

Systèmes multi-agents et phénomènes d'émergence : donner forme par actions et interactions

Florent Di Bartolo

DANS **SCIENCES DU DESIGN 2016/2 n° 4** , PAGES 63 À 71

ÉDITIONS **PRESSES UNIVERSITAIRES DE FRANCE**

ISSN 2428-3711

ISBN 9782130734475

DOI 10.3917/sdd.004.0063

Date de mise en ligne : 07/12/2016

Article disponible en ligne à l'adresse

<https://shs.cairn.info/revue-sciences-du-design-2016-2-page-63?lang=fr>



Découvrir le sommaire de ce numéro, suivre la revue par email, s'abonner...
Scannez ce QR Code pour accéder à la page de ce numéro sur Cairn.info.



Distribution électronique Cairn.info pour Presses Universitaires de France.

Vous avez l'autorisation de reproduire cet article dans les limites des conditions d'utilisation de Cairn.info ou, le cas échéant, des conditions générales de la licence souscrite par votre établissement. Détails et conditions sur cairn.info/copyright.

Sauf dispositions légales contraires, les usages numériques à des fins pédagogiques des présentes ressources sont soumises à l'autorisation de l'Éditeur ou, le cas échéant, de l'organisme de gestion collective habilité à cet effet. Il en est ainsi notamment en France avec le CFC qui est l'organisme agréé en la matière.

Systemes multi-agents et phenomenes d'emergence : donner forme par actions et interactions

Florent Di Bartolo

Maître de Conférences
à l'Université Paris-Est Marne-la-Vallée, France
Équipe CCAMAN, Laboratoire LISAA (EA4120)
florent.dibartolo@u-pem.fr

Presses Universitaires de France | Téléchargé le 04/06/2026 sur <https://shs.cairn.info> (IP: 216.73.217.142)

Mots-clés

Art numérique
Système multi-agents
Agent intelligent
Émergence
Programmation

Keywords

Digital art
Multi-agent system
Smart agent
Emergence
Programming

Résumé

La programmation est une activité à laquelle artistes et designers se trouvent aujourd'hui confrontés dans l'exercice de leur fonction. L'article suivant entend préciser les apports qu'une certaine pratique de la programmation est susceptible de leur apporter dans la définition de systèmes interactifs et la création de formes complexes. La programmation est abordée à travers la construction de systèmes multi-agents (SMA). Les propriétés des SMA sont analysées dans le but de définir une certaine façon de programmer qui ouvre sur des structures et des dynamiques vécues par leurs observateurs comme des émergences perceptives. Le concept d'émergence est redéfini relativement au point de vue de l'observateur à travers l'expérience qu'il est possible de faire d'installations interactives et de simulations multi-agents élaborées dans le champ des arts visuels.

Abstract

Programming as an activity has reached artists and designers in their daily life. This article aims to show how programming can help them to design interactive systems and generate complex structures. The design of multi-agent systems (MAS) made by media artists will be taken into consideration. Their attributes will be analyzed in order to define a certain way of programming that allows to create systems with emergent dynamics and behaviors. The concept of epistemic emergence will be used in order to outline the novelty that can emerge through evolutionary computer simulations in digital art.

Introduction

La programmation est une activité à laquelle de plus en plus d'artistes et de designers se trouvent confrontés depuis une quinzaine d'années dans leur pratique des médias numériques. Elle n'appartient plus exclusivement aux informaticiens. Elle s'est étendue à de nouveaux champs disciplinaires et profils d'utilisateurs notamment grâce à l'apparition de langages de programmation graphique⁰¹.

La programmation possède une dimension sociale que Pierre Lévy a su mettre en évidence en présentant l'écriture d'un logiciel comme le résultat des interactions entre un très grand nombre d'acteurs humains, techniques et sémiotiques. Pour cet auteur, les programmeurs figurent parmi les « architectes de l'écologie cognitive contemporaine » (1992). En effet, la programmation sert à définir les formes d'accès qui sont accordées aux données de notre époque. Les programmeurs modulent notre accès au savoir.

Il est possible de faire remonter l'usage de la programmation dans le champ des arts aux années 1960, à l'époque des premières recherches en art visuel (Lartigaud, 2011). Ces travaux ne seront pas ici évoqués. Cet article fait uniquement référence à des œuvres qui ont été réalisées à partir des années 90 à une époque où les premiers systèmes multi-agents (SMA) définis par des artistes vont être conçus, suite à l'apparition d'un nouveau champ de recherche interdisciplinaire alliant informatique et biologie : la vie artificielle, consacrée à la modélisation et à la simulation du vivant.

01.

La programmation est une activité qui peut se pratiquer à l'aide de logiciels appartenant à deux catégories différentes : les logiciels de programmation graphique comme Max/MSP, Usine Hollyhock, vvvv et les environnements de développement comme Visual Studio ou Éclipse. Les environnements de programmation invitent leurs usagers à définir un programme à partir de l'écriture de lignes de code tandis qu'un logiciel de programmation graphique propose de concevoir un logiciel en connectant des objets graphiques. Une nette distinction entre ces deux catégories de logiciels n'est toutefois pas à établir, car les interfaces des logiciels de programmation graphique permettent aussi de définir des algorithmes à partir de l'écriture de lignes de code et offrent à leurs usagers la possibilité de faire appel aux mêmes structures de contrôle.

Les propriétés de plusieurs SMA vont être ci-dessous analysées afin de définir une façon de programmer qui ouvre sur des structures et des dynamiques complexes. Bien que ces systèmes aient été élaborés par des artistes, il nous semble que leur examen est susceptible d'éclairer et d'orienter des pratiques qui relèvent du design.

La capacité des SMA à explorer voire à définir des champs de possibles à la recherche de formes, de structures, et de dynamiques nouvelles, représente un enjeu non seulement pour les artistes, mais aussi pour les designers.

Une attention particulière sera accordée aux intentions des artistes qui élaborent des SMA et insistent sur les propriétés émergentes de leurs systèmes afin d'interroger le caractère déterministe de la programmation et des processus auxquels elle donne une existence effective.

1. — Les systèmes multi-agents

Les premiers SMA élaborés dans le champ des arts sont apparus au cours des années 90. Ils ont été conçus par des artistes comme Karl Sims, Christa Sommerer et Laurent Mignonneau. Leurs installations interactives témoignent de la perspective pluridisciplinaire de leur travail qui associe l'art aux sciences du vivant, et à l'informatique. Le fonctionnement des SMA réalisés dans le champ des arts visuels s'inspire de celui des systèmes vivants. Leur élaboration repose sur une approche réactive qui s'appuie sur les recherches réalisées en biologie et en éthologie. L'approche réactive permet d'animer des créatures artificielles en définissant des modèles de comportements qui relie la perception à l'action sans faire usage de raisonnements symboliques. L'approche réactive s'oppose à l'approche cognitive qui vise à élaborer des agents cognitifs, c'est-à-dire des créatures artificielles dotées de capacités de raisonnement qui leur permettent d'agir sur des représentations symboliques de leur environnement (Richard, 2001).

Les SMA sont couramment employés pour définir des systèmes qui reproduisent le comportement d'animaux et de végétaux (Flake, 1998), mais aussi élaborer des simulations qui se détournent de ces modèles. Les SMA ne sont pas nécessairement employés pour reproduire des phénomènes observables dans la nature. Ils servent à construire des écosystèmes purement artificiels qui possèdent leurs propres règles et dont l'ensemble des paramètres peut être contrôlé par leurs concepteurs⁰². Les simulations de l'artiste Ian Cheng donnent un exemple des libertés que peuvent prendre les auteurs de SMA vis-à-vis des modèles biologiques et physiques que la nature propose pour attribuer à un environnement virtuel des propriétés singulières.

Exposée en 2015 au Musée d'Art Moderne de la ville de Paris à l'occasion de l'exposition *Co-Workers*, son installation *Droning Like a Ur* (2014) fait interagir au sein d'un même environnement différents agents autonomes qui prennent la forme d'objets 3D trouvés sur Internet ou appartenant à des jeux vidéo. L'émergence est un thème central du travail de Ian Cheng. Pour cet artiste, la programmation représente un outil qui lui permet de générer de la complexité à partir de la définition de lois comportementales et de propriétés élémentaires rattachées à des artefacts. Ian Cheng écrit peu de code. Il définit un comportement et établit des prédispositions à l'aide de scripts qu'il attribue individuellement aux objets que ses simulations mettent en scène. La programmation lui permet de créer un écosystème dans lequel vont interagir ces objets selon des règles qu'il a lui-même préalablement définies,

02. Le caractère totalement construit d'un SMA pousse ses auteurs à adopter un point de vue et à choisir les paramètres qui joueront un rôle déterminant dans le comportement et l'évolution de ses agents.

mais dont les répercussions vont progressivement lui échapper en s'additionnant les unes aux autres, pour générer selon ses propres mots : « une complexité inattendue » (Evers, 2015, p. 111).

Malgré son caractère déterministe, la programmation en tant qu'activité donne forme à des dynamiques et des structures complexes dont l'apparition, sous les traits de la nouveauté, est susceptible de surprendre ses usagers qui observent le résultat de leur travail en exécutant régulièrement le programme qu'ils sont en train d'écrire. Lorsqu'elle est employée pour concevoir des SMA, la programmation, et les logiciels qui servent de supports d'écriture, ne s'appréhendent pas comme des outils, mais comme de véritables partenaires avec lesquels il s'agit de composer pour définir de façon itérative des comportements qui ouvrent sur des structures macroscopiques et des dynamiques émergentes. La partie suivante de cet article énumère les propriétés des entités qui composent les SMA, et décrit comment les artistes leur attribuent des comportements.

1.1. — Décrire un comportement

La programmation permet de définir, sous la forme d'algorithmes, des fonctionnalités qui peuvent être exécutées manuellement par leurs auteurs, ou être attribuées à des entités dotées d'une certaine indépendance. Appelées agents, ces entités opèrent au sein d'environnements virtuels selon des modes de coopération, de conflit et de concurrence qui leur permettent de survivre voire de se reproduire. Un agent visera à satisfaire ses propres objectifs en fonction de ses compétences et des ressources qui sont à sa disposition, mais aussi de la représentation qu'il possède, à un moment donné, de son environnement. Un agent est capable d'agir seul, c'est-à-dire de prendre des décisions selon ses propres critères, sans l'intervention d'un autre agent ni d'un opérateur humain (Richard, 2001). Définir un agent autonome revient à concevoir une entité individualisée. C'est cette relative autonomie vis-à-vis d'un opérateur humain que met en exergue le *Pixel Blanc* d'Antoine Schmitt (1996) qui cherche à atteindre une destination qui change continuellement au sein d'un environnement bidimensionnel d'un noir absolu sur lequel il n'a aucune prise. L'artiste a utilisé la programmation pour définir un comportement, un « mode d'être » modélisé d'après une interprétation simplifiée du modèle freudien de l'appareil psychique et qu'il a attribué à un pixel (Schmitt, 2000).

Comme l'a écrit Antoine Schmitt, le *Pixel Blanc* « fonctionne pendant qu'on le regarde. [...] Rien n'est préenregistré, tout se passe en temps réel, ici et maintenant, dans le même temps que celui du spectateur » (2000). Les déplacements du *Pixel Blanc* ont lieu au moment où se déplace le spectateur dans l'espace d'exposition ou lorsque celui-ci le fixe du regard. Il s'instaure un dialogue entre la trajectoire du pixel blanc et le parcours du visiteur. La programmation confère à l'agent autonome une présence singulière. L'ordinateur n'est pas envisagé comme un simple moyen de communication ou un outil de synthèse d'images et de sons. Le travail d'Antoine Schmitt a la particularité de mettre en évidence la tension qui se crée entre une entité informatique dotée d'une certaine liberté de mouvement et les forces qui contraignent son déplacement. Elle se retrouve aussi bien mise en scène dans ses performances audiovisuelles les plus récentes comme *Tempest* (réalisée en duo avec Franck Vigroux et présentée en octobre 2015 à la Philharmonie de Paris) que dans ses simulations datant du début des années 2000 comme *Vexation 1*.

Vexation 1 soumet un agent autonome à des instructions qu'il doit exécuter dans un ordre prédéterminé afin de jouer indéfiniment la même mélodie, et ce malgré la liberté de mouvement qui lui a été aussi attribuée. Antoine Schmitt définit une entité qui joue inlassablement les mêmes sons, les uns à la suite des autres, en se cognant aux parois de l'espace dans lequel elle a été

insérée. Seul le tempo varie, témoin de la liberté de mouvement, toute relative, qui lui a été accordée. Le travail d'Antoine Schmitt interroge l'indépendance qu'il est possible d'accorder à des agents autonomes *via* la description de comportements. Il rappelle le caractère déterministe de la programmation, les algorithmes ne connaissant pas le hasard.

1.2. — Organisations artificielles

Les SMA rassemblent, au sein d'un même environnement, des entités individualisées pour lesquelles il est possible à tout moment d'indiquer une position. Les agents d'un SMA sont situés. Leur position détermine leur capacité à percevoir et agir. Contrairement aux systèmes experts de l'intelligence artificielle et à la majorité des autres programmes informatiques, les SMA n'ignorent pas leur environnement, ils constituent « de véritables sociétés d'êtres qui doivent se mouvoir, planifier, communiquer, percevoir, agir et réagir et, d'une manière générale, "vivre" et travailler dans un milieu dans lequel ils entrent parfois en conflit avec d'autres agents » (Ferber, 1995). L'élaboration d'un SMA se distingue des approches systémiques plus classiques en considérant l'action et l'interaction comme les éléments moteurs de la structuration d'un système dans son ensemble. Les SMA prennent le parti de l'émergence: une émergence épistémique relative à l'observateur qui apparaît sous les traits d'une nouveauté possiblement fondamentale.

Issu d'une collaboration entre Joseph Nechvatal et Stéphane Sikora le programme *Computer Virus 2.0* (2002) arbore les propriétés principales des SMA. Le logiciel met en scène un processus de destruction créatrice au cours duquel des images peintes et sélectionnées par Joseph Nechvatal sont attaquées par des colonies virus qui se nourrissent de la couleur de leurs pixels. Les colonies de virus correspondent à des communautés d'agents autonomes qui partagent les mêmes caractéristiques. Tous sont capables, à l'aide de capteurs, de percevoir les ressources disponibles dans leur voisinage afin de se diriger vers les pixels qui contiennent les couleurs dont ils peuvent s'alimenter. Ils sont aussi dotés d'effecteurs qui leur donnent la capacité d'agir sur leur environnement en prélevant ces couleurs afin d'avoir suffisamment d'énergie pour continuer à exister voire à se dupliquer. Le programme *Computer Virus 2.0* donne la possibilité à ses auteurs d'explorer des phénomènes et des dynamiques complexes issus de l'interaction entre ses agents situés. Il leur permet de définir une direction artistique en contrôlant l'impact des virus sur les images qui leur servent d'hôte via la définition en amont de leur comportement.

Les évolutions des SMA ne sont pas soumises à un déterminisme simple qui permettrait de les prédire dans leur ensemble à partir de l'analyse d'un état initial. Les SMA obéissent aux principes des phénomènes chaotiques. Les conséquences de la moindre modification des conditions de départ, ou les effets de la seule introduction de variables aléatoires se trouvent amplifiés par les interactions entre agents si bien qu'il est impossible de connaître à l'avance l'état ultérieur du système. Mais comme les simulations du programme *Computer Virus 2.0* le montrent, les évolutions des SMA ne sont pas non plus incontrôlables. Il est aisé au contraire d'observer des effets d'auto-organisation et de voir surgir des structures dont les évolutions sont parfaitement maîtrisées. Les SMA permettent d'élaborer des systèmes qui manifestent des comportements désirés, et s'adaptent à des situations initialement non prévues.

Parmi les premières installations interactives qui ont fait usage d'un SMA pour faire évoluer des créatures artificielles figure *A-Volve* de Christa Sommerer et Laurent Mignonneau (1994-97). Cette installation emblématique de l'art interactif des années 90 a été conçue de façon à permettre à ses usagers de dessiner la forme de créatures aquatiques (sur une surface tactile) afin de les voir prendre vie dans un bassin rempli d'eau. Ses auteurs souhaitaient

transformer à cette époque la réception des œuvres d'art en imaginant des dispositifs artistiques avec lesquels il était possible de communiquer, et le résultat de ces échanges se devait d'être imprévisible (Reichle, 2009).

Laurent Mignonneau a fait usage de la programmation pour modéliser en temps réel des animaux⁰³ à partir du dessin de leur silhouette, et définir des comportements de prédation. Le système mis en place permet aussi à ces créatures de se reproduire et de transmettre certaines de leurs caractéristiques à leur descendance (Sommerer et Mignonneau, 2009). À partir des comportements définis par ses auteurs en fonction de l'apparence de chaque animal, le SMA génère des agents de plus en plus aptes à survivre dans leur environnement. Un principe de sélection naturelle privilégie les entités qui possèdent une bonne condition physique, car elle leur permet de chasser et de se protéger des autres créatures plus facilement⁰⁴.

Le SMA produit de l'altérité en introduisant une part d'aléa dans le processus qui vise à simuler l'accouplement de deux créatures aquatiques. Les informations que transmettent les animaux à leur descendance pour définir leur apparence et leur comportement sont légèrement altérées afin de générer artificiellement de la diversité. Ces altérations entraînent des changements inattendus vis-à-vis des formes et du comportement des agents autonomes. Il est difficile d'évoquer le fonctionnement des SMA sans faire référence au monde du vivant tant ces systèmes s'inspirent de la théorie de la sélection naturelle énoncée par Darwin dans son livre *De l'origine des espèces* (1859). Le processus d'évolution d'agents situés est fréquemment qualifié de processus d'évolution génétique malgré la réduction que font subir ces systèmes à la notion de patrimoine génétique et aux mécanismes de reproduction sexuée.

Les artistes engagés dans l'élaboration de SMA s'inspirent de mécanismes moteurs de l'évolution des espèces comme la sélection naturelle, mais ils n'entendent pas en proposer une traduction parfaite. Au contraire, leur pratique artistique les pousse à prendre des libertés vis-à-vis de ces modèles afin de créer des systèmes qui possèdent leur propre logique. Les SMA représentent dans le champ des arts visuels une instance de l'« esprit distribué » que définit Roy Ascott au cours des années 90 pour évoquer les modalités d'une pratique artistique qui se construit par défaut en réseau :

« Nous sommes en train de créer une culture dans laquelle l'« artiste » devient un système complexe et largement distribué, dans laquelle la connaissance et la perception tant humaines qu'artificielles ont leur place ; un art qui émerge d'une multiplicité d'interactions dans un espace de données. » (1995)

Les SMA ont participé à créer des installations interactives prenant davantage en compte et de façon plus nuancée la présence de visiteurs dans un espace d'exposition. Les interactions entre agents, mais aussi avec les spectateurs, ne se résument pas à déclencher des actions de type on/off dont le résultat est immédiatement perceptible, et souvent décevant. Elles influent sur des processus qui peuvent s'étendre sur de longues périodes de temps. Les SMA élaborés dans le champ des arts invitent à ne pas se contenter d'observer, mais à s'immerger dans la variabilité fluide de bassins de données. Ils proposent un changement d'attitude et de perspective dont les conséquences sont radicales. Les SMA permettent aux artistes de proposer de nouvelles façons d'appréhender des propositions artistiques qui ne peuvent plus être considérées comme des objets aux formes stables et aux fonctionnalités arrêtées. Les SMA ouvrent leurs installations et environnements virtuels à la transformation et aux changements.

03.

Contraction des mots « animal » et « artefact », le terme animal a été inventé pour désigner des agents autonomes qui reproduisent des comportements animaux. Les créatures aquatiques définies par Christa Sommerer et Laurent Mignonneau ont pour objectifs principaux de se nourrir, d'éviter les prédateurs et de s'accoupler. Elles semblent agir de façon intelligente alors qu'elles font seulement preuve d'adaptabilité à leur environnement.

04.

La condition physique de chaque animal est intimement liée à sa forme qui détermine sa capacité à se déplacer rapidement pour fuir et chasser d'autres créatures. Elle est implicitement définie par les visiteurs qui choisissent de les dessiner dans le but de les voir intégrer le bassin numérique.

Employés par des designers, les SMA ouvrent sur des perspectives analogues notamment au cours de la phase de conception d'un projet. La rapidité avec laquelle la programmation permet de traiter d'importantes quantités de données augmente le temps que peut passer un artiste ou un designer à manipuler une matière informatique avant de fixer, au moins temporairement, sa forme. L'élaboration d'un SMA permet d'explorer des champs de possibles à la recherche de nouvelles structures et dynamiques. Certains programmes et librairies logicielles, permettant de définir le comportement d'agents autonomes et de les intégrer à des environnements virtuels, s'adressent expressément à des designers comme le montre la librairie iGeo. iGeo est une librairie logicielle conçue par Satoru Sugihara, fondateur de l'entreprise de design computationnel ATLV située à Los Angeles. Elle s'adresse à des designers ayant des compétences de base en programmation, et qui souhaitent générer des structures complexes en définissant le comportement d'agents situés. Un des principaux objectifs d'iGeo est de simplifier l'élaboration de SMA, en invitant ses usagers à choisir entre différents types d'entités individualisées (Sugihara, 2014). Chaque type d'agents autonomes possède son propre comportement et génère à un niveau macroscopique, par interaction avec son environnement, des structures qui lui sont propres. Les types d'agents prédéfinis correspondent à des primitives que les designers peuvent associer pour concevoir rapidement un SMA. Ils peuvent aussi choisir de créer de nouveaux types d'agents en étendant ou en réécrivant les fonctionnalités qui sont associées à ceux proposés par défaut. Ils pourront alors créer des formes qu'il n'était pas encore possible de générer à partir du canevas que représente iGeo.

2. — Phénomènes d'émergence

Les SMA offrent la possibilité d'explorer des champs de possibles, mais ils ne permettent pas nécessairement de les étendre ni de leur ajouter des dimensions. Les SMA élaborés dans le champ des arts représentent malgré leur capacité à générer des structures et des dynamiques complexes, des systèmes dont il est possible de circonscrire les évolutions. Les agents sont construits à partir de primitives dont la définition détermine l'étendue de l'espace formel pouvant être exploré. Leurs interactions ouvrent sur des espaces qui peuvent couvrir d'importantes étendues, mais qui demeurent néanmoins clos.

Cette clôture permet de maîtriser la part d'aléa que génèrent les interactions entre agents à un niveau macroscopique. Le caractère clos des espaces sur lesquels ouvrent les SMA peut aussi servir à redéfinir le concept d'émergence en distinguant deux formes de nouveauté : une nouveauté relevant d'un processus combinatoire et une autre issue de la création de nouvelles primitives. Peter Cariani qualifie d'émergence combinatoire (*combinatoric emergence*) l'ensemble de formes, de structures et de dynamiques, que les ordinateurs sont capables d'élaborer à partir de primitives définies à l'aide de langages de programmation. Il oppose cette forme de nouveauté à une émergence qu'il qualifie de créative (*creative emergence*) et qui repose sur des processus non computationnels capables de produire au sein d'un système donné de nouvelles primitives (2012).

Les SMA font partie des simulations computationnelles auxquelles Peter Cariani n'accorde pas cette émergence créative, car ils correspondent à des systèmes dont les évolutions peuvent être répliquées à l'infini : « Chaque

fois qu'elle sera exécutée avec les mêmes conditions initiales, une simulation empruntera la même trajectoire. La simulation computationnelle sera reproduite à l'identique; sa course ne déviara pas du modèle des possibles trajectoires que ses observateurs auront défini » (1991, p. 790).

Les émergences créatives de Peter Cariani correspondent à des émergences épistémiques capables de bouleverser notre vision du monde en faisant accéder leurs témoins à de nouvelles couches du sensible. Elles impliquent la constitution d'autres points de vue à partir de l'apparition de capteurs et d'instruments de mesure qui permettent de lire et interpréter le réel avec plus d'exactitude. Pour Peter Cariani, les simulations computationnelles comme les SMA ne peuvent pas produire de telles formes d'émergences, car elles sont bâties sur un ensemble de primitives qui n'évoluent pas.

Les propriétés des SMA évoluent effectivement selon des conditions qui ont été préalablement fixées par leurs auteurs. Mais qu'en est-il des évolutions qui se produisent lors de l'élaboration de ces systèmes? Leur apparition ne repose-t-elle pas sur des formes d'émergences créatives? La définition des entités qui composent un SMA dans le champ des arts n'est jamais arrêtée. Elle est sans cesse retravaillée par ses auteurs-observateurs à partir des structures et des dynamiques auxquelles elle donne forme à un instant donné. Les résultats des interactions entre agents ont le pouvoir de créer de nouveaux points de vue, de produire de nouvelles formes de perception au cours de la phase de conception des SMA qui revient à définir de nouvelles primitives. La programmation remplit cette fonction créatrice en participant à définir le comportement d'agents au sein d'un SMA et à faire émerger de nouvelles structures macroscopiques. Elle remplit aussi cette fonction de façon plus générale en facilitant l'accès et le traitement de données provenant de différents capteurs et interfaces numériques. ◀

RÉFÉRENCES

- ASCOTT, R. (1995), « Télénoïa », in L. Poissant (dir.), *Esthétique des arts médiatiques*, Québec, Presses Universitaires de l'Université du Québec, vol.1, p.363-383.
- CARIANI, P. (1991), « Emergence and Artificial Life », *Artificial Life II*, Vol.10, Redwood City, Californie, Addison-Wesley, p.775-796.
- CARIANI, P. (2012), « Creating New Informational Primitives in Minds and Machines », *Computers and Creativity*, Berlin, Springer, p.383-417.
- EVERS, E. (2015), « In Conversation with Ian Cheng », *Ian Cheng : Live Simulations*, Leipzig, Spector Books, p.101-115.
- FERBER, J. (1995), *Les systèmes multi-agents : Vers une intelligence collective*, Paris, InterEditions.
- FLAKE, G. W. (1998), *The Computational Beauty of Nature : Computer Explorations of Fractals, Chaos, Complex Systems and Adaption*, Cambridge, Massachusetts, The MIT Press.
- LARTIGAUD, D.-O. (2011), « Retour au code », in D.-O. Lartigaud (dir.), *Art++*, Orléans, HYX, p.15-23.
- LAUB, J., BOHNACKER, H., GROSS, B. ET LAZZERONI, C. (2011), *Design génératif : Concevoir, programmer, visualiser*, Paris, Pyramid.
- LÉVY, P. (1992), *De la programmation considérée comme un des beaux-arts*, Paris, La Découverte.
- REICHLI, I. (2009), « Art and Digital Evolution : The Genetic Art of Christa Sommerer & Laurent Mignonneau », in *Christa Sommerer Laurent Mignonneau : Interactive Art Research*, Vienne, Springer, p.56-59.
- RICHARD, N. (2001), *Description de comportements d'agents autonomes évoluant dans des mondes virtuels*, Thèse de Doctorat, École Nationale Supérieure des Télécommunications.
- SCHMITT, A. (2000), « Le Pixel Blanc », *Site personnel*, en ligne (consulté en août 2015) : <http://goo.gl/1h8qD0>
- SHIFFMAN, D. (2012), *The Nature of Code : Simulating Natural Systems with Processing*, The Nature of Code.
- SIKORA, S. ET NECHVATAL, J. (2011), « Le projet Computer Virus : Un équilibre entre art et complexité », in Z. Kapoula et L.-J. Lestocart (dir.), *Esthétique et complexité : Création, expérimentations et neurosciences*, Paris, CNRS, p.67-84.
- SOMMERER, C. ET MIGNONNEAU, L. (2009), « A-Volve », in *Christa Sommerer Laurent Mignonneau : Interactive Art Research*, Vienne, Springer, p72-87.
- SUGIHARA, S. (2014), « iGeo : Algorithm Development Environment for Computational Design Coders with Integration of NURBS Geometry Modelling and Agent Based Modelling », *Design Agency : Proceedings of the 2014 Association for Computer Aided Design in Architecture*, Los Angeles, p.23-32.
- WHITELAW, M. (2004), *Metacreation : Art and Artificial Life*, Cambridge, Massachusetts, The MIT Press.