



**HAL**  
open science

## **In Vitro**

Mikhail Malt

► **To cite this version:**

| Mikhail Malt. In Vitro. Journées d'Informatique Musicale, Jun 2001, Bourges, France. hal-02987405

**HAL Id: hal-02987405**

**<https://hal.science/hal-02987405>**

Submitted on 3 Nov 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## In Vitro

### Implémentation d'agents autonomes en MAX/MSP

Mikhail Malt  
*mmalt@ircam.fr*

**Résumé :** Nous exposerons dans cet article l'implémentation d'un modèle d'agents autonomes dans l'environnement MAX/MSP<sup>1</sup>. Cette implémentation faite dans le cadre du développement d'une installation temps réel, le projet « Traces », a par la suite donné lieu à une réflexion sur l'utilisation d'une certaine catégorie de modèles en composition musicale.

#### 1. Introduction

Un des problèmes qui se posent lors de la conception d'installations temps réel, est le choix du, ou des modèles génératifs. La majeure partie des modèles utilisés de nos jours reposent sur l'utilisation de l'aléatoire comme simulation du choix.

Dans notre cas précis, nous voulions trouver un modèle qui puisse être plus qu'une simple simulation des choix possibles d'un compositeur, qui puisse être un modèle de contrôle du matériau musical, en nous rapprochant le plus possible d'un modèle génératif simulant une écriture musicale.

L'utilisation d'un modèle basée sur des agents autonomes nous semblait être une solution intéressante, puisqu'il propose une solution fondée sur l'évolution de petites entités musicales, qui interagissent entre elles de manière à dégager une structure complexe.

L'unique modèle d'agent disponible pour une application musicale était « boids » de Craig Reynolds, « implémenté » dans MAX par E. Singer. Cependant, comme le modèle avait une orientation graphique explicite, il ne permettait pas son extension à d'autres utilisations, ce qui nous a amené à développer la présente application.

#### 2. Les agents autonomes

Depuis l'apparition de ce concept, au milieu des années soixante-dix dans le champ de l'intelligence artificielle distribuée, la dénomination d'« agent » est utilisée pour désigner soit des entités informatiques logiciels soit des entités informatiques matérielles (comme les robots). De ce fait il n'existe pas, de nos jours, un consensus général sur la définition « d'agent ».

Pour les besoins de notre travail, nous avons utilisé la définition donnée par M. Wooldridge:

« Un agent est un système informatique, situé dans un environnement et qui est capable d'actions autonomes dans cet environnement de manière à atteindre ses objectifs de conception » [Weiss 2000, 29].

<sup>1</sup> © Cycling74&Ircam, <http://www.cycling74.com>.



Nous ajouterons que « autonomie » signifie que l'agent doit être capable d'agir, réagir et interagir dans son environnement sans l'intervention d'autres systèmes (humains ou informatiques). Ceci nous amène au fait qu'un agent est une entité possédant : un état interne (dynamique) et des règles de comportement.

### 3. Pourquoi des « agents » ?

L'intérêt que peut avoir l'utilisation d'un modèle d'agents dans le contrôle d'évènements musicaux vient du fait qu'il fait partie des « modèles orientés individus » [Reynolds 2001]. Ces modèles sont des simulations basées sur les conséquences globales d'interactions locales entre les membres d'une population. L'action de contrôle se fait au niveau de l'individu, c'est-à-dire des règles qui commandent son comportement. Ceci revêt une importance majeure si on confronte ce type de modèle aux modèles statistiques. Bien que ces deux types de modèles aient comme vocation la génération d'informations de contrôle pour des situations globales, dans les modèles statistiques, le contrôle se fait sur des caractéristiques moyennes de la population, tandis que pour les « modèles orientés individus », le contrôle se fait plutôt au niveau des règles de comportement de l'individu.

Parmi les « modèles orientés individus », il existe aussi le modèle d'automate cellulaire, qui a été utilisé par plusieurs compositeurs [Miranda 1994]. Cependant, ce modèle n'étant pas capable de représenter des données complexes (ou multidimensionnelles), ni de représenter une « diversité » d'individus, puisque toutes les « cellules » sont semblables, n'était pas adapté à nos besoins. En revanche, le modèle d'« agent » permet une gestion d'espaces « n » dimensionnels et la représentation d'une grande diversité d'individus. De ce point de vue, ce modèle laissait entrevoir la possibilité de simuler d'une manière satisfaisante l'évolution d'« entités » musicales avec l'ensemble des paramètres nécessaires à leur représentation.

### 4. L'implémentation

#### 4.1 Le cadre artistique

Cette implémentation fait partie d'un projet de concert/installation/performance appelé « Traces », avec la collaboration du contrebassiste français Jean Pierre Robert. Ce projet qui se veut aussi une performance/reflexion humaine/numérique sur le concept de vie, se divise en deux parties : la première, « In vivo » (la performance) et la deuxième, « In vitro » (l'installation). La phase « In vitro » propose, en temps réel et sans intervention humaine, l'évolution musicale du matériau généré par l'instrumentiste et la machine dans la première phase « in vivo », de la même manière qu'on cultive des microorganismes dans une « boîte de Petri<sup>2</sup> ».

Chaque être musical naîtra, vivra, interagira avec son environnement, sera influencé par lui, aura un nom (une généalogie se construira), se reproduira et mourra.

La surface musicale générée, devra avoir comme but la formalisation de certains aspects de notre écriture musicale. Cette écriture se fonde sur l'utilisation de petits gestes musicaux qu'évoluent dans le temps [Malt 1996].

<sup>2</sup> Petite boîte stérilisée qui permet de cultiver et d'observer facilement les bactéries.



## 4.2 Le cadre technique

L'implémentation de ce modèle s'est faite dans l'environnement MAX/MSP. L'environnement des agents est une fenêtre MAX (figure 1), et chaque agent est un « patcher » chargé dynamiquement (figure 2).

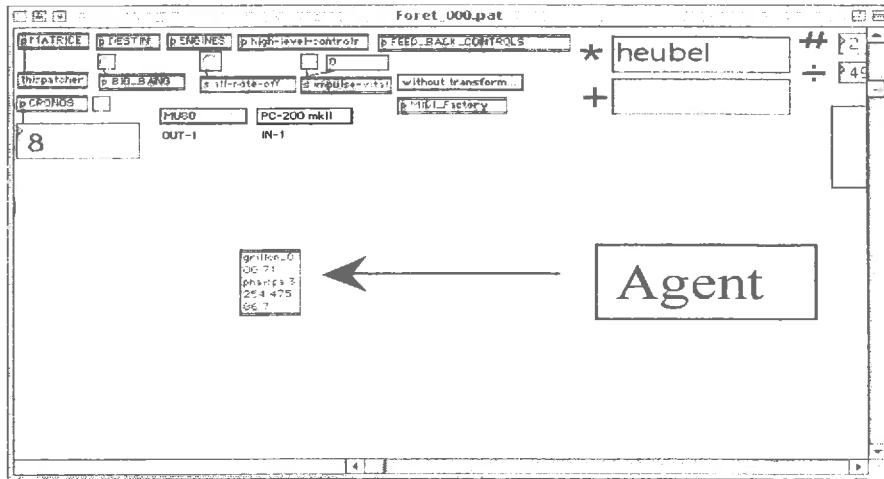


figure 1 : L'environnement

A sa naissance, chaque « individu » possède :

- Un nom, pour permettre entre autres son identification et de nommer les diverses variables pour l'envoi et la réception de messages « personnels ».
- Une position spatiale fixe, pour permettre de fixer des repères virtuels.
- Une note pivot, hauteur musicale pour déterminer le centre à partir duquel certains gestes seront joués.
- Un rayon d'écoute, pour restreindre l'appareil « cognitif » de chaque agent.
- Un âge maximal de vie, génétiquement déterminé ; au fur et à mesure qu'un agent vieillit, la probabilité de mort augmente.
- Un rôle, (un geste musical). Actuellement, il existe 10 rôles<sup>3</sup>, chaque rôle étant associé à un geste musical (l'événement simple, la répétition, une « proto-mélodie », éléments de synthèse, échantillons, etc.).
- Un mode de comportement. Chaque agent est capable de connaître les agents qui sont à l'intérieur de son rayon d' « écoute », et de modifier certaines caractéristiques de son comportement et de son rôle en fonction de ses voisins. Par exemple, dans un mode de comportement « sociable », chaque agent tend à faire évoluer son geste musical vers la moyenne des « notes pivot » de ses voisins. Par contre, un mode de comportement « anti-social » fera évoluer l'agent à l'opposé de ses congénères.
- La possibilité de se reproduire. Le deuxième tiers de la vie maximale prévue, est la « période fertile », à ce moment chaque agent peut se reproduire en transmettant une partie de son nom, son l'âge maximale, son rôle et son comportement.

Chaque agent a un temps de vie et son action dans l'environnement est toujours transitoire. Chaque agent peut aussi se reproduire de manière à transférer une partie de son patrimoine génétique (son nom, son rôle, son matériau, son comportement et sa position

<sup>3</sup> Les rôles actuels sont en train d'évoluer de manière qu'il est possible que ce nombre change prochainement.



spatiale) à sa progéniture. Le résultat final de l'interaction entre les divers agents est la construction d'une « surface musicale ».

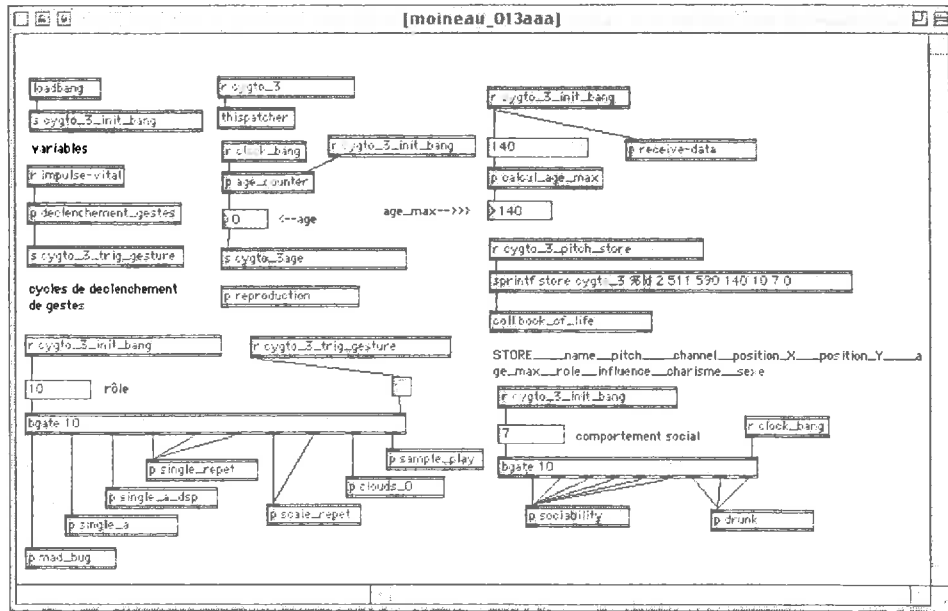


figure 2 : L'agent

Les « individus » sont générés de deux manières. La première est appelée « God\_generated » et la deuxième est une reproduction asexuée, « Self\_generated ». Le mode « God\_generated » permet d'initialiser le système, de contrôler le nombre d'agents (évitant la disparition de la « société » créée, ou une surpopulation), et également d'introduire de nouveaux éléments (gènes) en favorisant la variété. Le mode « Self\_generated » permet la reproduction des agents de manière à perpétuer des noms, des rôles, des modes de comportement et des matériaux génétiques divers (MIDI et audio).

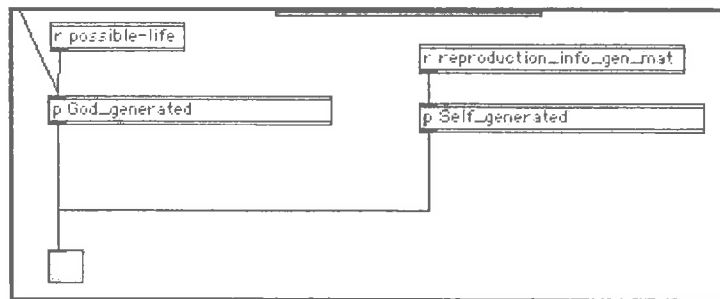


figure 3 : Les deux modes de génération

Ces deux modes de génération se fondent, d'un point de vue technique, sur la possibilité de l'environnement MAX de « charger » dynamiquement des « patchers<sup>4</sup> », à partir du message « newex » (voir figures 4, 5 et 6). Ce message prend comme paramètres la « commande » « newex », les positions relatives en pixels du patcher (X=511 et Y=590), des paramètres relatifs à la taille du patcher (largeur = 49 et taille de police, codée comme 196617 pour « geneva » 9), le nom du patcher (moineau\_013) et la liste de paramètres que nous voulons

<sup>4</sup> Un « patcher » est une « encapsulation » d'un processus dans un objet graphique telle que





associer à l'agent (note pivot, nom, canal MIDI, positions X et Y, âge maximal, index de rôle et index de comportement). Le « patcher » généré peut être vu figure 6.

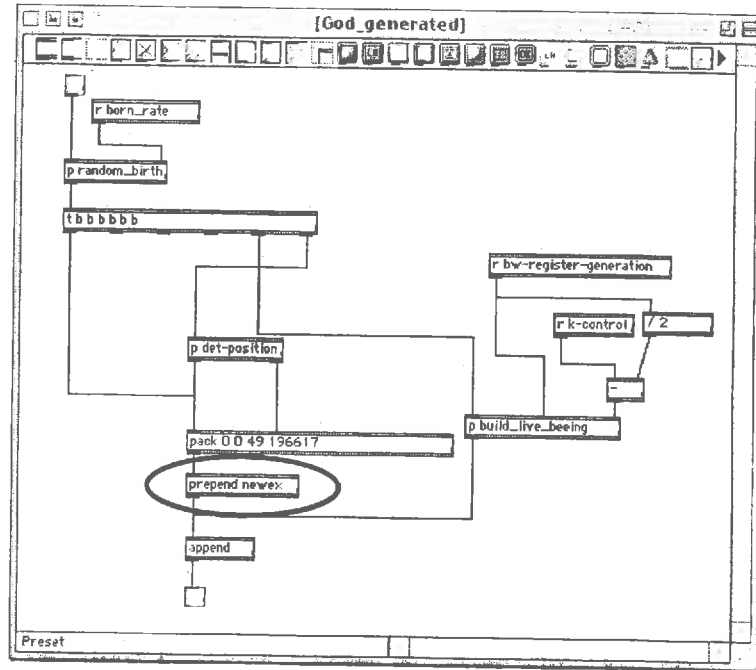


figure 4 : Le message « newex »

newex 511 590 49 196617 moineau\_013 75 cygto\_3 2 511 590 140 10 7 0 0

figure 5 : Le format de données pour « newex »

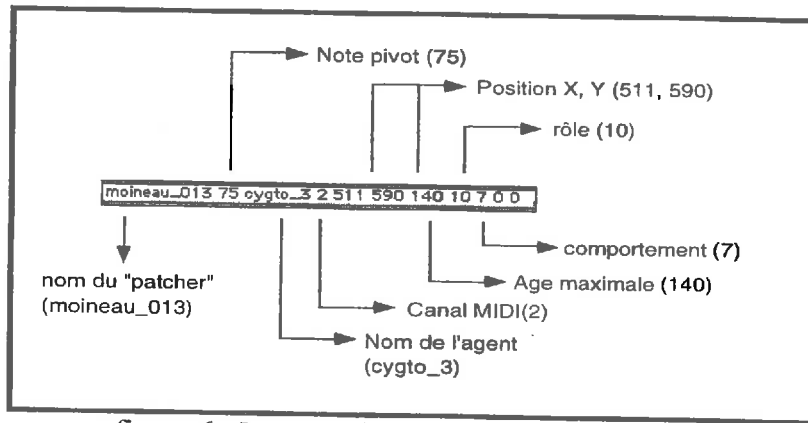


figure 6 : Le « patcher » généré par « newex »



## 5. Conclusions

Une des caractéristiques des modèles d'agents est leur pragmatisme implicite, c'est-à-dire, qu'ils sont très complexes<sup>5</sup> qu'il n'est possible d'appivoiser que par la simulation numérique. Dans un travail où le matériau de base est un ensemble de processus, contrôlés par des modèles mathématiques et d'objets ayant presque une vie propre, l'ordinateur se montre être l'outil approprié pour gérer le grand nombre de variables et de relations.

D'après les expériences que nous avons menées, nous pouvons émettre l'hypothèse qu'une « surface musicale », pourra être vue comme un système auquel correspondra une dynamique instable, mue par une multiplicité de forces en interaction. La composition sera vue comme un processus en permanent mouvement, une recherche permanente de « sens » entre les divers niveaux de l'espace musical considéré, avec des moments de stabilisation, des moments de déstabilisation et principalement des phénomènes d'émergence. La notion d'émergence, qui est fondamentale dans ces expériences, exprime l'apparition d'un sens nouveau lors de l'agrégation d'éléments au sein d'un contexte. Ce sens nouveau est explicitement absent des éléments individuels. Il est une nouvelle caractéristique qui résulte de l'interaction entre ces éléments. C'est finalement penser la composition comme étant le produit d'un tissu de relations avec elle même et avec le monde qui l'entourne. Relations aussi vivantes qui évoluent et se transforment pendant l'évolution de l'œuvre.

Le développement de ce travail se poursuit actuellement avec la recherche de représentations possibles pour les règles de comportement, l'implémentation de la reproduction sexuée, la représentation et le codage de divers matériaux musicaux (MIDI et audio) pour étudier les possibilités de transmission génétiques et la recherche des liens entre les structures de contrôles générées et le sens musical produit.

---

<sup>5</sup> La Complexité de laquelle nous parlons n'a aucun lien avec le "compliqué", mais plutôt avec son sens étymologique: *complexus*, ce qui est tissé ensemble. Des constituants divers associés, tissés ensemble pour produire une nouvelle unité.



## Références

- [Langton 1996] Christopher C. Langton, *Artificial Life, an overview*, Christopher C. Langton editor, MIT Press, 1996.
- [Malt 1996] M.MALT, « Lambda3.99 (Chaos et Composition Musicale) », in *Troisièmes Journées d'Informatique Musicale JIM 96*, Ile de Tatihou, Normandie, France, 1996.
- [Miranda 1994] E. R. Miranda, « Music composition using cellular automata », *Languages of Design*, Vol. 2, pp. 105-117, USA.
- [Miranda 2000] E. R. Miranda (Ed.), *Readings in Music and Artificial Intelligence*, Contemporary Music Series Vol. 20, Harwood Academic Publishers, Amsterdam, 2000.
- [Reynolds 1987] Craig W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model" in *Computer Graphics* 21(4) (SIGGRAPH 87 Conference Proceedings) pages 25-34. <http://www.red.com/cwr/boids.html>
- [Reynolds 1999] Craig Reynolds, "Steering Behaviors For Autonomous Characters" in *Conference Proceedings of the 1999 Game Developers Conference*, pages 763-782. <http://www.red.com/cwr/steer/>
- [Reynolds 2001] Craig Reynolds, « Individual-Based Models, an annotated list of links », <http://www.red3d.com/cwr/ibm.html>.
- [Weiss 2000] Gerhard Weiss, *Multiagent Systems, a modern approach to distributed artificial intelligence*. Edited by Gerhard Weiss, MIT Press, 2000.



