



Book Review “CODE/SPACE: Software and everyday life”

S. Widmer

Université de Neuchâtel, Switzerland

Correspondence to: S. Widmer (sarah.widmer@unine.ch)

Kitchin, Rob and Dodge, Martin: CODE/SPACE: Software and everyday life, Cambridge MA, MIT Press, 304 pp., ISBN-13: 9780262042482, € 28.99, 2011.

“Living beyond the mediation of software means being apart from collective life”. Relevé dans la préface de l’ouvrage de Rob Kitchin et de Martin Dodge, ce constat paraît à première vue extrême. S’ensuit pourtant une éloquente démonstration des activités et services auxquels nous renoncerions, si nous décidions de débarrasser nos existences de l’influence du code informatique. Le ton est donné, les quelques exemples cités (approvisionnement en eau, en électricité, services de santé, mobilité, etc.) suffisent à illustrer à quel point le code informatique sous-tend le fonctionnement de nos quotidiennetés.

Dans la lignée des géographes Nigel Thrift et Shaun French (2002) ou de Stephen Graham (2005), Rob Kitchin et Martin Dodge nous offrent ici une riche réflexion sur les spatialités que le “code” contribue à façonner. Le point de départ de Kitchin et Dodge est qu’un logiciel n’est pas quelque chose d’anodin (“*software matters*”). Le software a de l’importance parce qu’il a une capacité d’action dans notre monde; il “travaille” de façon autonome, traitant des capta¹ et réagissant à ceux-ci de façon instantanée: “ce code-barres correspond à tel prix”, “ce mot est mal orthographié”, “cette empreinte digitale ne permet pas l’accès à ce bâtiment”. Pour les auteurs, un logiciel a de l’importance parce que, au travers de son action, il modifie les conditions par lesquelles la société, l’espace et le temps se produisent.

L’ambition de Kitchin et Dodge est de documenter et de problématiser les façons par lesquelles les logiciels façonnent les nouvelles spatialités de nos quotidiennetés. Ils distinguent à cet égard les “*coded spaces*” (des espaces

façonnés par le code, mais dont l’existence ne dépend pas complètement de celui-ci) des “*code/space*”, dont la spatialité repose entièrement sur le bon fonctionnement du code (si ce dernier “plante” alors l’espace en question perd son sens). Les chapitres 7, 8 et 9 de l’ouvrage décortiquent trois de ces spatialités codées: les espaces du voyage aérien, de la maison, et de nos consommations.

“*Code/Space: Software and everyday life*” est structuré en quatre grandes parties. Faisant suite à une copieuse introduction, la partie II (“*The Difference Software Makes*”) est principalement consacrée à la présentation de l’approche théorique et conceptuelle retenue par les auteurs. En privilégiant une approche “ontogénétique” de l’espace, vu comme constamment en état de devenir, Kitchin et Dodge ancrent leur recherche dans la sociologie de Latour – on pense à sa définition “*performative*” du social (2007) – ainsi que dans la théorie non-représentationnelle de Nigel Thrift. Dans cette perspective, le code informatique produit une action dans le monde et contribue à constamment créer de la spatialité. Dans cette deuxième partie, Kitchin et Dodge introduisent également le concept de gestion automatisée (“*automated management*”) qu’ils comprennent comme une nouvelle forme de gouvernamentalité rendue possible par le code. Dans ce passage de l’ouvrage, le propos se problématise autour des notions de pouvoir et de surveillance; il s’appuie sur la pensée foucauldienne pour dévoiler la logique disciplinaire du code, imposant ses règles de conduite dans le monde. Et les auteurs d’évoquer la problématique fondamentale soulevée par Stephen Graham (2005) autour de la notion de “*software-sorting*”: le software classe, trie et distingue; il a le potentiel de créer ou de renforcer toute une série d’inégalités socio-spatiales, délimitant des droits, des vitesses et des accès différents.

La troisième partie de l’ouvrage (“*The Transduction of Everyday Spatialities*”) illustre ce que sont ces spatialités médiatisées par du code informatique. Trois exemples sont

¹Les auteurs ne parlent pas de données (data), mais de “capta” pour décrire les informations que captent, traitent et analysent les logiciels.

évoqués: les espaces du voyage aérien – succession de "code/space" tombant sous le coup d'une gestion automatisée; l'espace de la maison – connecté à de vastes réseaux codés; et l'espace de notre consommation – le code transformant aussi bien nos pratiques d'achat que la nature des produits que nous consommons.

Dans la quatrième partie du livre ("*Future Code/Space*"), Kitchin et Dodge évoquent les développements auxquels nous pourrions nous retrouver confrontés à l'avenir: ceux de l'"*everyware*", une informatique ubiquitaire se fondant dans notre environnement et régulant nos existences en se laissant oublier (l'exemple des habits munis d'une puce RFID, communiquant à la machine à quelle température ils doivent être lavés illustre bien cette autonomie des objets codés, travaillant entre eux sans supervision humaine). Le chapitre s'achève sur l'évocation des risques associés à l'"*everyware*": la dimension surveillante – et la possible dérive totalitaire – d'une société qui, en générant, analysant et stockant continuellement des données, n'oublie jamais rien.

L'ouvrage conclut par un chapitre intitulé "*A Manifesto for Software Studies*" dont le message est double. Les auteurs expriment, d'une part, la nécessité de mener des recherches plus approfondies sur les complexes réseaux d'acteurs intervenant dans la composition du code, ainsi que sur les effets du code sur nos pratiques quotidiennes. Mais, au-delà de cette ambition intellectuelle, Kitchin et Dodge plaident également en faveur de l'introduction d'une éthique de l'oubli dans les pratiques de programmation du code (cet argument fait notamment écho aux discussions actuelles autour du développement de "*privacy-enhancing technologies*". Voir à ce sujet: Goold, 2009).

L'ouvrage de Kitchin et Dodge constitue un compte-rendu très complet sur les spatialités que le code informatique contribue à façonner, ainsi que sur les risques associés à sa logique de gestion automatisée (en termes de surveillance, de contrôle et de "*software-sorting*"). L'une des grandes forces du livre est pourtant de ne pas sombrer dans le "*techno-pessimisme*" (Berthoud, 2002) le plus alarmiste, mais de souligner également tout ce que le code rend possible: les facilités qu'il occasionne, les mobilisations et les créations qu'il permet. Reconnaître l'ambivalence de l'informatisation de nos quotidiens, c'est également reconnaître la grande diversité et complexité du code, s'appliquant à des domaines très divers et comportant des risques variés (peut-on d'ailleurs décrire en des termes similaires un logiciel de traitement de texte et un logiciel "intelligent", profilant systématiquement ses utilisateurs en fonction de leurs requêtes passées?).

Souligner l'ambivalence du code, c'est aussi montrer que celui-ci est loin de tout déterminer. Comme les auteurs nous le rappellent à plusieurs reprises, les spatialités façonnées par le code informatique (ou dépendantes de celui-ci dans le cas des *code/space*) ne sont jamais complètement déterminées par celui-ci: des acteurs peuplent ces espaces codés, les pra-

tiquent et les éprouvent de multiples façons. Ainsi, la spatialité sans cesse réactualisée de ces espaces, se constitue de manière relationnelle et négociée. Dans le même ordre d'idées, Kitchin et Dodge suggèrent qu'un code/space est toujours un peu incomplet. Cette incomplétude – révélée au grand jour lors de "bugs" du système – nous rappelle que le code informatique est une production, somme toute, très contingente: un vaste "agencement" de connaissances, de subjectivités, de manières de faire, de sources d'autorités et d'intérêts – parfois divergents.

Bien que Kitchin et Dodge évoquent à diverses reprises le caractère contingent et négociable des espaces codés, la pertinence de ce constat peine toutefois à clairement émerger du reste du propos. D'abord, les spatialités *incomplètes* ou *sujettes à négociation* qu'évoquent les auteurs, ne s'accrochent pas toujours très bien d'un cadre théorique faisant explicitement recours au concept foucauldien de pouvoir disciplinaire – que Foucault définit comme réglementant tout, ne laissant rien échapper (2004). Enfin, on peut regretter que ces aspects ne soient que peu étayés par des données empiriques. Dans les trois chapitres illustrant les espaces codés, rares sont les exemples de "bugs" révélant les faiblesses de ces systèmes, au fonctionnement moins bien huilé que l'on pourrait le penser. Rares sont également les comptes rendus des façons par lesquelles le travail qu'accomplit le code peut être négocié ou contourné. Une étude centrée sur nos quotidiennetés gagnerait peut-être à considérer que "*le quotidien s'invente avec mille manières de braconner*" (De Certeau, 1990 XXXVI).

References

- Berthoud, G.: L'horizon d'une surveillance omniprésente?, Revue européenne des sciences sociales, Cahiers Vilfredo Pareto, 155–168, 2002.
- De Certeau, M.: L'invention du quotidien I. Arts de faire, Paris, Gallimard, 1990.
- Foucault, M.: Sécurité, territoire, population: cours au Collège de France (1977–1978), Paris, Gallimard Seuil, 2004.
- Goold, B.: Building it in: the role of privacy enhancing technologies (PETs) in the regulation of surveillance and data collection, in: New Directions in Surveillance and Privacy, edited by: Goold, B. and Neyland, D., London, Willan Publishing, 2009.
- Graham, S.: Software-sorted geographies, Prog. Hum. Geog., 29, 562–580, 2005.
- Latour, B.: Changer de société, refaire de la sociologie, Paris, La Découverte, 2007.
- Thrift, N. and French, S.: The Automatic Production of Space, Transactions of the Institute of British Geographers, 27, 309–335, 2002.