour symboliser le temps de la gauche vers la droite, ou pour exprimer visuellement une sorte de « partition » d'objets physiques, résultant d'une

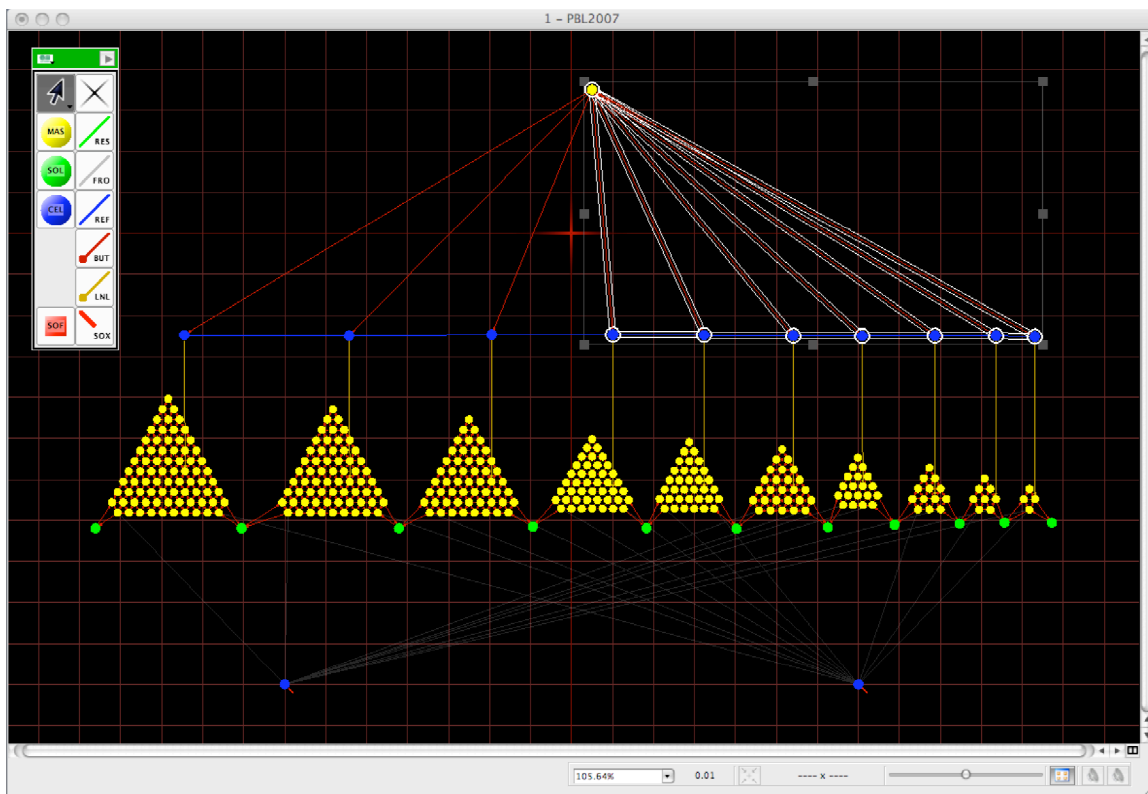


Figure 3. un modèle sur un établi (ici, sur plateforme Macintosh)

« composition » d'objets entre eux. De fait, les utilisations de la liberté de placement permise par l'établi varient d'un utilisateur à l'autre, d'un modèle à l'autre, ou même entre différentes parties d'un même modèle. De nouveaux usages sont régulièrement découverts et expérimentés par les utilisateurs.

Deux autres propriétés importantes de l'établi sont sa *taille* (quelques 100x100 m2) et, en conséquence, l'importance des *fonctionnalités de navigation* qui y sont offertes.

L'établi GENESIS s'inspire du principe des *interfaces zoomables* [14]. L'utilisateur peut zoomer en arrière ou en avant dans un facteur très important (10^5). G³ offre un ensemble de fonctionnalités de navigation optimisant le positionnement dans un tel espace. La représentation des modèles (ie : le niveau de précision dans la représentation d'une partie de modèle) dépend quant à elle du facteur de zoom et de la complexité du modèle suivant un algorithme heuristique visant à maximiser l'information pertinente affichée. De même, les possibilités de manipulation directe sont fonction de la complexité de la partie visible du modèle. En particulier, à certains niveaux de zoom les modules sont sélectionnés globalement sur la base de leur proximité sur l'établi ; alors qu'un zoom dans le modèle rend à nouveau chaque module manipulable de façon indépendante.

Enfin, une autre fonctionnalité originale sur l'établi est la possibilité d'y faire figurer des *commentaires hypertexte* ou « smart notes » afin d'annoter le modèle. Les liens hypertexte placés dans ces commentaires référencent des parties des éléments du modèle ou des éléments d'interface, à l'aide d'une syntaxe spécifique. Ils permettent, par exemple, de sélectionner rapidement

une partie de modèle, d'accéder immédiatement à un autre point de vue sur le modèle, etc.

6. LABELS

Nous avons consacré une attention particulière aux fonctionnalités permettant de manipuler et de maîtriser des modèles de grande taille. La première de ces fonctionnalités est bien sur la sélection, qui permet d'éditer globalement l'ensemble des modules sélectionnés. Elle n'est toutefois pas suffisante.

Dans l'objectif d'organiser des structures complexes, les solutions de loin les plus courantes, en particulier dans le contexte des outils d'informatique musicale, sont le *groupement* et l'*encapsulation*. Le groupement consiste à regrouper divers éléments de telle sorte qu'ils deviennent manipulables comme un tout. L'encapsulation ajoute au groupement le fait de cacher la complexité inhérente des éléments groupés en leur substituant une représentation iconique. Groupement et encapsulation permettent, à des degrés divers, d'organiser la complexité d'un document complet sous forme d'arbre. L'encapsulation permet en outre de simplifier l'organisation de l'espace de travail et est supposée réduire la charge cognitive de l'utilisateur en cachant des niveaux de complexité.

Toutefois, nous postulons que ni groupement ni encapsulation ne sont appropriés à GENESIS. Plusieurs raisons peuvent être avancées.

Tout d'abord, la position des modules sur l'établi est porteuse de sens et il est souhaitable qu'elle soit à tout instant modifiable de façon aisée.

Ensuite, il importe que chaque module reste accessible individuellement. Le cas des connexions

illustre cette assertion : alors que dans un modèle modulaire de signal il est possible de prédéterminer les entrées et sorties d'un « patch », n'importe quel <MAT> d'un modèle GENESIS est susceptible de recevoir de nouvelles connexions à n'importe quel moment. Par exemple, une « plaque » constituée de nombreux modules sonnera différemment suivant qu'elle est percutée au centre ou au bord ; prédéfinir le point d'excitation ne fait guère sens, et dans tous les cas l'utilisateur doit pouvoir le modifier aisément.

Par ailleurs, et plus fondamentalement, le processus de modélisation physique avec les réseaux CORDIS-ANIMA implique qu'il est le plus souvent impossible (et à *mimima* très difficile) de *strictement* isoler des sous parties dans le réseau. Dans un modèle physique, toute partie interagit bi directionnellement avec les autres – le cas échéant par parties interposées, bien sur. En conséquence, toute modification portant sur une partie du réseau, qu'elle soit paramétrique ou structurelle, nécessite pour conserver un fonctionnement cohérent du modèle de modifier en retour d'autres parties en interaction. De même, il n'est guère possible d'isoler des parties réutilisables en l'état : utiliser dans un modèle une partie conçue dans un autre modèle nécessite souvent des adaptations structurelles et / ou paramétriques en profondeur.

Enfin, de façon tout aussi importante, il apparaît que l'activité de modélisation nécessite de considérer tour à tour des parties de modèles qui se recoupent entre elles. Dans un cas, par exemple, il peut s'agir de déplacer globalement une partie sur l'établi ; dans d'autre, il peut s'agir de sélectionner uniquement certains modules dans cette partie pour leur appliquer un traitement qui leur est propre ; dans un troisième, il s'agira de considérer d'autres modules dans cette partie en même temps que certains modules d'autres parties pour modifier les paramètres physiques de ces modules ; etc.

En conséquence de cette analyse, il apparaît que *par nature* les réseaux physiques masse-interaction se prêtent mal à une approche structurelle en arbre, telle que portée par le principe de l'encapsulation. Aussi, nous avons consacré une attention particulière à la conception d'une fonctionnalité originale que nous appelons le *système de labellisation*.

Dans G^3 , un *label* est une chaîne de caractères qui désigne un module.

Un label peut comporter des *chiffres* ; le système a été optimisé pour traiter de façon extrêmement efficace des grands nombres de modules labellisés par des labels qui diffèrent les uns des autres d'un simple entier.

Un label peut également comprendre des séparateurs « / » en nombre quelconque. Ces séparateurs sont pris en compte par le système pour regrouper les labels entre eux sur la base de leur proximité syntaxique. On appelle alors *sous réseau* un ensemble de module qui partagent des labels ayant le même radical. Par exemple, si trois modules sont labellisés avec */filament/extrémités/extr1*, */filament/extrémités/extr2*, et */filament/module*, alors */filament* référence un sous réseau constitué de 3 modules et

/filament/extremities/ référence un autre sous réseau avec 2 modules.

L'utilisateur peut définir autant de labels pour un module qu'il existe de contextes, de tâches d'édition, de situations dans lesquelles ce module est susceptible d'être impliqué. Ainsi, un module pourrait être labellisé en même temps */filament/corde/m* et */sortiesAudio/module13*. De même, un module peut appartenir à autant de sous réseaux que nécessaire.

En conclusion, le système de labellisation permet d'organiser les *labels* des modules sous forme d'un graphe orienté, dans lequel les nœuds désignent des sous réseaux, et les feuilles des modules, mais ce *sans imposer une structuration au réseau lui même*.

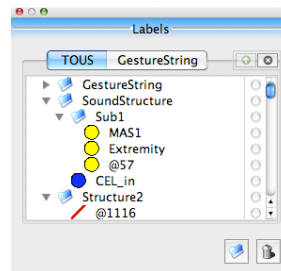


Figure 4. Fenêtre des labels

Dans GENESIS, le système de labellisation est relativement transversal. Il est notamment accessible dans plusieurs écrans, en particulier : dans la *fenêtre des labels* (figure 4), dans le menu contextuel de l'établi, dans les « smart notes » et dans le langage de modélisation PNSL (section suivante).

7. LANGAGE

La version actuelle de l'environnement MIMESIS [6] pour la création graphique incorpore un langage de script pour CORDIS-ANIMA qui intervient à plusieurs étapes clés du processus de modélisation. Les utilisations de MIMESIS ont montré que, pour de nombreux actes de modélisation, le recours à un langage (à la programmation) est particulièrement adapté. Ce n'est d'ailleurs pas chose étonnante puisque, par nature, la modélisation est une activité conceptuelle.

Dans le cadre de la conception de G^3 et plus généralement des bases de la future suite logicielle telle que présentée section 2, nous avons entrepris d'étendre les possibilités du langage MIMESIS et de réfléchir aux conditions de la collaboration de ce langage avec les outils de manipulation directe disponibles dans l'interface. Il s'agit, à terme, de réaliser une interface *multimodale* dans laquelle l'utilisateur pourra choisir le style d'interaction (langage / manipulation directe) le plus adapté à ses besoins du moment.

Le résultat de ce travail est la définition d'un nouveau langage, que nous appelons PNSL (Physics Network Scripting Language – Language de script pour les réseaux physiques). Cet article ne prétend pas présenter les quelques 67 commandes et 13 paquetages de PNSL ; nous dirons, toutefois, quelques mots sur PNSL et sur ce qu'il permettra de faire dans G^3 .

PNSL est un *langage de script* – construit sur le langage Tcl. A ce titre, il n'a pas été conçu pour *décrire* les modèles, mais bien pour les *programmer* – c'est à dire pour accompagner le processus de modélisation.

PNSL est un langage de modélisation. A l'inverse de nombre des langages accompagnant les interfaces, qui offrent des commandes permettant de manipuler l'interface, PNSL est focalisé sur le modèle et sa conception, suivant les règles du formalisme CORDIS-ANIMA. De la même manière que l'interface graphique de G³ permet de travailler au plus près du formalisme CORDIS-ANIMA, PNSL offre donc toutes les commandes permettant d'instancier, de connecter, de paramétrer, etc. les modules d'un modèle.

PNSL est *générique* dans le contexte de CORDIS-ANIMA ; il est adapté à n'importe quel type de modèle, quels que soient sa dimensionnalité (unidimensionnel, 1D, 2D, 3D), ou les catégories de phénomènes sensibles qu'il génère (image animée, son...). Le langage est ainsi amené à devenir une pièce centrale dans la suite logicielle en gestation (section 2).

PNSL a été conçu pour pouvoir être utilisé avec des *modèles de grandes tailles*, aussi bien en termes d'efficacité que de pouvoir expressif. En particulier, PNSL propose un système d'expressions régulières – les « *Labels Picker Expressions* » – qui permettent de sélectionner de façon concise des ensembles de labels.

Enfin, PNSL a été pensé pour « l'utilisateur final » du logiciel ; il pourra bien sûr être utilisé par les développeurs des logiciels pour réaliser rapidement et efficacement des fonctionnalités *ad hoc*, mais il est plus fondamentalement appelé à être pris en main par les utilisateurs – en particulier les musiciens – au même titre que l'interface de G³.

Dans la version actuelle de G³, et dans l'attente d'une intégration plus profonde du langage à l'interface, PNSL est proposé comme un *langage de macro*. Dans la *fenêtre des scripts*, l'utilisateur crée, modifie, édite et paramètre ses scripts. A tout instant, il peut exécuter un script sur un modèle en cours de conception. L'exécution du script modifie alors l'état du modèle, ce qui est immédiatement visible sur l'établi.

Les possibilités nouvelles offertes par PNSL sont pressenties comme étant importantes, mais ont tout juste commencé à être testées. Quelques exemples peuvent être d'ores et déjà cités, tels que :

Problème inverse : générer un modèle CORDIS-ANIMA de telle sorte qu'il dispose de propriétés reflétant un ensemble de contraintes souhaitées (par exemple : un spectre en fréquence), en utilisant des procédés d'analyse numérique ;

Edition des paramètres : calculer les paramètres d'un ensemble de modules de telle sorte qu'ils soient liés par certaines lois mathématiques ou physiques.

Position sur l'établi : placer automatiquement les modules sur l'établi de telle sorte que leur position reflète certaines propriétés des modèles qu'ils constituent. Par exemple : placement sur l'axe vertical en fonction de la fréquence propre du premier mode des

modèles, placement horizontal pour refléter l'instant auquel le modèle intervient dans une composition, etc.

8. AUTRES OUTILS

Les fonctionnalités fondamentales de la partie « *modeleur* » de GENESIS sont complétées par un ensemble d'outils qui permettent tour à tour d'analyser, de modifier, de générer ou d'organiser les modèles.

Sans présumer des développements à venir, nous listons ici les principaux outils disponibles dans l'actuelle version de G³.

La *fenêtre de génération* de topologie permet la génération de réseaux à topologie régulière (corde, plaque, etc.)

L'*analyseur modal* permet le calcul de l'ensemble des propriétés modales (fréquences propres, amortissement, déformée modale) d'un réseaux CORDIS, dès lors qu'il est linéaire. La *fenêtre d'analyse/accordage* offre plusieurs représentation du résultat des analyses modales et permet « d'accorder » à une fréquence donnée une structure linéaire préalablement analysée. L'accordage modifie les paramètres physiques de telle sorte que le mode considéré oscille à la fréquence voulue – les autres modes étant décalés en conséquence.

La *fenêtre d'organisation* propose un ensemble d'outil permettant de modifier le placement des modules sur l'établi en fonction de divers critères géométriques ou prenant en compte la topologie du réseaux de modules.

9. SIMULATION

Une particularité de GENESIS est que le modèle simulé peut donner lieu à deux types d'observation : visuelle (visualisation des déformations du modèle dans un espace 2D+1, c'est à dire comprenant le plan de l'établi additionné de l'axe perpendiculaire des mouvements) et sonore (écoute du signal sonore généré). Bien plus que le simple moyen de générer le signal sonore « final », la fonctionnalité de simulation est un outil essentiel dans l'activité de modélisation. L'utilisateur a très régulièrement recours à la simulation du modèle en cours de conception pour comprendre le comportement du modèle, évaluer le résultat obtenu, ou valider les actions de modélisation.

Avec G³, l'intégration des moyens de simulation dans le modeleur a franchi un degré de maturité.

Tout d'abord, nous avons maintenu le choix de deux écrans pour la conception et la simulation, soulignant ainsi que l'activité de l'utilisateur y est d'importance comparable, quoi que très différente. Ainsi, dans l'établi ont lieu les actions de conception et, dans la *fenêtre de simulation* (figure 5), le modèle prend vie.

La fenêtre de simulation propose plusieurs organisations suivant le besoin de l'utilisateur. L'organisation *écouter* permet de lancer le calcul de simulation, d'écouter le son obtenu et le cas échéant de visualiser la forme d'onde du signal sonore.

L'organisation *regarder* permet de visualiser les déformations du modèle au fur et mesure de son calcul et de contrôler la vitesse de calcul. Enfin, l'organisation complète (figure 5) offre un accès à l'ensemble de ces fonctionnalités.

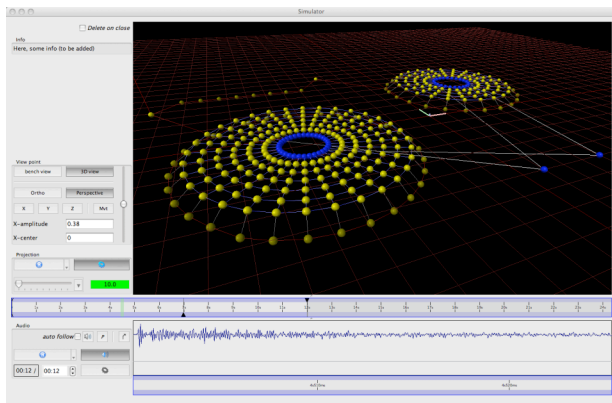


Figure 5. La fenêtre de simulation.

Un autre outil proposé, qui n'existait pas dans G1, est la *liste des simulations* qui liste l'ensemble des simulations réalisées depuis l'ouverture de l'application. Avec cette fenêtre, dans la mesure ou le lancement d'une simulation correspond de façon générale à un incrément significatif dans la conception du modèle, l'utilisateur garde une trace des étapes de son travail. Depuis cette fenêtre, pour chaque *item* de la liste, il est notamment possible de contrôler le processus de simulation, d'écouter les sons produits, de rouvrir la fenêtre de simulation, ou encore de rouvrir le modèle tel qu'il était lors du lancement de la simulation.

Enfin, notons que nous avons voulu dans G³ généraliser la possibilité d'un usage de plusieurs simulateurs CORDIS-ANIMA, de telle sorte que choisir le simulateur utilisé soit facilité. Cela permettra en particulier, dans un avenir proche, de connecter GENESIS au simulateurs TELLURIS et ERGON [7] conçus dans notre groupe de recherche. Ces simulateurs temps réel synchrones, multisensoriels, interactifs et utilisant les dispositifs gestuels à retour d'effort ERGOS permettront alors d'ouvrir GENESIS au temps réel et à l'instrumentalité, telle qu'elle est envisagée au laboratoire.

10. PROCESSUS DE CREATION

Utiliser GENESIS revient toujours à concevoir un modèle physique, puis à le mettre en vie. Contrairement à d'autres approches du modèle physique pour la création sonore et musicale, le logiciel ne repose pas sur des primitives toutes faites de morceaux d'instruments, réduisant la modélisation à une activité de juxtaposition et de paramétrage. GENESIS rend le musicien maître de l'activité abstraite de construction du modèle. Ainsi, les utilisateurs et utilisations de GENESIS démontrent qu'un musicien qui n'est pas scientifique est à même de s'appuyer sur une certaine « physique intuitive » pour

conduire seul un processus de modélisation physique qui peut être extrêmement complexe. Par ailleurs, avec ce logiciel la modélisation physique particulière devient bien un paradigme pour la musique, un langage pour « penser » et « écrire » le corps sonore.

Par contre, il est certain que les démarches mises en œuvre par les utilisateurs - compositeurs peuvent être très diverses. En particulier, une analyse des quelques 50000 modèles créés à ce jour avec G1 puis G³ au sein du groupe d'utilisateurs permet de mettre en évidence deux pôles complémentaires dans le processus de création mis en œuvre.

Dans le premier cas, GENESIS est utilisé essentiellement comme un générateur de sons. Les modèles peuvent être relativement simples, comprenant de quelques modules à quelques centaines de modules. Le son généré est lui-même relativement court – de l'ordre de quelques secondes à quelques dizaines de secondes. Il est ensuite exporté à la manière d'un « objet sonore », puis importé dans un autre environnement dans lequel il subira des transformations de montage et de modification, à la discrétion du compositeur. La composition de l'œuvre elle-même n'intervient donc pas précisément dans GENESIS, mais dans d'autres environnements, par exemple avec une démarche relevant de la musique concrète.

Dans le second cas, la totalité du processus de création musicale est réalisée dans GENESIS, en utilisant exclusivement le principe du modèle physique.

Ce processus est appelé par Cadoz la « composition (avec/des) modèles physiques » [2]. Il consiste à faire cohabiter dans un même modèle diverses catégories de structures physiques, certaines oscillant à haute fréquence (ie : produisant un son), d'autres à basse fréquence (typiquement : à la fréquence du geste). La simulation de telles structures en interaction génère alors une succession d'événements sonores – une « phrases sonore » – au lieu d'un son unique.

Cadoz montre qu'il est possible d'étendre considérablement cette démarche. Ainsi, des modèles très complexes, mêlant de nombreuses structures d'échelle variables et couvrant des fonctions musicales diverses, permettent de générer, lorsqu'ils sont simulés sans aucune interaction de l'utilisateur et sans qu'il n'y ait besoin de traitement ultérieur, non seulement un son, non seulement une succession de son, mais *une œuvre musicale de sons fixés complète et structurée*.

Dans ce processus de composition, le temps et la répartition temporelle des événements musicaux sont maîtrisés :

- Par les structures oscillant à basses fréquences qui remplissent la fonction de générateur d'événements. Concevoir une telle structure correspond à la conception, ou la « composition », des événements musicaux qu'elle génère lorsqu'elle interagit avec les structures sonores qu'elle active.
- Par des assemblages élémentaires composées d'un module MAS et d'une liaison BUT, elle-même connectée à une structure.

Nous appelons un tel composant un « déclencheur ». Lancé depuis une position initiale soigneusement choisie, et avec une certaine vitesse initiale, le MAS viendra « déclencher » la structure connectée à une date maîtrisée durant la simulation. Suivant les paramètres physiques utilisés pour le BUT, l'événement déclenché peut être une percussion, un étouffement, une excitation par plectre, etc. De même, la structure déclenchée peut être soit une structure sonore, soit une structure basse fréquence qui, à son tour, commencera son travail sur les structures avec lesquelles elle est en interaction. Quoi qu'il en soit, l'ensemble des « déclencheurs » d'un modèle permet de disposer d'un moyen simple mais adéquat d'écriture temporelle des événements sonores.

Le processus de « composition (avec/des) modèles physiques » a été mis en œuvre dans diverses œuvres au cours des années passées, notamment par Cadoz.

Un premier de ses avantages est qu'il permet d'étendre l'intérêt reconnu des modèles physiques en matière de qualité des phénomènes générés de l'échelle des sons à l'échelle des événements sonores, en rompant le paradigme du contrôle « top/down » porté par les systèmes de signaux. De fait, les structures génératrices d'événements étant perturbées par la rétroaction des structures « contrôlées », les événements sonores successifs qui sont générés présentent « naturellement » des différences légères dans leur placement temporel, dans la qualité de l'attaque ou encore dans leur timbre. Ces différences peuvent être musicalement pertinentes et porteuses d'expressivité ; elles sont, d'une certaine manière, à l'image des effets de l'interaction d'un musicien avec son instrument.

Plus fondamentalement, ce processus démontre que le potentiel du paradigme de modélisation physique particulière, lorsqu'il est envisagé dans le contexte de la création musicale, dépasse le cadre de la synthèse de son où il est traditionnellement cantonné, pour adresser l'ensemble du processus de la création musicale, incluant les problématiques compositionnelles. Avec ce processus, GENESIS offre au musicien la possibilité de « physicaliser » ses idées musicales et non seulement sonores ou instrumentales. Ainsi, GENESIS démontre que les modèles physiques sont porteurs d'une possible uniformisation des paradigmes pour la création et sont susceptibles d'apporter une solution à une « coupure ontologique » séculaire : celle du matériau (le son) et celle de la structure (la composition).

11. CONCLUSION

A l'occasion de la sortie de GENESIS³, nous avons voulu proposer une analyse des propriétés saillantes de GENESIS et préciser les processus de création avec l'environnement. Cet article a ainsi montré pourquoi GENESIS porte un « paradigme logiciel » original, comment les « modèles physiques » peuvent supporter une véritable activité de création musicale, bien au delà du rôle de génération du matériau sonore qui lui est souvent dévolu, et en quoi une telle activité de création à l'aide du modèle physique est nouvelle et intéressante.

GENESIS³ est actuellement en version alpha ; les tests sont en cours et une première version stable sortira très prochainement.

12. REFERENCES

- [1] Cadoz C., Luciani A. and Florens J. L., 1993: *CORDIS-ANIMA: A Modeling and Simulation System for Sound and Image Synthesis - the General Formalism*. Computer music journal 17(4):19-29.
- [2] Cadoz, C.: *The Physical Model as Metaphor for Musical Creation. pico..TERA, a Piece Entirely Generated by a Physical Model*, ICMC'02 proceedings, Sweden, 2002.
- [3] Castagne, N, Cadoz, C: *GENESIS: A Friendly Musician-Oriented Environment for Mass-Interaction Physical Modeling*. Proceedings of the International Computer Music Conference ICMC, Sweden, 2002.
- [4] Castagne N & Cadoz C: *A goals-based review of Physical modelling*, International Computer Music Conference - ICMC 2005 – Barcelonne - pp 343-346.
- [5] Chafe, C. *Case Studies of Physical Models in Music Composition*. International Congress on Acoustics, Kyoto, Japan, 2004.
- [6] Evrard M, Luciani A, Castagné N : *MIMESIS: Interactive Interface for Mass-Interaction Modeling*, Proceedings of CASA, Geneva, 2006, N Magnenat-Thalmann & al. editors. pp177-186.
- [7] Florens JL, Luciani A, Castagne N, Cadoz C. *ERGOS: a Multi-degrees of Freedom and Versatile Force-feedback panoply*. Proceedings of Eurospatics, pp356-360, Germany, 2004.
- [8] Henry C. *pmpd : Physical modelling for Pure Data* Poster, International Computer Music Conferences, november 2004, Miami, florida
- [9] Morrison JD, Adrien JM : *MOSAIC : a Framework for Modal Synthesis* – Computer Music journal vol 17/1 – MIT Press, 1993.
- [10] Pearson, M: *TAO: a physical modelling system and related issues*. Organised Sound, 1, 1996 , pp 43-50
- [11] Poyer F. & Cadoz C. : *Déformations de structures vibroacoustiques ; simulation par réseaux masse-interaction ; application à la synthèse de sons évolutifs*. In Actes des 14^{èmes} journées d'Informatique musicale – JIM-09, Grenoble, 2009.
- [12] Tache, O & Cadoz, C. : *Vers un instrumentarium pour les modèles musicaux Cordis-Anima*. In Actes des 14^{èmes} journées d'Informatique musicale – JIM-09, Grenoble, 2009.
- [13] Trueman D. and DuBois R. L., *PeRColate — A Collection of Synthesis, Signal Processing, and Image Processing Objects for Max, MSP, and Nato*, online: <http://www.music.columbia.edu/PeRColate/>
- [14] Ziat, M.: *Zoomable Interfaces*, in *Enaction and Enactive Interfaces, a Handbook of Terms*, edited by A. Luciani and C. Cadoz, Enactive System Books, 2007 – pp164.