



HAL
open science

Dérive décodée sur le logiciel

Anthony Masure

► **To cite this version:**

Anthony Masure. Dérive décodée sur le logiciel. Plan Libre. Journal de la Maison de l'Architecture, 2019. hal-04107399

HAL Id: hal-04107399

<https://hal.science/hal-04107399v1>

Submitted on 4 Jun 2023

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.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

Anthony Masure

Dérive décodée sur le logiciel

<http://anthonymasure.com>

Anthony Masure est responsable de la recherche à la HEAD – Genève (Institut de Recherche en Art & Design). Agrégé d'arts appliqués, ancien élève du département design de l'ENS Paris-Saclay, membre associé du laboratoire LLA-CRÉATIS de l'université Toulouse – Jean Jaurès. Ses recherches portent sur les implications sociales, politiques et esthétiques des technologies numériques.

⌘ .. Hors-Série p.3

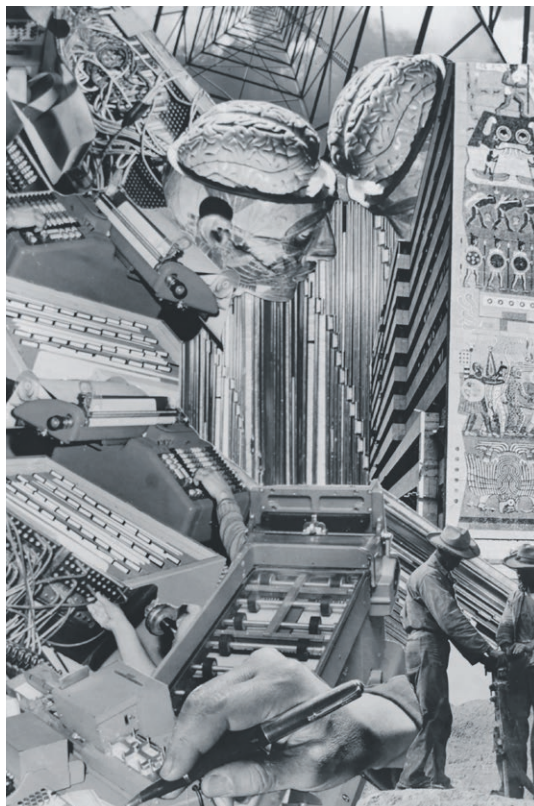
LONG FORMAT

Octobre 2019

Qu'est-ce qu'un logiciel ?

Il faut tout d'abord différencier un algorithme d'un logiciel. Un algorithme, c'est une suite d'instructions écrites pour arriver à un résultat déterminé. Mais un algorithme – même si l'on parle par abus de langage d'algorithme pour désigner un programme ou un logiciel – c'est quelque chose d'abstrait que l'on peut, en théorie, rédiger sur du papier. Les instructions d'un algorithme sont des règles logiques, pensées pour arriver à un résultat. Un algorithme se rapproche de l'algèbre ; il est plutôt de l'ordre de l'équation mathématique. Par contre, un logiciel (une suite de programmes) doit être écrit dans un « langage » machine, plus ou moins compliqué. En effet, pour que le logiciel puisse fonctionner, il faut que son code source, soit « interprétable » par un ordinateur, ce qui suppose une suite d'opérations techniques complexes pour que le langage, écrit à la base par un ou plusieurs êtres humains, soit compréhensible d'une machine. Le processeur d'un ordinateur ne « lit » pas des commandes rédigées dans du pseudo anglais, mais traite des suites de 0 et de 1, incompréhensibles pour les humains. Autrement dit, l'ordinateur, structurellement, consiste à empiler des couches de langages de plus en plus opaques.

Le deuxième aspect qui distingue l'algorithme du logiciel, c'est que dès lors que le logiciel est quelque chose qui est interprété ou exécuté par une machine, cela engage du hardware, des objets matériels. Depuis le mathématicien Alan Turing dans les années 1950, l'invention de l'informatique se base sur la séparation software/hardware, logiciel/matériel, mou/dur. Mais à vrai dire, cette séparation, qui a du sens conceptuellement, n'en a aucun dans la vie de tous les jours. En effet, il ne peut pas y avoir de logiciel sans une inscription matérielle, pour reprendre les idées du chercheur Friedrich Kittler. Autrement dit, le code source, pour fonctionner, est nécessairement dépendant des transistors, des puces, des écrans, de ces choses matérielles qui d'ailleurs engendrent beaucoup de pollution. Dire qu'avec les technologies numériques, on a une « dématérialisation » de la création n'est pas pertinent. Il



Photograph by Lola Alvarez Bravo – Collection Center for Creative Photography ©Center for Creative Photography, The University of Arizona Foundation

suffit de voir les décharges d'ordinateurs, tous les câbles sous-marins, les réseaux, etc. Le hardware est très présent. Concrètement, le logiciel a besoin du hardware pour fonctionner. Par exemple, si votre ordinateur se met à chauffer, cela va avoir une incidence sur le traitement des 0 et des 1, et donc cela risque de bugger. Un programme ne peut jamais être 100% fiable car ce n'est pas une équation abstraite. Dès lors qu'on peut avoir de la poussière, des insectes (bugs) ou de la surchauffe, le logiciel peut « bouger » physiquement. Les rendus d'architecture, gourmands en énergie, poussent matériellement les ordinateurs dans ses limites et peuvent dès lors les faire « planter ».

S'il y a le passage d'un langage à l'autre, cela implique des traductions. L'ordinateur ou le logiciel seraient-ils des traducteurs opaques permettant de traduire la pensée de l'architecte en dessin, donc en représentation du réel ?

Cette notion de traduction me semble très intéressante. Effectivement, on pourrait dire que l'ordinateur réalise une suite de traductions machine. Cependant, tout dépend de ce que l'on entend par traduction : dans la traduction, il y a l'idée d'une interprétation dans le passage d'un langage à l'autre. Ici, toutes les opérations sont automatisées. Donc oui, c'est peut-être une traduction mais c'est une traduction très mécanique qui n'est pas comparable à la traduction qu'un chercheur en littérature pourrait opérer sur de la poésie.

En ce qui concerne la traduction entre la pensée de l'architecte et le résultat obtenu, si l'on se replaçait du point de vue de l'architecte, l'opacité est encore plus grande : la plupart du temps, l'architecte n'a pas accès à la première couche de langage, à savoir celle qui est écrite par les développeurs. L'architecte achète (ou loue) des licences de logiciels qu'il n'a pas conçus. Étonnamment, alors que la profession d'architecte relève du champ élargi de la création, c'est une profession qui est très peu créative ou inventive à l'endroit des logiciels, outils ou « pseudo outils » – parce que je ne pense pas que le logiciel soit un outil – utilisés tous les jours. Il n'y a que peu de réflexion, de recul critique à l'endroit de ces médiations techniques sans lesquelles la profession ne peut aujourd'hui exister.

Si le logiciel n'est pas un outil, qu'est-ce donc ?

Effectivement, on aimerait bien que les logiciels soient des outils, parce qu'une définition simple de l'outil serait un prolongement de la main, ou quelque chose sur lequel on peut avoir une maîtrise, un maniement intellectuel et corporel maîtrisé. L'exemple le plus concret, c'est un marteau ou un crayon. Mais il me semble que le logiciel a une autre nature que le marteau, le pinceau, le crayon,

l'équerre ou le compas. Le logiciel embarque un programme. Il embarque des scripts, des instructions opaques. Il est programmable et reprogrammable. Un marteau n'est pas programmable. Certes, l'ergonomie du marteau va peut-être, suivant sa forme, prédéterminer certains gestes, mais ce que propose le marteau restera toujours en deçà de ce que propose – ou de ce qu'impose – un programme. Pour moi, il n'y a pas de continuité de fait entre l'équerre de l'architecte et Autodesk AutoCAD. Ce serait dangereux de dire que ce sont de simples outils car on risquerait de passer à côté de leurs différences et de ne pas comprendre ce qu'ils transforment dans la profession.

S'il l'on regarde la fenêtre de Dürer, machine mécanique qui automatise la perspective, n'est-on pas déjà dans cette logique du programme ? Le résultat et le geste sont déconnectés par une interprétation mécanique laissée à un dispositif. Si le programme n'est pas un outil, serait-ce un « dispositif » au sens de Giorgio Agamben, comme cette fenêtre pourrait l'être ?

En effet, il y a l'idée de distance, de séparation entre la conception et la production, de séparation entre la main et la pensée, mais l'idée de programmation et de reprogrammation est absente dans cette image. Il n'y a pas l'idée d'une suite d'instructions logiques écrites en langage mathématique même si d'une certaine manière la perspective pourrait s'y apparenter. Selon moi, la plupart des logiciels seraient davantage des « dispositifs » au sens du philosophe Giorgio Agamben, à savoir « des entités qui contrôlent les pensées ou les usages ou les libertés des êtres vivants », davantage que des outils sur lesquels on aurait une totale liberté, une totale maîtrise. Mais je ne crois pas que tout logiciel soit nécessairement un dispositif : c'est tout l'enjeu des professions de la création de faire en sorte que les logiciels ne soient pas seulement cela. Il peuvent aussi être « l'outil convivial » d'Ivan Illich ou « l'appareil » emprunté au philosophe Pierre-Damien Huyghe. Contrairement au dispositif, l'appareil serait une entité qu'on pourrait régler et

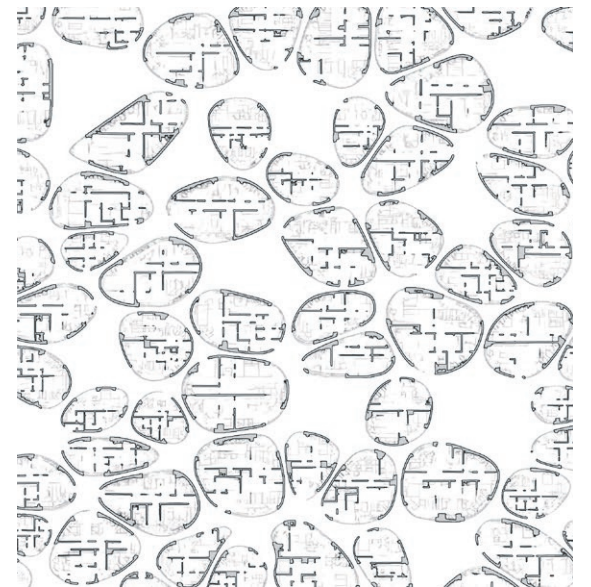
dérégler. Prenons par exemple un appareil photographique, en tout cas argentique, on peut régler la focale, l'exposition et d'autres paramètres de façon indépendante. On procède à une opération intellectuelle avant de produire une image. Avec un appareil photographique, on paramètre l'objet technique. En le réglant, on maîtrise l'image d'une certaine façon, mais en même temps, on ne la maîtrisera jamais tout à fait : quand l'appareil photographique opère une prise de vue, l'être humain est absent de cette opération technique. C'est dans cet équilibre entre maîtrise et abandon de maîtrise que Pierre-Damien Huyghe définit l'appareil. Ici, l'abandon est vécu positivement. Ce n'est pas un manque de maîtrise qui serait avilissant ou nous priverait d'une certaine liberté. Mais peut-on étendre les principes de l'appareil photographique argentique au logiciel contemporain ? Je pense que c'est possible et que c'est tout l'enjeu aujourd'hui. Malheureusement, je ne pense pas que ce soit souvent le cas de beaucoup de logiciels de création en architecture, notamment car les architectes sont souvent absents des processus de conception des logiciels, et que si l'on confie cela à des ingénieurs et à des entreprises dont le but est la rentabilité, on va davantage tendre vers le dispositif que vers l'appareil, pour rester dans ces distinctions conceptuelles.

Les architectes ont-ils toujours été absents des processus de conception de leurs programmes ?

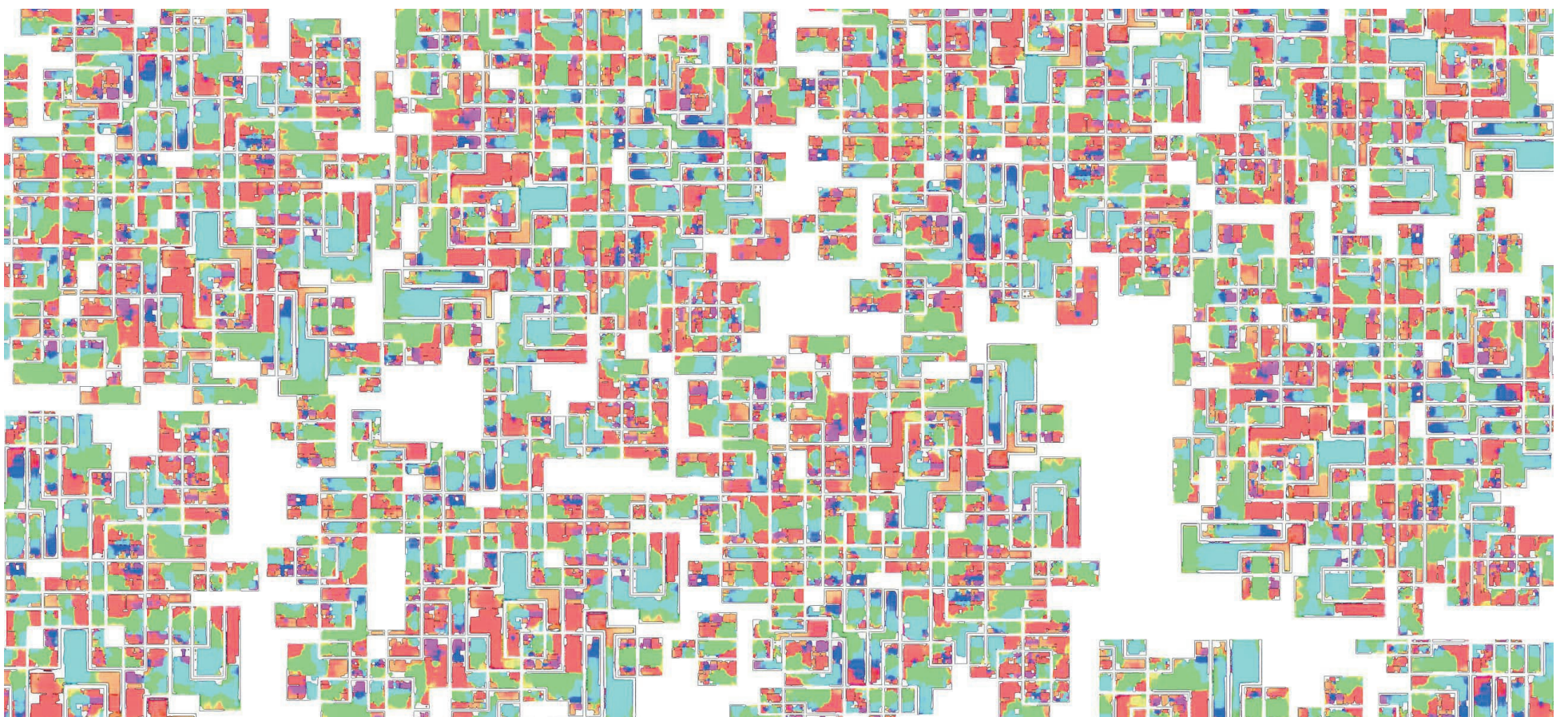
En réalité, l'architecture est le premier champ de la création qui s'est emparé des technologies informatiques ou numériques pour repenser ses pratiques, et ce, avant le design graphique. Selon un cycle de recherche sur « l'archéologie du numérique » effectuée au Centre Canadien d'Architecture (CCA Montréal), des pratiques pionnières ont été mises en évidence dès la fin des années 1980, c'est à dire au tout début de l'informatique personnelle. À cette époque, quelques architectes précurseurs comme Frank Gehry ou Peter Eisenman ont organisé leurs agences autour d'une double activité : d'une part, le développement de logiciels dédiés à la pratique architecturale et d'autre part, la conception de bâtiments,

de plans, de réflexions sur les matériaux, etc. L'entreprise Gehry est d'ailleurs scindée en deux : Gehry Technologies (qui édite des logiciels) et Gehry Architects. Je ne connais que peu d'exemples d'agences d'architecture qui ont une activité à la fois de développement informatique et de pratiques architecturale (Coop Himmel(1)au, MOS Architects, etc.). Certes, les agences n'ont pas les moyens de développer du logiciel – ce qui est vrai – mais il y a des choses plus simples qui existent, à savoir du hacking, du détournement ou de l'extension de programme ou plus simplement, par l'utilisation de plug-ins (Rhino Grasshopper, etc.).

Mais au-delà de la technique il y a le contenu critique : tout ce qui est culture numérique, culture informatique... On va former les futurs architectes à l'utilisation des logiciels, mais très rarement on va leur expliquer les valeurs qui sont embarquées dans le logiciel. La compréhension de la culture numérique et de l'histoire de l'ordinateur mérite d'être présente dans les cursus d'enseignement et de formation.



Extrait de plan généré par GAN © Stanislas Chaillou



Plan généré par GAN © Stanislas Chaillou



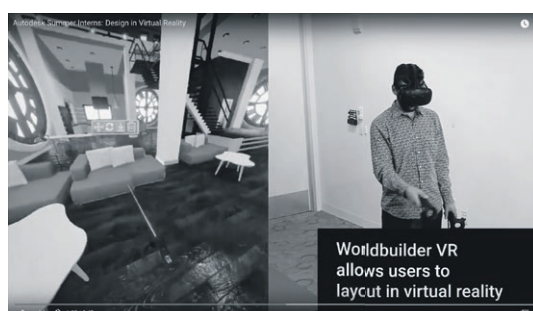
Joel Pennington (Autodesk): Virtual Reality in Design: Now and the Future, A talk from the Design Track at Augmented World Expo USA 2017

Si tous les architectes sont formés à la pratique de l'architecture sur les mêmes logiciels, n'y a-t-il pas un risque d'homogénéisation de la production architecturale ?

En effet, étant donné que certaines formes sont plus facilement tracées dans un certain logiciel grâce aux fonctions préinscrites qui permettent de générer ces formes, c'est possible. Mais malgré tout, cette homogénéisation doit être nuancée. Elle n'est pas totale. Evidemment que l'on peut avoir de multiples pratiques d'un même logiciel, cependant le champ de la création s'élargirait s'il était possible de recréer ou de réécrire ces logiciels à mesure de la conception. Malheureusement, l'étudiant-e comme l'architecte se désintéresse souvent de ces questions : les deux veulent souvent parvenir tout de suite à un résultat. Et développer des logiciels irait à l'encontre de l'efficacité (solution) recherchée. Et si dans les agences on n'utilise qu'un seul programme, pourquoi apprendre autre chose ? Cependant, si l'on réfléchit ainsi, la profession est figée. Il est important que des personnes s'emparent de ces questions pour proposer autre chose. Si on ne le fait pas à l'école, une fois pris dans des conditions de rentabilité en agence, cela sera de plus en plus compliqué. Ce qui s'applique aux logiciels de dessin se retrouve aussi dans le champ de la robotique, qui est en train de se développer considérablement dans le domaine de l'architecture et de la construction. Aujourd'hui des machines-outils, des robots peuvent permettre d'agencer des matériaux d'une manière totalement inédite et inaccessibles aux êtres humains sur les chantiers. De nouvelles formes de procédés de construction sont en voie d'être automatisées. L'automatisation peut à la fois remplacer les techniques existantes de construction mais aussi, et là, c'est beaucoup plus intéressant, développer des techniques de construction inexistantes (voir l'ouvrage *The Robotic Touch*, 2015. Il y a une réflexion de fond à mener sur cette architecture robotisée, automatisée, pour le meilleur comme pour le pire. Il me semble que nous sommes en train de prendre un retard crucial en France dans la prise en compte de ces nouvelles technologies dans l'enseignement et dans la qualité du discours critique à cet égard.

Robotique et automatisation d'un côté, drones, réalité augmentée et réalité virtuelle de l'autre c'est comme si les progrès technologiques avançaient plus vite que la critique que l'on puisse en faire.

Oui, et l'autre technologie qui pose aujourd'hui question, c'est l'intelligence artificielle (IA). Avec les réseaux de neurones du deep learning apparus au milieu des années 1990, on travaille avec des types de programmes qui sont d'une toute autre nature que ceux que j'évoquais auparavant, et qui étaient déjà mal compris par la profession. Ces programmes sont soi-disant auto-apprenants : à partir de données initiales qu'on leur fait ingérer, ils peuvent ressortir des productions ou des propositions formelles qui vont pouvoir automatiser des données initiales. En théorie, ce procédé automatisé, ne concerne donc plus seulement l'exécution de tâches sommaires, mais le domaine historiquement réservé à l'architecte, à savoir le dessin car le tracé « initial » pourrait être confié à une machine. Que deviendrait l'architecte dans ce scénario ? Il pourrait théoriquement disparaître de l'équation. On pourrait imaginer qu'un bureau d'études ou qu'un maître d'œuvre pourraient travailler avec un architecte simulé par une IA. Ce logiciel pourrait produire le dessin, les plans, les documents techniques et tous les actes relatifs à la construction sans passer par une agence. Ce scénario est-il souhaitable ? Je ne le pense pas, car ce que sait très bien faire ce type d'IA, c'est de la reproduction.



Joel Pennington (Autodesk): Virtual Reality in Design: Now and the Future

On lui donnera l'œuvre intégrale d'un architecte et il ressortira une œuvre inédite qu'aurait pu produire ce même architecte. En revanche, il ne sera pas capable d'inventer le prochain Wang Shu, ou le prochain Sou Fujimoto. C'est quelque chose que, pour l'instant, on ne sait pas faire. Mais si des agences bien « réelles » se positionnent spontanément ou par défaut sur de la reproduction de bâtiments existants et ne font œuvre que de peu de créativité dans leur dessin, elles risquent de rapidement disparaître du paysage au profit des machines. Ceci est un appel aux architectes, un appel positif finalement, à faire preuve d'invention et à valoriser d'autant plus la créativité dans les années à venir.

Ce même débat se pose pour tous les métiers de la création. Il faut sensibiliser les étudiant-e-s et les professionnel-le-s à ces fantasmes, ces promesses ou ces réalités de l'automatisation de la création – pour partie déjà en œuvre. Dans les 10 ou 15 années à venir, on ne travaillera certainement plus sur le paradigme actuel d'AutoCAD ou SketchUp : il y a déjà deux ans, le groupe Autodesk a annoncé une extension du logiciel AutoCAD permettant de la génération formelle automatisée (Netfabb). À ma connaissance, c'est encore limité à du design d'objets. Il me semble qu'avec les IA nous avons atteint un deuxième seuil de transformation profonde de la profession, après celui de la CAO (Conception Assistée par Ordinateur). Il faudrait qu'on tire des enseignements des mutations de la profession d'architecte qui ont été entamées dans les années 1980–1990 afin de comprendre et réagir au mieux face aux mutations actuelles. Il ne faudrait pas que la profession soit à nouveau dans le déni car ces débats sont pour l'instant abandonnés aux ingénieurs, ou au mieux à des comités d'éthique, mais les professions de la création ont aussi leur mot à dire dans ces transformations structurelles. En tout cas, si la profession se désintéresse de ces enjeux, elle risque de prendre de plein fouet une mise à l'écart d'un tiers voire de la moitié de la profession, du moins telle qu'elle est aujourd'hui organisée. Plus personne ne sera prêt à payer des gens quand une machine pourra le faire.

En attendant les logiciels de conception immersive en réalité virtuelle permettant de collaborer à distance sur des maquettes à échelle 1:1, le BIM est bien installé au sein des agences. Que penses-tu des solutions logicielles intégrant des procédés collaboratifs avancés ?

Le BIM (Building Information Modeling) permet un autre registre de transformation de la profession parce que le BIM inclut d'autres acteurs que les architectes, notamment ce qui concerne les aspects juridiques et techniques. L'obligation juridique de recourir au BIM dans les marchés publics présente un risque d'assujettissement de la profession à un logiciel contrôlé par les firmes privées, et ce, même si BIM est un protocole et pas un logiciel en tant que tel. On voit bien qu'assez vite ce protocole va être dépendant d'un ou deux éditeurs. On risque alors d'arriver à une sorte de « réglementation algorithmique », pour reprendre la formule du théoricien des médias Evgeny Morozov : le logiciel pourra prescrire non plus seulement des formes mais des formules, des façons d'habiter, des façons de se comporter les uns avec les autres, tout un tas de réglementations thermiques, environnementales, etc. sans qu'on n'ait aucune marge de manœuvre. L'architecte ne pourra plus valider son bâtiment sans passer par cette médiation technique. Il n'y aura plus de négociation humaine possible, ce ne sera plus discutable, on ne pourra plus en débattre. Le risque encouru, c'est une disparition de la responsabilité, et je pourrais même aller plus loin, une disparition de la morale, qui serait alors abandonnée à des programmes. Plus personne n'est responsable de rien, si ce n'est le logiciel. C'est aussi une possible perte ou fin de la démocratie qui est en jeu dans les chaînes de production type BIM, où la politique se réduit à du code et non pas à des négociations. Encore une fois, il faut des débats, des rencontres, des lieux d'expression ou de réflexion pour avancer sur ces sujets.

Au fantasme passé de l'humain moderne, taillé sur le modèle unique de l'humain industrialisé s'oppose aujourd'hui celui de la diversité, de la multiplicité et du modèle sur mesure.

Je reconnais dans notre discussion ces mêmes questions : ne peut-on pas concevoir nos solutions logicielles spécifiques ? Le libre ou l'open source peuvent-ils nous permettre d'éviter les stéréotypes ? Y'a-t'il des exemples de logiciels libres pour l'architecture ?

Il faut tout d'abord donner quelques précisions sur ce que l'on appelle « logiciel libre ». Le logiciel libre est théorisé, inventé et pratiqué par l'informaticien Richard Stallman au début des années 1980. Selon sa définition, le logiciel libre comprend 4 libertés, qui permettent d'exécuter le logiciel, d'étudier son code source, de le redistribuer, et de redistribuer des versions modifiées. Ces 4 libertés ne se retrouvent absolument pas dans les logiciels dominants de la profession type ArchiCAD. Google Sketchup est certes « gratuit » dans sa version de base, mais gratuit ne veut pas dire libre. On ne peut pas modifier Sketchup, et si l'on veut passer sur une version « avancée », il faut payer. Certes c'est une licence qui ne coûte pas très cher, mais une agence ne peut pas modifier Sketchup suivant ses propres besoins.

À ma connaissance, mis à part Blender mais qui est plus un logiciel de 3D pour la création ou pour l'objet, je ne crois pas qu'il y ait beaucoup de logiciels d'architecture placés sous licence libre. C'est effectivement un problème parce que cela crée une dépendance de la profession vis-à-vis d'acteurs privés. Cette fermeture des codes sources signifie aussi que même si l'on peut créer un plug-in, une extension à un programme propriétaire, dès lors qu'il est dépendant d'une base qui est privée – privatisée, ce plug-in peut être stoppé par l'éditeur, buggé, ne plus fonctionner, etc. Il faudrait inviter les architectes, les syndicats ou les représentants des architectes à s'investir davantage dans le logiciel libre pour que les agences n'aient pas à dépenser des dizaines de milliers d'euros dans des logiciels qu'ils ne peuvent même pas modifier. Si l'on avait une suite logicielle sous licence libre en architecture, adaptable vis-à-vis de tel client ou de telle agence, on aurait fait un gros progrès pour échapper aux médiations opaques que l'on évoquait au tout début de l'entretien ●

Entretien entre Anthony Masure et Joanne Pouzenc, enregistré en Septembre 2019.

