

# LOL

## UN ENVIRONNEMENT EXPERIMENTAL DE COMPOSITION

### CHOREGRAPHIQUE

(ce texte est la version auteur du texte publié dans la revue Ec/arts  
www.ecarts.org, vol 2:pratiques artistiques et technologiques, oct. 2001)

#### Chorégraphie, musique et logique

J'ai toujours fait en sorte d'éviter le lien de parenté entre musique et danse en préférant me consacrer à la musique dans ses aspects les plus spécifiques et les plus formels. Ce, tant dans mes recherches en ethnomusicologie au CNRS en Afrique et en Indonésie que dans mon travail avec les compositeurs dans les studios de l'IRCAM. J'emprunterai donc la réponse à Wittgenstein : « Dis moi *comment* tu cherches et je te dirais *ce* que tu cherches ».

Le *comment*, dans la musique contemporaine occidentale, hérite d'une longue tradition où l'écriture joue un rôle fondamental : non seulement celle-ci permet de conserver et de transmettre le produit d'une pensée, mais elle permet aussi l'expérimentation de la pensée par anticipation. La partition, avant d'être jouée, est écrite, suspendue par endroits, corrigée, voire effacée, réécrite pendant le temps de son écriture. Elle représente le résultat d'un processus ou de choix successifs de processus, exprimé par un système de symboles arbitraires. Le choix des symboles est arbitraire dans la mesure où chacun représente, *par*

*convention*, une procédure déterminée, un geste par exemple. Ainsi la partition musicale (ou chorégraphique) s'apparente à un programme pouvant être exécuté soit par des musiciens (des danseurs), soit par un automate mécanique ou informatique, mais ne représente qu'un aspect de l'écriture.

Je distinguerais donc un autre aspect, doté d'une fonction exploratoire et d'une fonction générative, plus à même de rendre compte du travail du compositeur ou du chorégraphe, à savoir l'écriture des processus eux-mêmes et des concepts impliqués. On retrouve initialement ces préoccupations en logique contemporaine, aboutissant d'une part à la nécessité d'une langue artificielle permettant l'écriture des concepts (cf. notamment la *Begriffsschrift* de Frege 1879) et, d'autre part, aux notions de « machine universelle » — prototype de l'ordinateur — (Turing 1936), d'automates (Von Neumann 1951) et de programmes informatiques. Dans la mesure où, pour chaque concept, il existe une procédure permettant de le vérifier par le biais d'une machine universelle — ce qui a été démontré —, il semblerait que toute forme de pensée susceptible d'être exprimée dans un langage peut, en théorie, l'être par une machine universelle et en l'occurrence par un ordinateur.

A ce titre, les programmes d'informatique musicale de CAO (composition assistée par ordinateur) s'apparentent souvent à des langages de programmation dits de haut niveau faisant appel à des « routines » de plus bas niveau. Je distinguerais deux types de programmes de composition :

d'une part, ceux pourvus d'une syntaxe spécialisée et d'objets (ou fonctions) orientés vers des concepts définis par une analyse experte. Ce type de programmes, s'ils ont une dimension exploratoire et générative, ne permet pas de créer, sinon par détournement de l'écriture, de nouveaux concepts et procédures. Il s'agit en quelque sorte de langages créés sur mesure et adaptés à un univers cognitif précis : celui de l'auteur (ou co-auteurs) du programme. Leur application la plus immédiate est soit représentationnelle, soit descriptive et orientée vers une notation à fin d'exécution (par un interprète, un programme, une machine).

Il existe d'autre part des programmes « ouverts » sans univers cognitif particulier, ni application exécutive immédiate, mais permettant à chacun de définir et expérimenter ses propres concepts, fonctions et objets dans un formalisme logique le plus général. De tels programmes se proposent donc plutôt comme des « environnements », sortes d'outils méta, proposant par défaut des fonctions et objets nécessaires à la construction d'univers aussi variés que possible. *Patchwork* et *OpenMusic*, développés à l'Ircam, (cf. Gérard Assayag, *Eccart* n°1) sont autant de tentatives réussies dans cette voie. Etant donné le niveau d'abstraction de ces programmes, il n'est pas étonnant que l'on puisse — en partie — y définir et expérimenter des concepts autres que musicaux, par exemple chorégraphiques. C'est ainsi que Myriam Gourfink devait expérimenter, il y a plus d'un an, ses premières investigations chorégraphiques avec *Patchwork* : il suffisait pour commencer — si l'on peut dire — de recourir

à ses fonctions d'abstraction et à la dimension arbitraire du symbole qui, finalement, signifie ce que l'on veut.

### Principes généraux

Cette première expérience initiée par Myriam Gourfink sur un programme initialement conçu pour des compositeurs de musique nous mit devant l'évidence qu'il fallait développer un environnement plus adapté à la chorégraphie sans pour autant se limiter à une esthétique précise. Cependant, en regard de la musicologie, la chorégraphie dispose de peu de théories générales. Il fallait commencer par définir, pour celle-ci, avec l'aide de Kasper T. Tøeplitz, ce que l'histoire avait retenu des représentations, notations, concepts et procédés fondamentaux utilisés en musique. C'est ce que nous avons nommé LOL, qui est à la fois un programme mais aussi une expérimentation grandeur nature sur le langage chorégraphique. Nous étions donc d'accord sur un point : l'urgence était non pas dans la représentation du geste chorégraphique, mais dans *la représentation des connaissances* utiles à la composition chorégraphique en général.

Un principe fondamental selon lequel je souhaitais programmer LOL repose sur la notion de catégorie, en prenant le parti selon lequel toute représentation cognitive est essentiellement catégorielle (ce qui était une évidence lorsque je travaillais avec des musiciens d'autres cultures). Les nombres (entiers, réels, etc.) autant que les couleurs, hauteurs musicales, gestes, et autres concepts sont des catégories qui prennent une valeur fonctionnelle (un

« sens ») selon qu'elles s'opposent, ou non, à d'autres catégories. La notion de catégorie pose directement la question de l'égalité ou de l'équivalence, laquelle peut être résolue par une catégorie d'ordre supérieur. Par exemple, si je considère les catégories *vert* et *rouge*, que l'on se représente bien comme deux concepts différents, elles sont identiques dès lors que je considère une troisième catégorie *couleur* dont *vert* et *rouge* seraient elles-mêmes des instances. Ainsi l'équivalence entre deux entités abstraites peut toujours être ré-évaluée — calculée — tant qu'il existe une troisième catégorie qui les englobe. Cela nous renvoie à la fois à la théorie des ensembles, mais surtout au calcul des prédicats, où attribut et fonction sont indissociables (Frege 1884, Russell 1962), et à la théorie des types. En quelque sorte, chaque catégorie est un symbole vide, ne signifiant rien directement (cf. à ce sujet Dan Sperber 1974) mais seulement *par opposition* avec d'autres symboles au sein d'une hiérarchie d'équivalences, lesquelles sont également des catégories d'un ordre supérieur.

Ce principe implique, *in fine*, que chaque objet quel qu'il soit est sa propre catégorie : le nombre 1 appartient à la catégorie des « 1 » et diffère du nombre 2 qui appartient à la catégorie des « 2 ». Ainsi on peut dire que 1 est différent de 2 sans recourir à une démonstration : il s'agit d'une définition, c'est-à-dire une représentation cognitive. A l'inverse, dans un système cognitif particulier, on peut créer une catégorie des *Iou2* selon laquelle  $1 = 2$ , le signe « = » représentant l'appartenance à la catégorie *Iou2* (ainsi une fonction est bien un attribut, un concept). La notion d'équivalence est donc purement conventionnelle; elle est fondée sur une définition d'ordre cognitif. Aussi peut-

on définir la notion de pertinence, à savoir si l'opposition possède un sens (au-delà de la contradiction) : 1 et 2 peuvent être comparés d'un certain point de vue s'ils ont à un concept commun — celui du point de vue — ou si l'un provient de l'autre au moyen d'une procédure existante (une fonction, soit un autre concept).

S'il paraît irréaliste de devoir définir chacune des catégories cognitives avant le travail de composition, il semble essentiel de permettre cette ouverture : différentes esthétiques sont le produit d'univers cognitifs très différents (parfois incompatibles) pouvant ignorer ou aller à l'encontre des théories scientifiques (que je considère, pour ma part, comme autant d'univers cognitifs particuliers reposant essentiellement sur des définitions, croyances, évidence ou sur le « bon sens », ce qui n'enlève rien à leur cohérence).

Ainsi devons-nous proposer un certain nombre de catégories fonctionnelles cohérentes permettant de commencer la composition chorégraphique. Plus générale parce que s'inscrivant dans une étude du mouvement, plus diffusée, plus éloignée d'une esthétique particulière, la notation Laban nous sert de base conceptuelle. Nous en avons extrait certains fondements, tous relatifs au corps, la notion d'espace étant exprimée par rapport à celui-ci.

A l'heure actuelle, nous avons retenu les notions catégorielles telles que le *corps*, les *parties du corps*, l'*état* (correspondant à un état du corps et de ses parties), la *séquence* (une succession d'états).

Le corps est défini comme une entité abstraite dont l'une des fonctions est d'identifier les seules parties nécessaires à l'investigation chorégraphique. Un corps peut donc n'être constitué que de deux jambes par exemple si un processus envisagé n'implique que ces parties du corps. On peut aussi considérer un corps comme un ensemble de corps qui viennent soit se compléter, soit se succéder dans une séquence (de la même façon qu'en musique, on ne considère pas dans un calcul les notes d'un accord qui ne sont pas jouées, ou encore qu'un « même » accord peut se présenter sous différents aspects). L'instance d'un corps est nommée en même temps qu'elle hérite, le cas échéant, des noms des autres corps, états et séquences dont elle dépend. Ainsi un corps peut être identifié à travers un état, une séquence ou un autre corps. Un corps, une fois défini, peut rester une entité abstraite et servir de corps « générique » (une nouvelle catégorie) susceptible par la suite d'instanciations différenciées.

De même, les parties du corps ne sont pas prédéfinies. A chaque instant, on peut définir une nouvelle partie du corps, la nommer, voire la subdiviser en d'autres parties. Chaque partie du corps peut donc être une partie de partie-du-corps et ainsi de suite. Chaque instance d'une partie du corps hérite par défaut des propriétés (modifiables) des parties du corps ou du corps dont elle est issue.

Neuf dimensions catégorielles permettent de décrire ensuite n'importe quel mouvement du corps : les appuis, l'élévation, la direction, la flexion, la rotation, la distance, l'enroulement, les contacts et les adresses. Puisque les valeurs sont elles mêmes catégorielles, chacune des dimensions peut être

discrétisée avec une précision à définir par le chorégraphe : intensité des appuis, degrés d'ouverture, de flexion peuvent être exprimés soit par des valeurs numériques, soit par des symboles. Un geste, un état ou une séquence constituent donc une proposition symbolique permettant (ou résultant de) un calcul propositionnel (pour le passage, le cas échéant, du symbolique au numérique, on fera appel à la théorie des sous-ensembles flous).

Mais déjà d'autres notions, plus particulières aux chorégraphies de Myriam Gourfink, semblent indispensables. La *direction de la pensée*, notamment, vient compléter les concepts relatifs au corps. Comme le remarquait Kasper Tøeplitz, permettre l'écriture de ce concept est essentiel puisqu'elle permet de ne pas la restreindre à une fonction purement procédurielle, exécutive. C'est par ailleurs ce type de considération qui devait amener à une crise de la notation musicale, laquelle n'est pas véritablement dépassée.

### Programmation

Le langage de programmation utilisé est le *Lisp*. Initialement créé pour les besoins de l'intelligence artificielle, ce langage donne facilement accès à un niveau méta dans la mesure où un programme, en Lisp, peut aisément s'écrire lui-même de façon dynamique. Sa syntaxe, fondée uniquement sur des expressions symboliques, se prête facilement à la représentation et à la manipulation de connaissances : une donnée, une déclaration, une fonction ou un programme est, en Lisp, une expression symbolique. Ainsi une variable, nécessaire à une fonction, peut elle-même être une fonction, tout comme une

fonction peut également produire une nouvelle fonction. Le lien avec le calcul propositionnel tel que définis en logique contemporaine est donc immédiat.

Le recours au Common Lisp Object System, extension orientée objet de Lisp, permet encore de faciliter le traitement des connaissances. Ainsi chaque nouveau concept sera représenté soit :

- par un nouvel objet abstrait (une classe), créé dynamiquement par l'utilisateur, pouvant hériter directement des propriétés d'autres objets précédemment définis;
- par une fonction (méthode), elle-même nommée et définie par l'utilisateur, admettant pour argument des objets (classes). En fait, en CLOS, une fonction est également un objet à part entière, une classe.

Selon le principe énoncé d'une représentation des connaissances purement catégorielle, la notion de variable peut disparaître au profit d'instances de classes différenciées : chaque nouveau symbole (autrement dit : nom) créé ou appelé par l'utilisateur est l'instance d'une classe prédéfinie. Le principe consiste donc à nommer tout objet en lui assignant, *en même temps*, une idée de ce qu'il est, c'est-à-dire sa place dans l'espace conceptuel invoqué (on imagine alors la possibilité de créer une classe *variable*, sorte d'ensemble fourre-tout qui permettrait de mémoriser temporairement un objet dont on ne sait ce qu'il est).

Le moyen privilégié d'effectuer ce type d'opération est le langage naturel : il s'agit d'énoncer simplement le nom de l'objet. Dans le dialogue avec la machine, lorsque l'appartenance catégorielle n'est pas préalablement définie

(le programme n'a pas de représentation « cognitive » du symbole), plutôt que de signaler une erreur, le programme peut au contraire demander ce qu'*est* (la valeur attributive) véritablement ce nouveau nom. On peut en effet inférer que l'assertion résultant du dialogue (« ceci est un x ») définit une nouvelle catégorie, et le programme peut créer ce nouveau symbole (« ceci ») en même qu'il crée la nouvelle classe désignée (« x »). Ainsi la question posée par le programme (« qu'est ce que ceci ? ») dénote son ignorance en même temps que son intention de connaître la catégorie invoquée, puis répond à l'intention implicite de lui faire connaître cette catégorie (« il y avait ceci»). Par la suite, une nouvelle assertion portant sur cette nouvelle classe permettra de la définir par rapport à une ou plusieurs classes préexistantes (« x est une couleur »). L'espace des classes représente donc un espace catégoriel plus ou moins hiérarchisé, projection de l'espace cognitif qui le crée, où chaque classe définit soit un prédicat, soit une action (foncteur).

L'accès le plus immédiat à LOL est donc le langage. On considère en effet que la structure syntaxique et le vocabulaire utilisé dénotent soit une représentation cognitive, soit une intention (Searle 1969) impliquant une action du programme (créer ou instancier une classe). Si un processus d'inférence est nécessaire, je ne pense pas qu'il doive être trop étendu : en effet, le but d'un tel programme est moins d'inférer de nouvelles connaissances que d'explicitier et d'organiser des connaissances qui seront utilisées dans un contexte dévoilé pas à pas par l'utilisateur. Cette explicitation est nécessaire au programme autant qu'à l'utilisateur pour expérimenter son univers conceptuel. D'une certaine façon, ce

type d'interface est une sorte d'ethnographe automatique posant des questions lui permettant de se représenter un nouvel univers. Une mauvaise question (et souvent l'ethnologue en pose !) sera donc une question non pertinente : soit il n'existe pas de méthode pour y répondre dans le contexte donné, soit l'objet évoqué n'existe pas ou ne possède pas la propriété requise. Rien n'empêche dès lors d'ajouter un nouvel objet, méthode ou propriété requis et de le nommer (ou à celui-qui-répond d'inventer une nouvelle histoire...). Cependant, d'un point épistémologique, la création de nouveaux concepts *ad hoc* n'est pas sans risque : la consistance d'un système (d'une théorie, d'une esthétique) est en effet inversement proportionnelle au nombre de concepts « ajoutés ». Mais ceux-ci sont indispensables et on imagine difficilement comment un système ne permettant aucune étrangeté serait susceptible de nouveauté.

Je tiens à remercier Myriam d'avoir suscité cette aventure et de déjà l'expérimenter.

Frédéric Voisin,

Paris le 27 avril 2000

### Bibliographie

Assayag, Gérard : «Du calcul secret au calcul visuel », *Interfaces homme-machine et création musicale*, Hermès, Paris, 1999.

- Arom S. & Voisin F. : « Theory and Technology in African Music », *The Garland Encyclopedia of African Music*, vol. 2, R. Stone (éd.), New-York, Garland.
- Frege, Gottlob : — *Idéographie*, Vrin, Paris, 1999, traduction de *Begriffsschrift*, Nebert, Halle, 1879.  
— *Les fondements de l'arithmétique*, Seuil, Paris, 1969, traduction de *Grundlagen der Arithmetik*, Marcus, Breslau, 1884.
- Feyerabend, Paul : *Against Method*, New Left Books, Londres, 1975.
- Von Neumann, John : « The General and Logical Theory of Automata », *Cerebral Mechanism in Behavior*, Lloyd A. Jeffries éd., Wiley and Sons, New-York, 1951.
- Popper, Karl : *The Logic of Scientific Discovery*, Hutchinson Co, Londres, 1959.
- Russel, Bertrand : *An Inquiry into Meaning and Truth*, Allen & Unwin, Baltimore, Penguin, 1962.
- Searle, John : *Speech Acts*, Cambridge University Press, Londres, 1969.
- Sperber, Dan : *Le symbolisme en général*, Hermann, Paris, 1974.
- Turing, Alan : « On Computable Numbers, with an Application to the Entscheidungsproblem », *Proceeding of the Mathematical Society*, II, vol. 42, 1936.
- Voisin, Frédéric : «Musical Scales in Central Africa and Java : modelling by synthesis», *Leonardo Music Journal*, vol. 6, Pergamon, New-York, MIT Press, 1995.
- Wittgenstein, Ludwig : *Remarques philosophiques*, Gallimard, Paris, 1975, traduit de : *Philosophische Bemerkungen*, B. Blackwell, Oxford, 1964.

