Comment l'IA inverse **l'obsolescence programmée**

Par **Christophe Leroy**

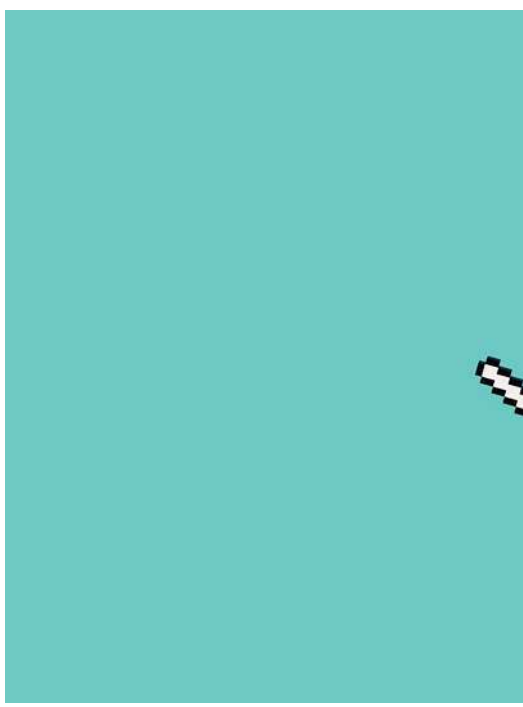
Ces derniers mois, les expériences de rétro-ingénierie assistée par l'intelligence artificielle se multiplient. Au point de faire reculer la date de péremption de nombreux logiciels?

Epurer l'interface d'une télévision de ses applications inutiles et de ses publicités. Ressusciter une imprimante sujette à l'obsolescence programmée, en créant un nouveau pilote. Déconnecter un aspirateur-robot du cloud de son fabricant, pour le faire tourner sur un serveur maison. Lancer des jeux vidéo actuels sur une ancienne console. Transformer un vieil iPad en assistant familial pour cuisiner. Depuis quelques mois, les expériences de rétro-ingénierie (*reverse engineering* en anglais) assistée par l'IA se propagent comme une traînée de poudre. Le terme désigne l'étude d'un produit ou d'un système existant, dans le but de déterminer son fonctionnement et, si besoin, de le modifier ou de le reproduire. La rétro-ingénierie existe donc depuis la nuit des temps dans d'innombrables domaines. Mais l'IA révolutionne potentiellement le temps et les moyens consacrés à la modification de nombreux logiciels ou appareils numériques.

Deux événements récents augurent de possibles bouleversements en la matière. Agacé par les prix exorbitants des licences, un ingénieur américain, Brandon Thomas, a publié début octobre une série de logiciels gratuits équivalant aux suites Adobe (Photoshop, Lightroom, Illustrator...), puis Microsoft Office (Word, Excel, PowerPoint...). Grâce à l'aide de l'IA Claude, ce résultat n'aurait nécessité que quelques mois de travail, même si plusieurs bugs persistent. Sur GitHub, célèbre plateforme collaborative pour la création de logiciels, un autre projet recourant notamment à l'IA est devenu particulièrement viral ces dernières semaines: *Reverse Engineer Anything* (REA). «Vous avez repéré une fonctionnalité dans une application que vous aimeriez intégrer à votre propre produit? Demandez à votre agent de l'étudier avec REA, exposent ses concepteurs. Cet outil peut analyser l'application sans accéder à son code source, expliquer le fonctionnement de la fonctionnalité, fournir des preuves et créer une version pour votre projet.»

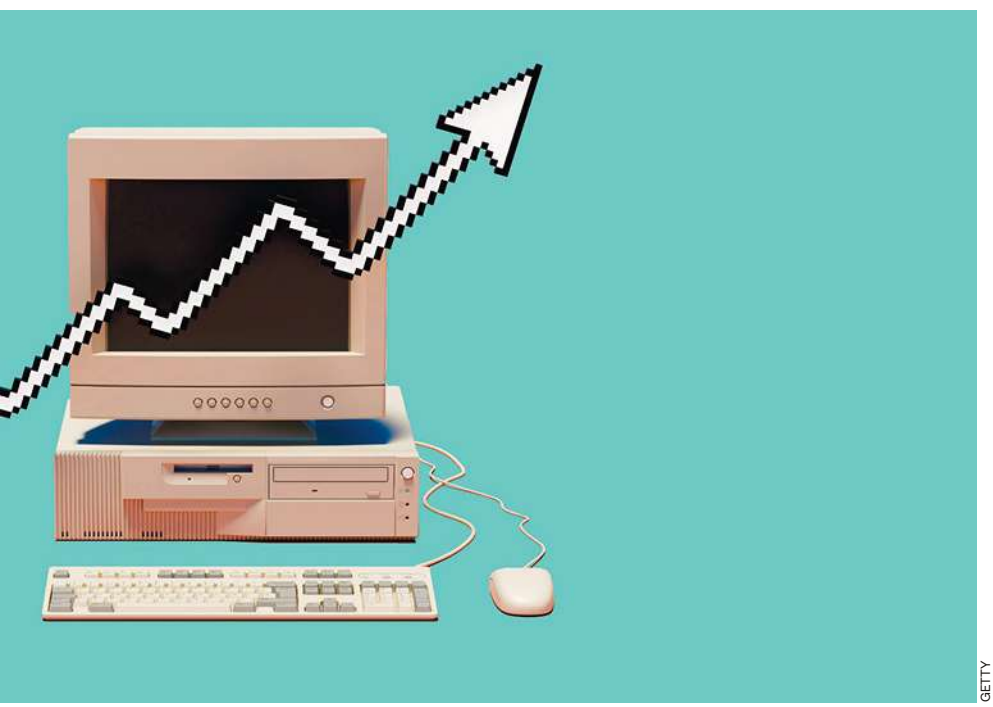
Contrer l'obsolescence logicielle

Dans certains cas, la philosophie consiste donc à démocratiser l'accès à certains logiciels –sans nécessairement copier le code, comme l'affirme par exemple Brandon Thomas. Dans d'autres, à outrepasser les limites d'un appareil, telles qu'imposées par son fabricant. «Pour le grand public, le plus simple est d'abord de demander à une IA si une solution existe pour un problème précis, expose Pierre Schaus, professeur d'informatique à l'UCLouvain. Encore la semaine dernière, l'IA m'a permis de trouver et d'installer en quelques minutes un logiciel gratuit pour retrouver les photos sur une carte SD formatée. La recherche et les tests nécessaires m'auraient sans doute pris des jours sans l'IA. Mais il est vrai qu'elle peut aller beaucoup plus loin. Moyennant un prompt plus pointu, vous pouvez lui demander de réécrire un code source, en décompilant les fonctionnalités d'un logiciel, à moins que son écriture soit protégée avec une clé cryptographique. C'est comme si vous aviez à votre disposition une armée de développeurs essayant de réparer votre appareil, pour un coût dérisoire.»



L'IA révolutionne le temps et les moyens consacrés à la modification de certains logiciels ou appareils numériques.

L'IA pourrait-elle devenir l'alliée d'une vertueuse lutte contre l'obsolescence programmée? C'est la question que posent désormais une série d'observateurs avisés. Si elle ne peut résoudre l'obsolescence matérielle, elle semble capable de redonner une seconde vie aux *hardwares* abandonnés par leurs concepteurs, ce qui incitait les utilisateurs à jeter leur appareil pourtant techniquement fonctionnel. «Les IA vont vraiment aider en ce sens, affirme Pierre Schaus. Il serait d'ailleurs opportun que certains équipements bénéficient d'un label autorisant la reprogrammation d'un *hardware*, afin de le personnaliser.



GETTY

Ce n'est sans doute pas une logique dans laquelle un acteur comme Apple voudrait rentrer. En revanche, cela pourrait constituer un argument marketing pour certains fabricants dans des systèmes plus ouverts.»

L'opportunité de triturer de la sorte un appareil ou un logiciel dépend non seulement de son ancienneté, mais aussi des connaissances de son utilisateur. Une telle opération rompt logiquement la garantie du fabricant. L'IA risque par ailleurs de produire un code dans lequel il sera d'autant moins possible de détecter les erreurs pour un réviseur humain. «L'aspect cybersécurité est un bémol, reconnaît Pierre Schaus. Quand vous installez un logiciel à un moment précis, il est possible que des failles de sécurité apparaissent par la suite, comme ce fut le cas la semaine dernière avec PHP. Là où un fabricant effectue des mises à jour pour y remédier, ce ne sera pas nécessairement le cas sur un logiciel amateur.»

«C'est comme une armée de développeurs capables de réparer un appareil, pour un coût dérisoire.»

Un «nouveau Far West»?

Créer un logiciel est une chose; le maintenir dans un environnement fonctionnel et sécurisé en est une autre. Plutôt que d'entrevoir la fin d'un modèle économique reposant notamment sur le paiement d'onéreuses licences logicielles, certains experts s'attendent plutôt à assister à une avalanche de clones gratuits, mais de mauvaise qualité. La rétro-ingénierie assistée par l'IA deviendrait alors un «nouveau Far West», comme la qualifie le site spécialisé 404 Media. Néanmoins, Pierre Schaus estime qu'il y a là matière à voir éclore un écosystème encadré. «A l'image des magasins réparant des téléphones, certains pourraient se spécialiser dans la reprogrammation et la personnalisation de vos *hardwares*.»

En Europe, la directive du 23 avril 2009 concernant la protection juridique des programmes d'ordinateur encadre certains grands principes de la rétro-ingénierie. Si elle autorise la décompilation d'un logiciel (c'est-à-dire la reconstitution de son code source à partir d'un programme exécutable) sous certaines conditions, celle-ci ne peut servir à la mise au point d'un «programme d'ordinateur dont l'expression est fondamentalement similaire.» En revanche, dans un arrêt rendu en 2021, la Cour de justice de l'Union européenne a considéré que l'acquéreur d'un logiciel pouvait le décompiler afin d'en corriger certaines erreurs pénalisant son bon fonctionnement. La légitimité future de la rétro-ingénierie assistée par l'IA dépendra donc non seulement de ses usages, mais aussi de l'interprétation du droit. ●