

Petits monstres

PAR JEAN-PIERRE CAHIER

Simple programmes, anecdotes souvent porteuses de gag, de démonstration, prouesses de technique ou de provocation, les virus informatiques ne sont pas tous méchants. Jusqu'ici, dans leurs manifestations médiatiques et réelles, ils ont affiché plus de virulence qu'accompli de méfaits. Certes. On pourrait même relativiser le phénomène et remarquer qu'ils ont surtout offert à des individus l'occasion de montrer, de démontrer, de rappeler qu'ils existent, qu'ils sont de bons spécialistes. Le discours dominant marginalise les auteurs et minore leur responsabilité, sans remarquer que ces "déviant" font eux-mêmes partie d'une catégorie sociale, celle des programmeurs, remise confusément en cause par l'automatisation, avec le génie logiciel. Ce serait déjà une raison de réorienter l'analyse : peut-être le problème des virus ne fait-il que commencer.



Une étude plus approfondie amène d'autres sources d'inquiétude. Si les virus restent démonstration, que montrent-ils ? Ils "montrent" un coin de ce qui ne se laisse encore voir qu'avec peine, un "monstre" au sens étymologique du terme (puisque "monstre" vient de "montrer"). Une hypothèse est que le phénomène "virus informatique" pourrait prendre dans l'avenir de nouvelles formes, plus pathologiques encore, si l'on n'y prend garde.

Pourquoi ? Nous voyons aujourd'hui deux types de raisons qui se recoupent dans un unique phénomène inquiétant et se conjuguent dans la (résistible ?) ascension du discours "biologique". La première raison est que le logiciel, au plan macroscopique, a changé. La seconde nous obligera à descendre au plan microscopique pour voir ce que le virus, ce "logiciel comme un autre", a à nous dire.

Sur fond d'"organisme"

Toile de fond sur laquelle apparaissent aujourd'hui les virus, le logiciel a changé. Il se développe en ensembles toujours plus grands, toujours plus communicants (par le développement des standards, les réseaux locaux, territoriaux, internationaux, les échanges de disquettes). "Il" ne se développe pas tout seul : çà et là, dans les grandes entreprises, les institutions de recherche, les états-majors militaires, il existe des hommes, organisateurs, coordinateurs, normalisateurs, responsables du "système d'informations", du "système de décision"... qui ont pour (dur) métier d'intégrer ces systèmes informatisés. L'accent est mis sur la modularité et la réutilisation des pièces d'un puzzle logiciel, et ce faisant, des procédures et des savoir-faire humains qu'ils encapsulent.

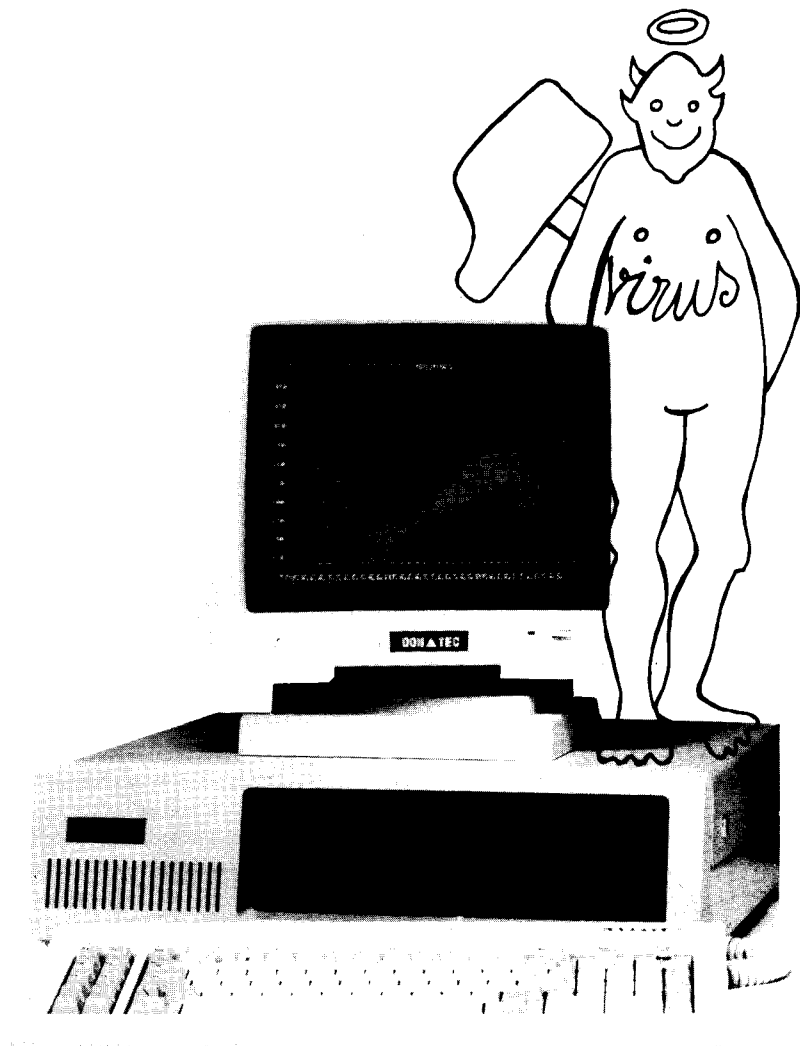
L'essor de la micro-informatique a peu modifié le phénomène : le volume de logiciels de télétraitement, de bases de données, d'aide à la décision qui communiquent entre eux dans les grandes organisations ne cesse de croître. Dans la conception même des ateliers de génie logiciel, utilisés pour fabriquer industriellement cette quantité croissante de programmes,

modulaires, réutilisables, interfaçables, la préoccupation d'intégration tient la première place. Dès lors qu'on veut mettre *tout* le système d'information, de décision et de communication sur informatique, ce qui est l'utopie des partisans de l'intégration, les jeux sont faits et la boucle est bouclée : c'est la fuite en avant dans "l'intégration ou le chaos".

L'intégration, et le chaos en prime

Cette utopie du réseau "tout-intégré" n'est pas seulement propagée par la quasi-totalité des fabricants d'"ateliers de génie logiciel intégré", à commencer par les Américains qui traversent l'Atlantique avec dans leur attaché-case les schémas pyramidaux de cette intégration tous azimuts : système stratégique/système de décision/système d'informations, référentiels (*repository*)... Tout communique dès la conception et la fabrication des programmes, et même dans l'utilisation de l'informatique. On retrouve aussi cette exaltation de l'intégration au cœur de l'offre de tous les grands constructeurs : Bull, IBM, Hewlett-Packard, Siemens. Tous vendent, en même temps que des ordinateurs et des "solutions", l'idée que des pans entiers de l'entreprise vue comme système (production, stocks, ordonnancement, études, méthodes, gestion, décision...) doivent être intégrés par l'informatique avec force réseaux. Et dans la réalité, c'est ce qui se passe. Le phénomène déborde le cadre fermé de l'entreprise, avec les réseaux territoriaux, les multiples services informatisés qui se délocalisent avec les EDI, les RVA, la télématique, les grands réseaux mondiaux comme Swift dans le domaine financier. A travers le réseau, un "organisme" systémique se met en place.

Le potentiel de représentation des logiciels de l'informatique classique, ou le réseau comme simple véhicule de données et de messages, sont des outils artificiels, chacun de leur côté maîtrisables. Pris séparément, ils donnent peu de prise aux virus. Réunis dans une intégration poussée, ils forment alors un complexe *d'une autre nature que la somme des parties*¹. L'outil devant lequel on est cède la place au système *dans* ou *sous* lequel on est. Justifiée ou non, cette mutation insensible a touché progressivement beaucoup d'organisations dans les années 80. Il suffit de regarder l'évolution des grands modèles et des architectures de réseau (OSI, SNA) – à la fois squelette, musculature et tissu nerveux mondial des grands ensembles logiciels-réseaux intégrés –, mastodontes qu'il ne reste plus qu'à habiller localement d'une fine, mais coûteuse, peau de logiciel d'application. IBM va même plus loin en proposant directement une architecture d'application "unifiée" (SAA, AUA) comportant une couche d'accueil, qui aura probablement valeur de standard, pour les ateliers de génie logiciel intégrés et les dictionnaires du système d'information (*repository*). L'utilisateur d'un réseau appréhende difficilement cette lourde infrastructure du logiciel de base, laquelle sait se faire "transparente", comme on dit curieusement dans le jargon pour désigner l'opacité maximale. Il ne perçoit pas de manière consciente pourquoi le glissement a lieu. Pourtant, un beau jour, il se rend compte qu'il a affaire à un ensemble plus puissant : qu'il n'est plus *devant*, mais *dans*.



La transformation est davantage qu'un effet d'échelle. Née du besoin de répondre d'une certaine façon à des situations complexes, l'intégration massive par les réseaux conduit à déporter le complexe dans la machine. Davantage de données, davantage de procédures, davantage de savoir-faire, de part dans la décision et la responsabilité, de connaissances. Davantage de circulation et de brouillage des responsabilités. Les réseaux, comme les récursions ou la prise en compte du temps dans le logiciel, se présentent alors comme des "facteurs de liberté" pour la représentation. Car le maillage ou l'étoilage existent sous forme de lignes, de protocoles au sens classique, géographique, de télécommunication ; ils existent aussi, comme structure, de plus en plus au sein du logiciel lui-même, par exemple dans les modèles relationnels des bases de données, ou les correspondances tous azimuts que permettent les hypertextes. En théorie, ils créent un espace immense de liberté : on ne peut prévoir toutes les relations qui peuvent se faire grâce à eux. Par suite, la multiplication des connexions logiques au sein du puzzle logiciel-réseau crée à son tour de l'indécidable. Elle acquiert les caractéristiques d'un phénomène naturel. Là est le terreau des virus et des vers, mais aussi d'autres phénomènes comme les "étreintes fatales" des réseaux de données ou les "boules de neige", qui ont affecté ces dernières années de grands réseaux comme Transpac.

Certes, l'ensemble est objectivement constitué de parties logiques, rationnelles, prévisibles. Mais qui osera prétendre que l'ensemble, dans la pratique, est encore garanti totalement logique, rationnel, prévisible ? Tel le catalyseur, un petit élément introduit par jeu, par malveillance ou par hasard, peut provoquer des

¹ Voir à ce sujet L. Sfez : Critique de la communication – Seuil, 1988.

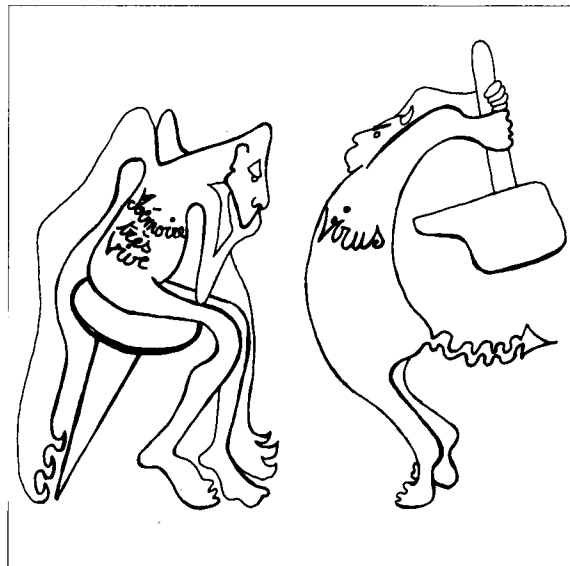
bifurcations, des saturations, des implosions, des conséquences sans commune mesure avec sa taille, des états "loin de l'équilibre". Les paramètres déclencheurs peuvent échapper aux plus louables efforts de conception, car la conception des systèmes complexes ne sait pas maîtriser les états "loin de l'équilibre". Tout juste sait-on les simuler dans certains cas, les provoquer... ou les subir. Les apologistes des phénomènes chaotiques² peuvent se réjouir : bien nommés, les phénomènes auto-organisés ne les ont pas attendus pour s'automanifester sur le terrain industriel.

La montagne accouche d'un éléphant

Le logiciel a changé, donc. Il a changé sous le feu de ce qu'on nomme la "problématique de la complexité". On n'a guère vu d'autre solution (mais en a-t-on vraiment envisagé d'autres ?) que de recourir massivement à la machine ; et de fil en aiguille, exclusivement à la machine : l'intégration ou le chaos. Alors on "intègre", on fusionne la représentation et le réseau. Mais la fusion apparaît alors susceptible d'avoir ces états limites où "on ne comprend rien à ce qui se passe", comme à la Bourse en octobre 87. L'intégration se révèle douée de chaos. On peut se dédouaner en constatant qu'aujourd'hui, heureusement, ces phénomènes se révèlent très marginaux, qu'ils prêtent encore le plus souvent à sourire (quoiqu'on se soit rendu compte, déjà, que les virus peuvent faire pas mal de ravages). C'est parce que l'intégration est encore très partielle. Toute avancée de l'intégration ne pourra qu'amplifier les phénomènes de type "virus".

Virus, greffe, organisme, squelette, système nerveux, pathologie... Il y a manifestement abus de termes renvoyant au vivant, de métaphores organiques. Le virus informatique, qui est toujours une fabrication intentionnelle et artificielle, est assurément mal nommé ; ses deux particularités étant de se dupliquer, mais aussi de voyager, il conviendrait de proposer des synonymes moins biologiques, tels que "pèlerin", utilisé par J.M. Truong³. Mais l'analogie biologique est révélatrice : un virus n'existe que sur fond d'organisme. Les ensembles intégrés logiciels-réseaux se rapprochent de plus en plus, par certains de leurs traits, d'organismes. D'ailleurs, le problème n'est peut-être pas tant le virus, dans la mesure où on peut espérer prémunir le système contre lui (en attendant de se laisser surprendre par un virus plus puissant), que la qualité de comportement "naturel", "d'organisme" qui se révèle à l'occasion de son apparition, qui laisse présager d'autres tempêtes.

Une petite créature, par exemple une souris, n'attaque pas l'homme. Supposons qu'elle trouve un terrain favorable dans un lieu que nous considérons habituellement comme une montagne pacifique, et que brusquement cette montagne se mette à bouger en tous sens, au risque de créer des accidents. Nous n'apprenons pas grand-chose en focalisant l'attention sur la souris. Tout au plus lui enverrons-nous un chat, quoique tous les chats du monde n'ont jamais réussi à faire disparaître l'espèce des souris. Nous apprendrons bien plus en observant la montagne, surtout si nous découvrons alors que cette montagne cache... une sorte d'éléphant (en tous cas suffisamment "éléphant" pour



avoir des réactions violentes en présence d'une souris !). C'est bien sûr l'éléphant – animal utile, mais néanmoins coléreux – qu'il faut tenir à l'œil au premier chef.

Tant qu'on ne comprend pas qu'il faut dépasser le stade du virus-anecdote, et se tourner vers ce qu'il révèle de l'ensemble, c'est comme tenter de soigner une maladie sans connaître le malade. Or nous connaissons encore peu de chose du logiciel dans ses aspects macroscopiques. Il faudrait développer, dans les années à venir, une approche scientifique de cette vaste question, s'appuyant aussi sur les sciences humaines, et réinterpréter à cette lueur des phénomènes comme les virus, les pannes, mais aussi l'architecture du logiciel et le génie logiciel.

Du microscope au microscope

L'aspect microscopique des virus est aussi révélateur. En d'autres termes, nous intéresser à l'éléphant ne nous dispense pas de regarder d'un peu plus près la souris qui le démange. Car si le "virus" est un programme, rédigé dans les langages de programmation habituels (assembleur, "C", lisp...), il présente déjà certaines caractéristiques, comme celle de pouvoir établir des connexions avec des programmes existants (logiciel de base ou d'application) qui n'ont pas été prévus pour cela. Le virus innove, en créant une relation non prévue par les concepteurs des autres programmes. Le virus utilise pour cela la "connaissance" de certaines normes d'interface qu'il est susceptible de rencontrer, de certains "points d'entrée" des logiciels d'exploitation ou de télécommunications, normalement connus des seuls développeurs. Pour que le virus soit efficace, il faut que son programme inclue une partie de cette connaissance.

En particulier, placé dans des conditions différentes (mis en présence de tel ou tel programme sur telle ou telle configuration de machine), le virus est amené à effectuer des "choix" : se greffer sur ce nouveau programme, ne pas s'y greffer, rechercher des programmes appelants ou appelés, attendre le déclenchement d'une certaine instruction, tenir compte du jour et de l'heure... Là encore, ces situations et ces choix peuvent avoir été prévus par le concepteur du virus, ou ne pas l'avoir été. La situation se corse, si l'on considère que le virus peut contenir des beugues, donc des spécifications-lapsus échappant à son auteur (ce qui semble

² Voir à ce sujet

J.C. Perez,

J.C. Bertille : Une

nouvelle famille de

réseaux neuronaux :

le modèle

holographique "chaos

fractal" et ses

applications

industrielles. Actes

Convention IA - Paris

1989 - Hermès.

³ Voir le roman

Reproduction

interdite de Jean-

Michel Truong

(Editions Olivier

Orban, 1989) qui y

fait jouer un rôle

important à un

"pèlerin", système-

expert baladeur - et

travaillant pour une

bonne cause -, virus

futuriste capable

d'échapper à toute

poursuite en se

déplaçant sur

n'importe quel

réseau et en se

logeant sur n'importe

quel hôte informa-

tique de rencontre.

avoir été le cas pour le fort dévastateur ver d'Arpanet en novembre dernier). L'idéal du virus (qui est encore, heureusement, loin d'être atteint par les virus actuels) est de pouvoir réagir dans n'importe quel cas, de manière à préserver sa "chaîne d'action" possible. Par exemple, un virus prévu pour l'univers Macintosh ou Unix, s'il parvient, par le hasard des connexions, en environnement PC-Dos, ne sera en général pas capable de contaminer ce système d'exploitation. Mais il peut reconnaître qu'il est en terrain non favorable, et "s'enkyster" à un niveau de protocole compatible avec sa survie, en attendant que le hasard le remette en communication avec son univers de départ, où il pourra de nouveau sévir.

Les virus logiciels ont rarement atteint un tel niveau de sophistication, mais l'apparition de certains virus capables de rester souterrains pendant certaines phases de leur parcours montre que cette voie est aujourd'hui ouverte. La possibilité d'introduire des techniques d'IA cognitive dans les virus (encapsulation dans le programme de bases de connaissances sur l'univers des machines) renforce leur pouvoir d'autonomie (l'arsenal des procédés du génie logiciel qui visent à la *réutilisation* des programmes va déjà dans ce sens). Par ailleurs, les techniques d'apprentissage et l'auto-organisation, qui ont marqué des points dans le secteur de la recherche, permettront demain d'aller plus loin : le virus acquerrait une autonomie et une incontrôlabilité accrue, étant programmé pour se diriger seul dans n'importe quel environnement. Ces techniques seraient d'autant plus redoutables que, dans le même temps, les architectures des machines se modifieraient pour traiter rapidement des connaissances, voire des automates cellulaires (processeurs IA, machines à parallélisme massif, connexion-machines). Plus les machines permettront de traiter efficacement de l'IA ou des réseaux de neurones formels, plus les programmes pourront exploiter cette structure pour intervenir ou apprendre sur leur environnement. Ce fantasme de démiurge, qui anime les grands chercheurs, est sans doute aussi de ceux qui "font courir" les concepteurs de virus informatiques.

Les idées fondatrices de cette approche se retrouvent notamment chez les partisans du connexionnisme et de l'auto-organisation, qui disposent déjà de quelques terrains d'expérimentation dans l'image de synthèse, la simulation, la reconnaissance des formes et la modélisation neurobiologique. L'une des plus brillantes percées théoriques en la matière est celle de Varela, qui a travaillé sur la théorie des systèmes auto-organisés et la notion de "closure" (ou autopoïèse). Pour schématiser, Varela a posé les concepts théoriques de systèmes fermés, disposant en leur sein propre de toute l'information nécessaire pour interagir avec des environnements changeants, non prévus à l'avance. Il a recherché de quel type d'information de tels systèmes devaient disposer. Ces recherches intéressent, semble-t-il, moins les biologistes que certains informaticiens, lesquels croient acquérir ainsi, à bon marché, une clé de "quelque chose qui ressemble beaucoup au vivant". On pressent alors pourquoi le problème des virus, mis en perspective sur une telle évolution d'ensemble de l'informatique, ne ferait que commencer. Les virus actuels, grossiers dans leurs techniques de conception

— bien que déjà très embêtants — ne seraient que les ébauches de prochains virus plus durs, réponse "frankensteiniennne" à une éventuelle évolution tout aussi "frankensteiniennne" des systèmes informatisés.

Besoin de bases solides

Ces hypothèses, macroscopiques et microscopiques, ne sont pas sans liens. Prolongées, elles présentent un ensemble inquiétant. Elles resteraient à approfondir scientifiquement, car seule une approche méthodique et scientifique pluridisciplinaire pourra permettre de réellement comprendre, et de faire la part entre fantasmes et réalités. Soulignons l'urgence d'une telle recherche approfondie. Nous aurions besoin de marcher sur un terrain plus solide. Trop d'éléments nécessaires à l'analyse sont encore masqués, tabous, ("forclos", diraient les psychanalystes). Certaines disciplines scientifiques dont nous aurions besoin sont trop insuffisamment vulgarisées, et les milieux concernés comme les médias se préoccupent bien peu de favoriser ces connaissances et ces débats. Faute de quoi nous risquons de comprendre dix ans trop tard ce qui se passe aujourd'hui. C'est ce monde fort clos qu'il s'agit d'ouvrir, et les virus logiciels sont peut-être pour cela un bon prétexte.

La réaction d'amusement qui prévaut assez souvent par rapport aux virus informatiques (hors des milieux directement touchés) est positive, car elle montre que, au moins par l'humour, l'homme garde une maîtrise de la situation. Mais elle cache une gêne, l'appréhension diffuse que le phénomène ainsi révélé soit plus grave — et à notre sens il l'est. En tous cas, ces virus ont le mérite de montrer à un large public que l'informatique a changé, et bien changé, en dix ans. Pour interpréter cette transformation dans l'univers des informaticiens, il faudra faire appel aussi à de nouvelles catégories de pensée, venues d'un univers non technique. Quant à la biologie et aux sciences du vivant, tellement à la mode ces temps-ci, contentons-nous de leur emprunter un peu de vocabulaire et quelques comparaisons, sans entretenir trop d'illusions sur la solidité de paradigmes que certains voudraient communs. ■

