

RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

La définition des performances dans les contrats de logiciels

Gerard, Jacques; Montero, Etienne

Published in:
Lamy droit de l'informatique

Publication date:
1990

Document Version
le PDF de l'éditeur

[Link to publication](#)

Citation for pulished version (HARVARD):
Gerard, J & Montero, E 1990, 'La définition des performances dans les contrats de logiciels', *Lamy droit de l'informatique*, p. 7-16.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Lamy droit de l'informatique

Informatique
Télématique
Réseaux

19
90

CAHIERS LAMY DU DROIT DE L'INFORMATIQUE
Supplément au N° 19 – Octobre 1990

La commande d'un logiciel est-elle un marché de dupes ?

par Claire LAROCHE VIDAL

L'un des mérites du projet de directive communautaire relatif à la protection des programmes d'ordinateur est, nous l'avons vu, d'avoir pris position sur certaines questions, aujourd'hui controversées. C'est également l'une des raisons de la défiance assez générale qu'il a suscitée; réaction qui n'a rien d'anormal dès lors que les solutions pour lesquelles il a ainsi opté ne peuvent que mécontenter les partisans des interprétations non-retenues.

Son article 2 qui, en cas de création de programmes d'ordinateur par contrats, attribue la titularité des droits d'auteur à l'auteur de la commande – et donc au détriment du fournisseur – n'échappe évidemment pas à la règle.

Maître Claire Laroche-Vidal, en présentant les arguments favorables et défavorables à la solution ainsi retenue permet par là-même de clarifier utilement les termes du présent débat.

La définition des performances dans les contrats de logiciels

par Étienne MONTERO et Jacques GERARD

Au titre des facteurs de réussite de toute informatisation, doit être relevée la nécessité d'une définition claire et précise des performances attendues du(es) logiciel(s). Or leur mise en forme contractuelle est souvent délicate. MM. Montero et Gerard, à la lumière de clauses issues de la pratique, envisagent à ce titre la disponibilité du logiciel, les exigences du client en matière de temps de réponse, les garanties relatives à l'évolutivité de la configuration informatique telles que: la compatibilité, l'extensibilité, la portabilité, la sécurité et la convivialité du logiciel.

La définition des performances dans les contrats de logiciels

INTRODUCTION

1. Les entreprises ne survivent que parce qu'elles sont compétitives. Or, l'informatique et la bureautique n'ont, dans l'entreprise, d'autre but que celui d'accroître sa rentabilité. Ainsi, logiquement, de nombreux chefs d'entreprise soucieux d'améliorer la productivité de leur société se sont tournés vers l'informatique.

Vecteur important de la rentabilité des entreprises, l'informatisation de celles-ci ne peut se faire au hasard. Sinon, loin d'être efficace, elle risque tout au contraire de créer des perturbations et d'être mal digérée par le personnel utilisateur. Il est impérieux de s'être ménagé au préalable de sérieuses garanties sur les plans technique et juridique. En effet, la jurisprudence témoigne que de nombreux échecs d'informatisation trouvent leur cause dans l'absence d'un contrat valablement rédigé.

Un bon contrat est indispensable, du moins pour les applications ayant pour but réel d'améliorer la rentabilité de l'entreprise concernée. Certes, cette garantie ne nous fera pas oublier les autres facteurs de réussite d'une informatisation : la viabilité du fournisseur, les qualités techniques réelles de la solution informatique adoptée, l'appréciation exacte des aspects sociaux liés à l'informatisation...

2. L'objet de notre propos n'est pas d'évoquer tous les facteurs de réussite d'une informatisation ni de commenter toutes les clauses qui doivent figurer dans un contrat d'acquisition de logiciel (1). Nous nous contenterons de proposer et d'analyser certaines clauses relatives aux performances du logiciel (2). On note en effet que les contrats plus récents se risquent parfois à définir dans des clauses techniques les performances attendues du logiciel. Nous ne manquerons donc pas de relever également certaines clauses issues de la pratique.

(1) La bibliographie générale sur les contrats informatiques est abondante. On citera notamment : A. Viricel, *Le droit des contrats de l'informatique*, éd. Moniteur, Paris, 1984 ; X. Linant de Bellefonds et A. Hollande, *Les contrats informatiques*, Masson, Paris, 1988 ; M. Coipel et alii, *Le droit des contrats informatiques, Précis de la Faculté de Droit de Namur* n° 4, Larcier, Bruxelles, 1983 ; M. Vivant et C. Le Stanc, *Lamy droit de l'informatique* 1990 ; P. et Y. Pouillet, *Les contrats informatiques. Réflexions sur 10 ans de jurisprudence belge et française*, J.T., 1982, p. 17 à 26 ; J.-P. Buyle, L. Lanoye et A. Willem, *L'informatique* (1976-1986). *Chronique de jurisprudence*, J.T., p. 93 à 104 et p. 113 à 123.

(2) Sur les autres clauses très spécifiques aux contrats informatiques, v. en particulier : X. Linant de Bellefonds et A. Hollande, *Droit de l'informatique*, Masson, 1988, p. 35 et s.

3. Nous appelons performances du logiciel, certaines propriétés et spécifications fonctionnelles générales qui contribuent à rendre le logiciel optimal sous le triple angle de son installation, de son utilisation et de son évolutivité. Les réflexions qui suivent concernent essentiellement les applications sensibles : logiciels développés sur mesure ou progiciels d'une certaine importance. Il s'agit des applications relatives aux opérations essentielles relevant de l'objet social de l'entreprise et dont l'arrêt ou le mauvais fonctionnement paralyse donc le déroulement de l'activité de l'entreprise. Il en sera ainsi par exemple du logiciel de gestion des réservations dans une entreprise de transport.

Les critères de performances sont assez difficiles à définir car ils comportent une part d'arbitraire. Il importe cependant aux parties contractantes d'arriver à déterminer en commun les qualités techniques du logiciel de manière telle qu'un accord soit possible entre elles. Signalons à cet égard, qu'à dessein, nous n'adopterons pas l'approche traditionnelle de la question en termes d'obligations de délivrance et de conformité, ou encore de garanties de bon fonctionnement et des vices cachés (3).

La mise en forme contractuelle des performances escomptées est particulièrement délicate car toute clause aboutit à figer des choix techniques en partie aléatoire et oblige à des concessions réciproques. Chaque clause traduit un équilibre à respecter entre deux positions apparemment inconciliables : celle du fournisseur qui hésite à s'engager fermement sur des qualités techniques et celle du client qui revendique certaines garanties qu'il juge essentielles. Dans ce sens, le contrat remplit une fonction de répartition consciente des risques, avec les implications financières de cette répartition (4). Une fois inscrits dans une clause du contrat, les choix opérés de commun accord font la loi des parties. Ils deviennent intangibles de sorte que le contrat a aussi pour but de prévenir les litiges. En effet, en cas de désaccord des parties contractantes sur l'existence ou le contenu d'une obligation, le contrat sert de référence pour les parties comme pour le juge. De nombreux conflits

(3) V. notamment sur ces questions : M. Vivant et C. Le Stanc, *Lamy droit de l'informatique* 1990, n° 482 et s. ; J. Huet, *Responsabilité du vendeur et garantie contre les vices cachés*, Paris, Litec, 1987, n° 182 et n° 208 et s. ; I. de Lamberterie, *Les contrats en informatique. Jurisprudence française*, 1983, n° 33.

(4) A ce propos, v. Y. Pouillet, *Quelques réflexions sur des questions contractuelles liées à l'informatisation des entreprises*, *Cahiers Lamy du droit de l'informatique*, avril 1990 (B), p. 3 et s.

peuvent donc être évités dans la mesure où leur solution a pu être clairement anticipée lors de la conclusion du contrat. C'est dire si, s'agissant d'aspects essentiels, il convient d'apporter un soin tout particulier dans la rédaction des clauses relatives aux performances du logiciel.

C'est d'abord dans l'intérêt du client, face au professionnel de l'informatique, que les clauses proposées ont été pensées. Ces clauses sont de celles qui collent de plus près à la technique : elles sont aussi par conséquent celles qui offrent au client les plus sérieuses garanties juridiques du point de vue technique.

4. Nous envisageons successivement comme performances du logiciel les aspects suivants : la disponibilité du logiciel (a), les exigences du client en matière de temps de réponse (b), les garanties relatives à l'évolutivité de la configuration informatique telles que la compatibilité (c), l'extensibilité (d), la portabilité (e), la sécurité (f) et enfin la convivialité (g) du logiciel.

I. — Disponibilité

5. Un logiciel est dit « disponible » s'il tourne correctement durant les périodes normales d'utilisation. Celles-ci correspondent habituellement aux heures d'ouverture des bureaux. La période d'indisponibilité commence donc « au moment où cesse l'état de marche et se termine au moment où le titulaire réassure l'état de marche » (5). Avant de préciser ses exigences en matière de disponibilité, le client doit mesurer la dépendance qu'il a par rapport au logiciel dont il est question. Ainsi, dans certains cas, il n'est pas tolérable qu'un logiciel reste trop longtemps indisponible. Les conséquences pourraient en être très sérieuses pour l'entreprise : retard dans le versement de salaires, baisse dans la production, irrégularité dans la gestion des dossiers, ... Il convient donc de parer à l'éventualité de délais d'intervention inadmissibles en cas d'arrêt prolongé de fonctionnement. Pour ce faire, le client peut fixer, sous la forme d'un pourcentage, une durée d'indisponibilité mensuelle maximum. Les experts estiment que le taux moyen de disponibilité d'un système est de l'ordre de 95 à 97 % du temps possible d'emploi, ce qui correspond environ à une journée d'immobilisation par mois (6).

(5) V. M. Vivant et C. Le Stanc, *Lamy droit de l'informatique* 1990, n° 484 et l'abondante jurisprudence citée ; J. Huet, *Responsabilité...*, op. cit., p. 172.

(6) Sur cette question, v. G. Dewulf, M. Schauss et R. Le Suisse, *La maintenance de logiciel : aspects juridiques et techniques*, Cahier du Centre de Recherches Informatique et Droit, n° 3, Bruxelles, Story-Scientia, 1989, p. 61 ; v. aussi « Règles de l'art. Experts expertisés. Dix ans : dix questions », *Expertises* 1988, n° 110, p. 336 et Paris, 22 décembre 1981, *Expertises* 1982, n° 40, p. 100.

6. Il faut cependant noter que le niveau de disponibilité est fonction du système dans son ensemble. Il est donc également lié au hardware et au logiciel de base. Il importe pourtant de pouvoir identifier la source exacte de l'indisponibilité afin de déterminer les responsabilités. Ce problème est particulièrement aigu lorsqu'il y a pluralité de fournisseurs. On ne peut automatiquement imputer une indisponibilité au fournisseur du logiciel. Le client ayant établi l'indisponibilité du logiciel, une présomption simple de responsabilité pèse sur le fournisseur du logiciel, en vertu du contrat. Il est alors admis à prouver le cas échéant que l'indisponibilité est due à un autre élément du système que le logiciel lui-même.

Encore reste-t-il à régler le sort des indisponibilités qui trouvent leur cause dans certaines incompatibilités éventuelles. En effet, une indisponibilité du système peut être due au fait que le logiciel objet du contrat est incompatible avec d'autres logiciels avec lesquels il tourne. Il peut également être incompatible avec la configuration « hardware » (sur la notion de compatibilité, v. *infra*, c). Dans la mesure où l'analyse fonctionnelle contient les informations suffisantes pour que le concepteur du logiciel sache avec quel(s) autre(s) logiciel(s) et sur quelle configuration hardware son logiciel est appelé à tourner, ce type d'incompatibilité lui est imputable. Une indisponibilité qui serait due à un tel problème d'incompatibilité rentre manifestement dans le cadre de ses propres obligations en matière de disponibilité. Cette responsabilité doit donc être prévue dans la clause qui définit ce que l'on entend par indisponibilité du logiciel, afin de bénéficier des sanctions qui s'y attachent.

7. Les paramètres de la clause relative à la disponibilité du logiciel sont donc les suivants :

- définir la période de référence pour l'évaluation de la disponibilité et déterminer les périodes normales d'utilisation du logiciel ;
- définir la notion d'indisponibilité (y inclure l'indisponibilité qui trouve sa cause dans un problème d'incompatibilité) et le taux d'indisponibilité maximum admis ;
- introduire une présomption simple de responsabilité du fournisseur et autoriser le fournisseur à faire la preuve contraire.

La clause peut être ainsi libellée :

Disponibilité : Le fournisseur garantit que le logiciel est disponible pendant la période de référence pour l'évaluation de la disponibilité du logiciel est le mois calendrier :

Le logiciel est déclaré indisponible si le client ne peut l'utiliser en raison d'un défaut ou d'une erreur affectant ce logiciel, alors que tous les autres éléments du système fonctionnent normalement.

Les termes erreur ou défaut au sens de cette clause incluent notamment les problèmes éventuels de compatibilité du logiciel avec son environnement informatique, tel que spécifié et défini dans l'analyse fonctionnelle du présent contrat (ou dans les annexes techniques).

Si le fournisseur conteste l'indisponibilité du logiciel, il est admis à prouver que l'indisponibilité est due au système, à l'exclusion d'une erreur affectant le logiciel.

Les périodes normales d'utilisation du logiciel sont fixées par jour ouvrable de (7 h 30 à 20 h 30).

La durée d'indisponibilité est la somme des durées pendant lesquelles le logiciel est inutilisable durant les heures normales d'utilisation.

La disponibilité du logiciel ne peut être inférieure à (95 %), ce qui signifie qu'il ne peut être indisponible par période de référence au-delà de (5 %) de la somme des périodes normales d'utilisation. »

8. Le client peut utilement compléter cette clause en préconisant le versement d'indemnités pour le cas où le logiciel reste trop longtemps indisponible (7). Il y a lieu d'indiquer, sous la forme d'un pourcentage, la durée maximale d'indisponibilité tolérable et de préciser le montant de l'amende qui sera due par heure d'indisponibilité. On peut éventuellement prévoir l'obligation pour le fournisseur de payer l'amende autant de fois qu'il y a de terminal.

Nous proposons la clause suivante :

« Amendes pour indisponibilité :

Lorsque la durée d'indisponibilité du logiciel dépasse les (5 %) des périodes normales d'utilisation pendant une période de référence, c'est-à-dire lorsque n'est pas respectée l'obligation de disponibilité de (95 %), l'adjudicataire sera tenu de payer une amende de (X FB) par heure (et par poste de travail). »

II. — Temps de réponse

9. Par temps de réponse, on définit communément la période de temps qui, en charge réelle, s'écoule entre l'envoi d'une commande de transaction à un poste de travail et l'apparition du premier caractère de la réponse à cette commande, étant entendu que cette réponse est fournie de façon continue et que le premier caractère est significatif.

La question du temps de réponse est importante car il y va de la commodité habituelle du travail des utilisateurs. Joint à la convivialité (v. *infra*, g), le temps de réponse est une qualité dont dépend étroitement l'acceptation par le personnel d'une entreprise d'un système informatique. Certaines

(7) A propos des clauses pénales en matière informatique, v. M. Coipel, et B. Hanotiau, *Les clauses relatives à la responsabilité civile contractuelle. Les clauses pénales*, in *Le droit des contrats informatiques. Précis de la Faculté de Droit de Namur* n° 4, p. 282 et s.

déficiences en matière de temps de réponse peuvent aussi avoir des conséquences néfastes sur le bon fonctionnement d'un service. Dans certains cas même, le temps de réponse ne peut souffrir aucune faille. Songeons à un logiciel de guidage d'un appareil médical. Dans tous les cas, il y a lieu d'inscrire les exigences attendues par le client quant au temps de réponse afin d'obtenir de sérieuses garanties juridiques à cet égard.

10. Le temps de réponse dépend de l'importance et de la puissance de la configuration, du réseau de télécommunication, du fait de l'existence ou non de plusieurs programmes tournant au même moment sur le même matériel, des procédures de sécurité mises en place (parfois complexes, elles peuvent ralentir les temps de réponse), ... Le client ne peut donc jamais exiger de temps de réponse qui soit inférieur au temps de réponse « hardware », compris au sens le plus large. Et il est entendu que le temps de réponse vaut pour la configuration telle qu'elle est définie dans l'analyse fonctionnelle (nombre d'ordinateurs, de terminaux, d'imprimantes, de logiciels tournant sur le système, etc.).

En principe, le fournisseur ne garantit plus ce temps de réponse en cas d'augmentation des charges sur la configuration informatique (augmentation du nombre de logiciels, du nombre d'utilisateurs, des fréquences d'intervention de chaque utilisateur, ou du volume d'informations à traiter, ...).

Au moment de déterminer les temps de réponse attendus, le client doit donc songer à l'éventualité d'extensions futures. Si, par exemple, il en est au début de son informatisation, ou s'il prévoit d'automatiser diverses autres fonctions dans un avenir prochain, ... il est préférable de prévoir une réserve de temps calcul et d'accès disques dans le système informatique. Il se ménage ainsi la possibilité d'installer de nouvelles applications tout en conservant un temps de réponse correct.

11. Il y a différentes manières d'exprimer les performances attendues quant aux temps de réponse.

Dans toutes les hypothèses, la clause relative aux temps de réponse doit comporter une définition générale du temps de réponse.

Dans la clause suivante, le client définit un temps de réponse donné pour toutes les transactions.

« Temps de réponse :

Par temps de réponse, on définit la période de temps qui, en charge réelle, s'écoule entre l'envoi d'une commande de transaction à un poste de travail et l'apparition du premier caractère de la réponse à cette commande, étant entendu que cette réponse est fournie de façon continue et que le premier caractère est significatif.

Un temps de réponse acceptable est exigé pour toutes les transactions.

Si le fournisseur estime que le non-respect du temps de réponse est dû à un autre élément du système à l'exclusion du logiciel, il est admis à en apporter la preuve.

Par temps de réponse acceptable, on entend un temps de réponse qui ne soit jamais supérieur à (3 secondes). »

Le client souhaite fréquemment fixer des temps de réponse différents selon le type de transactions. Il peut, par exemple, exiger un temps de réponse de 2 secondes maximum pour les transactions simples et de 5 secondes maximum pour les transactions complexes. La clause suivante se retrouve fréquemment dans la pratique :

« Par temps de réponse acceptable, on entend :

— pour les transactions simples, un temps de réponse qui ne soit pas supérieur à (2 secondes) ;

— pour les transactions complexes, un temps de réponse qui ne soit pas supérieur à (5 secondes). »

Cette distinction présente cependant un danger d'arbitraire dans la mesure où le fournisseur peut facilement décider qu'une transaction est complexe (dès qu'il est fait appel à plus d'un fichier, ou à plus d'un ordinateur, d'une imprimante, ou d'une banque de données, ...). En somme, il peut estimer qu'une transaction est complexe dès qu'il y a moyen d'invoquer un problème de file d'attente ou de « hand-shaking ».

Il y a file d'attente dès que trop d'utilisateurs veulent se servir simultanément du même processeur ou de la même base de données ou encore du même logiciel, ... de sorte que la vitesse de service du système devient insuffisante pour satisfaire immédiatement à toutes les demandes. Le « hand-shaking » est quant à lui le protocole de dialogue existant entre deux systèmes ou parties de système (par exemple, entre le processeur et un disque dur). Il faut éviter que par une formulation peu rigoureuse de la clause relative aux temps de réponse, le fournisseur ne puisse trop facilement éluder ses responsabilités en invoquant notamment ces phénomènes de l'hand-shaking et de la file d'attente.

Si l'on veut tout de même adopter la distinction entre transactions simples et transactions complexes, il y a lieu d'identifier exactement (et de manière limitative) ce que l'on entend par transaction simple. Par exemple : on appelle transaction simple une transaction qui nécessite le recours à un fichier au plus.

Une autre manière de procéder, qui permet de contourner le danger d'arbitraire décrit, est de prévoir un pourcentage limité de cas où le temps de réponse peut être plus élevé que le temps de réponse habituellement exigé. C'est ainsi que les parties contractantes ont procédé dans cette clause :

« Par temps de réponse acceptable, on entend :

— un temps de réponse moyen qui ne soit pas supérieur à (2 secondes) dans (90 %) des transactions ;

— le temps de réponse ne pourra en aucun cas dépasser (5 secondes).

Ce temps de réponse sera mesuré pendant une heure d'utilisation normale du logiciel à partir d'un poste de travail. »

12. Dans tous les cas de figure, il convient d'indiquer la sanction propre au non-respect du temps de réponse :

« Si l'obligation de résultat concernant le temps de réponse devait ne pas être respecté, le fournisseur serait contraint d'assurer par tous les moyens le rétablissement de cette obligation, soit en modifiant le logiciel, soit de toute autre manière appropriée, cela dans les plus brefs délais et à ses propres frais. »

13. D'autre part, le client doit veiller à ce que les tests des temps de réponse s'effectuent dans les mêmes conditions physiques que celles du système en temps normal. Il n'est pas réaliste par exemple de tester le temps de réponse d'accès à un fichier ne contenant que quelques informations alors qu'il est appelé à en contenir des millions. En effet, le temps de recherche de l'information est fonction du nombre d'informations. Il en est de même du nombre de logiciels tournant simultanément sur le hardware.

Le client a donc intérêt à définir très exactement les conditions de tests des temps de réponse.

On exigera que les tests soient réalisés avec des fichiers remplis artificiellement, de taille équivalente aux fichiers tels qu'ils existeront. Il est très facile de créer rapidement des fichiers artificiels de taille quelconque, à l'aide d'un programme itératif générant une information aléatoire.

On veillera aussi à installer un certain nombre de processus utilisant un temps calcul et un temps d'accès disque suffisants pour simuler une charge normale d'utilisation.

La clause suivante tient compte de ces différentes exigences :

« Le client se réserve le droit de tester le temps de réponse, à la livraison ou à tout autre moment. A cette fin, le temps de réponse du logiciel sera mesuré dans les conditions suivantes :

— des fichiers de taille équivalente aux fichiers prévus dans le contrat seront installés ;

— le nombre de processus de simulation d'un utilisateur sera égal au nombre moyen d'utilisateurs, tel qu'il est stipulé dans le contrat ;

— des processus réclamant un temps d'accès disque et un temps calcul équivalent à une réserve de (X %) du temps système seront installés. »

La dernière condition mentionnée est destinée à tenir compte des extensions futures du système au moment de mesurer les temps de réponse. Si aucune extension n'est prévue, cette condition de test peut être supprimée.

14. Comme en matière de disponibilité, il peut être judicieux de prévoir au contrat l'obligation pour le fournisseur de payer une amende en cas de non respect du temps de réponse. C'est ce qu'indique la clause suivante :

« Amendes pour non respect du temps de réponse :

A défaut de recouvrer un temps de réponse acceptable dans le mois de la notification du non respect du temps de réponse, le fournisseur, sauf pour lui à démontrer que le non respect est dû au fait du client ou d'un tiers, sera tenu de payer une amende de (X FB) par heure (et par poste de travail). »

15. Nous envisageons à présent diverses questions étroitement liées, qui toutes concernent l'évolution ou la modification de la configuration informatique. Il s'agit des notions de « compatibilité », d'« extensibilité » et de « portabilité ». Ces différentes notions sont interdépendantes : ceci est à ne pas perdre de vue même si, pour la clarté de l'exposé, nous sommes contraints d'aborder distinctement chacune d'entre elles.

III. — Compatibilité

16. Le logiciel objet du contrat est appelé à tourner sur une configuration donnée. Le client doit donc obtenir l'assurance qu'il tournera parfaitement sur le matériel en place et sans perturber le fonctionnement des autres logiciels. Ceci ne pose généralement pas de problème au niveau juridique. Ce qui par contre peut être utile à mentionner dans le contrat, c'est l'obligation pour le fournisseur de livrer les interfaces qui sont le cas échéant nécessaires à la compatibilité du logiciel. En effet, il se peut que l'utilisation du logiciel réclame obligatoirement l'écriture et la fourniture d'un logiciel d'interface.

Par exemple :

Imaginons la configuration suivante, déjà existante :

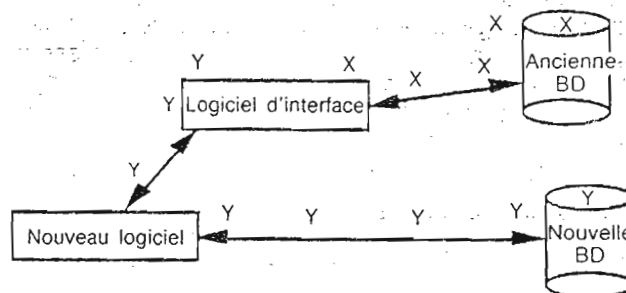
- le matériel ;
- les logiciels d'application ;
- une base de données dont les enregistrements ont une certaine forme (X).

Le client souhaite créer une nouvelle base de données (BD) et installer un nouveau logiciel destiné à utiliser cette dernière. Supposons alors que le client veuille que le logiciel nouvellement acquis puisse continuer d'utiliser l'ancienne base de données. Or, dans l'une et l'autre base de données,

les enregistrements n'ont pas la même forme. Ils ont, dans la nouvelle base de données, une forme différente (Y) qui s'imposait par une évolution du contexte factuel ou normatif ayant une incidence sur la présentation ou le contenu des données.

Un problème d'incompatibilité se pose alors. Le fournisseur devra écrire un logiciel réalisant l'interface entre l'ancienne base de données et la nouvelle.

Schéma :



C'est entre logiciels appelés à tourner en parallèle que se pose le plus souvent la question de la compatibilité. Par exemple, si un logiciel de calcul des salaires tourne simultanément à un logiciel de comptabilité, ce dernier doit pouvoir intégrer les résultats du premier.

Il importe donc de bien préciser, par une clause du contrat, que le fournisseur est tenu de fournir les interfaces nécessaires à la compatibilité du logiciel :

« Compatibilité :

Le fournisseur garantit la compatibilité du logiciel fourni avec la configuration matérielle et logicielle qui est actuellement celle du client.

Le fournisseur est tenu de fournir le cas échéant les interfaces appropriés. »

La compatibilité du logiciel doit par ailleurs être conservée malgré des extensions du système. C'est cette notion d'extensibilité que nous examinons à présent.

IV. — Extensibilité

17. A tout moment de la durée du contrat, le client doit pouvoir étendre le logiciel acquis (ajouter de nouvelles fonctions, apporter une modification de sorte qu'il puisse répondre à un besoin nouveau...). Cela signifie que la faculté d'ajouter ces nouvelles capacités au logiciel doit être garantie à la conclusion du contrat (8).

(8) Sur les problèmes nés de l'extension, v. Ph. Peters, Examen de quelques aspects particuliers des contrats particuliers, in *Le droit des contrats informatiques*, p. 426-7.

Par exemple :

Le client dispose d'un logiciel de comptabilité et il souhaite y adjoindre l'utilisation d'une banque de données ; ou encore, il utilise un logiciel de calcul des salaires et veut y ajouter une nouvelle fonction : le calcul des primes de fin d'années,...

Il importe tout d'abord que le client puisse obtenir du fournisseur ces extensions de logiciel. Encore faut-il ensuite que le logiciel étendu soit toujours compatible avec l'environnement informatique. Une garantie doit donc être obtenue à cet égard également. Bien évidemment le fournisseur n'est plus tenu de garantir la compatibilité du logiciel dans la mesure où les extensions sont faites par le client lui-même ou par un tiers.

Une clause du contrat doit dès lors prévoir les possibilités d'extension. Il convient d'y inscrire aussi que le logiciel étendu doit demeurer compatible avec l'environnement informatique.

« Extensibilité :

Le fournisseur garantit pendant toute la durée du contrat que le logiciel peut être étendu auprès du même fournisseur en fonction de la croissance des besoins du client et que ces extensions sont parfaitement compatibles avec le système en place et le réseau dont il fait éventuellement partie. »

e) Adjonctions et portabilité

18. Le client doit pouvoir adjoindre du nouveau matériel ou de nouveaux logiciels au système en place. L'adjonction est en fait « toute connexion mécanique, électrique ou électronique d'un équipement ou d'un accessoire à un matériel » (9). Il est en effet fréquent que le client soit confronté au besoin d'étendre son système informatique en fonction de la croissance de ses besoins, ou de l'augmentation du volume d'informations à traiter, etc. Il doit dès lors obtenir la garantie que le logiciel pourra être fondu dans le système étendu et parfaitement compatible avec lui.

La portabilité d'un logiciel est son aptitude à être utilisé sur des systèmes informatiques de type différent (10). Un logiciel est portable dans la mesure où il peut facilement être exploité sur une autre configuration que celle sur laquelle il est exploité actuellement. Le degré de portabilité est dès lors d'autant plus élevé qu'est moindre l'effort à accomplir pour qu'un logiciel puisse tourner sur une configuration différente en termes de matériel ou de système d'exploitation.

(9) M. Vivant et C. Le Stanc, *Lamy droit de l'informatique* 1990, n° 856.

(10) Sur la notion de portabilité, v. J. Hoorens, *Les contrats de développement de logiciel sur mesure, Précis de la faculté de Droit de Namur, Story-Scientia*, 1986, p. 90.

19. Deux qualités sont donc nécessaires pour qu'un logiciel soit portable.

La première vise à ce que le logiciel reste opérationnel malgré une extension du système ce qui suppose son indépendance suffisante par rapport au matériel.

Exemple :

Imaginons qu'un logiciel soit prévu pour être utilisé à partir de 15 postes de travail. Le client décide d'augmenter le nombre de terminaux afin qu'il y ait 25 postes. Or, chaque poste de travail génère un processus actif qui permet d'utiliser le logiciel. Celui-ci n'est pourtant capable de gérer simultanément qu'un nombre limité de processus. C'est pourquoi il est impérieux d'obtenir de la part du fournisseur qu'il modifie le logiciel en vue de gérer cette extension de matériel.

La seconde qualité d'un logiciel dit « portable » est son aptitude à fonctionner en autonomie par rapport à l'environnement logiciel.

Exemple :

Un logiciel utilise un serveur, c'est-à-dire un progiciel indépendant du système d'exploitation, utilisable par le logiciel d'application pour réaliser certaines fonctionnalités prédéfinies. Prenons un gestionnaire d'écrans, étant donné qu'il est dédié à un matériel bien précis. Si l'on doit changer de configuration matérielle, tous les appels au serveur deviennent caduques. Des modifications importantes de conception s'imposent donc.

La portabilité serait plus élevée si le logiciel utilisait son propre gestionnaire d'écrans. Dans l'hypothèse d'un changement de matériel, il suffirait alors, tout au plus, d'une simple recompilation.

Lorsqu'un logiciel est peu ou pas portable, il faudra le cas échéant opérer sa « conversion » pour qu'il devienne apte à servir dans l'environnement informatique modifié. Il sera nécessaire dans ce cas de déterminer à qui reviendra la charge des frais de reconversion.

Le choix d'une clause adéquate de portabilité est lié à la connaissance, suffisante ou non, qu'a le client de son plan informatique. Si ce dernier est très précis, le client pourra facilement spécifier des exigences concrètes en matière de portabilité. Si par contre, il est peu fixé quant à la manière dont son système informatique va évoluer, il pourra difficilement déterminer ses attentes précises quant à la portabilité. Au contraire, il risque dans ce cas de devoir ultérieurement envisager les extensions du système précisément en fonction des logiciels fournis et de leur portabilité.

20. Ceci dit, au plan contractuel, il est deux manières de définir le degré de portabilité exigé (i) (ii).

(i) La première suppose une connaissance approfondie des aspects techniques liés à la portabilité et consiste à spécifier en termes techniques le degré de portabilité exigé. Cette solution suppose généralement connues les acquisitions futures : de matériel et/ou logiciels de base et/ou logiciels d'application. Le client précisera les types de machines qui sont en place, ainsi que les types de systèmes d'exploitation, de serveurs, de logiciels d'application, ... Il indiquera les mêmes données pour les acquisitions futures, celles prévues par exemple lors de la phase suivante de son projet d'informatisation. La clause envisagée vise donc la portabilité du logiciel sur un système étendu par l'adjonction d'équipements nouveaux limitativement énumérés et connus du fournisseur dès la conclusion du contrat :

« Adjonctions et portabilité :

Le client se réserve le droit d'étendre le système en place par l'adjonction d'éléments nouveaux (qu'il se procure éventuellement chez un autre fournisseur).

Il est explicitement entendu que le logiciel objet du contrat ne sera portable que sur les équipements limitativement énumérés à l'annexe du présent contrat.

Par portabilité, on entend la possibilité de mise en œuvre opérationnelle du logiciel assurant un niveau de performances au moins équivalent, moyennant prise en charge des frais de conversion par le fournisseur. »

Cette autre clause ne diffère pratiquement pas de la première :

« Le fournisseur garantit qu'à tout moment de la durée du contrat, le logiciel est compatible avec les extensions du système, prévues et expressément indiquées dans le présent contrat.

En tout état de cause, les frais éventuels de conversion sont intégralement à charge du fournisseur. »

(ii) La seconde manière de procéder évite une approche technique et définit plus simplement un degré de portabilité à atteindre.

On peut concevoir comme suit la définition de degrés de portabilité :

- le logiciel peut tourner sur tout système informatique compatible avec plusieurs types donnés de systèmes d'exploitation ;
- le logiciel peut tourner sur tout système informatique possédant un système d'exploitation donné ;
- le logiciel peut tourner sur tout système informatique possédant un système d'exploitation donné et un ensemble de serveurs précisés ;
- le logiciel peut tourner sur plusieurs types donnés de matériel ;

- le logiciel peut tourner sur un seul type donné de matériel ;
- aucune portabilité.

Le client doit essayer de définir le niveau de portabilité qu'il peut raisonnablement exiger. Il tiendra compte du fait que le coût du logiciel et ses performances dépendront fortement du degré de portabilité garanti.

Cette seconde démarche dans l'écriture d'une clause relative à la portabilité donne lieu à des clauses très diversifiées, dont nous proposons trois exemples :

« Le client se réserve le droit d'adjoindre au système du matériel commercialisé par un autre fournisseur.

Dans ce cas, il notifie son intention par écrit à l'autre partie.

Celle-ci signalera, dans un délai de [30 jours] au client les inconvénients éventuels susceptibles de se manifester à la suite des adjonctions proposées.

Le fournisseur ne pourra invoquer les adjonctions pour s'exonérer de ses obligations en matière de compatibilité, de garanties ou de maintenance sauf s'il peut démontrer que ces adjonctions l'empêchent de satisfaire à ses obligations et en donne notification au client dans le délai imparti de 30 jours.

Il fournira toutes les informations nécessaires permettant au client d'effectuer ces adjonctions.

(En tout état de cause, les frais éventuels de conversion sont intégralement à charge du fournisseur.) »

« Le logiciel doit être portable sur tout système informatique qui utilisera un système d'exploitation équivalent. »

« Le logiciel doit être portable sur tout système qui utilise (tel système d'exploitation), moyennant une simple recompilation. »

21. Quant à déterminer qui supportera les éventuels frais de conversion, une option s'ouvre au client :

- s'il est capable de définir assez strictement ses exigences en matière de portabilité, il recommandera que les frais de conversion seront entièrement à charge du fournisseur, il en sera souvent ainsi lorsque le client a conçu une automatisation progressive en plusieurs phases. Il exigera que les logiciels — développés lors d'une des phases — soient portables sur le système étendu à l'occasion des phases ultérieures ;
- si le client n'a pas encore une connaissance suffisante des acquisitions futures, il doit se contenter d'une clause assez vague de portabilité. Dans ce cas, à peine d'y mettre un prix exorbitant (car trop d'hypothèses restent ouvertes), il est sage qu'il prenne à sa charge les frais éventuels de conversion à concurrence de ce qui excédera le niveau de portabilité stipulé.

22. En toute hypothèse, il convient que soient mentionnés dans une clause tous les cas concrets d'adjonctions futures connus par avance. Ces précisions présentent l'intérêt d'éviter des effets de surprise inutiles susceptibles de compromettre la bonne entente entre les parties.

Ainsi, par exemple, cette clause :

« Le logiciel, objet du contrat, doit être ouvert aux extensions prévues en annexe et être compatible avec elles, notamment la connexion prochaine au réseau (X) »,

ou cette clause :

« Le logiciel, objet du contrat, doit pouvoir être étendu de sorte que, dans un proche avenir, le système puisse être utilisée par toutes les secrétaires du service. »

V. — Sécurité

23. L'aspect sécurité couvre deux grands types de problèmes que nous envisageons successivement (11) :

- l'intégrité des données et des fonctions du logiciel (i) ;
- le contrôle des accès au logiciel et à la base de données éventuelle (ii).

(i) L'intégrité

24. L'intégrité vise la qualité de l'information du point de vue de sa cohérence, validité complétude, véracité,... L'intégrité du logiciel permet d'assurer que les données soient traitées, conservées, transmises et restituées de façon inaltérée.

Ainsi, par intégrité, nous entendons notamment :

- le fait que l'utilisation simultanée ou successive de plusieurs fonctions ne crée pas de perturbation ;
- le fait que le logiciel puisse utiliser la base de données sans y engendrer d'incohérence.

La cohérence de la base de données est conservée entre autres si toute interaction avec la base de données est ponctuelle et limitée dans le temps. Ceci implique que les modifications opérées dans la base de données doivent être effectuées de telle manière qu'au terme de chaque étape la cohérence de la base de données soit maintenue.

Par exemple :

Une base de données contient des renseignements relatifs au personnel d'une entreprise. Dans un premier fichier, on y trouve les renseignements

généraux. Dans un deuxième fichier, sont inscrits des renseignements techniques concernant le salaire, la fonction, les absences,... des mêmes personnes.

S'il y a lieu de supprimer une personne de la base de données, il est évident qu'il faut effacer les enregistrements dans les deux fichiers. A défaut, la base de données est devenue incohérente.

Un exemple tout à fait concret que nous exposons à présent, nous est fourni par une société de crédit hypothécaire. Le fichier « compte individuel des emprunteurs » comprend deux sous-fichiers :

- le premier donne la situation globale de l'emprunteur à la date de consultation, c'est-à-dire :
 - 1) le solde au premier janvier,
 - 2) le montant des intérêts dus à la date de consultation,
 - 3) les remboursements perçus,
 - 4) le solde à la date de consultation ;
- le second indique à tout moment quelle est la situation réelle de l'emprunteur demeurant titulaire d'une créance vis-à-vis de la société. Dans ce sous-fichier, on trouve principalement les dates auxquelles il a reçu des sommes et les intérêts dus sur les sommes reçues.

Dans les deux sous-fichiers, le montant des intérêts dus devrait être identique. Or, il n'en est rien : à certains moments et sans fournir d'explications, le logiciel fait apparaître des montants différents selon que l'on consulte l'un ou l'autre des sous-fichiers. Il s'en est déjà suivi pour cette société des pertes financières : ainsi, un client à qui la société a réclamé 500 FB d'intérêts – sur foi d'un des sous-fichiers – devait en réalité 20 000 FB.

Le concept de cohérence (de la base de données) présente une autre facette. En effet, lorsque plusieurs utilisateurs se servent des mêmes informations d'une base de données, il faut qu'un seul à la fois soit autorisé à modifier les informations. Dans ce cas, les autres utilisateurs doivent être avertis qu'une modification est en cours.

En outre, en cas de panne survenant pendant une intervention dans la base de données, le système doit être capable de revenir automatiquement à l'état antérieur.

Enfin, pour que l'intégrité soit maintenue dans le temps, le client doit se prémunir contre les pannes du système, incendies ou autres dégâts importants pouvant survenir dans ses locaux. A cette fin, il peut être utile de prévoir de une option de back up (sauvegarde des données sur un support indépendant).

(ii) Le contrôle des accès

25. Une autre composante de la sécurité vise à éviter que des flux d'informations ne parviennent à des personnes non autorisées.

(11) V. le chapitre relatif à la sécurité informatique in M. Antoine et M. Eloy, *La preuve face aux nouvelles technologies de l'information. Aspects techniques et juridiques du transfert et de la conservation électronique de documents*, à paraître dans la collection des *Cahiers du Centre de Recherches Informatique et Droit, Story-Scientia*.

Le contrôle des accès comprend essentiellement deux aspects :

- la définition des hiérarchies d'utilisation ;
- la protection des informations.

La hiérarchie d'utilisation définit pour chacun les possibilités d'utilisation des fonctions du logiciel. Celui qui se trouve au sommet de la hiérarchie peut accéder à toutes les possibilités. Il lui revient notamment de définir et de modifier la hiérarchie des priorités.

La première étape dans l'établissement d'une hiérarchie d'utilisation vise l'identification des utilisateurs. En effet, le client souhaite parfois réserver l'accès aux informations jugées confidentielles à certains utilisateurs.

La deuxième étape consiste à définir les droits d'accès pour chacun des utilisateurs identifiés.

Exemples :

- Le client souhaite que les membres d'une section ne puissent accéder qu'aux seules données relevant de cette section.
- Le client souhaite qu'à l'intérieur d'une section, l'accès soit encore plus sélectif (ex. : quant aux données concernant les employés, ou aux informations financières, ...).

La protection des informations tend à empêcher la lecture et/ou la modification des informations par les utilisateurs qui n'en ont pas droit. Par exemple, le client souhaite que les employés de tel et de tel niveau ne soient autorisés qu'à lire certaines informations tandis que les employés de niveau supérieur sont également autorisés à modifier la base de données (introduire de nouvelles données ou modifier des données existantes). Cette même protection sert également à se prémunir contre les violations de la base de données en provenance de l'extérieur. Il existe à cet égard différentes techniques de contrôle des accès : les mots de passe, les codes secrets, les clés magnétiques... Le fournisseur peut encore par exemple réaliser cette protection par un chiffrement des informations. Cette technique du chiffrement consiste à transformer l'information en un texte codé pour la rendre incompréhensible par tout utilisateur qui ne possède pas la clé de lecture (12). Si elle n'empêche pas la fraude, cette méthode garantit au moins l'inutilité des données acquises frauduleusement.

(12) La cryptographie est un procédé qui « consiste à coder un texte à l'aide de clés confidentielles et de processus mathématiques complexes (algorithmes) afin de le rendre incompréhensible à toute personne qui en prendrait connaissance sans avoir les moyens de procéder à son déchiffrement, opération symétrique qui rétablit le texte en clair » (B. Amory et X. Thunis, *Authentification de l'origine et du contenu des transactions commerciales assistées par ordinateur et responsabilité en droit continental*, Bulletin de l'IDATE, n° 29, 1987, p. 143).

La situation concrète d'une petite banque nous permet d'illustrer cette problématique du contrôle des accès : au départ de l'ordinateur, il n'est prévu qu'un seul accès à la base de données. Or, il existe deux types d'accès bien distincts :

- la possibilité de consulter les comptes des clients et
- le droit de modifier ces comptes (changements d'adresse, imputation de mensualités reçues sur un C.C.P., etc.).

D'après les responsables de la banque, l'expérience enseigne qu'il aurait fallu opérer une sélection entre la simple consultation des opérations financières et la gestion (pouvoir de modification) mensuelle, trimestrielle et annuelle des comptes.

D'autre part, quiconque peut accéder à la commande qui enclenche la clôture annuelle des comptes, opération par ailleurs irréversible. Il aurait fallu un code d'accès spécial, hautement sélectif, à cette commande « dangereuse ».

26. Une clause du contrat indiquera donc les attentes du client en matière de sécurité. Celles-ci peuvent concerner l'intégrité des données et fonctions du logiciel, comme dans cette clause :

« Le logiciel doit assurer l'intégrité des informations selon les spécifications suivantes :

- l'utilisation simultanée ou successive de plusieurs fonctions ne peut créer de perturbation ;
- le logiciel doit pouvoir utiliser la base de données sans y engendrer d'incohérence ;
- lorsque plusieurs utilisateurs se servent simultanément de la base de données, un seul à la fois sera autorisé à y modifier les informations ; les autres utilisateurs doivent être avertis qu'une modification est en cours ;
- le logiciel doit être doté d'une fonction de sauvegarde automatique des données sur un support indépendant. »

Les mesures de sécurité attendues peuvent également viser le contrôle des accès au logiciel, comme dans cette autre clause :

« Les mesures de sécurité doivent prévoir :

- l'accès aux données à partir d'un service limité aux seules données relevant de ce service ;
- un accès sélectif conforme à la hiérarchie des priorités telle que définie en annexe, pour les données (X, Y). »

VI. — Convivialité

27. La convivialité du logiciel concerne les qualités de dialogue avec l'utilisateur. Un logiciel est dit « convivial » dans la mesure où il est agréable et facile à utiliser (13).

(13) V. J. Hoorens, *op. cit.*, p. 95 à 101.

Ainsi, par exemple, tel logiciel présente un défaut de convivialité dans la manière dont il utilise la base de données. Dans sa fonction de recherche sélective par le nom du client, le logiciel devrait être à même d'aller chercher dans la base de données des propositions sur base des premières lettres du nom. Et si un nom est mal orthographié, l'écran devrait tout de même afficher une série de noms voisins.

Dup → Dupond Dominique
 Dupond Jules
 Dupont Antoine
 etc.

Or, le logiciel, dans notre exemple, n'affiche jamais qu'un seul nom à l'écran, celui-là seul qui a été appelé et dans la mesure où il a été correctement orthographié.

Cet exemple permet de conclure qu'il n'est pas inutile d'inscrire dans une clause tous les facteurs de convivialité auquel on songe compte tenu de l'expérience et des données propres à l'entreprise.

De manière schématique, la convivialité suppose notamment certaines des caractéristiques suivantes, du reste liées entre elles :

- les écrans sont suffisamment communicatifs au sens où ils fournissent des messages faciles à interpréter ;
- il existe des messages d'avertissement si l'utilisateur pousse sur une touche inadéquate ou viole une spécification ;
- le logiciel doit être agréable à utiliser au sens où les résultats sont générés sans perte de temps ou d'énergie pour l'utilisateur et sans le démotiver (14) ;
- il existe un menu d'aide applicable à tout moment et facile à utiliser. Les fiches explicatives sont pertinentes, claires et bien structurées ;
- le logiciel est simple à utiliser : les temps d'apprentissage sont courts ;
- le logiciel est compréhensible, dans la mesure où il contient suffisamment d'informations pour permettre à l'utilisateur de déterminer ou de vérifier les objectifs, les hypothèses, les contraintes, les entrées, les sorties et les composantes ;
- le logiciel est écrit dans un style concis : il ne contient pas d'informations excessives ;
- etc.

A titre d'exemple, relevons une clause relative à la convivialité du logiciel :

(14) A mettre en relation avec la problématique du temps de réponse (v. *supra*, b).

(15) A propos de la maintenance : v. R.L. Glass et R.A. Noiseux, *Maintenance du logiciel*, Masson, Paris, 1983, traduit de l'anglais par M. Gabriel, G. Dewulf, M. Schass et R. Lesuisse, *op. cit.*

« Le logiciel devra satisfaire aux exigences suivantes :

- les écrans sont hautement communicatifs : les messages seront concis, facilement compréhensibles, faciles à interpréter ;
- il sera prévu un menu d'aide appelable à tout moment ;
- le logiciel est facile à utiliser : les temps d'apprentissage seront courts ;
- des messages d'avertissement en cas d'utilisation violant une spécification s'inscriront sur l'écran. »

CONCLUSION

28. Enfin, il est de l'intérêt du client de se ménager la possibilité d'exiger le maintien des performances telles que définies au contrat, pendant une période de temps donnée (2, 3, ... ans) se situant au-delà de la réception définitive. C'est d'ordinaire dans le cadre d'un contrat de maintenance conclu avec le fournisseur que ces performances seront garanties. La notion de « durée de contrat » offre à cet égard, selon nous, une base juridique pour revendiquer un droit à la maintenance. D'autre part, la maintenance (15) — destinée notamment à assurer le maintien à niveau du logiciel — se trouve également finalisée par la référence aux obligations initiales qu'implique le concept de durée du contrat. En effet, par ce biais, le niveau de performance à maintenir est très exactement défini : il s'agit des performances stipulées au contrat. A cet égard, on relève les clauses suivantes :

« Le fournisseur garantit que le logiciel répondra aux exigences contenues dans le présent contrat pendant une période de trois ans à compter de la réception définitive. »

« Le contrat prend cours à la signature pour se terminer cinq ans plus tard jour pour jour à partir de la réception définitive. »

Étienne MONTERO et Jacques GERARD,
Centre de Recherches Informatiques et Droit.
 (Facultés Universitaires
 Notre-Dame de la Paix, Namur)

LAMY DROIT DE L'INFORMATIQUE

Directeur de publication : J.-M. DETAILLEUR
 Rédacteur en chef : L. COSTES

Abonnement annuel : 1 575,00 F TTC
 Bulletin mensuel

N° Commission paritaire : 71 519
 Édité par LAMY S.A. au capital de 10 692 000 F
 155, rue Legendre, 75850 PARIS CEDEX 17
 tél. : (1) 46.27.28.90 - télex : 214 398 - téléfax : 42.29.86.81
 C.C.P. Paris 20 581 11 H
 Dépôt légal LAMY S.A. éditeurs - N° d'édition : 2684
 N° d'impression : 1928 - Octobre 1990

Toute reproduction, même partielle, des articles parus dans LAMY DROIT DE L'INFORMATIQUE, est interdite sans notre autorisation (loi du 11 mars 1957).

Laboureur - 16, rue Montgolfier, Z.I. 93115 Rosny-sous-Bois Cedex