Design
Arts
Médias

Systèmes : logiques, graphies, matérialités

Victor Guégan
Kim Sacks

2021-07-16

Pour son quatrième « Dossier thématique », la revue en ligne *Design, Arts, Médias* lance un appel à contributions autour des formes de logiques computationnelles au sein des pratiques du design, des arts et des médias.

Le dossier, dirigé par Victor Guégan et Kim Sacks, entend réunir des articles qui s'efforceront de questionner les infra-structures logiques et techniques qui gouvernent les pratiques esthétiques dans leurs relations à l'émergence de la création assistée par ordinateur. Ce numéro mettra en valeur les articles abordant le concept d'implémentation – entendu comme la transposition d'une logique de système dans un médium/artefact – et analysant les idéologies gouvernant les productions graphiques assistées.

1. Le sujet, les enjeux, les pistes

1.1. La thématique du dossier

> *« À la fin des années 1950, il y a eu une tentative profonde et sérieuse de transformer le design en une activité scientifique, de le rationaliser. Cette approche est née à la Hochschule für Gestaltung d'Ulm et a trouvé l'une de ses sources de force dans la "nouvelle" science de la cybernétique, qui était en son temps, à la manière dont nous, les humains, recherchons la réponse universelle, ambitieusement promue comme une nouvelle science qui nous permettrait de résoudre tous nos problèmes. Elle était donc, de toute évidence, significative pour le design[1]. »*

Comme en attestent ces quelques remarques de Ranulph Glanville, le design, la cybernétique et les théories de l'information et de la communication ont une histoire intriquée, se nourrissant réciproquement avec, semble-t-il, pour objectif commun la rationalisation des idées qui les gouvernent. Les années qui suivent la fin de la Seconde Guerre mondiale ont été marquées par des transformations majeures dans le champ des pratiques esthétiques – design, arts et médias. Qu'il s'agisse de la publication de *Understanding Media*[2] de Marshall McLuhan, des expositions de la *Nouvelle Tendance*[3], *Arte Programmata*[4], la célèbre *Cybernetic Serendipity*[5], ou encore la constitution de l'*Experiments in Art and Technology* (E.A.T.), toutes ces instances contemporaines illustrent un champ disciplinaire en effervescence ayant pour ambition de traiter des problèmes de l'époque. Au travers des conférences de Macy[6], une volonté se dessine : celle de faire apparaître un socle commun de pensée qui rendrait possible une sorte d'unification de différents champs disciplinaires, de la philosophie aux mathématiques, de l'anthropologie à la sémiotique, de la linguistique aux arts. Cette volonté de production d'une première théorie de la cybernétique prend corps dans l'interdisciplinarité : l'enjeu est de ré-établir les causalités (circulaires) dans les sciences sociales et la biologie. L'effet de la pensée cybernétique est flagrant, elle se popularise très rapidement probablement en ce qu'elle touche à des questions qui éveillent l'imaginaire contemporain : les machines « intelligentes », les automates, mais surtout de façon plus générale, le contrôle des systèmes, des humains comme des machines.

Dans le design, les arts et les médias, l'approche transdisciplinaire appelle à des définitions de notions héritées des champs précédemment cités, souvent complémentaires ou quelquefois contradictoires. L'enjeu du dossier est de questionner les définitions de programmes et d'algorithmes comme paradigmes de pensée, l'information comme médium, l'implémentation comme méthode, les pratiques assistées par des systèmes de conception et les nouvelles interconnexions entre les différentes disciplines qu'elles suggèrent.

1.2. Les axes

Trois principaux axes de réflexion sont proposés :

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

Axe 1 : Quelles sont les logiques (abstraites) qui gouvernent les pratiques esthétiques systématiques ?

Ce premier axe traite des machines computationnelles en elles-mêmes et de leurs logiques internes. Il aborde le lien entre l'émergence de l'art computationnel et l'importance des théories du langage, de l'information et de la cybernétique. L'articulation entre deux concepts centraux des relations entre art, design et pratiques computationnelles sera débattue, algorithme et programme, au travers de la notion d'implémentation qui sera l'occasion de chercher à mieux définir les logiques (abstraites) qui gouvernent les pratiques esthétiques systématiques.

Axe 2 : Comment s'articulent langages algorithmiques, logiciels et programmes (typo)graphiques ?

Ce second axe traite des logiciels et programmes (typo)graphiques, en ce qu'ils donnent à voir les infra-structures de la pensée systématiques, par les formes qu'ils produisent. Il s'agira de documenter et de questionner la conception des programmes et des algorithmes dans le domaine de la communication et de la poésie visuelle, du design graphique et de la typographie à partir des années 1960, que ce soit comme outil technique, modèle de réflexion ou comme point de départ pour une recherche en art et en design. L'axe traitera de l'esthétique des pratiques systématiques et des ses outils (logiciels) en questionnant l'influence de ces derniers sur les artefacts, les langages graphiques et les systèmes sémiotiques produits.

Axe 3 : Comment la pensée systématique déborde-t-elle des pratiques esthétiques ?

Ce dernier axe traite de la pensée systématique au sein d'environnement complexes. Il sera le lieu d'un questionnement sur les enjeux politiques de la distinction software/programme et des structures de la pensée systématique dans l'imaginaire de machines utopiques. Il permettra d'aborder la pensée systématique en ce qu'elle détermine les méthodes de conception, d'implémentation et l'usage d'artefacts programmatiques. La pensée systématique sera abordée dans un ensemble de software échappant à la représentation, constituant une matrice (peut-être difficilement perceptible) de gouvernance du vivant et du non-vivant.

2. Les modalités de l'appel

- **20 septembre 2021** : Envoi des propositions (3000 signes) auprès des coordinateurs du numéro aux deux adresses suivantes : **vguegan(at)esad-orleans(dot)fr** et **kim.sacks(at)univ-paris1(dot)fr**

Les propositions doivent comporter un titre, le prénom et le NOM de l'auteur ainsi que quelques lignes de présentation (qualité, rattachement institutionnel ou lieu d'exercice de la profession, une ou deux références bibliographiques). La revue *Design, Arts, Médias* accepte des contributions dans toutes les langues.

- **4 octobre 2021** : Réponses aux auteur.e.s après examen, de façon anonyme, des propositions

- **6 décembre 2021** : Remise des articles complets (de 30 000 signes à 50 000 maximum)

- **17 janvier 2022** : Retours aux auteur.e.s après expertise en double aveugle

- **18 février 2022** : Remise des articles (version définitive)

- **fin mars 2022** : Mise en ligne du dossier thématique

3. Orientations, axes, réflexions

Les pratiques esthétiques – arts et design – et les mathématiques ont une histoire entremêlée et

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

jalonnée de moments clés : la théorisation de la perspective, de la symétrie, des rapports de proportions, ou encore le nombre d'or. Dans la société industrialisée du tournant du 20e siècle, l'apparition progressive des machines au potentiel computationnel, puis au cours du 20e siècle des ordinateurs, s'est traduite par l'imbrication technique entre ces dispositifs et des formes émergentes au sein des pratiques du design, des arts et des médias. Les potentiels qu'offrent ces nouvelles pratiques sont nombreux : application de modèles mathématiques aux formes graphiques, utilisations d'instructions, algorithmie, définition systématique, approche d'une logique informatique, synthèse visuelle, simulation, permutations, ou combinaisons. Ces potentiels constituent un corpus de logiques que nous qualifierons de systématique en ce qu'elles répondent à des règles prescrites qui prennent forment par l'exécution.

La pensée des programmes et l'émergence d'une gouvernance systématique semblent se décliner en trois temps :

1) de la guerre à 1956 : la convergence de plusieurs champs disciplinaires aboutit à l'émergence d'une pensée unifiée de l'information, à savoir la cybernétique et la théorie (mathématique) de la communication et de l'information.

2) entre 1956 et 1970 : les systèmes et les programmes deviennent indissociables d'un travail d'intégration et de conception dans un objet, une réalité matérielle, tangible. Comme l'attestent les expositions de la *Nouvelle Tendance* ou d'*Arte Programmata*, l'artefact et les logiques sous-jacentes à celui-ci ne se pensent pas de façon indépendante. Le programme formalise la pensée dans un objet codifié.

3) à partir de 1970 : L'exposition *Software* et les réflexions de Jack Burhnam autour du même sujet marquent un tournant. Le concept de software se détache de celui de programme ; il n'est plus la formalisation d'un processus dans un médium donné mais il s'entend, de manière plus abstraite, comme une logique sans nécessité d'artefact. Pour conséquence, la pensée des systèmes tend vers une pensée des programmes *aniconic,* qui échappent à la représentation[7].

Notre dossier cherchera à mettre en lumière comment ces infrastructures logiques prennent forme, se traduisent dans le langage, les signes et les symboles visuels, et comment ces modes de pensées mettent en tension l'abstraction et sa formalisation – son implémentation, ses artefacts et ses médias. Nous souhaitons questionner les enjeux historiques et les conséquences contemporaines, à savoir, la production des images à lire, des signes à traduire, bref des pratiques esthétiques qui émergent à une époque où l'information et la communication se sont imposées à la fois comme véhicule des idées, des images, comme outil de production et comme « théorie » de ces canaux de diffusion, voire peut-être d'infrastructure idéologique commune.

3.1. Axe 1 : Quelles sont les logiques (abstraites) qui gouvernent les pratiques esthétiques systématiques ?

Ce premier axe souhaite, par différentes contributions, analyser les idées qui gouvernent les productions computationnelles (cybernétiques) : l'information, l'implémentation, l'abstraction, le programme/l'algorithme, le traitement syntaxique/sémantique, les relations software/hardware, le code. Ces concepts ont donné lieu à de nombreuses définitions, dans les différents champs disciplinaires au sein desquels ils se déploient, comme les sciences informatiques, l'ingénierie ou encore la linguistique. Cet axe propose d'explorer la généalogie de ces concepts, leurs définitions et leurs évolutions théoriques ainsi que la porosité de ces différentes définitions.

Les principes de la computation précèdent historiquement leurs implémentations techniques dans des machines computationnelles, qu'il s'agisse d'un système de règles ou d'instructions, de règles de jeu, de musique ou d'architecture, de *De re aedificatoria* (1452) de Leon Battista Alberti au *Underweysung der Messung* (1525) d'Albrecht Dürer[8]. Il est communément admis que les travaux d'Ada Lovelace et de Charles Babbage ont été déterminants pour ce qui s'annonçait comme l'avènement des machines computationnelles et de l'automatisation du traitement mathématique. La *machine analytique*, le commentaire de Luigi Federico Menabrea et sa traduction par Lovelace

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

(incluant la fameuse *note G*) marquent le passage d'une logique mathématique à une mécanisation de l'algorithme, et en conséquence, font émerger la nécessité de rendre le traitement du « code » automatisable. Ce qui nous intéresse particulièrement dans ces premières recherches est qu'il s'agit là des prémisses des questionnements autour de la machine et de son potentiel d'exécution[9] qui dessine alors entre les lignes une opposition conceptuelle : les limites d'une machine exécutante face à son potentiel génératif. Cette dichotomie anime différents champs disciplinaires dont il nous semble pertinent d'étudier les relations ; les informaticiens en ce qu'elle va cadrer les limites techniques de la computation ; les pionniers du computer art[10] - artistes, mais aussi designers, graphistes, musiciens, danseurs, écrivains et poètes - en ce qu'ils essaieront de repousser ces limites en générant des œuvres à partir d'exécutions machiniques ; la linguistique computationnelle en ce qu'elle s'attachera à « ouvrir » le langage aux machines, par le biais d'un traitement syntaxique.

Cette porosité s'avère déterminante dans l'élaboration des premières notations formelles d'un système de décision algorithmique : les théories d'Axel Thue[11], d'Emil Leon Post[12] ou de Turing ont posé les éléments théoriques nécessaires à la création d'une passerelle entre la logique formelle, la linguistique théorique et la possibilité d'une implémentation technique du traitement syntaxique de chaînes de caractères. Cette approche par un traitement syntaxique formel va permettre de nourrir les méthodes de traitement des chaînes de caractères et d'appliquer la logique formelle au développement naissant des langages de programmation, à l'instar d'ALGOL 60 qui s'appuie sur la *Forme de Backus-Naur*[13]. Notre dossier propose d'interroger l'apparition d'un langage « coder », lisible par les machines, en la mettant en relation avec les recherches en linguistique théorique, notamment la description des structures syntaxiques du langage naturel de Noam Chomsky[14] et le développement dans la seconde moitié du 20e siècle de la linguistique théorique formelle. Ainsi, nous invitons les contributeurs à étudier l'émergence dans les années 1960 d'avancées notables, simultanées, dans la science informatique (apparition des premiers langages de haut-niveau) et du déploiement d'un modèle de pensée qui s'attache à formaliser l'abstraction mathématique dans le but de rendre possible le traitement par des machines computationnelles.

Comme en attestent les travaux mathématiques d'Alonzo Church, de Kurt Gödel, de Stephen Cole Kleene ou encore d'Alan Turing, le concept d'algorithme fonde l'abstraction des machines et se manifeste dans de nombreuses implémentations computationnelles. La notion d'algorithme a été définie informellement par Andrey Markov (1954)[15], Stephen Cole Kleene (1967)[16], ou encore de façon plus approfondie par Donald Knuth (1968)[17]. En 1968, ce dernier détaille les cinq propriétés qui caractérisent le concept d'algorithme : la *finitude*, la *définition précise*, les *entrées*, les *sorties* et le *rendement*[18]. Ces différents principes, s'ils se traduisent souvent par l'écriture d'un arbre logique d'étapes interdépendantes afin de les rendre intelligibles, présentent surtout l'algorithme en ce qu'il relève d'une logique décorrélée de sa mise en œuvre, de son écriture. L'algorithme définit une procédure non-ambiguë. Knuth précise en outre que les sciences de l'informatique cherchent non seulement le développement d'algorithmes, mais surtout de *bons*[19] algorithmes. Ce faisant, l'auteur indique que le passage de l'algorithme au programme peut prendre de nombreuses formes, certaines répondant d'un rendement plus important, autrement dit, qui nécessite un nombre d'itérations moins important pour chaque étape. De cette façon, il distingue le programme de l'algorithme en spécifiant que le programme est « l'expression d'une méthode computationnelle dans un langage de programmation[20] ». Le langage de programmation présente une forme syntaxique, une notation conventionnelle qui formule les algorithmes et qui permet la production d'artefacts qui les appliquent (ou qui les exécutent dans le cas du logiciel). Comme suggéré par les analyses de William J. Rapaport[21], ce passage du champ syntaxique au champ sémantique est l'implémentation. L'implémentation n'est jamais abordée de manière autonome, elle est toujours l'implémentation de quelque-chose. En ce sens, il s'agit d'une notion relative. La maison est l'implémentation du plan, un programme est l'implémentation d'un algorithme. Le point, qui nous semble essentiel, est que, ce quelque-chose dont il est question, ce dont le programme est l'implémentation, est ce que Rapaport qualifie d'*abstraction*[22]. Cette abstraction, qui n'est ni nécessairement concrète ni nécessairement abstraite, s'opère exclusivement dans un médium.

Dans cette mesure, les propositions pour cet axe pourraient questionner les relations entre

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

l'implémentation, le médium et l'abstraction en ce qu'elles induisent (peut-être) des effets esthétiques pré-déterminés. Elles pourraient traiter la question suivante : comment se traduit et s'articule l'abstraction logique avec la production d'artefacts, avec l'implémentation dans un médium ? A titre d'exemple, les propositions pourront aborder l'information hors-contexte et son traitement syntaxique comme médium produisant une sémantique dans un artefact.

Dès les années 1950 et particulièrement au début de la cybernétique, il n'existe pas de consensus sur la définition de l'information, ni sur la façon avec laquelle celle-ci s'articule avec la notion d'abstraction. L'opposition émergente entre les théories de l'information de Claude Shannon et Norbert Wiener et celles relatives à la *structural information* (thèse défendue par Donald MacKay), se traduit par les distinctions suivantes :

1) Pour Shannon et Wiener, l'information est la possibilité du choix. Elle peut être entendue comme purement quantifiable puisqu'abstraite de son contexte (définissant ainsi directement ce que l'information *est*). La réification de l'information (objet stable et constant) lui permet d'être décorrélée du contexte mais simultanément coupe celle-ci de la possibilité de la représentation[23]. Le design serait alors l'abstraction de l'implémentation, à savoir, l'application sémantique d'une syntaxe logique décorrélée de toute représentation.

2) Pour MacKay, l'information peut être entendue comme structurelle en ce qu'elle se détermine non pas par la possibilité d'évaluer une quantité d'information, mais par sa capacité opérationnelle à produire (et altérer) des représentations[24]. Celle-ci ne peut se faire qu'en considérant le contexte, et en conséquence, octroie à la représentation sa capacité à produire la connaissance (de l'information intrinsèque).

Historiquement, c'est bel et bien l'information abstraite de son contexte qui l'emporte, sans doute parce qu'elle sut être, en son temps, la réponse technique la plus sensible, la plus pragmatique en somme. Pourtant, l'argument de MacKay en faveur d'une information structurelle nous semble important à re-questionner aujourd'hui, dans le sens où il préfigure des modes de représentations propre à la cybernétique : l'opération dont il est question est celle d'une sélection parmi un corpus de représentations préétablies, et non d'une production de représentation[25], faisant ainsi écho au passage d'une logique de programme à une logique de software souligné par Jack Burnham[26]. Nous avons donc une définition de l'information quantifiable, produisant des artefacts par le biais de l'implémentation opérée dans un médium, et qui répond aux conditions des logiques mathématiques puisque nous pouvons leur appliquer des raisonnements abstraits, quantifiables, avec des objets logiques variables. Pourtant, en reprenant notre questionnement sur l'implémentation, Rapaport exclut la possibilité de traduire l'abstraction autrement qu'au travers d'un médium, celui-ci conditionnant inévitablement le contexte de l'opération.

Les questions qui se posent alors, et que nous invitons à traiter, sont les suivantes : quelle « valeur » donner aux pratiques esthétiques si celles-ci découlent d'un procédé pré-figuré ? Les effets produits par les programmes sont-ils le reflets d'une (infra)structure déterministe ?

3.2. Axe 2 : Comment s'articulent langages algorithmiques, logiciels et programmes (typo)graphiques ?

Sans s'y limiter, notre dossier thématique invite à apporter des éléments de réponse à ces questions dans le champ de la communication et de la poésie visuelle, qui par la manipulation et l'invention constantes de systèmes et de relations sémiotiques qui caractérise cette discipline, nous apparaît comme un espace privilégié de l'articulation entre langages algorithmiques abstraits, programmes, formes graphiques ou plastiques et *software*. L'automation fait passerelle entre le design et l'algorithme : les grilles, la typographie, les compositions peuvent dorénavant être soumises à des règles mathématiques complexes, combinatoires, épuisant les possibles. Les formes évoluent et les qualités graphiques des productions traduisent les logiques machiniques qui rendent possibles ces objets esthétiques : lignes rigoureusement droites, courbes sinusoïdales « parfaites », aplats de couleurs uniformes, symétries et rapports de proportions minutieusement pré-déterminés. Nous laisserons de côté les questions concernant la réception, en particulier

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

celles concernant l'écran, qui nous semblent avoir été traitées dans un nombre de travaux conséquent, pour centrer notre attention sur les processus de travail en art et en design, de la conception à la production[27].

Pour cela, nous proposons tout d'abord de poursuivre les recherches concernant la rencontre des cultures visuelles avec les logiques algorithmiques et computationnelles, à partir de la fin des années 1950[28]. Si, dès la fin des années 1960, des expositions, des publications et des recherches ont présenté et étudié ce type d'expériences en art et en architecture, les tentatives de designers et de chercheurs pour documenter et évaluer avec rigueur l'apport des nouveaux langages algorithmiques et des méthodes programmatiques dans le domaine du design graphique, de la typographie, du dessin de lettres ou de la poésie visuelle pendant ces années-là, restent en friche[29]. Contrairement au monde des Beaux-Arts qui reste attaché à une multiplicité de médiums, l'ordinateur s'impose pourtant entre les années 1960 et 2000 comme un outil indispensable pour la conception graphique et la communication visuelle. Grâce au développement d'interfaces graphiques, puis de logiciels dits de « PAO » [*Publications Assistée par Ordinateur*], des machines de calcul conçues au départ pour aider les scientifiques, les militaires et les comptables ont été adoptées par un nombre grandissant de concepteurs et se sont transformées en de véritables outils d'écriture et d'aide à la création graphique, participant à l'esthétique communicationnelle et archivistique de la seconde partie du 20e siècle. En outre, alors que l'émergence d'un art de la programmation informatique se construit en parallèle d'un intérêt croissant des mondes de l'art pour les théories de l'information, de la communication et de la linguistique, il est urgent d'interroger une discipline qui s'est construite depuis le début du 20e siècle sur des rapports dynamiques entre texte et image et une réflexion sur la sémiotique, les frontières entre le lisible et le visible, sans oublier les problématiques techniques des processus industriels liés à l'impression et la diffusion de masse.

Le deuxième axe de ce dossier thématique se donne donc comme objectif de documenter et d'interroger l'articulation entre d'un côté les « programmes » graphiques et typographiques dans un contexte de création industrielle, artistique et poétique et de l'autre, la culture informatique, scientifique et technologique dans laquelle ils prennent source et, en particulier les langages computationnels. Le rôle des programmes sera ici questionné doublement, d'une part comme matrice conceptuelle, interrogeant en particulier la manière dont la structure de langages algorithmiques et mathématiques fondée sur des séries d'instructions inspire les méthodes de création graphique et, d'autre part, dans leurs réalités matérielles, comme outils permettant de générer des formes graphiques nouvelles. Nous ouvrons également la voie à des contributions qui évoquent des cas où de nouveaux usages ou langages computationnelles pourraient venir d'expérimentations dans le domaine de la conception graphique et des arts visuels.

Actant du caractère transdisciplinaire des expérimentations autour des différents médias depuis l'après-guerre, ce deuxième axe du dossier propose de se centrer sur la question suivante : la manière dont, depuis les années 1950, les designers graphiques, les typographes, les artistes, les poètes visuels, les publicitaires, les chercheurs et les industriels se sont appropriés les langages informatiques et les logiques computationnelles en les implémentant dans des programmes qu'ils ont eux-mêmes conçus ou contribué à inventer. Nous proposons de revenir tout d'abord sur les projets menés dans le sillage de courants avant-gardistes interdisciplinaires comme l'art concret, l'art optique, Fluxus, la poésie visuelle et bien-sûr la cybernétique qui renvoient à des questions relatives au lien entre visible et lisible, à l'écriture et aux outils graphiques propres aux métiers de la communication visuelle. Des projets poétiques et conceptuels aux frontières des arts visuels comme les expérimentations de Friedrich Nake, Herbert Franke ou mystico-conceptuelles du poète et artiste Carl Fernbach-Flarsheim (*The Boolean Image* / *Conceptual Typewriter*, 1970) témoignent du fait que cette question de la retranscription écrite de la langue passée à la moulinette de langages computationnels essaime largement hors du champ des professionnels de la typographie et de l'informatique, dans le cadre de programmes poétiques et artistiques créés à partir non plus seulement de mots mais également d'instructions entrées dans des langages informatiques comme Fortran ou Algol. Ensuite, dans le domaine du graphisme, ce dossier sera l'occasion de questionner également l'apparition de nouvelles méthodes de conception systématiques fondées sur des rapports de calcul et des séries d'instructions, qui souvent

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

précéder l'usage de l'ordinateur. A titre d'exemple, pourront être interrogés les travaux issus des principes de l'art concret de Max Bill[30] ou Richard Paul Lohse[31], le système de grille de Josef Müller-Brockmann[32], ou encore les programmes de Karl Gerstner[33]. L'ensemble des démarches menées en Suisse sous le nom de « style international », qui pourraient regrouper les démarches citées précédemment sont d'autant plus intéressantes à étudier qu'elles déploient, avant puis en parallèle de l'usage progressif d'outils computationnels dans la communication visuelle, des logiques proches de celles utilisées dans la programmation (esthétique procédurale, grille, permutation, redondance, aléatoire)[34]. Sans que l'appel se limite à ce moment et cette zone géographique, la rencontre entre art concret, design, nouvelles théories de l'information et communication, poésie visuelle, expérimentations avec des ordinateurs et des langages de programmation à la Hochschule für Gestaltung à Ulm et autour de Max Bense à Stuttgart[35] ou encore au sein du GRAV [Groupe de recherche en art visuel] à Paris, constitue un terrain d'étude évident pour ces expériences. Les différentes expositions marquantes de cette génération à la fin des années 1960 (« Cybernetic Serendipity », « Software », « Tendencije 4 ») pourront être interrogées selon l'angle de la communication visuelle et de la lettre. De même dans des revues d'avant-garde ou professionnelles (*Bit*, *Futura*, *Rot*, *Werk*, *Page*, *Semiosis 19*, *Computers and Automation*) un certain nombre d'articles évoquent de manière plus ou moins frontale cette question des liens entre conception (typo)graphique et programmatique, alors que la manière de concevoir graphiquement les couvertures et la mise en page de ces revues cherche souvent à proposer un reflet esthétique aux discussions contenues à l'intérieur de la revue. Ensuite, dans le domaine généralement plus réservé du dessin de caractères, l'ouverture à divers champs disciplinaires des débats visant à concevoir des formes d'écriture « électrifiées », lisibles à la fois par l'être humain et la machine [le *New Alphabet* de Wim Crouwel et la contre-proposition des typographes Gerhard Unger dans la collection Kwadraat-Blad ; l'« Information-Typographie » du théoricien et artiste Kurd Alsleben ; le *Line Up* de Timothy Epps et Christopher Evans ; les formats commandités par différents industriels comme l'OCR-B conçue en 1968 par Adrian Frutiger d'après les recommandations de l'European Computer Manufacturers Association (ECMA) ou encore les expériences autour de caractères de reconnaissance magnétiques (MICR) par IBM] suggère l'apparition de nouveaux types de collaborations et "chaînes" de travail entre dessinateurs de caractères, informaticiens, scientifiques et centres de production qu'il nous semble important de documenter et de questionner[36]. Comment articuler langages informatiques et conception de systèmes graphiques? Lorsqu'il est seul face à la machine, le typographe est-il contraint de se former à la programmation ? Est-il possible de transposer les logiques du code dans un nouveau langage graphique ? Lorsqu'un designer ou un poète utilise des programmes réalisés par d'autres, avec une interface graphique ne nécessitant pas d'entrer dans le code de l'outil, reste-t-il maître des formes et des lettres qu'il manipule?

L'ensemble de ces questions ne pourra trouver de réponse convaincante qu'en documentant et définissant avec plus rigueur l'apport des nouveaux outils informatiques qui sont devenus incontournables dans la plupart des studios de graphisme et des fonderies. L'un des partis pris de ce dossier thématique est de combler le manque de connaissances et parfois même d'intérêt constaté dans les recherches en esthétique et en histoire du design graphique et de la typographie, pour le code informatique et l'ordinateur comme outil de conception graphique. A titre d'exemple, il semble difficile de questionner l'esthétique du dessin de lettres dans les années 1980, sans comprendre comment un logiciel comme Metafont (Donald Knuth) est pensé et structuré[37]. De la même manière, si l'interface d'un logiciel qui fait le lien entre le code et les usages de designers non formés à la programmation, est dite "graphique", c'est bien parce qu'elle est elle-même un objet graphique et, donc, un nouvel objet de recherche pour la discipline. Enfin, les concepteurs qui construisent leurs systèmes graphiques à l'écran ont souvent besoin de penser la réalisation matérielle de leur projet, grâce à différentes techniques de traceurs et d'impression. Ce passage d'une machine de calcul à une machine d'impression nécessite également des programmes capables de faire dialoguer les opérations de conception et de production, de transposer des séries d'instruction logiques à des machines qui ne fonctionnent pas toujours avec les mêmes langages et les mêmes procédures d'activation. En bref, ce dossier thématique souhaite contribuer à l'émergence de nouveaux objets d'étude dans le champ du graphisme, de la typographie et plus généralement des arts visuels : les langages algorithmiques, les programmes, les logiciels et leurs interfaces.

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

3.3. Axe 3 : Comment la pensée systématique déborde-t-elle des pratiques esthétiques ?

Ce dernier axe est l'occasion de questionner ce qui, par la prolifération des pratiques esthétiques gouvernées par les programmes, participe de la structuration d'un imaginaire des machines, utopique, fantasmé, débordant du cadre de la représentation.

Dans les premières lignes du texte d'Umberto Eco pour l'exposition *Arte Programmata*, l'harmonie mathématique est mise en parallèle avec la richesse de la chance et de l'aléatoire[38]. Le texte reprend la dichotomie classique, entre règle mathématique et hasard. Pourtant, comme l'atteste les travaux de Leslie Mezei[39], faire en sorte de contourner la régularité, l'uniformité, la prédictibilité de machines déterministe se fait par détour. L'implémentation de systèmes aléatoires s'oppose fondamentalement aux idéologies guidant l'utopie de la Machine Universelle de Turing : le paradoxe fascine (programmer le non-programmable) et s'illustre par un écart grandissant entre le programme exécuté et ses effets, scindant symboliquement la machine de ses artefacts graphiques, faisant inévitablement écho au texte de Burhnam et la séparation conceptuelle du *software* et du *programme*. Si cette question nous paraît intéressante, c'est qu'elle touche à la constitution d'un imaginaire des machines, capable de tout (ou presque).

Les machines computationnelles stratifient les logiques. Elles proposent un *ordre*[40]. Elles procèdent par niveaux d'abstractions (*Levels of Abstraction*[41]), entendus comme des implémentations techniques de fragments de logiques, potentiellement interdépendants d'autres strates – gouverner par des logiques potentiellement différentes. Cette ordonnance peut être appréhendée par la formulation des niveaux d'abstractions de Giuseppe Primiero, composée ainsi : *Intention – Specification – Algorithm – High-level programming language instructions – Assembly/machine code operations – Execution*[42]. Dans le champ du design, cela se traduit pour Raymond Turner de la façon suivante : *Fonction – Design – Structure – Implémentation – Artefact*[43]. L'auteur va décliner cette logique afin de mettre en lumière le procédé qui mène au programme en tant qu'artefact technique : *Requirements – Design – System specification – Implementation – Programs as Artifacts*[44]. Or, les niveaux d'abstractions communiquent entre eux et, pour ce faire, répondent à une pratique du design de systèmes techniques interopérables, en réseaux ou en strates mais au sein duquel l'étape artefact n'est pas toujours présente. Se pose alors la question : quels sont les limites et les enjeux d'un design de systèmes aniconiques ? Comment le design de protocole s'appuie sur la pensée systématique pour prévoir l'usage du software (par l'utilisateur et par d'autres software) ? Comment l'usager perçoit un programme sans artefact ?

Qu'il s'agisse de la machine utopique de Turning, du déterminisme catégorique de l'algorithme ou encore de l'abstraction formelle et les limites de la computation[45], la production par la machine ne relève pas seulement du champ de la machine (par son implémentation dans un artefact) mais aussi de la constitution d'un protocole de gouvernance : des machines elles-mêmes et de la conduite de projet s'appuyant sur des machines. L'on perçoit aisément que nos problématiques d'implémentation et d'abstraction sont au cœur de l'enjeu du design de programmes, que l'on retrouve dans le process cybernétique explicité par Gordon Pask[46] mais aussi dans le célèbre modèle conceptuel du design d'expérience utilisateur élaboré par Jesse James Garrett[47]. Ces méthodes participent à l'idée d'un artefact qui exécute une logique insondable, en ce les strates logiques de bas niveaux résistent à l'entendement (par la nécessité d'aborder les boîtes noires par leurs matérialités). Mais ils soulèvent également une question idéologique : si le procédé de création oscille entre l'abstraction et son implémentation concrète dans un artefact, comment peut-on par l'observation des systèmes, comprendre leurs logiques internes ?

En prémisses de la cybernétique, Norbert Wiener[48] ne l'entendait pas comme un champ disciplinaire strictement limité aux mathématiques de l'information. Il souhaitait que celle-ci soit une théorie avant tout politique qui définisse les structures idéologiques de la gouvernance, des hommes et des machines, du vivant et du non-vivant. De cette façon, les modèles issus de la cybernétique devaient fournir les outils conceptuels pour appréhender un monde transformé par la

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

guerre[49] et par les choses qui l'ont rendu possible. Si l'on observe le débat entre Wendy Hui Kyong Chun[50] et Alexander Galloway[51], il semblerait que le software entendu comme idéologie suggérerait que l'abstraction, l'opération et la prescription déterminent un imaginaire de la relation homme-machine, de la relation au hardware par l'interface. L'idée d'une machine inexplicable – *black box* – n'est pas nouvelle et découle en grande partie des travaux d'Ashby en 1956 puis de Wiener en 1961 sur les méthodes d'analyses de systèmes fermés. Elle repose sur l'idée (reprise plus récemment par Galloway) que le software se dissimule (technique) et se dévoile simultanément (par ses effets esthétiques). Cette résistance à la connaissance – intrinsèquement ou par défaut d'outils d'analyses adéquates – semble avoir ouvert la voie à tout un champ d'études sur les conséquences sociales et politiques du software comme culture, à l'instar des nombreux travaux de Mackenzie[52], Eubanks[53], Pariser[54], O'Neil[55] ou Pasquale[56]. Si certains de ces textes fondent leurs argumentations sur les relations entre une technique invisibilisée et leurs conséquences sociales (qu'ils cherchent à rendre visible, comme les inégalités sociales automatisées par l'usage d'algorithmes), ils s'attardent en particulier sur ce qui vient *après* la computation, ce qu'elle produit en guise d'infrastructure de contrôle[57]. Nous proposons de questionner ces imaginaires *avant*, *pendant* et *après* la computation, à savoir, comment ces idéologies – entendues comme infra-structures de la pensée – caractérisent l'implémentation dans un médium, que celui-ci soit *iconique* ou *aniconique*. En ce sens, le dossier pourra aborder les protocoles de conduite de projets, en design, arts et médias, en questionnant ce qui dans les méthodes pré-déterminés répond d'une logique programmatique. Le dossier propose aux auteurs de mettre en relation les structures du design process au regard de l'exploitation de méthodes formelles, comme par exemple l'application de projections probabilistes au sein des calculs de rentabilité des projets en design[58].

Ce dernier axe de l'appel invite donc des textes qui sondent les machines, les programmes, le software, les algorithmes, dans la relation entre une dimension technique (peut-être insondable) et une dimension politique. Les textes pourraient interroger comment s'élabore un dialogue entre la résistance des interfaces à la connaissance et la prolifération d'artefacts techniques. Le dossier cherchera à mettre en valeur les propositions dont les arguments critiques sont accompagnés d'un regard analytique sur les infrastructures (technique et politique) des pratiques esthétiques en design, arts et médias.

4. Bibliographie indicative

Antonelli, Paola (dir.),*Talk to Me: Design and the Communication between People and Objects*, cat. d'exp., New York, MoMA, 2011.

Archer, Bruce L., *The Structure of Design Processes,* Thesis (Ph.D.) Royal College of Art, London, 1968.

Armstrong, Helen, *Graphic Design Theory: Readings from the Field*, New York, Princeton Architectural Press, 2009.

Ashby, W. Ross, *An Introduction to Cybernetics*, London, Chapman & Hall LTD, 1957.

Banham, Reyner (ed.), *The Aspen Papers : twenty years of design theory from the International Design Conference in Aspen*, Londres, Pall Mall Press, 1974.

Bense, Max, *Aesthetica*, Paris, éditions du Cerf, 2006

Bill, Max, « La pensée mathématique dans l'art de notre temps » dans : *Max Bill*, Thomas Maldonado, Buenos Aires, éditions Nueva Vision, 1955.

Breton, Philippe, *L'Utopie de la communication*, Paris, La Découverte, 1997.

Burnham, Jack, *Software - Information Technology: Its New Meaning for Art*, cat d'exp., New York,

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

Jewish Museum, 1970.

Ceruzzi, Paul E., *A History of Modern Computing – 2nd ed.*, Cambridge, MA, The MIT Press, 2003.

Chomsky, Noam, *Syntactic Structures*, Princeton, New Jersey, Mouton and Co., 1957.

Chun, Wendy Hui Kyong, *Programmed Visions : Software and Memory*, Cambridge, MA, The MIT Press, 2011.

Church, Alonzo, « On the Concept of a Random Sequence », *Bulletin of the American Mathematical Society 2*, 1940, pp. 130–135.

Cladders, Johannes (dir.)., *Programm, Zufall, System: ein neuer Zweig am alten Konzept der Sammlung Etzold*, cat. d'exp., Städtisches Museum Mönchengladbach, 1973

Colburn, Timothy R., *Philosophy and Computer Science*, Armonk, NY, M. E. Sharpe Inc., 2000.

Dartnall, Terry (dir.), *Artificial Intelligence and Creativity : an Interdisciplinary Approach,* Dordrecht, Springer Netherlands, 1994.

Davis, Stephen Boyd, Gristwood, Simone, « Computing, Design, Art :Reflections on an Innovative Moment in History. », *History and Philosophy of Computing. IFIP Advances in Information and Communication Technology,* Springer, ITA, (2016), pp. 101-115.

Davis, Stephen Boyd, Gristwood, Simone, « The Structure of Design Processes : Ideal and Reality in Bruce Archer's 1968 Doctoral Thesis. », *Proceedings of DRS 2016, Design Research Society 50th Anniversary Conference,* Brighton, UK, 27–30 June 2016.

Drucker, Johanna, *SpecLab : digital aesthetics and projects in speculative computing*, Chicago, University of Chicago Press, 2009.

Drucker, Johanna, *Visualisation. L'interprétation modélisante*, Paris, Éditions B42, 2020.

Eubanks, Virginia, *Automating Inequality : How High-Tech Tools Profile, Police and Punish the Poor*, New York, St Martin's Press, 2018.

Fischer, Thomas, Herss, Christiane M. (dir.), *Design Cybernetics : Navigating the New*, Cham CH, Springer Nature, 2019.

Floridi, Luciano (dir.), *The Routledge Handbook of Philosophy of Information*, London, Routledge, 2016.

Flusser, Vilém, *Post-histoire*, Monlet, T&P Work Unit, 2019.

Fuller, Matthew, *Behind the Blip : Essays on the Culture of Software*, New York, Autonomedia, 2003.

Frank, Helmar G., *Informationsästhetik, kybernetische Ästhetik, Ästhetokybernetik : eine Einführung*, Berlin, Institut f. Kybernetik, 1997.

Franke, Herbert ; Herzogenrath, Wulf ; Nierhoff-Wielk, Barbara (dir.), *Ex machina - frühe Computergrafik bis 1979 : die Sammlungen Franke und weitere Stiftungen in der Kunsthalle Bremen*, cat. d'exp., Berlin / Munich, Deutscher Kunstverlag, 2007.

Garrett, Jesse James, *The Elements of User Experience: User-Centered Design for the Web and Beyond (2nd Edition) (Voices That Matter)*, Berkeley, CA, New Riders Press, 2010.

Gerstner, Karl, *Designing Programmes*, Teufel, Niggli, 1968 (2e édition).

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

Gerstner, Karl, *A Compendium for Literates: A System of Writing*, Cambridge, The MIT Press, 1974.

Gerstner, Karl, « Think program : synopsis of the exhibition "designing programs/programming designs" », Directed by Emilio Ambasz, Museum of Modern Art, New York, February 5 - March 30, 1973, The Museum of Modern Art, 1973.

Gomringer, Eugen (ed.), *konkrete poesie* : *deutschprachige autoren*, Stuttgart, Reclam, 1972.

Grau, Olivier, *MediaArtHistories,* Cambridge, MA, The MIT Press, 2007.

Hagengruber, Ruth, Riss, Uwe V. (dir.), *Philosophy, Computing and Information Science*, London, Pickering & Chatto, 2014.

Hayles, Katherine, *How We Became Posthuman : Virtual Bodies in Cybernetics, Literature, and Informatics*, Chicago & London, University Press of Chicago, 1999.

Hofstadter, Douglas R., *Gödel, Escher, Bach: An Eternal Golden Braid*, New York, Basic Books, 1979.

Kittler, Friedrich, *There is no software*, ctheroy.net, 1995, [en ligne, http://ctheory.net/articles.aspx?id=74, consulté le 09 juin 2021]

Kleene, Stephen Cole, *Mathematical Logic*. New York, Wiley, 1967.

Knuth, Donald E., *The Art of computer programming volume I, Fundamental Algorithms*, (third edition), Massachusetts, Addison-Wesley, 1997.

Knuth, Donald E., *Computer and Typesetting*, Addison-Weslay, Reading / Massachusetts, 5 vol., 1984.

La Monte Young ; Mac Low, Jackson (dir.), *An Anthology of Chance Operations*, New York, autoédité, 1962.

Lartigaud, David-Olivier (dir.), *Art*++, Orléans, Hyx, 2011.

Lee, Pamela M., *Chronophobia : On Time in the Art of the 1960s*, Cambrige MA, The MIT Press, 2004.

Mackenzie, Adrian, *Machine Learners : Archaeology of a Data Practice,* Cambridge MA, The MIT Press, 2017.

Maeda, John, *Design by Numbers*, Cambridge MA, The MIT Press, 1999.

Maldonado, Tomas, *Avanguardia e razionalita : Articoli, saggi, pamphleti 1946-1974*, Turin, Einodi, 1974.

Maldonado, Tomas, *Environnement et idéologie : (vers une écologie critique)*, Paris, Union générale d'édition, 1972.

Manovich, Lev, *Software Takes Command*, New York, Bloomsbury Academic, 2013.

Markov, Andrey, « Theory of algorithms » Tr. Mat. Inst. Steklov 42, 1–14. traduction Edwin Hewitt, *American Mathematical Society Translations*, Series 2, Vol. 15 (1960), 1954.

Masure, Anthony, *Design et humanités numériques*, Paris, Éditions B42, 2017.

McLuhan, Marshall, *Understanding Media : The Extensions of Man* (1964), Cambridge MA, The

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

MIT Press, 1994.

Medosch, Armin, *New Tendencies : Art at the Threshold of the Information Revolution (1961-1978)*, Cambridge, MA, The MIT Press, 2016.

Meneguzzo, Marco (dir.), *Arte Programmata : Italian Kinetic Art From the 1960s*, M&L Fine Art, London, 4 October - 20 December 2016.

Migayrou, Frédéric (dir.), « Coder le monde », cat. d'exp., Paris / Orléans, Centre Pompidou / Hyx, 2018.

Minsky, Marvin (dir.), *Semantic Information Processing*, Cambridge, Massachusetts, The MIT Press, 1968.

Minsky, Marvin, *Computation : Finite and Infinite Machines*, Englewood Cliffs NJ, Prentice-Hall, 1967.

Minsky, Marvin, *Computation: Finite and Infinite Machines,* New Jersey, Prentice Hall Inc., 1967.

Minsky, Marvin, *The Society of Mind,* New York, Simon & Schuster, 1986.

O'Neil,Cathy, *Weapons of Math Destruction :How Big Data Increases Inequality and Threatens Democracy*, New York, Crown Publishers, 2016.

Pariser, Eli. *The Filter Bubble : What the Internet Is Hiding from You*, New York, Penguin Press, 2011.

Pasquale, Frank, *The Black Box Society : The Secret Algorithms That Control Money and Information*, Cambridge, Harvard University Press, 2015.

Pias, Claus (dir.), *Cybernetics, The Macy Conferences, 1946-1953, The Complete Transactions*, Zürich/Berlin, diaphanes, 2003, (University of Chicago Press, 2016).

Peirce, Charles S., *Graphen und Zeichen : prolegomena zu einer apologie des pragmatizismus*, Stuttgart, Hansjörg Mayer, 1971.

Peirce, Charles S., *Ecrits sur le signe*, Gérard Deledalle (éd.), Paris, Le Seuil, 1978.

Rapaport, William J., « Implementation as Semantic Interpretation : Further Thoughts », *Journal of Experimental & Theoretical Artificial Intelligence*, 17(4), 2005 pp. 385–417.

Rapaport, William J., « Implementation Is Semantic Interpretation », *The Monist*, 82(1), 1999, pp. 109–130.

Rapaport, William J., « Understanding Understanding: Syntactic Semantics and Computational Cognition », Tomberlin, James E. (dir.), *Philosophical Perspectives*, Vol. 9 : AI, Connectionism, and Philosophical Psychology, Atascadero, CA, Ridgeview, 1995, pp. 49–88.

Reichardt, Jasia (dir.), *Cybernetics, Arts and Ideas*, London, Studio Vista, 1971.

Reichardt, Jasia, *Cybernetic Serendipidity. The Computer and the Arts : a Studio International Special Issue*, London, Studio International, 1968.

Rieder, Bernhard, *Engines of Order : A Mechanology of Algorithmic Techniques*, Amsterdam, Amsterdam University Press, 2020.

Rojas, Raúl, Hashagen, Ulf (dir.), *The First Computers – History and Architectures*, Cambridge, MA, The MIT Press, 2000.

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

Rosen, Margit (dir.), *A little known story about a movement, a magazine and the computer's arrival in art : new tendencies and bit international, 1961-1973*, Cambridge, Massachusetts, The MIT Press/ZKM, 2011.

Shanken, Edward A. (dir.), *Telematic Embrace : Visionary Theories of Art, Technology, and Consciousness by Roy Ascott*, Berkeley CA, University of California Press, 2003.

Shanken, Edward A., *Systems*, London, Whitechapel Gallery, Cambrige MA, The MIT Press, 2015.

Shannon, Claude E., Weaver, Warren, *A Mathematical Theory of Communication*, Urbana, University of Illinois Press, 1964.

Solt, Mary Ellen, « Charles Sanders Peirce and Eugen Gomringer. The Concret Poem as a Sign », *Poetics Today*, n° 3, été 1982, p. 197-209

Turner, Raymond, *Computational artifacts : Towards a philosophy of computer science*, Berlin Heidelberg, Springer. 2018.

van Heijenoort, Jean, *From Frege to Gödel : A Source Book in Mathematical Logic, 1879-1931*, Cambridge MA, Harvard University Press, 1967.

von Foerster, Heinz, *Understanding Understanding : Essays on Cybernetics and Cognition*, New York, Springer-Verlag, 2003.

Wark, McKenzie, A Hacker Manifesto, Cambridge, Harvard University Press, 2004.

Weibel, Peter (dir.), *Beyond art : a Third Culture. A Comparative Study in Cultures, Art, and Science in 20th Century Austria and Hungary*, Wien, Springer, 2005.

Wiener, Norbert, *Cybernetics, or control and communication in the animal and the machine*, Cambridge, Massachusetts, The Technology Press, New York: John Wiley & Sons, Inc., 1948.

Zenil, Hector (dir.), *A Computable Universe : Understanding and Exploring Nature as Computation*, London, World Scientific Publishing Company – Imperial College Press, 2013.

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

1. Traduction des auteurs : « In the late 1950s, there was a profound and serious attempt to turn design into a scientific activity, to rationalise it. This approach originated at the Hochschule für Gestaltung in Ulm, and found as one of its sources of strength the “new” science of cybernetics, which was at the time, in the way in which we humans look for the universal answer, ambitiously promoted as a new science that would allow us to solve all our problems. It was, therefore, obviously significant for design. », Glanville, Ranulph, « Try Again. Fail Again. Fail Better : The Cybernetics in Design and the Design in Cybernetics » dans Thomas Fischer, Christiane M. Herr (dir.), *Design Cybernetics : Navigating the New*, Cham CH, Springer Nature, 2019, p. 29.
2. Marshall McLuhan, *Understanding Media : The Extensions of Man* (1964), Cambridge MA, The MIT Press, 1994.
3. Darko Fritz, « La notion de “programme” dans l’art des années 1960 – art concret, art par ordinateur et art conceptuel » dans Lartigaud, David-Olivier (dir.), *Art++*, Orléans, Hyx, 2011, pp. 27-40.
4. Umberto Eco, « Arte Programmata » dans Rosen, Margit (dir.), *A little known story about a movement, a magazine and the computer’s arrival in art : new tendencies and bit international, 1961-1973*, Cambridge, Massachusetts, The MIT Press/ZKM, 2011, pp. 98-101.
5. Jasia Reichardt, *Cybernetic Serendipidity. The Computer and the Arts : a Studio International Special Issue*, London, Studio International, 1968.
6. Claus Pias (dir.), *Cybernetics, The Macy Conferences, 1946-1953, The Complete Transactions*, Zürich/Berlin, diaphanes, 2003, (University of Chicago Press, 2016).
7. Jack Burnham, « Notes on art and information processing », *Software :Information technology : its new meaning for art*, New York: Jewish Museum, 1970, p. 11.
8. Peter Weibel, « It Is Forbidden Not to Touch: Some Remarks on the (Forgotten Parts of the) History of Interactivity and Virtuality » dans Olivier Grau, *MediaArtHistories*, Cambridge, Massachusetts, The MIT Press, p. 24.
9. Mario Verdicchio, « The Role of Computers in Visual Art », Fabio Gadducci, Mirko Tavosanis (dir.), *History and Philosophy of Computing*, Cham CH, Springer Nature, 2016, p. 290.
10. Voir les travaux de Ben F. Laposky, Lillian Schwartz, Frieder Nake, Charles Csuri, Ken Knowlton, A. Michael Noll, Billy Klüver.
11. Axel Thue, « Probleme über Veränderungen von Zeichenreihen nach gegebenen Regeln », Christiania Videnskabs-Selskabs Skrifter, I.Math.-naturv. Klasse, 10, 1914, et James F. Power, « Thue’s 1914 paper : a translation », Department of Computer Science, National University of Ireland, Maynooth,Co. Kildare, Ireland. August 28, 2013.
12. Emil Leon Post, « Introduction to a general theory of elementary propositions » dans Jean van Heijenoort, *From Frege to Gödel : A Source Book in Mathematical Logic, 1879-1931*, Cambridge MA, Harvard University Press, 1967, pp. 264-283.
13. La Forme de Backus-Naur, théorisée par John Backus et Peter Naur, permet la description de la syntaxe d’un langage de programmation. Elle servira à Chomsky pour sa grammaire hors-contexte et conséquemment sera cruciale dans le développement d’ALGOL 60.
14. Noam Chomsky, *Syntactic Structures*, Princeton, New Jersey, Mouton and Co., 1957.
15. Andrey Markov, « Theory of algorithms » Tr. Mat. Inst. Steklov 42, 1–14. traduction Edwin Hewitt, *American Mathematical Society Translations*, Series 2, Vol. 15 (1960), 1954.
16. Stephen Cole Kleene, *Mathematical Logic*. New York, Wiley, 1967.
17. Donald E. Knuth, *The Art of computer programming volume I, Fundamental Algorithms,*

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

(third edition), Massachusetts, Addison-Wesley, 1997.

18. Traduction des auteurs, « Finiteness », « Definiteness », « Input », « Output », « Effectiveness », *Ibid.*, pp. 4-6.
19. *Ibid.*, p. 7.
20. Traduction des auteurs, « An expression of a computational method in a computer language is called a *program*. », *Ibid.*, p. 5.
21. William J. Rapaport, « Implementation Is Semantic Interpretation », The Monist, 82(1), 1999, pp. 109–130. et William J. Rapaport, « Implementation as Semantic Interpretation : Further Thoughts », Journal of Experimental & Theoretical Artificial Intelligence, 17(4), 2005 pp. 385–417.
22. *Ibid.*, p. 386.
23. Katherine Hayles, *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics*, Chicago, The University of Chicago Press, 1999, p. 56.
24. Donald MacKay, « In Search of Basic Symbols » dans Claus Pias (dir.), *Cybernetics, The Macy Conferences, 1946-1953, The Complete Transactions*, *op. cit.*, p. 480.
25. *Ibid.*, pp. 481-482.
26. Jack Burnham, « Notes on art and information processing », *Software : Information technology : its new meaning for art*, New York: Jewish Museum, 1970, p. 11.
27. Lire notamment : Jennifer Helfand et John Maeda, *Screen: essays on Graphic Design, New Media and Visual Culture* (ed.), Princeton Architectural Press, 2001 ; Lev Manovich, *The Language of New Media*, The MIT Press, 2011 ; Annick Lantenois (éd.), *Lire à l'écran. Contribution du design aux pratiques et aux apprentissages des savoirs dans la culture numérique*, B42, Paris, 2011. Par ailleurs, la revue *Back Office* consacre un de ses numéros à la thématique "écrire l'écran", dont les articles sont accessibles en ligne à l'adresse suivante : http://www.revue-backoffice.com/numeros/03-ecrire-lecran [consulté le 15/07/2021].
28. Christoph Klütsch, *Computergrafik. Ästhetische Experimente zwischen zwei Kulturen ; die Anfänge der Computerkunst in den 1960er Jahren*, Springer, New York / Vienne, 2007
29. Bien que trop confidentiels, un certain nombre de travaux dans ce domaine ont servi de base à notre réflexion pour ce dossier thématique, dont une sélection est présentée en bibliographie. On citera, à titre d'exemple, les travaux de Johanna Drucker, Jessica Helfand, John Meada et, en France, les recherches de Pierre-Damien Huyghe, David Olivier Lartigaud, et plus récemment d'Anthony Masure et de la revue *Back Office* qu'il codirige avec Kevin Donnot et Elise Gay.
30. Max Bill, « Die mathematische Denkweise in der Kunst unserer Zeit », Werk, n° 3, Winterthur, 1949. En 1980, Max Bense publie un article sur Max Bill en prenant comme grille de lecture les théories du linguiste Charles S. Peirce (Max Bense, "L'essai de Max Bill "La pensée mathématique dans l'art de notre temps"", Semiosis 19, n° 3, 1980, pp. 29-36.
31. Richard Paul Lhose, « Art in the age of technology » dans : *Richard Paul Lhose 1902-1988*, cat. d'exp., Wilhem-Hack-Museum, Ludwigshafen am Rhein, 1992.
32. Josef Müller-Brockman, *Grid Systems in graphic design*, Niggli, Teufen, 1981.
33. *Cf*. notamment : Karl Gerstner, Programme entwerfen, Niggli, Teufen, 1964 ; Kalte Kunst, Niggli, Teufen, 1957.
34. Sur l'histoire de ce mouvement, lire la remarquable monographie de Richard Hollis, Swiss Graphic Design, Yale University Press, 2006.
35. *Cf*. notamment : Max Bense, *Aesthetica*, éditions du Cerf, Paris, 2007.
36. Lire une synthèse de ces débats dans l'article de Sarah Owens, « Electrifying the alphabet », *Eye Magazine*, n° 62, hiver 2006

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143

37. Donald Knuth, *Computer and Typesetting*, Addison-Weslay, Reading / Massachusetts, 5 vol., 1984

38. Umberto Eco, « Arte Programmata » dans Margit Rosen (dir.), *A little known story about a movement, a magazine and the computer's arrival in art : new tendencies and bit international, 1961-1973*, *op. cit.*.

39. Leslie Mezei, « Randomness in computer graphics » dans Jasia Reichardt (dir.), *Cybernetics, Arts and Ideas*, London, Studio Vista, 1971, pp. 165-174 et Peter Weibel (dir.), *Beyond art : a Third Culture. A Comparative Study in Cultures, Art, and Science in 20th Century Austria and Hungary*, Wien, Springer, 2005, pp. 342-343.

40. Bernhard Rieder, *Engines of Order : A Mechanology of Algorithmic Techniques*, Amsterdam, Amsterdam University Press, 2020, p. 55.

41. Timothy R. Colburn, *Philosophy and Computer Science*, Armonk, NY, M. E. Sharpe Inc., 2000, p. 174.

42. Giuseppe Primiero, « Information in the philosophy of computer science » dans Luciano Floridi (dir.), *The Routledge Handbook of Philosophy of Information*, London, Routledge, 2016, pp. 90–106.

43. Raymond Turner, *Computational artifacts : Towards a philosophy of computer science*, Berlin Heidelberg, Springer. 2018, p. 27.

44. *Ibid.*, p. 57.

45. Alonzo Church, « On the Concept of a Random Sequence », *Bulletin of the American Mathematical Society 2*, 1940, pp. 130–135.

46. Gordon Pask, « A Comment, a case history and a plan » dans Jasia Reichardt (dir.), *Cybernetics, Arts and Ideas*, *op. cit.*, p. 76.

47. Jesse James Garrett, *The Elements of User Experience: User-Centered Design for the Web and Beyond (2nd Edition) (Voices That Matter)*, Berkeley, CA, New Riders Press, 2010.

48. Norbert Wiener, *Cybernetics, or control and communication in the animal and the machine*, Cambridge, Massachusetts, The Technology Press, New York: John Wiley & Sons, Inc., 1948.

49. Philippe Breton, *L'Utopie de la communication*, Paris, La Découverte, 1997.

50. Wendy Hui Kyung Chun, « Contrôle et liberté : logiciel et plaisir causal » dans David-Olivier Lartigaud (dir.), *Art++*, *op. cit.*.

51. Alexander R. Galloway, « Oubliez la langage, il se surveille lui-même : logiciels et idéologie » dans David-Olivier Lartigaud (dir.), *Art++*, *op. cit*..

52. Adrian Mackenzie, *Machine Learners : Archaeology of a Data Practice*, Cambridge MA, The MIT Press, 2017.

53. Virginia Eubanks, *Automating Inequality : How High-Tech Tools Profile, Police and Punish the Poor*, New York, St Martin's Press, 2018.

54. Eli Pariser. *The Filter Bubble : What the Internet Is Hiding from You*, New York, Penguin Press, 2011.

55. Cathy O'Neil, *Weapons of Math Destruction : How Big Data Increases Inequality and Threatens Democracy*, New York, Crown Publishers, 2016.

56. Frank Pasquale, *The Black Box Society : The Secret Algorithms That Control Money and Information*, Cambridge, Harvard University Press, 2015.

57. Lev Manovich, *Software Takes Command*, New York, Bloomsbury Academic, 2013.

58. Bruce L. Archer, *The Structure of Design Processes*, Thesis (Ph.D.) Royal College of Art, London, 1968, chapitre 5.

© La revue DAM
téléchargé le 2026-11-09 17:05:45, depuis le 216.73.217.143