

Le courrier des lecteurs

À propos du "bogue de l'an 2000", n° 545, mai 1999

— J.-M. BARDIN (72)

La lecture de l'article d'Erik Egnell à propos de l'an 2000 dans le numéro de mai m'a plongé dans un sentiment à mi-chemin entre la stupeur et l'indignation. Après avoir hésité à répondre, je me suis résolu à le faire, non pas pour défendre mon statut de responsable de projet an 2000 dans une grande banque française, mais pour rétablir quelques vérités mises à mal dans cet article, qui pourrait inciter à un certain relâchement dans les efforts, pourtant encore insuffisants, engagés dans notre pays pour maîtriser les risques liés au bogue de l'an 2000.

- Le bogue de l'an 2000 n'est pas une invention d'informaticiens désœuvrés. C'est une réalité à laquelle beaucoup d'entre nous ont été confrontés ces derniers temps. Je n'en prendrai pour exemple que ces milliers de terminaux de paiement qui, au début de l'année 1998, refusaient les cartes de paiement à échéance 2000, car ils les pensaient périmées depuis quatre-vingt-dix-huit ans. Mais il y a quantité d'autres exemples plus discrets.

- La confusion sur les années se traduit par des erreurs dans des calculs ou des tris faisant intervenir des dates, une des données les plus fréquentes dans les programmes (90 % des programmes d'une banque manipulent des dates!). Les systèmes rentrent alors dans des séquences d'instructions jamais testées : au mieux, ils s'arrêtent, ce qui permet de détecter l'anomalie ; au pire, ils génèrent des erreurs sournoises, qui, non décelées immédiatement, sont très difficiles à corriger quand on s'en aperçoit bien plus tard.

- Ce n'est pas parce qu'un programme de prêts immobiliers sait depuis vingt ans traiter des échéances postérieures à l'an 2000 qu'il est vacciné contre le bogue. En effet, celui-ci peut affecter de manière différenciée chacun des types de dates traitées dans un programme : les dates concernant les échéances à venir peuvent ne pas poser problème, alors que celles correspondant au prochain appel de fonds peuvent en poser un.

- Les systèmes informatiques ne sont pas les seuls concernés. Tous les systèmes électroniques sont susceptibles d'être affectés par le bogue, quand bien même il n'y aurait pas de dates apparentes. Ce peut être le cas d'ascenseurs, dont l'électronique peut gérer les dates des visites de maintenance, et qui peuvent s'arrêter en cas de délai excessif depuis la dernière visite.

- L'informatique et l'électronique ont envahi notre monde ; toute notre société en dépend. Vous ne seriez sans doute pas fâché si votre centre d'impôts oubliait votre existence suite au déclenchement intempestif d'un programme de purge de données. Vous le seriez sans doute davantage, si vous appreniez que les installations de réanimation de votre hôpital ou que les dispositifs de surveillance des centrales nucléaires d'Europe de l'Est pouvaient ne pas fonctionner correctement au passage à l'an 2000.

- Il est vrai que le bogue de l'an 2000 est finalement un bogue comme un autre, comme on en rencontre tous les jours dans nos systèmes. Sa seule particularité est qu'il risque de frapper au même moment dans de multiples endroits, rendant insupportable

tables des problèmes qui, survenant isolément, pourraient être résolus, sans même que vous vous en soyez aperçu. De plus, c'est un problème systémique, c'est-à-dire que, même si vous avez tout fait pour adapter et vérifier vos systèmes, vous risquez d'être mis en difficulté par vos fournisseurs ou vos clients.

- Il est vrai que certains sont allés loin, trop loin, dans le catastrophisme millénariste, et ont fini par décrédibiliser le problème réel. Les médias à sensations avides de sujets croustillants, les cabinets d'avocats américains habiles à tirer des millions de dollars d'un fait sans importance, certaines sociétés informatiques peu scrupuleuses ont cherché à tirer avantage en rajoutant. Certaines entreprises en font même un argument commercial, en dénonçant l'insuffisante préparation de leurs concurrents. Mais il ne faut pas jeter le bébé avec l'eau du bain. Si le sujet est correctement traité, il peut être maîtrisé, mais, s'il est ignoré, il peut avoir dans certains cas des conséquences dramatiques. Comment réagissez-vous, quand vous apprenez, après l'incendie du tunnel du mont Blanc, qu'un rapport avait dénoncé quelques mois plus tôt la sécurité insuffisante de cette installation ?

- Il est vrai que les informaticiens portent la responsabilité originelle du problème et ils n'ont jamais cherché à le nier. Il peut sembler surprenant qu'ils n'aient pas plus anticipé un problème inéluctable connu depuis toujours. La raison en est simple : il y a trente ans, les premiers informaticiens ne pensaient pas que les systèmes et les programmes qu'ils fabriquaient

dureraient plus d'une dizaine d'années et ils ont surtout cherché à économiser les ressources mémoire et machine, alors rares et chères, en codant les années sur deux caractères. Dans les faits, les programmes ont duré beaucoup plus longtemps qu'ils n'avaient imaginé, et les nouveaux systèmes n'étaient jamais construits *ex nihilo* mais empruntaient souvent à la génération précédente, d'où la propagation jusqu'à nos jours de cet anachronisme.

- Mais pourquoi leur reprocher d'avoir alerté le reste de la société sur ce sujet, au prétexte qu'ils en étaient à l'origine? Fallait-il attendre que des problèmes graves se produisent pour réagir? Auriez-vous préféré qu'ils imitent les organismes de transfusion sanguine, qui ont attendu que des milliers de personnes soient contaminées avant de reconnaître leur responsabilité? ■

— J.-P. VIAL (65)*

Informaticiens de l'an 2000 Félicités? Remerciés?

Il paraît presque superflu de rappler dans ces colonnes la teneur du "bogue de l'an 2000". Il existe, dans les ordinateurs, comme dans les fichiers qu'ils utilisent, des limites à la capacité d'affichage des données. La capacité d'affichage est *finie*. Pour les dates comme pour les autres données.

Ce problème d'affichage est, par nature, un problème *informatique*, mais il n'est pas pour autant un problème uniquement électronique. Il existe, dans tous les instruments que nous utilisons, des limites d'affichage. Nombreux sont les affichages qui se font sur un nombre de chiffres limité et choisi arbitrairement. Par exemple, le kilométrage des voitures.

Dans les voitures des *années 50*, le compteur kilométrique comportait cinq chiffres, parce que la plupart des voitures de l'époque avaient une durée de vie estimée inférieure à 100 000 km. Depuis, on a construit des voitures plus solides, si bien que l'affichage du kilométrage à six chiffres s'est généralisé.

Il est certain que la précision d'un affichage kilométrique ne revêt pas une importance capitale. Certains indéliçats pourraient même y voir un avantage, en sous-évaluant l'ancienneté d'une voiture qu'ils comptent ainsi revendre à meilleur prix. Mais pourquoi les détracteurs du "bogue de l'an 2000" choisissent-ils justement cet exemple-là pour en minimiser les risques informatiques? Il y a tant d'autres situations dans lesquelles l'affichage d'une donnée revêt une importance bien supérieure, économique ou même vitale.

Les compteurs d'eau, de gaz, d'électricité ont eux aussi leurs limites d'affichage, et on peut raisonnablement supposer que les compagnies qui distribuent les ressources correspondantes sont hautement intéressées, financièrement, à une certaine exactitude des informations affichées.

Et qui monterait à bord d'un avion si, tous les 10 000 pieds, l'altimètre repassait par le zéro?

Admettons donc, pour l'instant, l'existence de limites de capacité d'affichage : une information affichée sur n chiffres s'affiche *modulo* 10^n .

Comment exprimer les dates?

Pour exprimer une date quelconque, il faut d'abord choisir une date de référence. La limite de capacité se traduit ici par un *nombre de jours*, compté positivement ou négativement, que l'on sait représenter à partir de la *date de référence*.

Selon les dates de référence choisies, et les nombres de jours permis, les systèmes informatiques peuvent tomber en panne à des dates diverses. Certains sont déjà tombés en panne en 1970 ou en 1999. D'autres tomberont en panne en 2011, 2038 ou 2071. Sans action préventive, beaucoup pourraient tomber en panne en 2000.

Seuls quelques spécialistes, comme les astrophysiciens ou les géologues, ont besoin de toutes les

échelles de temps, depuis le *big bang* originel jusqu'à la fin supposée de l'univers. Pour eux, la seule représentation du temps qui convienne est la représentation scientifique, que les informaticiens appellent représentation en *virgule flottante*, c'est-à-dire :

$$\pm n \cdot 10^{\pm p}$$

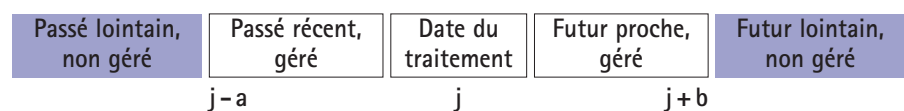
En informatique de gestion, comme dans la vie courante, on s'intéresse à une tranche de temps, une *fenêtre de temps*, plus ou moins centrée sur la date du traitement. Du moins, la fenêtre de temps contient-elle, en plus de la date du traitement, un certain laps (ou *horizon*) de temps dans le futur et un certain laps de temps dans le passé. C'est ce que figure le schéma ci-dessous.

Les paramètres (- a, + b) de la fenêtre de temps varient en fonction des besoins des applications informatiques. Par exemple, la gestion de prêts bancaires aura sans doute un horizon (+b) de quinze ou vingt ans. Pour gérer la date de naissance des emprunteurs, le passé (- a) pourra avoisiner un siècle.

Comme les traitements informatiques sont répétitifs, la date du traitement change continuellement. Il en résulte que la tranche de temps gérée se déplace uniformément vers le futur. L'informatique de gestion a besoin d'une tranche, ou fenêtre, de temps *glissante*.

Les paramètres (- a, + b) de la fenêtre sont choisis à un moment donné, en fonction des besoins, des habitudes, des normes. Il est possible que les choix effectués judicieusement à un moment donné ne soient plus valables ultérieurement. Mais pourquoi serait-ce la seule faute des informaticiens?

Par exemple, si les gendarmes viennent déranger une mamie de 106 ans, dans sa maison de retraite, pour la conduire à l'école *manu militari* comme une gamine de six ans, c'est d'abord et avant tout une erreur informatique (c'est "la faute de l'ordi-



nateur”). Mais on pourrait tout aussi bien soutenir que c’est une erreur de la médecine, qui s’évertue à prolonger la durée de vie de l’espèce humaine, alors qu’il y a à peine de quoi nourrir l’espèce tout entière.

Sans aller aussi loin – je risquerais de me fermer définitivement les portes des maisons de retraite – les informaticiens ne sont pas les seuls coupables. L’homme de la rue, les médias, notre culture commune nous font tous commettre la même erreur. On parle des *années 60* ou des *années 90*. Personne ne dit les *années 1960* ou les *années 1990*. Même à l’approche de cette fin de siècle et millénaire. Même sous la menace du “bogue de l’an 2000”.

Pour les dates, et pour les années en particulier, le problème – car c’en est un – prend tout à coup une acuité démesurée, parce que *tous* les systèmes risquent de tomber en panne *en même temps*, à la fin de l’année 1999.

J’entends au fond de la salle quelques cartésiens qui murmurent “mais n’y a-t-il pas une norme de représentation des dates?”

Ah! s’il n’y en avait qu’une! Dans le langage courant, il y a une norme de fait : les années s’énoncent avec deux chiffres. Dans les *années 60*, comme dans le langage courant, il y avait aussi une norme informatique de fait : les années se représentaient communément sur deux chiffres. La modestie ou le manque de vision des informaticiens de l’époque les empêchaient d’entrevoir que, non seulement leurs programmes, mais aussi, et surtout, leurs fichiers – voir plus loin – survivraient jusqu’en l’an 2000 et au-delà.

La norme ISO 8601 traite de la représentation des dates. Mais de quand date-t-elle? Notre calendrier lui-même date seulement d’un peu plus de quatre siècles, et son universalité est tout juste – est-ce un hasard? – contemporaine de l’informatique. Combien de temps restera-t-il en vigueur sous cette forme? Est-il seulement raisonnable d’affirmer, aujourd’hui, que le 1^{er} janvier 2100 tombera un vendredi, comme le prévoyait le calcul?

Le problème n’est pas tant de normer que de *renormer*. C’est un problème de volume.

Quelles sont les formes du “virus an 2000”?

Il existe *grosso modo* trois variantes du “bogue de l’an 2000”.

- Les *saisies* ou les traitements informatiques vérifient les années traitées. Aucune saisie ni aucun traitement n’ont été prévus au-delà de 1999. Dans le futur, le système informatique se bloque de lui-même.

C’est le cas de figure le plus “informatiquement correct”. Le système informatique n’a pas été prévu pour le futur, et il s’autoprotège. Néanmoins, s’il faut l’utiliser après 1999, on doit le corriger.

- Les années ne sont pas vérifiées, et le système les laisse passer sans contrôle.

C’est déjà un cas de laisser-aller informatique caractérisé. Le système peut calculer les résultats les plus aberrants. Et même peut-être les bons; allez savoir? Et, pour le savoir, il faut inspecter les programmes de fond en comble.

- Certaines années, comme 00 ou 99, prennent une signification spéciale qui interdit de les utiliser en tant que telles. Souvent, 00 signifie *donnée inconnue*, valeur non communiquée. 99 représente l’*infini*, l’absence de limite supérieure.

C’est le cas de figure le plus défavorable. Encore que, pour 99, le problème – s’il y a problème – est déjà réglé.

Dans tous les cas de figures, il faut inspecter les programmes de fond en comble, pour y débusquer l’une ou l’autre forme du “virus an 2000”, puis les décontaminer. Avant le réveillon, s’il vous plaît.

Les risques de dysfonctionnements sont nombreux.

• Erreurs de comparaison

Les erreurs de comparaison sont liées au traitement des années *modulo 100*. Par exemple, une carte bancaire dont l’année d’expiration est 00 risque d’être rejetée en 99, sous prétexte que 00 est inférieur à 99 – cela s’est déjà vu. Les erreurs de comparaison risquent de provoquer la

destruction, à tort, de données non expirées. Si une telle erreur n’est pas détectée rapidement, elle peut être très grave (destruction d’archives).

• Erreurs de tri

Les erreurs de tri, ou de classement, sont de même nature que les erreurs de comparaison, mais elles sont “statistiquement” moins graves, en ce sens que leur probabilité de passer inaperçues est plus faible. Par exemple, si notre mamie, née au siècle dernier, se retrouve mêlée à la jeune génération, et convoquée à l’école primaire, il reste exceptionnel qu’aucun *recoupement* ne permette de détecter le problème, quelque part dans le système informatique, avant que l’information erronée ne sorte du système. Mais pourtant cela arrive.

• Erreurs de calculs

Les erreurs de calculs sont sans doute les plus *spectaculaires*.

Statistiquement parlant, elles peuvent, comme les erreurs de comparaison, produire les résultats les plus aberrants et, comme les erreurs de tri, conduire à des problèmes plus facilement décelables par recoupements. Par exemple, si vous souhaitez la bonne année à l’un de vos proches, en l’appelant un peu avant minuit le soir de la Saint-Sylvestre, et si vous raccrochez le combiné à l’aube de l’année 2000, il n’est pas exclu – dans le cadre de cet article, bien sûr! – que le système informatique de votre opérateur téléphonique vous décompte un siècle de communication (entre fin 99 et début 00). Une telle erreur est possible, mais elle a peu de chances de passer inaperçue. Soit le système informatique respecte les règles de l’algèbre, et votre opérateur vous crédite d’un avoir de quelques millions d’euros, soit il ne s’embarrasse pas de considérations de signes, et vous facture froidement un montant équivalent. Dans un cas comme dans l’autre, c’est la banque qui saute.

Quels sont les risques pour l’entreprise?

Pour l’entreprise, le risque est avant tout une *désorganisation majeure* de l’outil informatique. Tant que l’on n’a pas inventorié et corrigé

les programmes contaminés par le virus, le mal peut frapper n'importe où.

Y a-t-il des applications informatiques plus sensibles que d'autres? Certainement.

La gestion actuarielle, celle des prêts bancaires, celle des contrats d'assurance sont des applications qui traitent le *long terme*, sur des fenêtres de temps par nature très larges. Pour elles, l'an 2000 a pu être un problème dans le passé; mais, en 1999, le problème est très certainement déjà résolu. Ces applications à horizon lointain peuvent être, éventuellement, sensibles à un allongement de la vie humaine, mais ce n'est déjà plus tout à fait le même problème. Comme ces applications traitent le long terme, leur correction est moins urgente: les informaticiens disposent d'un temps, sinon confortable, du moins suffisant pour apporter les corrections nécessaires.

À l'autre bout de l'échelle de temps, les applications qui contrôlent, en *temps réel*, des processus industriels sont finalement peu sensibles aux *dates*, même si elles sont sensibles aux *durées*. Il est possible qu'elles tombent en panne dans la nuit de la Saint-Sylvestre. Mais, si on les relance immédiatement, il y a de fortes chances qu'elles fonctionnent de nouveau, parfaitement, à l'aube de l'an 2000. Le risque pour l'entreprise est "simplement" de gâcher un cycle de production. Le risque économique est plus ou moins important (cas de la sidérurgie, par exemple), mais il reste limité. Existe-t-il un risque physique de mort d'homme? Sans doute, dans quelques cas extrêmes (nucléaire, hôpitaux), mais un tel risque semble, là aussi, très limité.

Mon opinion est que les applications informatiques les plus sensibles au "virus an 2000" sont celles qui traitent le *moyen terme*, comme la planification, la gestion des stocks, la distribution, les réservations de places de transports. La raison en est que le virus n'a pas encore eu l'occasion de se manifester dans ces applications. Si le virus apparaît, ce sera vers l'automne 1999. Or, corriger ces

applications, qui traitent le moyen terme, risque d'être *plus long* que leur horizon de temps. Les incidents risquent de survenir plus vite que les corrections ne seront rendues disponibles. Les programmeurs chargés de corriger pourraient buter sur un *mur du temps*, comme les ondes sonores butent sur le mur du son. Les services informatiques pourraient se retrouver engorgés, étranglés, par une cadence d'incidents supérieure au rythme des corrections.

Industriels qui travaillez en *flux tendu*, craignez l'an 2000! Ce sont les relations avec vos fournisseurs qui risquent d'être tendues.

Quelles corrections appliquer?

Bien entendu, inspecter un programme et le corriger pour le rendre "compatible an 2000" ne constitue pas un exploit hors de portée. Mais alors, pourquoi ne pas l'avoir fait au cours des dernières années?

La difficulté tient au *nombre* de programmes, à la méthode et au cadencement de l'opération.

Tandis que le comportement d'un seul programme est parfaitement déterministe et, en principe, reproductible, le *nombre* des programmes et leur combinatoire dans les systèmes informatiques rendent aléatoire le comportement du système informatique tout entier.

La *méthode* cartésienne suggère d'appliquer la norme: puisque les années s'expriment sur quatre chiffres, modifions les programmes pour qu'ils traitent des années à quatre chiffres.

Le problème, c'est qu'il ne suffit pas de modifier les *programmes*. Il faut aussi et surtout modifier les *données*, c'est-à-dire les fichiers informatiques. Un programme traite rarement une seule date. Il en traite communément plusieurs, dans plusieurs fichiers. Disons trois fichiers, pour donner un ordre de grandeur réaliste.

Les fichiers servent d'interfaces, de structures d'échange entre programmes. Lorsqu'on modifie un fichier, pour y changer la structure des dates, la modification concerne donc plusieurs programmes. Au

moins deux. Et, de plus en plus, les *bases de données* prenant progressivement la place des *fichiers*, les données concernent un nombre croissant de programmes. Lorsqu'on modifie la structure d'une base de données, la modification concerne couramment des centaines de programmes.

Si bien que la modification de notre programme d'origine peut maintenant concerner des dizaines d'autres programmes *connexes* qui échangent des dates avec le premier programme. Et, comme il serait dommage de s'arrêter en si bon chemin, autant corriger aussi ces programmes connexes, liés au premier, qui eux-mêmes sont liés chacun à quelques dizaines d'autres programmes connexes, etc.

En l'absence de bases de données, on pourrait encore espérer trouver des sous-ensembles de programmes connexes *disjoints* dans le système informatique, ce qui simplifierait notre problème. Mais il ne faut pas se faire trop d'illusions: les bases de données sont devenues omniprésentes, et le système informatique est ainsi devenu un seul ensemble connexe de programmes.

Notre méthode cartésienne, si rassurante au départ, nous conduit en fin de compte à un problème dont la complexité croît, par récurrence, de façon exponentielle, et est seulement limitée par la taille du système. Véhiculé par les flux de données entre programmes, le "bogue de l'an 2000" se diffuse dans le système informatique comme un virus dans un organisme vivant. La correction consiste maintenant à décontaminer tout le système. Il faut éradiquer le virus.

L'interaction entre les programmes et leurs données est si forte que c'est bien la *survivance des données*, dans les fichiers puis les bases de données, qui explique la survivance des programmes dans lesquels on exprime encore aujourd'hui les années *modulo 100*.

Ce n'est rien de modifier un programme. Par contre, c'est une toute autre affaire de modifier les fichiers ou les bases de données, de façon *synchrone*, dans tout un centre infor-

matique ; ou, qui plus est, entre centres informatiques partenaires (téléinformatique). Pour donner quelques ordres de grandeur, disons qu'il est courant, dans les grands systèmes informatiques, de compter les programmes par milliers et les fichiers par dizaines de milliers.

Par ailleurs, il semble excessif d'imposer, en tant que norme, des années à quatre chiffres. C'est, d'une certaine façon, nier la continuité du temps. Nulle application informatique courante n'a besoin de traiter tout le calendrier depuis son origine en 1582. Ce n'est pas l'an 2000 qui crée le besoin de normer les années. L'an 2000 est, tout au plus, une occasion de le faire. Mais le moment n'est sans doute pas le mieux choisi.

Les partisans des années à quatre chiffres arguent aussi des meilleures performances des systèmes informatiques, en faisant observer que les calculs de dates sont simplifiés par les années à quatre chiffres. Et il est vrai qu'avec des années à deux chiffres les calculs doivent tenir compte de la représentation *modulo 100*, et sont donc un peu plus complexes. Mais est-ce là un argument suffisant ? C'est sous des prétextes d'optimisation analogues que l'on codait, dans les années 60, les années sur deux chiffres, pour éviter de gaspiller quelques *bits*. Avec le succès que l'on connaît. À quoi bon, aujourd'hui, grappiller quelques *Mips* ou quelques *Flops* ?

C'est ici, selon moi, que les informaticiens se sont offert un petit plaisir, en abondant dans le sens de la norme, à l'occasion des "projets an 2000". Il suffisait, dans la plupart des cas (mais cela ne saurait, non plus, être une règle absolue), de continuer à exprimer les années *modulo 100*, et d'attribuer à ces années, au moyen d'une programmation adéquate, une *signification* compatible avec le changement de siècle. Au lieu de cela, un certain nombre d'informaticiens de tous bords, sociétés de services, entreprises clientes, gourous et autres technocrates, ont cru bon de sonner le glas de la continuité du temps. L'an 2000 informatique connaîtrait le temps absolu.

Il en résulte qu'au lieu de modifier progressivement, l'un après l'autre, les programmes concernés par le problème, en faisant en sorte que les programmes attribuent à chaque date la *signification* qui lui revient, dans la *fenêtre de temps* impartie, nos défenseurs de l'année à quatre chiffres ont solidairement alourdi de façon démesurée les "projets an 2000", en imposant de modifier les fichiers, les bases de données et la plupart des programmes. Tous en même temps. Et avant le réveillon s'il vous plaît.

Et en rendant, au passage, monstreux le futur "bogue de l'an 10000".

Comment conduire un "projet an 2000" ?

Il me semble inopportun d'en dire plus sur la façon de mener un projet "an 2000". Non pas que le sujet manque d'intérêt – et je crois, à tort ou à raison, avoir acquis une certaine expérience des conversions informatiques, indépendamment de l'an 2000 – mais il est tout simplement *trop tard* pour se lancer maintenant dans un projet important.

Les campagnes de sensibilisation, auxquelles on assiste actuellement, ne peuvent plus concerner que l'informatique individuelle, ou celle de PME ayant un parc informatique réduit. Pour les grandes entreprises ou les administrations qui ont un parc informatique d'une certaine importance :

- soit elles ont commencé (et, je leur souhaite, largement avancé) leur projet,
- soit leur informatique court de sérieux risques – voir plus haut.

Il n'est plus temps de se soucier, en 1999, d'assouplir les conditions de passation des marchés publics, pour préparer telle ou telle administration à franchir l'an 2000.

Enfin, faut-il féliciter ou... remercier les informaticiens ?

Partant d'un problème simple, à l'échelle du programme individuel, la combinatoire rencontrée dans les systèmes informatiques fait que leur modification, ou leur conversion, constitue un projet complexe.

Oui, l'an 2000 présente un risque informatique certain. Et, comme l'informatique est partout, le "bogue de l'an 2000" peut se manifester n'importe où. Il se présente comme un virus multiforme, en ce sens qu'il frappe à l'improviste, et que son identification puis son éradication ont quelque chose de médical.

Les risques pour l'entreprise sont une *profonde désorganisation informatique*, une perte majeure de crédibilité, une image de marque en chute libre.

Sans doute, les informaticiens sont-ils en grande partie responsables du problème, mais ils ne sont peut-être pas les seuls. Qu'ils travaillent dans les sociétés de services ou les entreprises clientes, les informaticiens se sont tous fait un peu plaisir en transformant, si commodément, un problème simple de conversion – qui pouvait raisonnablement s'effectuer *progressivement* – en un problème complexe de conversion de type *big bang*. Mais peut-être sont-ils seulement coresponsables de ce détournement.

Ils ont voulu, à cette occasion, appliquer une norme, sans doute judicieuse, qui leur a peut-être été imposée, et ils ont seulement péché par omission, en masquant la charge que représente un tel changement de norme.

Rendez-vous le samedi 1^{er} janvier 10000 – si notre calendrier n'a pas changé entre-temps – pour vérifier si l'expérience acquise, à la fin de notre millénaire, pour corriger le "bogue de l'an 2000", est réutilisable pour traiter celui de l'an 10000... ■

* Auteur du progiciel "Software ISO 2000".

Rectificatif

Paul BONNET, ancien directeur du Centre d'études nucléaires du CEA de Saclay, auteur dans le Courrier des lecteurs de juin-juillet 1999, page 78, est de la promotion 44 et non 35 comme indiqué par erreur.