

L'INFORMATIQUE

P R O F E S S I O N N E L L E

Dossier

Développement

Christophe Blaess, Alain Fernandez, Jean-Marc Berlioux, Gilles Mergoil, Jean-Marc Lejeune, Yves Constantinidis

Internautes

Souriez, vous êtes pistés

Eric Larcher

Achats informatiques

Diminuer les risques et négocier

Gilles Blanchard

Refacturation des réseaux

La diplomatie s'impose

Edward Younker, David Neil

Mensuel • Numéro 189 • décembre 2000

ont aussi collaboré à ce numéro :

André Bors

Eric Boulanger

Sommaire complet en pages 4 et 5

Gartner

Bouhot & Le Gendre
Produits et Services



Développeurs : la fin des temps !

Fini, périmé, dépassé, obsolète... Si l'on devait croire certains discours, c'en serait fini du développement et de la programmation.

Face à l'offre florissante en matière de progiciels, d'outils bureautiques et de services ASP, le programmeur serait une race en voie de disparition. Il suffirait en effet d'installer un produit bien choisi et de définir rapidement un paramétrage adapté, ou de se connecter à un fournisseur de services, pour accéder au nirvana du système d'information. Qui plus est, le progiciel apporterait dans ses bagages un catalogue de "bonnes pratiques" permettant l'amélioration des processus. Et le management pourrait toujours faire porter au progiciel le chapeau des évolutions de l'organisation, ce qui lui éviterait de se retrouver en première ligne pour conduire le changement. Enfin, les outils bureautiques aux mains des utilisateurs suffiraient à combler les espaces laissés vides par les grands progiciels.

La réalité est toute autre. Les gros progiciels sont des outils précieux, mais ils réclament souvent un paramétrage très lourd. Il s'agit alors de traduire des spécifications fonctionnelles sous une forme technique, qui ne ressemble certes pas à Cobol, mais constitue tout de même une variété de code source. De plus, dès lors qu'une organisation veut différencier quelque peu son service, il lui faut souvent aller au-delà des fonctions offertes : d'où le recours à des développements avec des langages "classiques" du style Java, C ou C++.

Par ailleurs, l'accélération du cycle des technologies conduit souvent les organisations à entamer des projets avant que les outils de base ne soient totalement mûrs ou intégrés. Il faut bien alors réaliser l'intégration et, là encore, il y a du pain sur la planche pour les programmeurs. Quant à réutiliser telles quelles, dans des systèmes décisionnels, les données issues des systèmes de gestion, cela s'avère bien difficile du fait de la faible qualité des données existantes. Il faut donc réaliser des logiciels ad hoc chargés de filtrer et de corriger les données avant injection dans les datawarehouses.

Enfin, les outils en principe destinés aux utilisateurs avertis sont peu utilisés et c'est encore aux développeurs de réaliser les requêtes SQL ou Business Object ou les macros Excel un peu complexes.

Au total, les activités de développement ont encore de beaux jours devant elles. Elles ont seulement tendance à se complexifier et à réclamer des outils et des compétences de plus en plus pointus. Les technologies et les méthodes au service du développement sont donc toujours d'actualité.

Dans ce numéro, L'Informatique Professionnelle vous en fournit les principales clés.

Jean-Marc Berlioux

Développement

Développer sous Linux

p. 6

Comment faire, comment choisir ?

Le choix de la plate-forme Linux pour effectuer un développement logiciel est une décision importante. Mais, outre le système d'exploitation, il faudra aussi choisir le langage et les bibliothèques.

Christophe Blaess

Décisionnel

p. 11

La collecte des données est essentielle

La collecte des données est une étape majeure d'un projet d'informatique décisionnelle. Pour réussir, les règles et les objectifs ne doivent pas être perdus de vue.

Alain Fernandez

Orientations

p. 17

La réduction fonctionnelle s'impose

La suppression des fonctions non indispensables est un acte de saine gestion. En identifiant les fonctions dont le coût est supérieur à la valeur, l'analyse de la valeur permet de décider.

Jean-Marc Berlioux

Architectures à composants

p. 19

Savoir-faire et organisation

Le développement et l'implémentation d'une architecture à composants imposent une réorganisation complète et une coordination des acteurs du projet.

Gilles Mergoil

La mort de DCOM

p. 23

Un SOAP opéra !

La mort de DCOM ne fait plus de doute. En 2003, plus de 70 % des e-services invoqués au travers de l'internet le seront en utilisant SOAP de Microsoft.

Jean-Marc Lejeune

L'utilisateur

p. 25

Un acteur de qualité

Les exigences des consommateurs doivent être placées au cœur de la qualité des développements. Reste à qualifier leur vision et à endiguer leur immaturité.

Yves Constantinidis

e-business

Internauts

p. 30

Souriez, vous êtes pistés !

Attention, sur l'internet, la liberté est surveillée. Que ce soit à l'intérieur ou à l'extérieur des entreprises, des yeux et des oreilles observent en permanence le moindre de vos déplacements.

Eric Larcher

Management

Achats informatiques

p. 35

Diminuer les risques et négocier

Les achats informatiques coûtent cher. Pour diminuer les risques et les coûts, acheteurs et informaticiens doivent travailler de pair.

Gilles Blanchard

Ressources humaines

Concepteurs

p. 37

La double compétence est de rigueur

Pour gérer l'indispensable compromis entre les besoins, la technique et l'organisation, le concepteur doit disposer de la durée et se former aux impératifs des métiers.

André Bors

Télécommunications

Refacturation des réseaux

p. 41

La diplomatie s'impose !

Plusieurs méthodologies de refacturation des télécommunications existent. Une seule est vraiment efficace : la diplomatie. Etude de cas.

Edward Younker, David Neil

Arrêts et tendances

Indemnités de licenciement

p. 46

De la subjectivité à l'objectivité

Déclarations. Le régime fiscal des indemnités de licenciement a changé. Gare aux redressements !

Eric Boulanger

En bref

A suivre...

p. 49

Les e-mails sont-ils des correspondances privées ? • Portage de .NET sur Linux • USA : doublement des visas high-tech • Oracle9i : Marketing et poudre aux yeux • Batteries Dell : au feu ! • Internet et grandes oreilles : La Chine vous salue bien !

L'Informatique Professionnelle - Mensuel publié par Gartner • Directeur des rédactions : Olivier Le Gendre • Directeur de la publication : Norbert Miconnet
 Rédacteur en chef : Jean-Marc Berlioux • Rédacteur en chef délégué : Jean-Michel Atzel • Comité éditorial : Jean-Pierre Corniou, Serge Leclerc, Catherine Leloup, Christian Morfouace, Jacques Pantin, Frédéric-Georges Roux, Pierre Lora-Tonet, Serge Yablonsky • Secrétaire de rédaction/Maquette : Céline Holleville • Gestion des abonnements : Sylvie Garfalo • Siège social : Gartner, Immeuble Défense Bergères - TSA 40002 - 345 avenue Georges Clemenceau - 92882 Nanterre cedex 9 - Tél : 01 41 35 15 15 - Fax : 01 41 35 15 10 • Abonnements : France 2 450 FF HT (tva 2,10 %) - Etranger 2 660 FF • Commission Paritaire 61050 • RC 350 624 102 - SARL au Capital de 1.012.500 F • Imprimerie Moderne de Bayeux - Zone Industrielle - 7 rue de la Résistance - BP 133 - 14401 Bayeux cedex - Tél. 02 31 51 63 20.

Développer sous Linux

Comment faire, comment choisir ?



Christophe Blaess

Ingénieur en informatique, il intervient comme conseil et développeur auprès de compagnies aériennes, d'aéroports ou de centres d'études.

Pour assurer la réussite d'un développement applicatif sous Linux, il est important de bien cibler les objectifs. Ceux-ci conditionneront à la fois les outils et les langages de programmation à utiliser.

Le choix de la plate-forme Linux pour effectuer un développement logiciel est une décision importante

Le choix d'effectuer le développement d'un logiciel en langage C ou C++ sous Linux recouvre en réalité deux prises de décision. En premier lieu, on fait le choix d'un système d'exploitation. L'utilisation de Linux comme environnement de développement est une option nouvelle dans le milieu industriel. La seconde décision concerne le langage et les bibliothèques employées. D'autres langages sont en effet disponibles, et il faut être en adéquation avec le but final de l'application et avec le public visé. Nous allons examiner ici les possibilités qu'offrent les outils disponibles sous Linux pour une bonne réalisation logicielle en langage C ou C++.

Développement sous Linux

Nous avons indiqué que le choix de la plate-forme Linux pour effectuer un développement logiciel est déjà une décision importante. Plusieurs motivations peuvent être à l'origine de ce choix. Tout d'abord, Linux constitue probablement le meilleur système d'exploitation pour la diffusion de logiciels libres. L'esprit qui anime la communauté Linux est tel qu'un nouveau projet est généralement bien accueilli et l'enthousiasme des participants permet la progression rapide d'un logiciel, pour peu que l'équipe soit suffisamment motivée.

Ce type de choix n'est plus seulement le fait d'amateurs ou d'étudiants : cer-

taines entreprises ont déjà compris l'intérêt immédiat que présente pour elles l'adhésion à la logique des logiciels libres. C'est le cas, par exemple, pour certains fabricants de périphériques pour ordinateurs : ils diffusent les spécifications techniques de leur matériel et fournissent une assistance pour le développement de drivers libres permettant le fonctionnement sous Linux. Le choix du système Linux comme environnement de développement peut aussi être motivé par la qualité des outils présents "librement" et la richesse de la documentation disponible sur Internet. Dès son installation, une distribution Gnu/Linux comprend tous les outils logiciels nécessaires au programmeur – éditeurs, compilateurs, débogueurs – et ce pour l'ensemble des langages informatiques couramment employés.

De plus, le portage des applications vers d'autres environnements Unix est souvent très rapide, puisqu'il ne nécessite qu'une simple recompilation des fichiers source. Il est alors possible de disposer de stations de développement de type PC, puissantes et économiques. Notons également que Linux, loin d'être limité aux PC, est disponible sur un large éventail de machines. Les applications tournant sous ce système disposent ainsi automatiquement d'une portabilité "interne" sur une gamme complète d'architectures et de processeurs différents.

Cependant, la motivation la plus fréquente pour le développement sous Linux, est tout simplement le choix de ce système comme plate-forme applicative finale. Cette solution est depuis plusieurs années retenue dans de nombreux laboratoires, centres d'études, etc. L'utilisation de Linux comme environnement principal dans le domaine des applications industrielles est plus récente, mais une fois les réticences

initiales vaincues, on assiste généralement au sein d'une entreprise à une progression lente mais continue du parc de machines fonctionnant sous Linux.

Pour assurer la réussite d'un développement applicatif sous Linux, il est important de bien cibler dans quelle catégorie s'inscrit le produit : logiciel libre diffusé sur Internet ou réservé à un usage interne ; application destinée à être largement portée sur d'autres systèmes d'exploitation ou au contraire limitée à une utilisation uniquement sous Linux. Ceci influe à la fois sur les outils et langages de programmation à employer, ainsi que sur les buts prioritaires que l'équipe de développeurs devra se fixer.

Choix du langage

Dans le monde industriel, l'un des points forts les plus appréciés du système Linux est sa stabilité. Les stations configurées correctement peuvent présenter des temps de fonctionnement interrompu très longs. Les seuls arrêts du système sont alors dus aux composants matériels, et non au logiciel. D'où des taux de disponibilité du système particulièrement élevés, ce qui est très précieux dans certains types d'application.

L'exemple le plus fréquemment cité est celui du serveur web. Le logiciel libre Apache est très utilisé pour constituer des serveurs performants et peu coûteux sous Linux. Pour configurer et personnaliser un serveur réseau, on emploie souvent le langage Perl, qui permet d'écrire rapidement de petits scripts puissants, pour dynamiser le contenu des pages web, programmer des robots logiciels gérant des listes de diffusion ou encore trier et répartir le courrier ou les articles Usenet.

Pour des applications nécessitant une implication technique plus importante,

On assiste à une progression lente mais continue du parc de machines fonctionnant sous Linux

L'un des points forts les plus appréciés du système Linux est sa stabilité

Le noyau Linux et tous les utilitaires système sont écrits en C

les langages les plus utilisés sont le C et le C++. La rapidité et la puissance des applications produites constituent des arguments de choix. En contrepartie, la portabilité est un peu plus restreinte, essentiellement en ce qui concerne l'interface utilisateur.

En fait, le noyau Linux, ainsi que tous les utilitaires système, sont écrits en C. L'emploi de ce langage permet donc d'accéder facilement aux ressources de bas niveau indispensables dans certaines applications (interfaçage, écriture de pilote de périphérique, etc.) Le langage C offre des avantages techniques en termes de puissance et de rapidité convenant tout particulièrement aux applications scientifiques et industrielles. De plus, en tant que langage compilé, il fournit un fichier exécutable autonome, ce qui assure la confidentialité des algorithmes utilisés dans les sources. Cet aspect constitue, dans certaines circonstances, un avantage réel sur les langages interprétés.

L'environnement graphique utilisé sous Linux, comme pour les autres Unix, est le système X-Window. Ce système très puissant est particulièrement intéressant sur des machines en réseau, car il permet de déporter des affichages sur des stations distantes. X-Window n'impose que très peu d'obligations en terme d'aspect ou de comportement des applications. C'est pourquoi de nombreuses bibliothèques graphiques différentes coexistent, compliquant parfois quelque peu le portage des applications.

Sous Linux, deux environnements graphiques principaux existent actuellement : le système Gnome, issu du projet Gnu, et le système KDE. L'un comme l'autre sont installés automatiquement avec la plupart des distributions actuelles, et c'est l'utilisateur qui choisit son environnement favori lors de la connexion. Ces environnements

jouent sur l'aspect et le comportement des fenêtres (menu système, bordures, réaction à la souris, etc.), mais ils ont aussi un rôle important dans la communication entre les applications.

Choisir d'utiliser l'environnement Gnome ou KDE pour l'interface utilisateur d'une application n'est donc pas une action innocente. D'autant qu'il faut alors être conscient que la portabilité de l'application hors de l'univers Linux est largement diminuée. Une autre possibilité est d'employer une bibliothèque portable sur l'ensemble des systèmes Unix. L'environnement Motif s'est imposé comme standard de fait dans les systèmes informatiques industriels. Il en existe plusieurs implémentations commerciales sous Linux, ainsi qu'une implémentation libre nommée Lesstif. L'utilisation de ces bibliothèques permet d'assurer une certaine portabilité des applications graphiques, tout en conservant l'efficacité et les performances d'un langage compilé proche de la machine.

Outils de développement

Les outils de développement disponibles librement sous Linux sont étonnamment puissants. Au cœur de la programmation en langage C ou C++ se trouve naturellement le compilateur. Celui que l'on emploie sous Linux se nomme GCC. Il s'agit d'un compilateur produit par le projet Gnu, dont le but est de fournir librement des logiciels de remplacement pour tous les utilitaires système et outils de développement sous Unix. Le compilateur GCC n'a rien de spécifique à Linux, il est disponible sur de nombreux autres Unix, et un portage a même été réalisé sous Dos. La qualité de GCC est telle que certains le préfèrent aux compilateurs natifs vendus sur les stations Unix classiques. Il représente également une alternative intéressante aux

Choisir Gnome ou KDE n'est pas une action innocente

compilateurs très coûteux, puisqu'il permet d'équiper facilement et à moindre frais toutes les stations des développeurs d'une équipe. GCC se décline en plusieurs versions installées conjointement, selon que l'on préfère utiliser le langage C, le C++, voire le dialecte Objective C. Les très nombreuses options disponibles pour ajuster la configuration du compilateur permettent d'adapter son comportement aux désirs du programmeur, notamment en ce qui concerne les respects des normes les plus courantes (Posix, Iso C99, etc.)

Le compilateur GCC est un outil fonctionnant en ligne de commande, qu'on peut donc lancer directement "à la main" en indiquant les options désirées et les fichiers sources à compiler. Cet utilitaire n'effectue qu'une seule tâche, mais il la réalise au mieux, sans que des considérations d'interface utilisateur ne viennent le distraire de son rôle. Pour améliorer l'ergonomie du poste de travail programmeur, il existe toutefois un certain nombre d'environnements de développement intégré, qui regroupent un éditeur de texte pour la saisie des fichiers sources et des composants graphiques (boutons, menus...) pour invoquer directement le compilateur en arrière-plan, en s'assurant de l'emploi des options correctes. Ces environnements permettent aussi l'appel du débogueur Gnu, nommé GDB. À l'instar du compilateur, cet outil fonctionne par une invocation depuis une ligne de commande puis il offre une interface utilisateur minimale. Les environnements de développement intégré l'encadrent et présentent une interface graphique plus attrayante, en permettant l'exécution d'un programme instruction par instruction, l'inspection des variables, etc.

Le projet Gnu regroupe une panoplie complète d'outils destinés à assister le

travail du développeur. Citons, par exemple, les utilitaires d'archivage permettant de créer des bibliothèques de fonctions personnalisées, les enjoliveurs de code source pour uniformiser l'aspect des fichiers créés par les différents membres d'une équipe, les outils RCS ou CVS qui assurent le contrôle de version et permettent ainsi le travail simultané par plusieurs développeurs sur le même projet en minimisant les risques de modifications concurrentielles.

Documentation et exemples

La richesse de la documentation disponible concernant l'environnement Linux fait partie des caractéristiques de ce système qui sont les plus attrayantes pour les programmeurs. On peut trouver sur Internet des sites regroupant des informations de grande qualité dans tous les domaines intéressant le développeur.

En ce qui concerne l'environnement Linux en général, l'installation et la configuration des applications ou l'administration du système, on consultera les documents HOW-TO, traduits en français, qui regroupent des informations sur la plupart des sujets importants pour l'utilisateur. Pour vérifier la syntaxe d'appel d'une routine de la bibliothèque C, pour connaître la signification d'un appel système (c'est-à-dire un service offert par le noyau Linux lui-même) ou encore pour étudier le détail de toutes les options proposées par une application, il existe une commande MAN dont l'invocation est très commode. Elle donne accès à des fichiers indépendants, que l'on nomme habituellement "pages de manuel" par analogie avec les classeurs de documentation livrés avec les Unix traditionnels. Chaque page (qui peut représenter parfois une dizaine de pages imprimées) contient la documen-

Les très nombreuses options disponibles pour ajuster la configuration du compilateur permettent d'adapter son comportement aux désirs du programmeur

Le projet Gnu regroupe une panoplie complète d'outils destinés à assister le travail du développeur

Les fonctionnalités offertes par le noyau Linux et les routines de plus haut niveau proposées par la bibliothèque C standard ne sont que rarement suffisantes

tation pour une routine de la bibliothèque C, un appel système ou un utilitaire. Il existe environ mille pages de manuel pour Linux actuellement traduites en français.

Le programmeur pourra toujours consulter à son gré les sources du noyau ou celles de la bibliothèque C, ce qui représente non seulement une source d'information exacte en ce qui concerne une fonctionnalité précise, mais également des exemples performants et souvent élégants, d'implémentation d'algorithmes⁽¹⁾.

Les fonctionnalités offertes par le noyau Linux et les routines de plus haut niveau proposées par la bibliothèque C standard ne sont que rarement suffisantes pour la réalisation d'une application complète. Nous avons déjà abordé la question de l'interface graphique généralement indispensable. Mais il est souvent nécessaire de faire appel à d'autres bibliothèques réalisant des tâches diverses. Citons, par exemple, l'existence de bibliothèques intégrant des algorithmes de compression, de cryptage, de lecture de fichiers graphiques dans la plupart des formats usuels. Citons aussi la possibilité d'accéder aux objets disponibles sur un bus Corba ou encore d'utiliser des passerelles vers d'autres langages de développement comme Python, Perl ou Tcl/Tk. Toutes ces bibliothèques sont disponibles sous forme de code source librement consultable et adaptable à des besoins spécifiques.

Pour conclure cette présentation rapide, j'aimerais insister sur l'aspect intellectuellement motivant que représente, pour un programmeur, la possibilité de travailler dans un environnement où il est possible d'avoir accès au fonctionnement interne du système dans ses moindres détails, où rien ne reste dissimulé à qui désire obtenir des informations, et où la qualité des outils et l'as-

sistance disponible via Internet font que l'on ne reste jamais bloqué face à un dysfonctionnement d'un utilitaire système.

Christophe Blaess
ccb@club-internet.fr

BIBLIOGRAPHIE

Programmation système en C sous Linux, Christophe Blaess, éditions Eyrolles.

Sur le Net

<http://perso.club-internet.fr/ccb>

C'est le site de Christophe Blaess sur lequel on peut retrouver notamment les pages de manuels Linux traduites en français. Autres adresses utiles :

[www.linux-center.org/ fr](http://www.linux-center.org/fr/) : liens de documentation française sur Linux

www.linux.org/ : ensemble de liens internationaux sur Linux

www.gnu.org/home.fr.html : le projet Gnu

Votre avis...

**Votre avis
nous
intéresse,**

écrivez-nous !

atzeljm@worldnet.fr

Revue d'auteurs, l'Informatique Professionnelle accueille des opinions qui n'engagent pas la rédaction.

J'aimerais insister sur l'aspect intellectuellement motivant d'avoir accès au fonctionnement interne du système dans ses moindres détails

1 - Ceci est d'ailleurs vrai pour l'ensemble des applications disponibles sous Linux, la plupart des distributions fournissant d'emblée plusieurs centaines de mégaoctets de fichiers sources immédiatement consultables.

Décisionnel

La collecte des données est essentielle

Capitale, la collecte des données d'un système décisionnel est une opération particulièrement délicate. Accessibilité, nettoyage, consolidation, qualité, traçabilité et coûts ne doivent jamais être perdus de vue. Les freins et les réticences des personnes non plus !



Alain Fernandez

Consultant indépendant, il intervient depuis plus de 15 ans auprès des grands comptes et des PME sur la conception des systèmes d'information stratégiques

La collecte des données dans l'entreprise est une étape majeure du projet d'informatique décisionnelle. Lors des premières réalisations (projet EIS, Infocentre ou Data Warehouse), les concepteurs traitaient un peu rapidement cette question. Ils partaient du postulat qu'il "suffisait" de rapatrier un maximum de données en un point accessible. Le décideur serait alors à même de dénicher les informations pertinentes et essentielles cachées au sein des bases de données. Le mythe du décideur-orpailleur des bases informationnelles a perduré.

Ce mythe est à l'origine de nombreux systèmes décisionnels inutilisables. Pour éviter l'échec d'un tel projet, il ne

faut donc pas se limiter exclusivement aux aspects techniques. La question de la collecte des données doit être reconsidérée pour prendre en compte les besoins fondamentaux en terme d'utilisation des données.

Adopter une démarche projet

Par définition, toutes les données de l'entreprise n'ont pas la même valeur pour tout le monde. Cette notion de valeur est fortement dépendante de l'usage et de l'utilisateur. Certaines données seront importantes pour un utilisateur en fonction de ses préoccupations et insignifiantes pour d'autres.

Avant de commencer les opérations de collecte, il est donc indispensable

Toutes les données de l'entreprise n'ont pas la même valeur pour tout le monde

Le vieil adage " Qui peut le plus peut le moins " est souvent faux en informatique

Les systèmes d'information contiennent de nombreuses données erronées et inutilisables sur le plan décisionnel

SUR LE TERRAIN

Il y a quelque temps, je suis intervenu comme " pompier " sur un projet de Data Warehouse en dérouté, censé répondre à des besoins de pilotage et d'études clientèle dans le service marketing d'un équipementier. La base construite comportait d'énormes quantités de données techniques relatives à la conception et à la fabrication des produits. Ces données, sûrement riches d'enseignements pour les hommes du bureau d'études, de la production ou des méthodes, demeuraient purement absconses pour les gens du marketing. Pourquoi les avait-on placées dans la base décisionnelle ? Tout simplement parce que ces données étaient déjà structurées et aisées à collecter. Personne ne s'était préoccupé de l'enseignement que pouvaient en tirer les utilisateurs, en l'occurrence les gens du marketing, par nature peu versés dans les questions de mécanique.

d'identifier avec précision les besoins des utilisateurs et les limites du projet. Il ne sert à rien de mettre à la disposition des utilisateurs potentiels des quantités faramineuses de données fort éloignées de leurs préoccupations.

Une donnée devient une information lorsqu'elle est porteuse de sens. La perception du sens dépend des utilisateurs et des préoccupations du moment. Nous ne sommes pas égaux devant l'information et cette dernière n'est pas porteuse d'un sens universel. Seuls les utilisateurs, en fonction de leurs préoccupations, sont à même de transformer les données en informations. La simplicité de l'accessibilité technique ne doit pas être l'unique critère de collecte. Il faut adopter une démarche projet afin de bien cerner les objectifs et les frontières des différents domaines d'intervention en fonction du besoin présent. La base décisionnelle pourra par la suite être enrichie, au fur et à mesure des nouveaux projets. Ne perdons pas de vue que le vieil adage "Qui peut le plus peut le moins" est souvent faux en informatique.

Collecte, Consolidation et maîtrise des coûts

Une fois la problématique bien définie et les domaines d'intervention circonscrits, la question purement technique de la collecte peut alors être abordée. En règle générale, les concepteurs se heurtent à trois difficultés :

- l'accessibilité des données en raison de l'hétérogénéité du système d'information,
- le nettoyage des erreurs et aberrations contenues dans les bases,
- la consolidation nécessaire pour rendre les données utilisables.

Les systèmes d'information de nos entreprises n'ont pas été conçus en six jours. L'approche a toujours été parcelaire et étalée dans le temps. Chaque projet était lancé pour répondre à un besoin fonctionnel précis et ponctuel, limité le plus souvent à une unité, une division ou un service. Personne ne considérait à leur juste mesure les besoins futurs en terme d'accès aux données essentielles. Notons que les différents rapprochements, rachats et fusions d'entreprises viennent complexifier encore la question de la cohérence du système d'information.

Les opérations de nettoyage constituent la seconde difficulté rencontrée par les concepteurs. Les systèmes d'information contiennent de nombreuses données erronées et inutilisables sur le plan décisionnel (par exemple, des erreurs, des valeurs aberrantes, des omissions, etc.). Les applications de production pleinement opérationnelles et recueillant la satisfaction des utilisateurs n'en sont pas exemptes.

La présence d'erreurs au sein de systèmes d'information parfaitement rodés s'explique simplement. Les données qui n'influencent pas les résultats,

n'ont pas à être vérifiées trop soigneusement. Cependant, ces données peu significatives pour les tâches de production sont peut-être porteuses d'un sens informationnel beaucoup plus riche pour les décideurs. Lors de la collecte, il faudra contrôler la validité de toutes les données devant jouer un rôle décisionnel dans le projet. On ne peut se permettre de laisser les utilisateurs prendre des décisions à partir d'informations erronées car dans ce cas le système serait très rapidement mis au rebut.

Les travaux de consolidation constituent la troisième difficulté à considérer. Du fait des anciennes habitudes de cloisonnement et de découpage des entreprises en centres de profits autonomes, les règles de gestion élémentaires sont rarement standardisées au niveau du groupe. Ceci est d'autant plus vrai lors de fusion et de rachat, avec le rapprochement d'entreprises conservant leurs propres modes de calcul. Pensons simplement aux différentes façons de calculer un chiffre d'affaires. Chaque entreprise utilise sa propre méthode (avec ou sans les ristournes, les escomptes, les commissions, etc.).

Comment peut-on intégrer ou comparer les éléments d'activité lorsque ceux-ci sont calculés différemment ? Il est du ressort de l'architecte du projet de mettre à la disposition des décideurs des données cohérentes.

Ainsi, lors de la collecte, il faudra résoudre chacun de ces points. La tâche est conséquente et grèvera significativement les budgets "financier" et "temps" du projet. Pour éviter de se lancer dans des opérations trop complexes, il faudra toujours garder en ligne de mire le paramètre coût. Pour chaque point difficile de la collecte, nous nous poserons alors la question de la rentabilité en mettant en regard l'ap-

port sur le plan décisionnel des données à collecter et le coût de l'opération. Bien que fortement subjectif (il est difficile d'exprimer a priori l'apport décisionnel d'une information), ce questionnement permettra d'éviter les dépassements de coûts et de délais inconsidérés.

Une gestion centralisée

L'accroissement de la quantité des données en circulation dans les entreprises suit une loi exponentielle. Avec l'incertitude ambiante et la rapidité du changement, les hommes sont avides d'informations et chaque unité produit de plus en plus de données. Il est désormais temps de gérer les données non seulement en termes d'accessibilité mais aussi en termes de qualité et de traçabilité.

De plus en plus d'acteurs de l'entreprise sont concernés par la prise de décision. Le processus de banalisation de la fonction décisionnelle va rapidement s'accélérer avec l'essor attendu des portails informationnels d'entreprise (EIP). Déjà, les projets décisionnels se multiplient. Que ce soit pour des questions d'aide au pilotage, de gestion de relation client ou d'analyse qualité, de plus en plus d'applications voient le jour. Il existe une forte tendance à la mise en œuvre de Datamarts spécifiques pour répondre localement à un problème posé. Paradoxalement, cette solution satisfaisante quant aux résultats, contribue fortement à la dispersion des données dans l'entreprise. Ainsi, on peut retrouver dans des bases décisionnelles distinctes, des données redondantes à différents stades de transformation et de fraîcheur.

Pour limiter les erreurs et plus généralement pour mieux maîtriser les coûts de nettoyage et de mise en forme des données, il est temps de placer un observatoire centralisant non pas les

Il est du ressort de l'architecte du projet de mettre à la disposition des décideurs des données cohérentes

Il est temps de gérer les données en termes de qualité et de traçabilité

Il faut créer un observatoire centralisant non pas les données mais leur(s) définition(s), leur(s) parcours et leurs utilisations

Pour prendre les décisions, il faut pouvoir accéder à des données situées au sein de bases dépendantes d'autres activités et d'autres services

LES OUTILS DE COLLECTE

Les outils comme Datastage d'Ardent ou Genio de Hummingbird automatisent la collecte et le nettoyage des données provenant des bases de données et des ERP les plus courants. Une fois le schéma d'extraction modélisé pour les différentes sources, l'utilisateur définit les transformations nécessaires pour rendre les données disponibles à des fins décisionnelles (calculs d'agrégat, contrôle des valeurs, élimination des aberrations...). Ces outils gèrent aussi le référentiel de méta-données centralisé. La grande majorité des bases sont accessibles en mode natif et il est possible de développer sa propre passerelle spécifique pour les sources de données non supportées par le produit.

données mais leur(s) définition(s), leur(s) parcours et leurs utilisations. Pour cela, Il faut au préalable construire un référentiel centralisé à même de répondre à des questions du genre : d'où provient cette donnée ? Quand est-elle mise à jour ? Comment est-elle calculée ? Quelles sont les précautions d'usages ?

On appelle "métadonnées" les données sur les données apportant des réponses à ces questions. Pour une parfaite gestion, tous les modules fonctionnels du système d'information assurant le stockage, l'extraction, le traitement et la présentation, devraient utiliser et mettre à jour le référentiel.

Jusqu'à récemment encore, la définition d'un format standard demeurait le dernier obstacle à la généralisation des métadonnées. Deux formats étaient en lice : l'OIM (Open Information Model) proposé par le Meta Data Coalition et soutenu par Microsoft, à l'origine de sa définition, et le CWM (Common Warehouse MetaData) supportée par l'OMG (Object Management Group), Oracle et IBM, entre autres. Les deux formats s'appuient sur UML (Unified Modeling Language) pour la phase de

modélisation et XML (Extensible Markup Language) pour les formats de description et d'échanges. Fin septembre 2000, l'OIM a choisi de baisser les bras et d'abandonner sa norme pour contribuer à la prochaine version de l'OMG, le futur format unifié.

Décloisonnement et pouvoirs "parallèles"

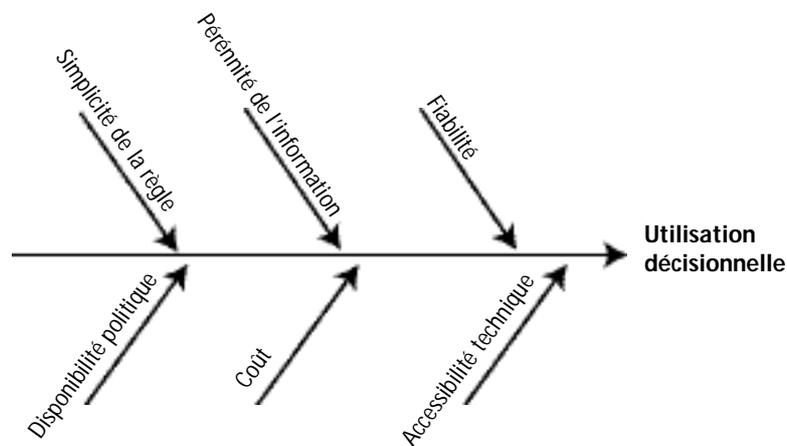
Les besoins en matière de collecte ne se limitent pas aux données présentes dans la sphère de prérogatives des utilisateurs du système décisionnel. Pour prendre les décisions, il faut pouvoir accéder à des données situées à l'extérieur, au sein de bases dépendantes d'autres activités et d'autres services. Là encore, il ne faudra pas limiter notre étude à l'aspect exclusivement technique mais bien s'informer auprès des producteurs et utilisateurs habituels des données concernées.

A ce stade, on aborde la difficulté majeure du projet. Prenons un exemple : récupérer les chiffres censés décrire l'activité d'une filiale peut présenter quelques difficultés techniques qui demeureront rarement insurmontables. Ces données seront alors dispo-

GERER LA DONNEE A PRIORI

Dans tous les cas, il faudra envisager la potentialité informationnelle des données gérées par les applications de production dès le lancement des projets système d'information. Nous perdons beaucoup trop de temps à récupérer des données peut-être secondaires pour une application de production mais importantes pour les besoins décisionnels. Rappelons-nous qu'une des principales récriminations des utilisateurs d'ERP concernait l'absence d'éléments décisionnels. Avec l'essor de l'e-business et plus généralement de l'importance stratégique liée à l'éclatement de l'entreprise et au développement de la supply-chain, l'information joue un rôle clé.

FIGURE 1 - LES CRITERES DE SELECTION DES INFORMATIONS



(D'après Fernandez, les Nouveaux Tableaux de bord des décideurs)

Le culte de la compétition individuelle encourage les hommes à thésauriser leurs informations

nibles. En revanche, nous ne disposons d'aucune garantie sur leur validité et surtout sur les précautions d'usage à respecter avant d'en tirer un enseignement.

Chaque filiale, chaque service, chaque activité définit ses propres règles de gestion. Avant d'utiliser une donnée et de chercher à en extraire une quelconque information, il est préférable au préalable d'en référer aux détenteurs de cette connaissance, les seuls à même de nous informer et de nous mettre en garde. Mais les détenteurs de cette connaissance seront-ils prêts à la communiquer à d'autres décideurs ? C'est toute la question du décloisonnement et du partage de la connaissance qui se pose. Le culte de la compétition individuelle, encore en vigueur dans de trop nombreuses entreprises, encourage les hommes à thésauriser leurs informations pour asseoir plus confortablement leurs pouvoirs. Ils ont, le plus souvent, tout intérêt à ne pas communiquer les recommandations nécessaires à l'usage des données... Surtout si personne ne le leur demande explicitement !

Cette question cruciale est à traiter avec précaution. D'autant plus que l'information n'est pas répartie uniformément dans l'entreprise. Certaines

personnes placées aux nœuds d'informations disposent d'un pouvoir de fait significatif. L'architecte du système décisionnel devra faire preuve de psychologie pour inciter les hommes à communiquer et éviter de heurter de front les sensibilités. Il prendra aussi garde à ne pas asseoir encore plus confortablement le pouvoir des hommes clés en entérinant techniquement les domaines informationnels.

Sur ce type de projet, l'engagement de la direction est indispensable ! Sans son réel appui, le projet est irréalisable. Seuls les dirigeants pourront expliquer les avantages de la coopération dans un esprit gagnant-gagnant au niveau de l'entreprise. Et pour les causes perdues, ils seront aussi les seuls en mesure de "faciliter" l'ouverture des portes des rebelles à l'entreprise communicante.

Sélectionner les données

Prenons le cas de la construction de tableaux de bord de pilotage. Pour chaque indicateur inclus⁽¹⁾, nous sélectionnerons les données nécessaires à sa construction en valorisant chacun des critères essentiels (voir figure 1). Il s'agit d'un travail de groupe, et chacun donnera son avis pour chaque critère à l'aide d'une note comprise entre 1 et 4.

L'architecte du système décisionnel devra faire preuve de psychologie pour inciter les hommes à communiquer

1 - Pour la méthode de choix des indicateurs, se référer à l'article du même auteur dans le numéro 176 de l'Informatique Professionnelle.

Orientations

La réduction fonctionnelle s'impose



Jean-Marc Berlioux

Directeur des programmes de Bouhot & Le Gendre. Rédacteur en Chef de L'Informatique Professionnelle. Il intervient depuis plus de vingt ans dans le domaine des systèmes d'information.

Pour tenir les délais, les budgets et les coûts, alors que les développeurs expérimentés sont de plus en plus rares et chers, mieux vaut remettre à plus tard la réalisation des fonctions non indispensables. L'analyse de la valeur est un outil qui permet de choisir en connaissance de cause.

Les responsables de développements applicatifs le savent bien : une bonne partie des fonctionnalités réclamées par les maîtrises d'ouvrage ne servent finalement pas. Il s'agit d'un gaspillage manifeste. Du coup, les fonctions utiles elles-mêmes sont pénalisées : elles coûteront plus cher, arriveront plus tard, comporteront plus de bogues, etc. Or nous entrons dans une période où la tenue de délais courts va devenir de plus en plus un critère majeur de succès des projets...

Pendant qu'on y est...

Très souvent, la maîtrise d'ouvrage cède à la tentation de "profiter de l'occasion", "pendant qu'on y est", pour

raccrocher au projet tout ce qui peut avoir un quelconque rapport, proche ou lointain, avec lui. Certaines maîtrises d'ouvrages ne sont pas sûres de pouvoir relancer un autre projet de sitôt. Elles préfèrent donc réclamer des fonctions sans être certaines de leur utilité, mais "au cas où" elles en auraient besoin d'ici leur prochain projet.

D'autres maîtrises d'ouvrages, confrontées à l'angoisse légitime du risque d'erreur dans la conception d'un nouveau système, essaient de se garantir en "taillant large". D'autres encore ont une envie latente de perfection et rêvent d'un système d'information idéal. D'autres demandent toutes les fonctions qui leur paraissent utiles,

Certaines maîtrises d'ouvrages préfèrent réclamer des fonctions sans être certaines de leur utilité

Quand on ne paye pas les fonctions supplémentaires, on peut être tenté de s'attribuer une part un peu plus grande du gâteau informatique

sans se rendre compte que la valeur produite par certaines fonctions sera inférieure à leur coût. D'autres, enfin, se trompent de bonne foi sur la valeur de certaines fonctions, dans le cas de la mise en place d'un nouveau processus, par exemple.

Le phénomène ne date pas d'hier. On aurait pu espérer que, à force d'expériences douloureuses, les utilisateurs auraient su en tirer les leçons et s'imposer à eux-mêmes une modération pragmatique.

Mais les maîtrises d'ouvrage sont souvent tentées de justifier a posteriori les fonctions supplémentaires, soit par l'évolution du monde dans l'intervalle entre la conception et la livraison de l'application, soit par une application avisée du principe de précaution. En fait, il s'agit là, le plus souvent, de modes de défense. Le fait est que les méthodes de conception "classiques" les ont autorisés à se laisser aller à ce genre de dérive.

L'absence de refacturation du coût des projets aux maîtrises d'ouvrage est d'ailleurs un facteur d'aggravation de cette tendance. Quand on ne paye pas les fonctions supplémentaires, on peut être tenté de s'attribuer une part un peu plus grande du gâteau informatique (au détriment des autres maîtrises d'ouvrage).

Une part notable des demandes des utilisateurs est donc largement irrationnelle. Les concepteurs fonctionnels doivent être conscients de cette tendance et disposer d'un outil méthodologique pour lutter contre les dérives qu'elle implique.

L'analyse de la valeur

L'analyse de la valeur fournit une base méthodique à la réduction fonctionnelle. Le principe de base est le suivant : à chaque fonction est associée une "valeur" estimée.

On évalue ensuite le coût de la fonc-

tion. Ce coût inclut les charges de développement, d'installation, de formation, etc. spécifiques à la fonction, mais aussi l'impact de cet ajout sur les délais de livraison de l'ensemble, le risque de dégradation de la fiabilité et de la qualité de l'application, l'allongement de l'apprentissage, etc.

Toutes les fonctions dont le coût est supérieur à la valeur sont éliminées ou reportées. Au total, on essaie donc d'optimiser le rapport valeur/coût.

Des fonctions auxiliaires attrayantes

Il arrive qu'une démarche de réduction fonctionnelle parvienne à réduire les coûts et les délais à un niveau inférieur au budget. Deux options sont alors envisageables. Soit repêcher quelques fonctions éliminées de justesse. Soit introduire des fonctions destinées à répondre à des besoins latents importants.

L'introduction de telles fonctions, pour autant qu'elles soient bien choisies, peut permettre de renforcer sensiblement la satisfaction des utilisateurs. Il n'existe pas de baguette magique garantissant le choix pertinent de fonctions destinées à répondre à des besoins latents. Cependant, on peut rappeler que les ressorts qui sous-tendent la satisfaction "supplémentaire" des utilisateurs tournent le plus souvent autour de la communication, de l'accélération, de la liberté de créer des fonctions auxiliaires (par exemple d'extraction ou de représentation), de l'aide à la compréhension des faits ou des données, de la réduction des travaux fastidieux et du confort d'utilisation.

A titre d'exemple, une interface permettant de récupérer des données dans des outils bureautiques de type Excel est souvent un facteur de satisfaction.

Jean-Marc Berlioux

Toutes les fonctions dont le coût est supérieur à la valeur sont éliminées ou reportées

Architectures à composants

Savoir-faire et organisation



Gilles Mergoil

Président de Neoxia
(www.neoxia.com).
Cabinet de conseil
en architectures
informatiques
distribuées.

Développer et implémenter une architecture à composants obligent à repenser les méthodes et les outils. Rien de plus simple ! Le vrai défi réside dans la nouvelle organisation des équipes de développement.

L'intérêt actuel que suscitent les architectures distribuées aussi appelées "architectures à composants" démontre que fournisseurs et utilisateurs de technologies ont su capitaliser sur les succès d'hier et tirer enseignement des erreurs passées.

En effet, nous disposons aujourd'hui de solutions qui concilient la fiabilité des systèmes centralisés avec la souplesse des systèmes client-serveur, et ce sans hériter de la rigidité des premiers ou des coûts élevés de déploiement des seconds.

Parmi ces nouvelles architectures, les offres J2EE⁽¹⁾ de Sun et DNA⁽²⁾ de Microsoft encouragent la centralisation des objets métiers, la factorisation des

services techniques -sécurité, déploiement, ...- tout en permettant une forte réactivité aux nouveaux besoins des utilisateurs.

L'objectif de ces nouvelles offres est simple : permettre aux équipes informatiques de concentrer leur attention sur les attentes des utilisateurs en s'abstrayant de la complexité technique de mise en œuvre des nouveaux services.

Cette louable intention n'est certes pas nouvelle, elle est simplement aujourd'hui devenu réalisable... en s'entourant d'un minimum de précautions !

Marier l'ancien et le moderne

Un bref rappel de l'évolution des architectures permet de se rendre compte

Nous disposons aujourd'hui de solutions qui concilient la fiabilité des systèmes centralisés avec la souplesse des systèmes client-serveur

1 - Java 2 Enterprise Edition.
2 - Distributed interNet Architecture.

Les architectures client-serveur ont accru le pouvoir des équipes métiers

que ces fameuses “nouvelles technologies” tirent une partie de leur légitimité de techniques plus anciennes, largement éprouvées et extrêmement efficaces lorsqu’elles sont utilisées à bon escient.

Il n’est nul besoin de rappeler les qualités des systèmes centralisés en matière de sécurité, de support transactionnel et, dans une certaine mesure, de “scalabilité”⁽³⁾. Longtemps critiqués, ils peuvent tirer gloire aujourd’hui de l’incapacité des architectures client-serveur à les égaler sur ces points précis.

Pour leur part, les architectures client-serveur ont contribué fortement à élargir la place qu’occupe l’informatique dans nos entreprises. Elles ont accru le pouvoir des équipes métiers sur les systèmes d’information ... mais probablement un peu trop.

En privilégiant l’utilisateur final, nous avons obtenu des progrès immenses en terme d’ergonomie et acquis une liberté quasi totale pour accéder à l’information et la traiter.

Mais ce transfert du centre de décision du technique vers le fonctionnel est souvent à l’origine de nombreux blocages. Il a eu les conséquences que nous connaissons : failles de sécurité, incohérence de l’information, manque d’interopérabilité, incapacité à monter en charge, coûts et délais de déploiement élevés.

Les nouvelles architectures nous proposent de concilier le meilleur des technologies précédentes et surtout de mieux définir le partage des rôles entre équipes techniques et équipes métiers. D’une architecture à base d’applications, le système d’information doit passer à une architecture en services métiers.

A chaque domaine métier significatif de l’entreprise correspond un service métier qui factorise un savoir-faire sous forme d’un ensemble d’objets

métiers. Les objets et services métiers viennent s’intégrer dans une architecture mettant à leur disposition des services techniques de haut niveau à travers des interfaces normalisées et – plus ou moins – universelles.

Les applications traditionnelles sont remplacées par des applications légères, qui intègrent différents services métiers pour proposer à l’utilisateur les fonctionnalités attendues.

Des ORB aux serveurs d’applications

La mise en place de telles architectures pose des problèmes de conception classiques mais également de nombreux problèmes d’ordre technique. Parmi les défis majeurs, on trouve la sécurité de niveau applicatif, la prise en compte de la montée en charge, l’intégration de l’existant, mais également la garantie de l’intégrité des données et la localisation des services et des objets métiers.

Ces dernières années ont vu l’émergence des ORB⁽⁴⁾ incarnés par la spécification CORBA⁽⁵⁾ et ses implémentations d’une part et la technologie Microsoft DCOM⁽⁶⁾ d’autre part.

Ces ORB ont commencé à offrir des moyens techniques permettant la mise en place d’architectures en services métiers. Pourtant, ces technologies se sont rapidement révélées techniquement très complexes et difficiles à mettre en œuvre dans l’entreprise. Ce constat a contribué à l’apparition récente des serveurs d’applications qui ont pour objectif de faciliter le développement des objets métiers, en éloignant le plus possible tous les problèmes techniques de bas niveau.

Les serveurs d’applications sont représentés sur le marché par l’architecture Microsoft DNA d’une part et la spécification Java 2 Entreprise et ses implémentations d’autre part.

Des serveurs d’applications ont pour objectif de faciliter le développement des objets métiers en éloignant le plus possible tous les problèmes techniques de bas niveau

3 - La “scalabilité” (*scalability* en anglais) est la capacité à supporter une montée en charge.

4 - Object Request Broker : middleware orienté objet permettant le dialogue inter-processus entre objets.

5 - Common Object Request Broker Architecture.

6 - Distributed Component Object Model.

Les serveurs d'application

Un serveur d'applications intègre un ensemble de services nécessaires au fonctionnement des applications distribuées.

Le service d'annuaire rend possible l'indépendance à la localisation. Il permet de localiser un service ou un objet métier à partir de son nom. Le service de transaction permet de garantir l'intégrité des données. Il rend possible les transactions distribuées. Ce type de service est déjà connu sous le nom de moniteur transactionnel. Le service de sécurité de niveau applicatif permet de déclarer de manière centralisée les autorisations d'utilisation des objets métiers et ceci jusqu'au niveau des méthodes. Le service de message permet la communication, sous forme de message, entre composants du système d'information. Ce service est un représentant de la famille des MOM⁽⁷⁾. Le service de déploiement permet l'installation des objets métiers et autres composants dans le serveur d'application. Le service web, qui n'est ni plus ni moins qu'un serveur HTTP, permet l'accès au système d'information par une interface web.

C'est l'intégration de tous ces services, plutôt que chaque service pris isolément qui fait le véritable intérêt des serveurs d'application. Enfin, la capacité des serveurs d'application à se mettre en grappe rend possible la montée en charge et la tolérance à la panne, et ceci sans développement supplémentaire. C'est également une de leurs grandes forces.

Le mythe de la programmation distribuée

Pour ceux qui ont fait ou feront le choix d'adopter les architectures à composants, la difficulté provient moins du développement que de la capacité à comprendre les concepts fondamentaux de l'informatique distribuée.

L'évolution des technologies, en informatique comme ailleurs, est ainsi faite qu'il ne s'agit pas toujours de "faire comme avant, plus facilement" mais plutôt de penser différemment !

Une croyance répandue est que l'on programme une application distribuée exactement comme on le fait en local, la technologie se chargeant de tout. Il n'en est rien. Les appels aux méthodes métiers se font via le réseau. Les temps d'appel de bout en bout sont donc considérablement plus longs qu'en programmation locale. Ceci reste vrai même si l'appelé et l'appelant sont logés sur la même machine, l'appel donnant lieu à une coûteuse communication inter-processus. Cet élément technique fondamental est à prendre en considération dès la conception afin d'éviter de construire un système d'information aux performances inacceptables.

Ce simple exemple met en lumière le besoin d'une nouvelle répartition des rôles au sein de l'organisation en charge du système d'information.

Les équipes projet au service des utilisateurs doivent bénéficier d'une architecture technique fiable et de procédures de développement et d'intégration. Elles doivent également disposer d'un support technique pointu pour éviter de mettre en péril la bonne marche du système.

Maîtrise d'ouvrage : priorité au métier

Penser différemment met en cause l'organisation des équipes et redistribue les rôles et les responsabilités parmi les acteurs, tant techniques que fonctionnels. Les équipes fonctionnelles vont ainsi devoir penser et formaliser leur savoir-faire et leurs attentes en tenant compte des aspects dynamiques de leur métier. Elles doivent définir les bases pérennes de leur métier (les objets métiers persistants, c'est-à-dire stockés et parta-

La difficulté provient moins du développement que de la capacité à comprendre les concepts fondamentaux de l'informatique distribuée

Les équipes fonctionnelles vont ainsi devoir définir les bases pérennes de leur métier

⁷ - Message Oriented Middleware.

L'ingénierie de projet est prise en charge par des équipes disposant d'une compétence technique intermédiaire

gés), ainsi que les processus répétitifs (les services métiers fondamentaux utilisés par tous). Elles doivent également formaliser leurs besoins spécifiques et court terme (objets et services volatiles, à durée de vie limitée et peu réutilisables) qui donneront lieu simplement à de nouvelles interfaces vers d'autres services plus stables.

Pour cela, les acteurs fonctionnels devront apprendre à utiliser un formalisme universel comme UML et ses cas d'utilisation, permettant ainsi un dialogue efficace avec leurs homologues techniques.

Maîtrise d'œuvre : une nouvelle organisation

Pour leur part, les équipes de maîtrise d'œuvre doivent se structurer pour prendre en charge trois fonctions : répondre efficacement aux besoins utilisateurs, concevoir l'architecture applicative partagée, assurer la maîtrise des technologies utilisées.

L'ingénierie de projet (c'est-à-dire la conception et le développement des applications) est prise en charge par des équipes disposant d'une compétence technique intermédiaire et, chacune, d'une bonne vision d'un métier particulier. Leur rôle est de comprendre et de définir clairement les besoins immédiats des utilisateurs et de développer les applications nécessaires à partir des objets et services métiers existants. Une de leurs responsabilités est également de détecter et de formaliser les anomalies et les attentes tacites ou explicites des utilisateurs afin d'alimenter les processus d'évolution du système d'information. Leur objectif est l'efficacité (qualité, coûts, délais) au service des utilisateurs dans le respect de l'architecture du système d'information.

L'architecture applicative (c'est-à-dire la conception et le développement des objets métiers) est conçue par une

équipe disposant de fortes compétences techniques et d'une bonne vision globale des métiers de l'entreprise. Son rôle est de définir, avec la maîtrise d'ouvrage, les objets, composants et services structurants du système d'information. Les membres de cette équipe auront la charge de transposer sur le plan technique les spécifications fonctionnelles retenues en veillant à maintenir la qualité générale de l'architecture applicative. En effet, ils conçoivent et développent les éléments fondateurs du système d'information et doivent donc en assurer la fiabilité, la cohérence, la sécurité ainsi que la capacité à monter en charge.

Ils ont donc la charge de l'évolution du système d'information. En prise directe avec les métiers fondamentaux de l'entreprise, ils bénéficient également du retour d'information en provenance des équipes ingénierie. Leur objectif est la pérennité du système d'information et l'assurance de sa capacité à répondre efficacement à l'évolution des métiers de l'entreprise.

Quant à l'équipe "architecture technique", véritable task force composée d'architectes et d'experts techniques, elle permet de garantir la maîtrise des risques technologiques. Ses membres étudient, proposent et valident les fondements techniques de l'architecture du système d'information. Ils mettent à disposition de l'ingénierie les outils techniques et les procédures nécessaires à l'intégration des futurs objets et services métiers au sein des applications. Ils conseillent techniquement les équipes d'ingénierie et d'architecture applicative, apportent leur expertise et les font bénéficier d'un transfert de compétences, si cela se révèle nécessaire.

Gilles Mergoïl

L'équipe "architecture technique" permet de garantir la maîtrise des risques technologiques

La mort de DCOM

Un SOAP opera !



Jean-Marc Lejeune

Directeur de programme chez Gartner.

Il a occupé des fonctions de direction informatique dans les domaines de la logistique et du transport aérien.

En 2003, plus de 70 % des e-services qui seront invoqués au travers de l'Internet, le seront en utilisant SOAP de Microsoft. SOAP permet à deux applications de communiquer au moyen de messages XML transportés par HTTP, et ce, quel que soit le système d'exploitation, le langage de développement et le modèle d'objet utilisé (COM, EJB, etc.).

Les Américains ont le secret des soap operas, ces feuilletons à l'eau de rose dont ils inondent le marché. L'informatique n'y échappe pas ! Ainsi, il fut longtemps question de la mort de J.R., la Java Revolution. Aujourd'hui, la question qui se pose est de savoir si SOAP, qui signifie Simple Object Access Protocol, annonce la fin de DCOM (Distributed Component Object Model), le modèle objet distribué de Microsoft.

Un peu d'histoire

Pour bien comprendre ce qu'apporte SOAP, il est utile de revenir un peu en arrière, au début des modèles objets de Microsoft. En 1996, Microsoft intro-

duisit DCOM comme une extension à son modèle objet COM, lui permettant de fonctionner en mode distribué. DCOM présupposait qu'aux deux bouts de la communication il y avait des objets de type COM. DCOM permet donc de communiquer, mais si tout se passe dans le monde propriétaire de Microsoft.

Au contraire, le modèle CORBA (Common Object Request Broker Architecture) rendait les mêmes services, mais dans un monde plus ouvert. Cette communication COM – COM devait être en mode synchrone. Les deux systèmes communiquant par ce biais devaient établir une session entre eux. Ils étaient alors qualifiés de forte-

DCOM permet de communiquer dans le monde propriétaire de Microsoft

ment couplés (tightly coupled). Le middleware utilisé dans ce mode de communication était MS-RPC, la version Microsoft du DCE de l'Open Software Foundation.

Evolution des besoins

Les besoins en matière d'interconnexion de systèmes évoluèrent depuis 1996, principalement suite à la pression qu'exerce l'Internet sur les architectures techniques. C'est ainsi qu'aujourd'hui on ne peut plus se contenter de systèmes fortement couplés et propriétaires quand il s'agit d'ouvrir son architecture technique au monde extérieur pour, par exemple, s'interconnecter avec ses partenaires et qu'en sus, cette communication se fasse au travers des firewalls.

Microsoft franchit une première étape avec COM+ qui nous est arrivé dans Windows 2000. COM+, à côté des RPC admet d'autres middlewares et notamment MSMQ, le système de "message queuing" de Microsoft. Ceci permet de bâtir des interconnexions qui soient faiblement couplées (loosely coupled).

De telles interconnexions s'affranchissent de l'établissement d'une session entre les deux systèmes communiquant, et peuvent même se passer de la présence permanente du réseau entre eux, grâce aux mécanismes sécurisés des systèmes de "message queuing". Cependant, il est toujours nécessaire d'avoir du COM aux deux extrémités.

Une étape suivante est franchie grâce à SOAP. SOAP est défini par Microsoft, mais Microsoft l'a proposé comme standard à l'IETF, l'organisme qui veille à la destinée de l'Internet. Le but de SOAP est de permettre à deux applications de communiquer au moyen de messages XML transportés par HTTP, et ce, quel que soit le système d'exploitation, le langage de développement et le modèle d'objet utilisé (COM, EJB,

etc.). XML doit cependant être adapté pour pouvoir décrire les interfaces de services. Une telle interface dans une architecture orientée services contient non seulement la définition de la requête et de la réponse, partie pour laquelle XML n'éprouvera aucune difficulté, mais aussi la définition de l'identité du service, de son comportement et du traitement d'exceptions, c'est-à-dire tout ce qui est traditionnellement décrit au travers d'IDL (Interface Definition Language).

SOAP propose de réaliser la conversion des IDL en XML. SOAP ne signifie donc pas la mort de DCOM, mais il vient le compléter en l'ouvrant au monde extérieur.

Microsoft fait la course Internet en tête. Certes, SOAP n'a pas encore été adopté en tant que standard par l'IETF, il s'agit toujours d'une proposition. Cependant au Gartner nous pensons que cette proposition a de fortes chances de passer et que, en 2003, plus de 70 % des e-services qui seront invoqués au travers d'Internet, le seront en utilisant SOAP comme mécanisme sous-jacent.

Microsoft, malgré des débuts très lents, fait maintenant la course Internet en tête. De plus, il le fait de manière ouverte, en publiant et proposant sa technologie comme standard IETF, tandis que d'autres s'acharnent à breveter des clics de souris. Serait-ce là une conséquence bénéfique de l'action du juge Jackson ?

Jean-Marc Lejeune

Votre avis...

**Votre avis
nous intéresse,
écrivez-nous !
atzeljm@worldnet.fr**

On ne peut plus se contenter de systèmes fortement couplés et propriétaires

Le but de SOAP est de permettre à deux applications de communiquer au moyen de messages XML transportés par HTTP

Qualité

Vision utilisateur et maturité



Yves Constantinidis

Consultant à BSGL Conseil, membre actif d'ADELI, il intervient auprès des grandes entreprises, en France et à l'étranger, sur des missions de diagnostic et de conseil.

Face à l'immaturité générale des processus, l'imperfection des méthodes, le flou des concepts et les failles des produits, peut-on espérer développer un jour des logiciels de qualité ? Oui, mais à condition que les maîtres d'œuvre respectent des contraintes indispensables et que les maîtrises d'ouvrage tiennent mieux leur rôle propre dans le projet.

Imaginons qu'un commandant de bord annonce le message suivant aux passagers d'un avion de ligne : "Mesdames et Messieurs, nous avons maintenant atteint notre vitesse de croisière, que je n'ai pas le droit de vous communiquer. J'ai une vue superbe sur le mont Blanc, que vous auriez pu distinguer à votre gauche si le concepteur de l'appareil avait bien voulu prévoir des hublots. Je dois vous annoncer que des retards sont à prévoir, pour des raisons techniques que vous ne pouvez pas comprendre. L'appareil que je suis en train de piloter est sûr et fiable. Je vais même vous le montrer tout de suite en exécutant un looping. J'espère que cela vous fera plaisir".

Une telle annonce est impensable dans un avion de ligne. Mais c'est pourtant ce genre de discours que tiennent nombre de projets informatiques à leurs maîtrises d'ouvrage et leurs utilisateurs.

Immaturité

Un seul mot peut résumer la situation : immaturité. Celle des processus, des méthodes, des concepts, des produits, des moyens et de la vision utilisateur.

Processus. On ne s'étendra pas ici sur le modèle de maturité des processus de développement (modèle CMM). Rappelons simplement que la majorité des entreprises en est encore au niveau 1 : le processus de développement n'est pas contrôlé et sa qualité

Un seul mot peut résumer la situation : immaturité.

Le processus de développement n'est pas contrôlé et sa qualité repose entièrement sur la compétence et la bonne volonté des hommes

Dans un contexte hautement compétitif les outils qui supportent la construction du logiciel sont mis à très rude épreuve

repose entièrement sur la compétence et la bonne volonté des hommes. Et cette situation perdurera, très probablement, pour de nombreuses années. L'évolution sera lente, car, pour que les processus changent, il faut que les mentalités changent. Le frein est culturel et humain.

Méthodes. On s'accorde tous à dire que les méthodes ne sont pas parfaites mais qu'elles ont "le mérite d'exister". Cet argument est pervers. Dans l'industrie, lorsqu'une méthode de fabrication est inefficace, on la change ou on l'améliore. La méthodologie devrait être un processus continu et non un ensemble figé de normes. Mais, aujourd'hui, la méthodologie est élaborée sous stress. Les techniques évoluent très vite et nous voulons que les méthodes évoluent au même rythme.

Prenons un exemple : trois gourous se sont réunis pour mettre au point la méthode de "l'orienté objet". Il en a découlé non pas une méthode, mais une notation, un "langage unifié de modélisation" (UML, Unified Modeling Language). C'est un progrès, mais moins ambitieux que l'objectif de départ. De fait, la notation a été normalisée avant les concepts, ce qui ne va pas sans problèmes.

Concepts. Pour reprendre l'exemple d'UML, la notation laisse entrevoir un certain nombre de concepts. Sont-ils clairement définis ? Chaque nouveau terme est-il précisé ? Si les mots "béton" et "armature" était aussi vaguement décrit que le mot "agrégation" et "association" dans UML, les tours de la Défense ne resteraient pas longtemps debout !

Moyens. Dans un contexte hautement compétitif, où techniques et méthodes évoluent à un rythme accéléré, les outils qui supportent la construction du logiciel sont mis à très rude épreuve. Leurs éditeurs doivent nécessairement

faire des arbitrages douloureux. Les outils devraient être pensés de façon cohérente et globale, en tenant compte de l'ensemble des activités de construction du logiciel. C'est rarement le cas. La course-poursuite que mènent les éditeurs d'outils les empêche de prendre le recul indispensable à une conception rationnelle.

Produits. Les produits sont-ils adaptés aux besoins, sont-ils utiles, sont-ils effectivement utilisés ? Ce n'est pas toujours le cas. Les défauts évoqués ci-dessus sont évidemment pour beaucoup dans l'inadéquation des produits. Mais ce ne sont pas les seules causes. La qualité des processus est une condition nécessaire, mais non suffisante, à la qualité des produits qui en résultent. Un processus parfaitement au point peut donner lieu à des produits inutilisables. En particulier, aucun processus ne peut garantir que les besoins des utilisateurs ont été bien compris.

Vision utilisateur. L'utilisateur est encore trop souvent considéré comme le passager à qui l'on peut cacher les paramètres de vol et qui ne pourra regarder le paysage qu'à l'arrivée. Et, même lorsqu'il le souhaite, l'utilisateur ou le maître d'ouvrage n'a pas les moyens d'avoir un regard sur les paramètres du projet et sur la qualité du produit. Cependant, l'utilisateur aurait dès aujourd'hui la possibilité de connaître, et même d'influencer, la qualité du produit livré.

Lorsque nous parlons d'immaturité, ne jetons la pierre à personne. L'informatique est une activité jeune. Il est naturel qu'elle soit immature. Mais la jeunesse n'explique pas tout.

Une machine molle et invisible

Le mot software résume à lui seul à la fois l'essor considérable de cette industrie et la disproportion entre sa fulgu-

rante progression et la faible maturité des produits qui en résultent. Le logiciel est mou, impalpable, aucune contrainte ne pèse sur lui. Le béton obéit aux lois de la physique et de la chimie. On ne peut pas construire un pont de n'importe quelle forme, n'importe où, n'importe comment. La mise sur le marché d'un produit industriel ne peut se faire qu'après des contrôles, de plus en plus stricts, et dans le respect de normes.

Avec le logiciel, il en va autrement. Les seules contraintes sont imposées par le marché : coûts, délais, prix, concurrence. L'absence de contraintes physiques sur le logiciel a des conséquences importantes sur la maturité des processus et des produits. Le logiciel étant mou, il peut être facilement modifié. D'où la difficulté, à tous les niveaux, de contrôler et de gérer les versions successives.

De plus, la structure interne d'une application ne peut pas être vue ni même devinée de l'extérieur, comme c'est le cas pour un ouvrage d'art ou pour un bâtiment. Les modifications sont invisibles, faciles, rapides et ne coûtent pas cher. Elles restent inaperçues, mais peuvent être lourdes de conséquences.

Se fabriquer des contraintes

Serait-il donc utopiste d'imaginer développer des logiciels de qualité ? Je ne le crois pas. Pour y parvenir, la solution consiste à fabriquer de toutes pièces un ensemble de contraintes, et à s'y conformer.

Enoncée de cette manière, la démarche peut paraître artificielle, voire saugrenue. Pour la comprendre, il suffit de renverser la question : quelles sont les contraintes qu'on devrait appliquer en amont pour que le logiciel soit satisfaisant en aval ? La réponse est alors triviale : ce sont celles qui permettront au produit final de satisfaire à tous les cri-

tères de qualité. Or, les critères de qualité d'un produit logiciel, nous les connaissons. Ils sont précisément définis par la norme internationale ISO/IEC 9126. Ils sont au nombre de six.

La capacité fonctionnelle. Le logiciel doit apporter un ensemble de fonctions. C'est la capacité fonctionnelle. Les résultats doivent être exacts, c'est-à-dire conformes à ce qu'on attend. Le logiciel doit respecter les lois et règlements en vigueur et les exigences de sécurité.

La fiabilité. C'est l'aptitude du logiciel à maintenir son niveau de service. Un logiciel fiable est un logiciel qui a peu de défaillances. Il peut être rapidement remis en état. Il sait maintenir un certain niveau de service en cas de défaillance et recouvrer un état acceptable après remise en état.

L'utilisabilité. Elle concerne l'effort nécessaire à l'utilisation. Un logiciel utilisable est un logiciel facile à apprendre et à utiliser. Il s'adapte au profil de l'utilisateur et réduit les risques d'erreur. L'utilisabilité est étroitement liée à l'ergonomie.

Le rendement. Il concerne l'utilisation des ressources (processeur, mémoire, etc.). Il mesure les performances du logiciel.

La maintenabilité. Un logiciel maintenable est un logiciel facile à analyser, à modifier et à tester et sur lequel les modifications n'ont pas d'effets inattendus.

La portabilité. La portabilité correspond à la facilité d'installation, au transfert aisé d'un environnement à un autre. Le logiciel portable est apte à remplacer un autre logiciel et à être remplacé par un autre logiciel.

Ces six caractéristiques sont utilisées pour vérifier la qualité d'un produit livré (progiciel ou application). Elles peuvent donc nous aider à identifier les contraintes à appliquer en amont.

Le logiciel est mou, impalpable, aucune contrainte ne pèse sur lui.

La solution consiste à fabriquer de toutes pièces un ensemble de contraintes et à s'y conformer.

Les exigences de fiabilité, de portabilité et de rendement sont rarement décrites

Modifier la vision utilisateur

Lorsque vous faites construire une maison, vous allez visiter le chantier. Même si vous n'êtes pas architecte, vous avez la possibilité de vérifier non seulement la couleur des fenêtres, mais aussi les matériaux de construction et les techniques utilisées.

En matière de construction de logiciel, les choses fonctionnent autrement. L'utilisateur a un droit de regard et aussi des possibilités de contrôle, mais ceux-ci restent superficiels. Il peut effectivement vérifier la couleur des fenêtres (affichées à l'écran) mais les exigences de fiabilité, de portabilité et de rendement sont rarement décrites et quand, par hasard, elles le sont, c'est le plus souvent de façon floue ou incomplète.

Un simple particulier qui fait construire sa maison de campagne peut avoir des exigences très précises quant à la forme ou à la manière dont la maison s'insère dans le paysage.

En informatique, un maître d'ouvrage ou un donneur d'ordres a rarement des exigences aussi précises. Et ceci pour deux raisons : parce qu'il ne sait pas comment les exprimer et surtout parce qu'il ne sait pas quel type d'exigences il est en droit d'exprimer, ni comment il les fera contrôler et respecter. Par exemple, peu de cahiers des charges expriment des exigences de maintenabilité. Lorsqu'ils le font, ils expriment plus souvent les moyens à utiliser (méthodes et outils) que les objectifs à atteindre (par exemple, un encadrement de l'effort moyen consacré à la correction d'un bogue).

Un outil de dialogue

Il est donc important que les caractéristiques de qualité, qui sont actuellement un outil de contrôle, deviennent également un outil de dialogue et le fondement des spécifications des produits. Pour cela, il faut que l'utilisateur, le

maître d'ouvrage, le donneur d'ordres, modifient la perception qu'ils ont du logiciel. Aujourd'hui, le logiciel est une boîte noire. A défaut de devenir une boîte transparente, il faut qu'elle se transforme au moins en une boîte "de couleurs", où les différentes caractéristiques soient identifiables.

La première étape consiste donc à donner un vocabulaire commun aux producteurs et aux consommateurs de logiciel. Avoir un vocabulaire commun, centré sur les objectifs plus que sur les moyens, permet de construire plus vite un cahier de charges plus clair et de se concentrer sur le contenu. Ceci permet également à chaque acteur de jouer pleinement son rôle : le produit fini au maître d'ouvrage, le travail pour y parvenir au maître d'œuvre.

Vers une échelle de maturité de la maîtrise d'ouvrage

Il existe une échelle de la maturité pour les organisations de développement du logiciel. De la même façon, on pourrait imaginer une échelle de maturité pour les organisations consommatrices, qui pourraient ainsi s'autoévaluer. Cette échelle comporterait également cinq niveaux :

Niveau 1 : le cahier des charges est absent ou embryonnaire, les besoins ne sont pas formalisés.

Niveau 2 : l'organisation établit un cahier des charges minimum, purement fonctionnel, ou en délègue la rédaction à un consultant.

Niveau 3 : l'organisation coopère avec le consultant pour établir un cahier des charges global, tenant compte des six caractéristiques citées plus haut, ou l'établit elle-même selon des règles précises et un savoir-faire éprouvé.

Niveau 4 : l'organisation mesure l'efficacité de la maîtrise d'œuvre, grâce à des audits, des diagnostics, des évaluations, des techniques d'analyse de la

La première étape consiste donc à donner un vocabulaire commun aux producteurs et aux consommateurs de logiciel

valeur, et ce à tous les stades du développement ; le produit demeure une boîte noire, mais sa valeur est connue en temps réel.

Niveau 5 : l'organisation a les moyens et la volonté de maîtriser l'efficacité de la maîtrise d'œuvre, en faisant ou non appel à des compétences externes.

Bénéfices communs

Une plus grande maturité apporterait aux organisations consommatrices une maîtrise croissante du rapport qualité/prix.

Au niveau 1 : la qualité est aléatoire ; les besoins sont satisfaits par hasard, donc rarement.

Au niveau 2 : les besoins sont partiellement satisfaits et de façon peu prévisible.

Au niveau 3 : les besoins sont largement satisfaits, avec néanmoins des points aveugles.

Au niveau 4 : la satisfaction des besoins est quasiment garantie dès les phases amont, la non-satisfaction de certains besoins est formalisée.

Au niveau 5 : l'organisation connaît a priori le rapport entre le coût et la qualité du produit qu'elle va faire développer et elle en a la maîtrise.

Les bénéfices d'une telle autoévaluation seraient également partagés par les organisations de développement (maîtres d'œuvre, sous-traitants) : celles-ci disposeraient d'une vision réaliste des qualités et des limites de leurs maîtrises d'ouvrage, et elles pourraient se prémunir en conséquence.

En matière de logiciel comme ailleurs, les exigences des consommateurs seront sans aucun doute les meilleurs stimulants de la qualité.

Les exigences des consommateurs seront sans aucun doute les meilleurs stimulants de la qualité

Yves Constantinidis
yconstan@easy.net.fr

Revue d'auteurs, l'Informatique Professionnelle accueille des opinions qui n'engagent pas la rédaction.

Ne manquez pas ce numéro

Au sommaire vous trouverez :

- Les factures du 21ème siècle
- Les moyens de paiement sur internet
- Les mécanismes du traitement des cartes de crédit sur internet
- Mise en place des paiements sur internet
- Critères d'évaluation des solutions de paiement sur internet
- Les standards basés sur le langage XML : une saine concurrence
- Espace de confiance
- Les problèmes de sécurité des paiements
- Les moyens de paiement futurs du commerce électronique
- Paiements électroniques : les agents électroniques auront les pleins pouvoirs
- La facturation électronique : l'importance de la relation client
- Les portefeuilles électroniques

Chaque mois, dans *Architecture Informatique et Télécom*, un dossier complet pour orienter vos choix techniques



Bouhot & Le Gendre
Produits et Services

Pour tout renseignement ou pour recevoir *Architecture Informatique et Télécom*, contacter Sylvie Garofalo au tél. : 01 41 35 15 18 - Fax : 01 41 35 15 10



Internautes

Souriez, vous êtes pistés !



Eric Larcher

Ingénieur ESME-Sudria, titulaire du Mastère spécialisé en sécurité des systèmes informatiques et des réseaux de l'ENST, il est actuellement en charge de la sécurité des systèmes d'information au sein d'une grande entreprise française.

Les messageries et les sites web sont des portes ouvertes pour les malveillants en général et les pirates en particulier. Mais attention, des parades existent !

N'importe qui peut envoyer un message avec n'importe quelle adresse d'expéditeur

Pour tout un chacun, “sécurité Internet” rime souvent avec firewall, VPN, passerelle antivirus, SSL, etc. Pourtant, en se focalisant uniquement sur la sécurité des flux, des infrastructures réseaux, des serveurs internes (messageries, applications, etc.) ou externes (sites web, DNS, etc.), on oublie que l'utilisation irraisonnée et/ou incontrôlée d'un simple logiciel de messagerie ou navigateur web peut mettre en péril les informations confidentielles de l'entreprise, relatives à ses activités ou à ses salariés.

Nous allons nous intéresser ici aux deux services les plus utilisés sur Internet, à savoir la messagerie et le web.

Lettres ouvertes aux pirates

Rappelons quelques réalités simples !

Un jour, un utilisateur de ma société m'envoya un mail, contenant l'en-tête SMTP d'un courrier électronique qu'il venait de recevoir et qui le laissait quelque peu perplexe : il s'agissait d'un message “anonyme”, dont l'expéditeur avait indiqué une adresse source invalide.

Mon utilisateur s'étonnait qu'il soit possible de faire ce genre de choses. Je dus alors lui expliquer que la messagerie, telle que des centaines de millions de personnes l'utilisent chaque jour dans le monde, n'assure en standard aucun service d'authenticité. Cela signifie que n'importe qui peut

envoyer un message avec n'importe quelle adresse d'expéditeur.

Et même si un individu envoie un e-mail sous sa propre identité, rien ne l'empêche de nier cet envoi ultérieurement. On appelle ceci la répudiation. La messagerie Internet ne permet pas non plus la non-répudiation d'un courrier électronique.

Par ailleurs, il faut savoir qu'un e-mail peut être modifié en cours de route (que ce soit intentionnellement ou par accident), sans que son destinataire ait un quelconque moyen de s'en rendre compte. On dit alors que son intégrité n'est pas garantie.

Enfin, tout message circule "en clair" sur les réseaux qu'il emprunte lors de son transport du serveur SMTP de l'expéditeur vers celui de son destinataire. Ainsi, n'importe qui peut prendre connaissance de n'importe quel message, du moment qu'il est capable de l'intercepter en cours de route. Il n'y a donc aucune confidentialité.

Résumons-nous : confidentialité, intégrité, authenticité et non-répudiation, autant de services qui paraissent indispensables mais qui ne sont malheureusement pas assurés sur les messageries électroniques.

Avec la place que prend la messagerie aujourd'hui dans les échanges entre salariés, partenaires, clients et fournisseurs, il paraît donc déraisonnable de se contenter d'un système aussi peu fiable et qui date ... de vingt ou trente ans !

S/MIME

Alors, comment faire pour sécuriser un outil dont plus personne ne peut se passer. Il existe une solution qui s'appelle S/MIME⁽¹⁾ (Secure/Multipurpose Internet Mail Extensions). Il s'agit de l'extension sécurisée de la norme MIME (permettant d'attacher à un message toutes sortes de fichier). S/MIME, reconnue en standard au sein

des principaux clients de messagerie Internet, comme Netscape Messenger ou Microsoft Outlook (Express), a recours :

- Au chiffrement pour assurer la confidentialité du courrier électronique ;
- de la signature numérique pour mettre à disposition les services d'intégrité, authenticité et non-répudiation que nous évoquons plus haut.

Quand on parle de chiffrement et/ou de signature numérique, on pense automatiquement à la cryptographie, a fortiori à clés publiques. Et qui dit clés publiques dit généralement certificats. En effet, il faut bien s'assurer de l'authenticité d'une clé et de l'identité de son titulaire, sinon on ne pourrait pas parler réellement d'authenticité ou de non-répudiation d'un message électronique signé.

Le standard S/MIME repose justement sur des certificats numériques à la norme X.509 v3. En pratique, pour activer les fonctionnalités de chiffrement et de signature numérique dans un logiciel compatible S/MIME, il suffit de récupérer un certificat pour chaque individu souhaitant pouvoir s'échanger des messages "sécurisés".

Ainsi, pour signer un e-mail, il faut disposer d'un certificat personnel valide. Cette opération nécessite l'emploi de la clé privée associée au certificat individuel que l'on possède. Pour chiffrer un message à l'attention d'une autre personne, il faut donc au préalable disposer de son certificat afin de connaître sa clé publique, nécessaire aux opérations de chiffrement.

A l'arrivée, le destinataire suit la démarche opposée : il utilise sa clé privée (liée à la clé publique de son certificat) afin de déchiffrer le message qu'il a reçu, et emploie la clé publique de l'expéditeur (contenue dans son certificat) afin de vérifier la signature de ce message.

Un e-mail peut être modifié en cours de route sans que son destinataire ait un quelconque moyen de s'en rendre compte

Pour signer un e-mail, il faut disposer d'un certificat personnel valide

¹ - Voir RFC 2633 sur www.rfc-editor.org.

On peut alors créer une autorité de certification interne à l'entreprise, capable de délivrer ses propres certificats

Même si vous êtes le seul utilisateur de votre ordinateur, les administrateurs systèmes de votre société pourront, le plus souvent à votre insu, accéder aux mêmes informations...

2 - On peut alors en vérifier l'authenticité et l'intégrité grâce à la clé publique de l'autorité.

Pour une utilisation ponctuelle ou dans une population limitée, on peut se contenter de se connecter au site web d'une autorité de certification et demander, en ligne, un certificat, dont le coût annuel sera de l'ordre d'une centaine de francs pour les certificats les plus basiques. Ces derniers seront alors générés et signés par cette autorité⁽²⁾.

En revanche, s'il faut permettre à l'ensemble des salariés d'une société de pouvoir envoyer des messages sécurisés, il devient nécessaire d'avoir recourt à ce qu'on appelle une infrastructure à clés publiques ou PKI (Public Key Infrastructure). On peut alors créer une autorité de certification interne à l'entreprise, capable de délivrer ses propres certificats. Ceux-ci pourront alors être vérifiés non seulement par les salariés mais également par les partenaires, clients ou fournisseurs de l'entreprise.

Ainsi, grâce à S/MIME, aux certificats et éventuellement aux PKI, il est possible d'échanger des correspondances électroniques en toute sécurité, sans risquer la divulgation ou l'altération d'informations confidentielles et sensibles.

Des traces sur le Web

Lorsqu'on surfe sur le Web pour la première fois, on découvre un monde (voire un univers) d'une richesse insoupçonnée, où des millions de services sont rendus accessibles par un simple clic de souris.

Mais on est, en général, loin de se douter que chaque site accédé, chaque page, fichier ou image récupérés, chaque musique ou vidéo que l'on joue sur son ordinateur s'accompagne de toute une série d'enregistrements aussi précis que nombreux.

En local tout d'abord, c'est-à-dire sur le poste de l'utilisateur, les navigateurs web enregistrent de façon horodatée, les adresses de toute page accédée,

dans un fichier ou répertoire appelé historique. Ce dernier permettait historiquement (si l'on peut dire) à l'utilisateur de différencier à l'écran les liens qu'il avait déjà visités durant les jours précédents. Depuis les versions 4.0 des navigateurs de Netscape et Microsoft, le système de "complétion" ou remplissage automatique des adresses en cours de saisie dans la barre d'adresses repose également sur cet historique. Ces adresses, entrées à la main, font d'ailleurs l'objet d'un enregistrement spécifique par le navigateur.

Mais tout ceci n'est rien à côté du cache. Le cache a pour objet de mémoriser presque tous les éléments (textes, images, animations, sons, etc.) constituant chaque page visualisée via le navigateur. Il permet de garder en mémoire (vive et sur le disque dur) les fichiers récemment récupérés, afin d'en accélérer l'affichage en cas d'accès ultérieurs à condition que les sites dont elles sont issues n'aient pas été modifiés.

Le cache, l'historique ainsi que la liste des adresses saisies permettent donc de savoir précisément, non seulement les adresses des documents visualisés par une personne, mais également le contenu de ceux-ci. Dans le cas d'une machine utilisée par plusieurs personnes (dans un cybercafé ou au bureau), il n'est donc pas très difficile de savoir précisément ce que faisait monsieur X à 14 h 32, deux jours plus tôt. Et même si vous êtes le seul utilisateur de votre ordinateur, les administrateurs systèmes de votre société pourront, le plus souvent à votre insu, accéder aux mêmes informations...

Traces distantes

Mais là ne s'arrête pas le pistage. En effet, chaque serveur web génère nombre d'informations, dans des "fichiers logs" contenant les noms des ressources utilisées par les clients dis-

CONSEILS

Une PKI peut être “insourcée” ou “outsourcée”. Dans le premier cas, elle repose entièrement sur des infrastructures internes à l’entreprise. Dans le second, elle est partiellement ou totalement sous-traitée à une société spécialisée. Le choix de l’une ou l’autre de ces solutions s’effectue en fonction des besoins de l’entreprise, de son budget, son organisation et des compétences dont elle dispose en interne. Dans tous les cas, il ne faut surtout pas se contenter d’une approche technique mais également s’intéresser aux questions organisationnelles liées à l’implantation, l’utilisation et l’exploitation de la PKI.

- Les fichiers logs doivent être déclarés à la CNIL ainsi que les coordonnées de toutes les personnes ayant accès à ces fichiers.
- Une charte d’utilisation encadrant l’utilisation des accès à Internet est indispensable.
- L’utilisation de serveurs proxy filtrants et/ou d’anonymat permet de réduire les risques de divulgation d’informations qui n’ont aucune raison de quitter l’entreprise. Notez cependant que certains sites ont besoin de ces informations ou utilisent des technologies à risques (scripts, applets, etc.) et peuvent ne plus fonctionner correctement via ces proxies.

tants, les versions des navigateurs et des systèmes d’exploitation utilisés par ceux-ci, l’adresse de la page où a été trouvé chaque lien accédé, etc. Ces informations sont autant de traces distantes, sur lesquelles l’utilisateur n’a que peu ou pas du tout de contrôle.

Voici une ligne d’un fichier log, extraite du site web internet-securise.com.

```
proxy.superisp2000.fr - -
[25/Jun/2000:08:45:06 +0100] "GET /
HTTP/1.1" 200 3620 "http://www.lar-
cher.com/eric/guides/" "Mozilla/4.51
[fr] (Win98; I)"
```

Examinons chaque champ. Le premier correspond à l’adresse (IP ou DNS si disponible) de la machine ayant accédé au site considéré. Viennent ensuite deux tirets indiquant qu’aucune authentification n’a été utilisée (“accès anonyme”). Ces champs sont renseignés lorsque vous saisissez un couple nom d’utilisateur - mot de passe dans une fenêtre d’authentification, permettant ainsi d’accéder à une partie privée d’un site web.

On trouve ensuite la date et l’heure exactes auxquelles la requête a été reçue par le serveur web. Juste après, on note la commande envoyée par le navigateur : “GET/HTTP/1.1” qui signifie “Je comprends la version 1.1 du protocole HTTP, envoyez-moi la

page d’accueil (/) du site”. Cette séquence est suivie d’un code (200) indiquant que la requête précédente s’est déroulée correctement (on aurait pu trouver ici le fameux code 404 si la page n’avait pas existé) ainsi que la taille en octets des données envoyées.

Ce que nous venons de décrire jusqu’à présent est le minimum d’informations générées par tout site web lors de l’accès à n’importe quel fichier qu’il met à disposition. Cela signifie que si la page d’accueil du site comporte trois images, nous verrions apparaître trois lignes (donc requêtes) de ce type (en plus de celle correspondant à la page d’accueil en elle-même), seul l’argument de la commande GET variant d’une requête à l’autre (ex. : GET/images/accueil.jpg).

Cependant, nous pouvons constater que d’autres informations apparaissent dans la ligne du fichier log reproduite plus haut. C’est tout simplement parce que nous avons activé le niveau maximal de logging. On distingue en effet deux chaînes. La première contient l’URL de la page à partir de laquelle l’utilisateur a cliqué sur un lien pointant vers la page d’accueil du site dont la ligne de log est issue. En clair, il existe un lien sur la page d’adresse <http://www.larcher.com/eric/guides/> pointant vers le site [Ces informations sont autant de traces distantes, sur lesquelles l’utilisateur n’a que peu ou pas du tout de contrôle](http://www.inter-</p>
</div>
<div data-bbox=)

Lorsque vous surfez nombre de traces sont générées par votre navigateur en local et les administrateurs des serveurs consultés savent exactement quelles pages vous avez visualisées

Les logs générés par les proxies internes peuvent porter atteinte à la vie privée des salariés

net-securise.com. On appelle ceci le champ Referer.

La seconde chaîne décrit précisément le système ainsi que la version du navigateur utilisés. En l'occurrence, il s'agit ici de Netscape Communicator (nom de code "Mozilla") en version française 4.51 sous Windows 98.

Quels risques ?

Ainsi, lorsque vous surfez sur n'importe quel site web, nombre de traces sont générées par votre navigateur en local et les administrateurs des serveurs consultés savent exactement quelles pages vous avez visualisées, quel jour et à quelle heure, avec quel navigateur et système d'exploitation. Ils savent également d'où vous venez. Notez que ces informations peuvent être également enregistrées au sein des serveurs proxy dont votre entreprise dispose. Dans ce cas, le log du proxy va indiquer en tête de chaque ligne : l'adresse de la machine interne accédant au site web externe et le log du serveur ainsi consulté comprenant celle du proxy (puisque'il se situe entre le client et le serveur).

Tout ceci n'est pas sans poser un certain nombre de problèmes. Sur le plan juridique tout d'abord, les logs générés par les proxies internes (ainsi que les informations stockées en local) peuvent porter atteinte à la vie privée des salariés puisqu'ils mémorisent exactement les mêmes données que celles que nous avons décrites plus haut. De plus, il s'agit d'informations nominatives puisqu'elles permettent "l'identification des personnes physiques auxquelles elles s'appliquent⁽³⁾" (via l'adresse indiquée en tête de chaque enregistrement).

Ensuite, vis-à-vis des serveurs web distants, il peut être dangereux de révéler tant d'informations sur les systèmes et navigateurs utilisés en internes et sur

les sites d'où est issu l'utilisateur.

En effet, la connaissance des versions exactes des logiciels et systèmes employés permet de lancer des attaques contre ceux-ci, autorisant parfois l'accès à des fichiers disponibles en local sur la machine du surfeur. De plus, le champ Referer peut révéler les adresses de serveurs intranet de l'entreprise si un lien vers un site web externe est proposé sur un tel serveur interne...

Des solutions antitraces

Heureusement, il est possible d'effacer voire de désactiver les traces locales rien qu'en modifiant les paramètres de configuration des navigateurs. Concernant les traces distantes, les possibilités sont plus complexes. On peut masquer l'adresse de la machine ou du proxy accédant à un site web via un proxy d'anonymat (comme www.anonymizer.com ou www.proxy-mate.com). Il est également possible de réduire le nombre et la précision des informations générées via l'emploi de proxies filtrants locaux tels que AdSubstract, Webwasher ou encore Junkbuster.

Dans tous les cas, l'établissement d'une charte d'utilisation des accès internet, éventuellement secondée par un contrôle d'accès au niveau proxy et un anti-virus bloquant les scripts malveillants, permettent de limiter les informations sensibles pouvant insidieusement quitter l'entreprise.

Eric Larcher

(info@internet-securise.com)

BIBLIOGRAPHIE

L'Internet sécurisé, Éric Larcher, mars 2000, Eyrolles.

3 - Art. 5 de la loi du 6/1/78.

Achats informatiques

Diminuer les risques et négocier

En informatique, le management des achats relève d'appréciations spécifiques et de pratiques communes. Outre la négociation tarifaire, l'acheteur professionnel peut aider l'informaticien à réduire le risque fournisseur et le risque contractuel.

Gilles Blanchard

Responsable des achats informatiques dans une grande banque privée.

On le sait, les achats informatiques coûtent cher. Il n'est pas rare que leur montant atteigne le tiers des frais généraux de nos entreprises.

Leur maîtrise est donc devenue un objectif prioritaire du management.

Les achats informatiques conjuguent plusieurs particularités qui les différencient des achats traditionnels. Ces particularités sont liées essentiellement à l'aspect stratégique de l'informatique dans nos entreprises, à l'importance des dépenses et à la rapidité de l'évolution de la technologie.

Remarquons aussi que l'informatique se traite souvent en mode projet. Ces projets comportent quasi systématiquement un volet "achats", pour des

licences, du matériel, des prestations de service...

L'organisation des achats

Peu d'entreprises ont aujourd'hui réellement donné aux services achats les moyens d'intervenir sur les dépenses informatiques. Les enjeux sont pourtant importants et une bonne prise en compte des meilleures pratiques permettrait de réaliser des gains substantiels et d'éviter les erreurs liées à une mauvaise appréciation du risque.

Mais, bien sûr, les freins sont nombreux. De fait, l'introduction de la problématique des achats dans nos projets informatiques relève d'une gestion délicate du changement, car tradition-

Peu d'entreprises ont réellement donné aux services achats les moyens d'intervenir sur les dépenses informatiques

Les bonnes pratiques en matière d'achats informatiques exigent un fonctionnement en équipe entre les acheteurs et la direction du projet

nellement les chefs de projet sont seuls maîtres à bord et s'occupent généralement aussi de la fonction achat.

Pourtant les bonnes pratiques en matière d'achats informatiques exigent un fonctionnement en équipe entre les acheteurs et la direction du projet. Que le projet informatique soit complexe, ou qu'il concerne la simple homologation d'un matériel qui sera approvisionné au fil des besoins, chacun a son rôle à jouer.

Plus un achat est stratégique, plus la direction du projet pilote et décide en connaissance de cause. Le service achats a, dans ce cas, la charge de proposer des stratégies d'acquisition et de négociation fournisseur. Plus l'achat est banal et concerne des produits de commodité, plus le service achats sera autonome.

Pourquoi une telle organisation ? Parce que l'aspect prix ne peut pas être le seul élément à prendre en compte pour les achats informatiques. Ainsi, lorsque l'on acquiert une solution influant directement sur la compétitivité de l'entreprise sur ses marchés, le prix ne peut évidemment pas être le critère principal.

Les risques "achat"

En fait, si le principe générateur de la fonction achat se situe dans la maîtrise des dépenses, le rôle de l'acheteur ne se limite pas à cela. Dans les achats structurants, comme c'est souvent le cas en informatique, l'acheteur a une autre responsabilité importante. Il doit absolument limiter un double risque lié à cet achat : le risque fournisseur et le risque contractuel.

Le risque fournisseur

Diminuer le risque fournisseur est clairement une tâche du service achats. Certes, il est vain de penser qu'un service achats peut suivre la santé financière de tous les fournisseurs. Mais il n'est pas impossible de définir la liste des dix ou vingt fournisseurs princi-

paux et d'organiser son service achats de façon à connaître en permanence l'évolution de ce "hit parade" d'entreprises. Pour cela, il convient de :

- mener des réunions périodiques entre les acheteurs et les responsables commerciaux et administratifs par exemple,
- assurer le suivi des affaires en cours avec l'avis des chefs de projet sur le déroulement des prestations,
- attribuer une notation qualité à jour,
- faire réaliser des audits fournisseurs par des organismes spécialisés.

Voici quelques idées pour diminuer le risque fournisseur. Il n'en reste pas moins que souvent la crise chez un fournisseur se déclenche de façon brutale parce que des éléments d'appréciation ont été soigneusement dissimulés.

Le risque contractuel

Si la responsabilité de la négociation des contrats se retrouve habituellement au sein du service achats, en ce qui concerne l'informatique la réduction du risque contractuel ne pourra se faire qu'en connaissance de cause. Comment en effet négocier un risque contractuel si on ne comprend pas le cahier des charges ?

Face à ce problème, certains répondent qu'il s'agit là d'une préoccupation du chef de projet. Justement non ! Le chef de projet connaîtra dans sa vie peut-être deux ou trois projets majeurs, alors que l'acheteur en négociera des dizaines. Ce dernier aura une expérience d'une autre nature que celle du chef de projet. Il pourra alors négocier des engagements fournisseurs à partir d'une expérience certes moins en profondeur dans les projets, mais bien plus diversifiée. C'est, là encore, la collaboration entre l'acheteur et le chef de projet qu'il faut privilégier.

Gilles Blanchard

Revue d'auteurs, l'Informatique Professionnelle accueille des opinions qui n'engagent pas la rédaction.

Concepteurs

La double compétence est de rigueur



André Bors

Diplômé d'une école d'ingénieurs (Ponts 1973), André BORS a été chef de projet, puis responsable d'un centre de développement et d'exploitation informatiques. Il est aujourd'hui conseiller informatique au Ministère des Affaires étrangères.

Toute application nouvelle est le fruit d'un compromis entre les besoins, la technique et l'organisation. A la croisée des chemins, le concepteur doit disposer de la durée pour devenir aussi un bon connaisseur du métier de l'utilisateur.

Tout le monde s'accorde sur la nécessité de coupler la conception des applications informatiques à un véritable reengineering de l'organisation et des procédures existantes. Car il va de soi que l'informatique n'est qu'un moyen parmi d'autres pour l'exécution de processus de gestion ou de production, dans lesquels toutes les tâches interagissent entre elles. C'est avec l'objectif d'améliorer le plus possible l'efficacité d'ensemble d'une unité opérationnelle que doivent être menées les études de conception. Là est la condition première d'une bonne rentabilité des investissements informatiques, et cela nécessite un travail d'étude en profondeur de chaque domaine concerné par des

développements informatiques.

C'est aussi une tâche quasi permanente que de veiller à la meilleure synergie entre les fonctions du système d'information et le reste des processus opérationnels. Car le système d'information doit régulièrement évoluer, tant pour tirer parti de l'évolution technique que pour prendre en compte l'évolution des métiers des utilisateurs, lesquels doivent souvent répondre à des exigences ou à des opportunités nouvelles.

Ces principes étant admis, il faut en tirer toutes les conséquences quant à l'organisation des travaux d'étude et aux moyens humains mis en œuvre. Les concepteurs sont les premiers concernés.

C'est avec l'objectif d'améliorer le plus possible l'efficacité d'ensemble d'une unité opérationnelle que doivent être menées les études de conception

Même avec une solide équipe de consultants, la contribution des utilisateurs reste une condition sine qua non du succès des projets

L'idée de confier aux directions et services utilisateurs les tâches de conception et la maîtrise des choix de solutions présente de sérieux inconvénients

La fonction organisation

De prime abord, la solution coule de source : elle consiste à doter l'entreprise non d'un simple service informatique, mais d'un service informatique et organisation. Concrètement, ce service est alors doté d'une équipe dédiée à la fonction organisation, avec pour mission de travailler le plus possible en synergie avec les équipes de développement.

C'est là une solution qui a longtemps eu la faveur de la profession. Son principal avantage est d'affirmer clairement une volonté politique d'efficacité et de rationalisation. Ainsi le service informatique et organisation se trouve-t-il dûment investi pour étudier et développer des solutions susceptibles de faire évoluer l'organisation existante. Un autre avantage de la formule est de disposer en interne de consultants qui, pour peu qu'ils disposent de la durée, acquièrent progressivement une très bonne connaissance du domaine sur lequel ils interviennent. C'est, pour ces intervenants, un précieux atout, dont ne disposent pas le plus souvent les consultants externes.

Malgré cela, les résultats ont souvent été décevants et la formule est largement tombée en désuétude.

La première raison tient sans doute à ce que les moyens humains affectés à la fonction ont rarement été à la hauteur des besoins. Faute d'une conscience suffisante de l'enjeu, la tentation l'a parfois emporté de donner dans la demi-mesure. On a recruté en nombre insuffisant des consultants du niveau adéquat, ou bien on s'est contenté de faire intervenir des consultants extérieurs ponctuellement sur les projets les plus stratégiques.

Quoi qu'il en soit, la formule présente un inconvénient majeur, à savoir qu'elle fait peu de cas du rôle et de la contribution des utilisateurs pour la

conception et la mise en œuvre des solutions. Car même avec une solide équipe de consultants, la contribution des utilisateurs reste une condition sine qua non du succès des projets. Or, dans ce schéma, la mobilisation des utilisateurs ne va pas de soi. Parce que ceux-ci se trouvent alors dans une position un peu "inférieure", ou ressentie comme telle, et que cela ne favorise pas l'instauration d'un véritable partenariat. En fait, beaucoup d'utilisateurs comprennent mal qu'un service qu'ils perçoivent d'abord comme un service technique se trouve ainsi investi d'un pouvoir d'intervention dans leur travail, un pouvoir qui sous-tend peu ou prou celui de l'évaluation. Le risque est grand, dans ces conditions, que les utilisateurs ne s'investissent pas suffisamment dans des projets qu'ils ne ressentent pas comme étant les leurs.

Le problème reste ainsi posé de bien répartir les rôles entre utilisateurs et informaticiens.

L'implication et le rôle des utilisateurs

Si l'on ne peut se permettre de cantonner les utilisateurs aux seconds rôles, il faut éviter de trop "inverser la vapeur" au détriment du rôle et des prérogatives du service informatique. Ainsi l'idée de confier aux directions et services utilisateurs les tâches de conception et la maîtrise des choix de solutions présente de sérieux inconvénients.

De même que les informaticiens ne peuvent être au même niveau que les utilisateurs dans la connaissance de leur métier, ces derniers n'ont pas la capacité d'appréhender les aspects techniques des choix de solutions. L'expérience montre qu'il est très difficile d'expliquer aux utilisateurs les tenants et aboutissants de la mise en œuvre d'un nouveau système d'exploitation ou d'une nouvelle filière de

développement. Fort logiquement, l'utilisateur ne s'intéresse qu'à la couche supérieure du système, celle qu'il voit. Il ne peut pas appréhender des questions comme celles de la pérennité du système d'exploitation ou de la dépendance vis-à-vis d'un fournisseur.

C'est pour cela qu'il faut aussi exclure la possibilité, parfois évoquée, de faire rédiger les cahiers des charges par les seuls utilisateurs. Il est indispensable que les concepteurs informaticiens y prennent une part prépondérante.

Confier aux directions utilisatrices les tâches de conception et la maîtrise des choix de solutions, c'est aussi s'orienter vers une segmentation du système d'information qui risque fort de mettre en péril sa cohérence d'ensemble. Souvent préoccupés d'aller vite, les utilisateurs ont naturellement tendance à "faire leur marché" et à pousser à l'acquisition d'équipements (matériels et/ou progiciels) sans se préoccuper de l'existant en matière de filières de développement ou de réseaux, sans parler des contraintes de sécurité. Il est alors difficile au service informatique de faire prendre en compte ses arguments techniques et de jouer son rôle d'architecte du système d'information.

Des concepteurs compétents dans leur domaine

En fait, il n'y a guère d'alternative ! Le service informatique doit s'astreindre à appréhender le mieux possible le travail des utilisateurs et à concevoir des solutions qui allient la richesse fonctionnelle au réalisme technique. Il s'ensuit que les concepteurs doivent parvenir à joindre les deux bouts que constituent le métier des utilisateurs d'une part, l'état de l'art informatique d'autre part. Pour ce faire, il ne suffit pas qu'ils soient "à l'écoute" des utilisateurs.

En effet, la mission d'un concepteur ne consiste pas à enregistrer servilement les demandes "fonctionnelles" de ceux qu'on leur a désignés comme interlocuteurs. Il n'existe pas de besoin fonctionnel qui se concevrait et s'exprimerait indépendamment des techniques disponibles. Une nouvelle application est de plus en plus souvent issue de la rencontre entre une possibilité technique nouvelle et un besoin, parfois latent.

Pour construire les solutions les mieux adaptées aux besoins, la conception doit aussi être un processus itératif. Elle nécessite que s'instaure entre les concepteurs et les utilisateurs de tous niveaux, des échanges continus et un travail en symbiose. Ainsi peuvent se faire, au jour le jour, les arbitrages sur les fonctionnalités des applications, mais aussi sur les calendriers de développement ou diverses modalités de mise en œuvre.

La voie à suivre

Cette constatation induit deux conséquences principales quant à l'organisation des études :

1. Concernant la formation des concepteurs. Quand ils ne possèdent pas déjà une vraie compétence dans le domaine sur lequel ils interviennent, les concepteurs d'origine informatique doivent l'acquérir. Il vaut mieux, pour cela, ne pas se satisfaire d'une formation sur le tas, au travers d'échanges avec les utilisateurs ou d'une documentation parcellaire. On ne doit pas hésiter à investir dans une formation ad hoc. Le principe doit être de développer réellement une double compétence ciblée et que la fonction ne dérive pas vers un profil de généraliste. A l'inverse, les personnes dont la formation initiale est non technique, par exemple des diplômés d'écoles de gestion, doivent s'imposer de suivre un cycle de

Fort logiquement, l'utilisateur ne s'intéresse qu'à la couche supérieure du système, celle qu'il voit

Le service informatique doit s'astreindre à appréhender le mieux possible le travail des utilisateurs et à concevoir des solutions qui allient la richesse fonctionnelle au réalisme technique

Les concepteurs doivent constituer le noyau stable qui assure un service permanent d'étude, de suivi et de "gestion client" des utilisateurs

formation informatique qui fasse d'eux des professionnels. Dans tous les cas, une réelle capacité d'adaptation est nécessaire et un niveau élevé de formation initiale est très souhaitable.

2. Concernant la gestion des équipes d'étude. Dès lors que des concepteurs ont été affectés sur un domaine, il est souhaitable de les y maintenir pendant une période suffisante pour justifier l'investissement consenti. D'ailleurs, même avec une formation ad hoc, ce n'est que progressivement qu'ils pourront s'immerger dans le métier des utilisateurs.

Cela ne signifie nullement qu'ils doivent se trouver en sous-charge entre deux phases de développement d'une application. D'une part, le domaine confié à un ou plusieurs concepteurs comportera généralement un ensemble d'applications dont les développements, les évolutions, les refontes doivent être planifiés de manière à lisser leur charge de travail. D'autre part, les concepteurs doivent, entre deux phases de développement, rester très présents auprès des utilisateurs. Pourquoi ? Pour, entre autres choses, prendre en compte la maintenance indispensable, négocier le calendrier des fonctions qui pourront être reportées, faire le bilan des gains par rapport aux coûts, préparer les futures versions, étudier les évolutions dans le travail des utilisateurs. Ainsi, même si les concepteurs affectés à un domaine peuvent être en nombre très réduit, ils doivent constituer le noyau stable qui assure un service permanent d'étude, de suivi et de "gestion client" des utilisateurs. Il faut donc en prendre son parti. Les concepteurs constituent une ressource peu mobile, surtout par rapport aux développeurs qui doivent pouvoir passer rapidement d'une application à une autre.

Une objection assez courante à cette approche est qu'elle mettrait les

concepteurs dans une situation de "propriétaire" de leurs applications. On y répondra en précisant simplement que disposer de la durée n'est pas synonyme d'inamovibilité. Même avec des changements d'affectation très espacés, le principe de mobilité doit s'appliquer, tout comme celui de l'évaluation. Précisons enfin que la mission ainsi définie pour les concepteurs ne laisse guère de place à d'autres tâches, en particulier à l'encadrement des développeurs. Celle-ci doit revenir aux chefs de projet ou aux chefs d'application, des fonctions à bien distinguer de celle de concepteur.

Tout compte fait, la fonction de concepteur tend à se confondre, surtout pour les plus expérimentés, avec celle de consultant. Et compte tenu de cette proximité avec la fonction de conseil, on doit pouvoir attendre d'eux qu'ils assurent aussi un rôle pédagogique auprès des utilisateurs.

André Bors

Revue d'auteurs, l'Informatique Professionnelle accueille des opinions qui n'engagent pas la rédaction.

Au sommaire du prochain numéro

Dossier Governance

Governance et NTI

Faire plus avec moins

Balance Score Card

Un modèle fédéral

Sourcing

et aussi...

Pilotage des intranets

E-mail et vie privée

L'encadrement des développeurs doit revenir aux chefs de projet ou aux chefs d'application

Refacturation des réseaux

La diplomatie s'impose !

Avec la multiplication des réseaux, l'explosion des coûts n'a pas tardé. Quand on veut remettre de l'ordre et canaliser les ardeurs, la refacturation des réseaux aux utilisateurs est souvent utilisée. Plusieurs méthodes existent mais une seule réussit vraiment : la diplomatie. Etude de cas.

Edward Younker
et David Neil

Gartner

Les DSI et autres responsables informatiques sont fortement impliqués dans les décisions concernant les dépenses informatiques et la plupart cherchent également à mettre au point des stratégies efficaces de refacturation. Gartner présente ici une étude de cas concernant une entreprise américaine. Cette entreprise s'était aperçu que sa logique de récupération des coûts WAN (ou réseaux étendus) ne répondait pas à ses besoins et ne fonctionnerait plus dans ses futures architectures de réseau. Un projet destiné à choisir les meilleures politiques de refacturation fut donc lancé. Sa conclusion fut qu'il faudrait engager de gros efforts pour parvenir à mettre en place un sys-

tème de refacturation des WAN fondé sur l'usage.

Les méthodologies de refacturation

Durant les années 90, les réseaux, et en particulier les réseaux étendus d'entreprise, ont changé radicalement en termes de technologie et d'importance stratégique. Mais les mécanismes administratifs, comme par exemple la refacturation, n'ont pas suivi. De plus, les entreprises essaient de mettre en place de nouveaux systèmes de réseau IP complexes tout en conservant leurs budgets informatiques au même niveau. La refacturation est un des mécanismes qui permet de faire face à la croissance

La refacturation est un des mécanismes qui permet de faire face à la croissance rapide du coût global des réseaux étendus

rapide du coût global des réseaux étendus. Mais comme la plupart des systèmes de refacturation ont été conçus pour gérer des réseaux anciens, il est nécessaire de chercher une nouvelle approche de la refacturation des réseaux étendus.

Les divers types de refacturation en vigueur aujourd'hui sont les suivants :

- un coût fonction du nombre d'utilisateurs ;
- la mesure de l'utilisation effective ;
- le coût direct ;
- le forfait par niveau.

Un coût fonction du nombre d'utilisateurs

Cette logique de facturation est utilisée par de nombreux services informatiques comme une méthode passe-partout dans des domaines où les métriques sont encore mal maîtrisées ou trop coûteuses à mettre en œuvre. Cet "abonnement" couvre un ensemble de services de base et de moyens permettant aux utilisateurs de faire leur travail. Cela comprend par exemple : l'usage du helpdesk, les opérations sur les PC, les services liés aux réseaux locaux ou étendus, les services d'impression, etc.

La mesure de l'utilisation effective

Cette méthodologie de refacturation se fonde sur des mesures représentatives de l'utilisation effective, tels que le nombre de MIPS, les secondes de CPU, les volumes de stockage sur disque ou bien sur les paquets échangés et les transactions effectuées. Certains responsables informatiques, maniaques de la mesure, vont "tout" mesurer pour pouvoir calculer plus précisément les coûts imputables à chaque acteur et pour garantir une égalité de traitement. Dans le cadre du passage des réseaux privés à des réseaux partagés, la

mesure de l'utilisation effective est devenue la forme la plus courante de facturation des réseaux.

Le coût direct

Certains services informatiques adopteront ou conserveront des méthodologies de coût direct car nombre d'entre eux hésitent à adopter des méthodologies de refacturation plus novatrices. La plupart découperont arbitrairement les coûts relatifs à un serveur partagé à partir d'un chiffrage fait une fois par an (par exemple, sur la base de la mémoire disque, des entrées/sorties, du nombre de connexions par mois ou du nombre de transactions). Ces méthodologies de coût direct peuvent alimenter une sous-capacité chronique et accroître les coûts.

Les tarifs forfaitaires à plusieurs niveaux

La méthodologie de refacturation sur la base d'un tarif forfaitaire à plusieurs niveaux est généralement liée à différents niveaux de service garantis pour le réseau. Par exemple, avec ce système, à chaque niveau de disponibilité correspondra un tarif particulier. La disponibilité sans backup pourrait représenter le premier niveau. La disponibilité avec backup téléphonique pourrait constituer un second niveau et des réseaux totalement redondants représenter le niveau le plus élevé.

Le cas d'une entreprise

Une grande entreprise, géographiquement dispersée, cherche à savoir comment optimiser les performances de son réseau et améliorer ses niveaux de service. Elle a regroupé ses réseaux anciens sous forme d'un réseau IP (1200 routeurs et 60 000 utilisateurs) et a réussi à réduire le coût global de son réseau étendu en dépit des mises à niveau de l'infrastructure, des contrain-

Certains services informatiques découperont arbitrairement les coûts relatifs à un serveur partagé à partir d'un chiffrage fait une fois par an

A chaque niveau de disponibilité correspondra un tarif particulier

tes budgétaires et du manque de personnel technique.

Le trafic sur le réseau devrait doubler dans les deux prochaines années en raison de l'utilisation accrue des outils de planification, de l'amélioration de la sécurité et de l'introduction des nouvelles applications. L'entreprise doit donc dès maintenant investir fortement dans un nouveau réseau. Mais si l'usage du réseau n'est pas soigneusement suivi et contrôlé, les coûts risquent fort de s'envoler.

Les politiques de refacturation et les pratiques administratives correspondantes sont un des moyens essentiels pour parvenir à une maîtrise des coûts.

Les méthodes actuelles de récupération des coûts consistent en une association de différentes méthodes : des affectations de coûts à un niveau très global ; un système de refacturation simple (avec des accords sur les niveaux de service) ; une refacturation réelle des coûts, qui ne permet de récupérer globalement qu'environ 50 % du coût annuel d'un réseau étendu, auprès des clients internes et externes.

Au plan organisationnel, le service chargé des réseaux étendus est structuré et financé comme un centre de coût dépendant du budget informatique central.

Les résultats

A la mi-1999, l'entreprise a lancé un projet destiné à réétudier l'ensemble de ces méthodes de refacturation. Elle a examiné les meilleures méthodes en vigueur dans les autres entreprises et mis en place des recommandations concernant sa politique future de refacturation.

Les principaux résultats de l'étude ont été les suivants.

Les systèmes de refacturation de l'utilisation effective, sur la base du nombre de paquets transmis, ne sont pas encore mûrs. Moins de 1 % des grosses entre-

prises ont mis en place de tels systèmes de refacturation. L'entreprise continuera à travailler avec des systèmes forfaitaires ou fondés sur la nature des services fournis, jusqu'à ce que les modes de mesure de l'utilisation effective aient mûri, vers les années 2002-2003.

Les directions opérationnelles de l'entreprise considèrent souvent les dépenses de réseaux étendus comme de l'argent perdu. Dans ce cas, elles pensent évidemment que tout le travail de suivi, de mesure et d'évaluation des coûts liés à leur utilisation est inutile.

L'adoption de systèmes de refacturation des réseaux étendus IP n'est pas encore très répandue et l'on constate une ignorance assez générale des avantages de la refacturation, même chez ceux qui la mettent en pratique.

La refacturation des réseaux étendus peut constituer une méthode efficace pour maîtriser l'utilisation de la bande passante. Elle permet de promouvoir la révision de certaines méthodes de travail et elle facilite le dialogue entre les divers partenaires pour la mise au point d'accords sur les niveaux de service. Cependant, si elle est mal mise en œuvre, une telle refacturation sera inefficace.

La logique et les pratiques de refacturation doivent être harmonisées avec la structure de l'entreprise, les évolutions prévues, la stratégie du département informatique et les principes modernes de gestion financière.

Ainsi :

- il faut définir clairement les objectifs visés et les bénéfices attendus de la mise en place d'une refacturation ;
- la refacturation devra être utilisée dans le cadre d'une stratégie globale destinée à s'assurer que les ressources sont utilisées de la façon la plus efficace et ne deviennent pas une arme contre les utilisateurs ;
- le soutien de responsables de haut niveau est indispensable ;

Les politiques de refacturation et les pratiques administratives correspondantes sont un des moyens essentiels pour parvenir à une maîtrise des coûts

L'entreprise continuera à travailler avec des systèmes forfaitaires jusqu'à ce que les modes de mesure de l'utilisation effective aient mûri

- les utilisateurs doivent être tenus responsables des niveaux de service qu'ils exigent.

Les facteurs clés de succès

Cette entreprise est dans une phase cruciale de transition vers la mise en œuvre d'une politique de refacturation de son réseau et la mise en place de structures organisationnelles permettant de gérer la prochaine génération de réseaux. L'étude a montré qu'il n'existait qu'un tout petit nombre d'entreprises qui mettaient en œuvre des méthodes de refacturation fondées sur l'utilisation effective. Les autres entreprises utilisaient des méthodes de refacturation mises au point pour des environnements plus anciens, qui sont beaucoup moins complexes du point de vue de la refacturation.

Les premières tentatives de refacturation des réseaux étendus menées par cette entreprise n'ont pas été très réussies car elles n'avaient pas été "bien vendues" à l'intérieur de l'entreprise.

On aurait pu mieux surmonter les résistances initiales. Pour cela, il est indispensable que le directeur informatique (ou le directeur technique de l'entreprise) se fasse l'avocat de la refacturation. Il faut arriver à un consensus chez les responsables hiérarchiques des utilisateurs et s'assurer que les utilisateurs et les informaticiens comprennent les objectifs à court terme de la refacturation du réseau. Cela nécessite une approche presque "commerciale" du problème afin de communiquer et de former l'ensemble des équipes de l'entreprise.

La première priorité devrait être de faire évoluer les règles de refacturation, leurs conséquences organisationnelles et les pratiques administratives. Ensuite, il faudra donner la priorité aux outils, à la structure du modèle de refacturation ainsi qu'au calcul des taux et des charges.

Il est important d'explicitier et de définir la future structure organisationnelle ainsi que les services qui seront fournis. Dans la plupart des entreprises, le passage à de nouvelles structures de travail (par exemple, à l'utilisation d'un prestataire de service indépendant) devra se faire progressivement. De fait, le passage d'un centre de coût interne à un prestataire de services externe est une opération qui prend du temps et qui coûte de l'argent.

Les utilisateurs et les informaticiens doivent être formés aux objectifs et aux avantages de la refacturation du réseau étendu. Chaque méthode de refacturation est spécifique. Toutes possèdent leurs avantages et leurs inconvénients. La structure d'un modèle de refacturation des réseaux en entreprise sera hybride, et combinera des approches fondées sur l'utilisation effective associées à des approches fondées sur des affectations de coûts ou sur des tarifs forfaitaires.

Il serait intéressant d'envisager un modèle forfaitaire à plusieurs niveaux, fondé sur les besoins en bande passante et sur les niveaux de service. Ceci pourrait constituer un système transitoire en attendant de passer à un système fondé sur les consommations réelles. Cependant, les accords sur les niveaux de service et les coûts doivent être précisés pour chaque groupe d'utilisateurs et pour chaque système. Il s'agit là de tâches qui ne sont pas faciles à mener.

En conclusion

Cette entreprise a pris conscience du fait que, parallèlement aux mutations des techniques des réseaux étendus, l'utilisation efficace de ses infrastructures exige de nouveaux mécanismes de contrôle, tels que la refacturation. L'entreprise a conclu que la mise en œuvre d'une refacturation constitue

Les premières tentatives de refacturation des réseaux étendus menées par cette entreprise n'ont pas été très réussies

La première priorité devrait être de faire évoluer les règles de refacturation, leurs conséquences organisationnelles et les pratiques administratives

Indemnités de licenciement

De la subjectivité à l'objectivité

Eric Boulanger

Avocat à la Cour,
Alain Bensoussan-Avocats,
Directeur du département
Fiscalité et Droit des sociétés

Le régime fiscal des indemnités de licenciement a changé. Ce qui relevait de l'appréciation du contribuable, des services fiscaux ou du juge est dorénavant fixé dans la loi. Plus objectif, le nouveau régime est toutefois moins malléable.

Il était particulièrement difficile d'évaluer financièrement la part de l'indemnité qui correspondait à l'indemnisation d'un préjudice

L'article 3 de la loi de finances pour 2000⁽¹⁾ a instauré un nouveau régime fiscal des indemnités de licenciement, applicable pour les revenus perçus en 1999. En effet, le précédent régime laissait place à une grande incertitude puisqu'il n'exonérait l'indemnité de licenciement de l'impôt sur le revenu que pour les sommes qui compensaient un préjudice autre que la perte de rémunération, c'est-à-dire les sommes versées correspondant à l'indemnisation d'un préjudice.

Ainsi, les tribunaux avaient mis en place un certain nombre de critères qui permettaient de déterminer la part de l'indemnité correspondant à l'indemnisation d'un préjudice. Ils prenaient en

considération, par exemple, des éléments comme l'âge du salarié, la perte de statut social, d'éventuelles difficultés de réinsertion pour évaluer la part exonérée de l'indemnité.

Le montant de l'exonération dépendait donc de l'appréciation souveraine du juge. Celle-ci étant particulièrement subjective, il pouvait en résulter, pour le contribuable, de "désagréables surprises" et des redressements fiscaux importants. En effet, il était particulièrement difficile d'évaluer financièrement la part de l'indemnité qui correspondait à l'indemnisation d'un préjudice. Le contribuable se retrouvait donc dans une situation délicate lors de l'établissement de sa déclaration de

1 - Loi n° 99-1172 du 30 décembre 1999, JO du 31 décembre 1999

revenus puisqu'il devait seul effectuer cette évaluation. Des contribuables de bonne foi pouvaient dès lors mal évaluer la part de leur indemnisation exonérée. Aux "tracas" du licenciement pouvaient alors s'ajouter ceux d'une procédure de redressement fiscal. Plus grave, cette évaluation pouvait être très différente entre les juridictions et engendrer ainsi des "inégalités fiscales" importantes.

La loi nouvelle a le mérite de présenter clairement le régime fiscal des indemnités de licenciement et de distinguer celles qui sont imposables et celles qui sont exonérées d'impôt sur le revenu.

Elle rappelle tout d'abord le principe de l'imposition des indemnités de licenciement. Sont ainsi imposables tous les types d'indemnités : l'indemnité compensatrice de préavis, l'indemnité compensatrice de congés payés, l'indemnité de non-concurrence. Elle prévoit ensuite un certain nombre d'exonérations. Celles-ci ne concernent toutefois que les salariés qui ont au moins deux ans d'ancienneté dans l'entreprise et ceux dont les employeurs occupent habituellement au moins onze salariés. Ainsi sont intégralement exonérées, les indemnités versées en cas de licenciement abusif ou irrégulier, celles versées en cas de violation de la procédure de licenciement et celles versées dans le cadre d'un plan social.

La loi précise, le cas échéant, un mode de calcul objectif de la part exonérée de l'indemnité de licenciement, ne laissant plus de doute quant à une éventuelle qualification ou une appréciation subjective du préjudice subi par le salarié licencié. Elle fixe enfin un montant maximal pour la part exonérée : la moitié de la première tranche d'imposition de l'ISF, soit 2,35 millions de francs.

S'agissant des indemnités exonérées, celles-ci ne le sont que dans la limite du plus élevé des trois montants suivants :

- le montant prévu par la convention collective de branche, l'accord professionnel ou interprofessionnel ou, à défaut, par la loi ;
- le double de la rémunération annuelle brute perçue par le salarié au cours de l'année civile précédant celle de la rupture de son contrat de travail ;
- la moitié du montant total des indemnités perçues.

Toutefois, la part exonérée ne peut excéder la moitié du seuil d'imposition de l'ISF, soit 2,35 millions de francs⁽²⁾.

Prenons l'exemple d'un salarié dont la rémunération annuelle brute de l'année civile précédente s'établit à 600 000 francs, et qui a perçu une indemnité égale à 1 600 000 francs, dont 400 000 francs correspondant à l'indemnité conventionnelle. En application des dispositions nouvelles, l'indemnité sera exonérée à hauteur de 1 200 000 francs et imposable pour le surplus, soit 400 000 francs.

En définitive, le nouveau régime instaure une plus grande égalité entre les contribuables en édictant des règles précises, qui ne sont susceptibles d'aucune interprétation. Ces règles éviteront, à n'en pas douter, de nombreux contentieux.

Ces nouvelles dispositions enlèvent toutefois au juge tout pouvoir modérateur ou correcteur quant à l'imposition des indemnités de licenciement. Ce pouvoir souverain du juge permettait, en effet, d'exonérer de l'impôt sur le revenu une part importante ou même la totalité de l'indemnité dans tous les cas où le licenciement entraînait des conséquences particulièrement graves, notamment pour les salariés dont les possibilités de reclassement étaient faibles. Il n'en demeure pas moins que les tribunaux préciseront leur position dans le futur, notamment au regard du traitement des indemnités à caractère de dommages et intérêts.

Eric Boulanger

Revue d'auteurs, l'Informatique Professionnelle accueille des opinions qui n'engagent pas la rédaction.

Des contribuables de bonne foi pouvaient dès lors mal évaluer la part de leur indemnisation exonérée

La loi fixe un montant maximal pour la part exonérée

2 - Le seuil d'imposition de l'ISF est actuellement fixé à 4,7 millions de francs

BIBLIOGRAPHIE

La Gestion fiscale de l'entreprise 1999, Éditions Tissot 1999.

Référencement

Les bons comptes...



Jean-Marc Berlioux

*Directeur des programmes
de Bouhot & Le Gendre.*

*Rédacteur en Chef de
L'Informatique
Professionnelle.*

*Il intervient depuis plus de
vingt ans dans le domaine
des systèmes d'information.*

Comment perdre de l'argent en voulant en gagner ? Facile, voici une petite histoire édifiante ! Toute similitude avec une situation existante ou ayant existé serait purement fortuite...

Réduisons nos coûts, ça améliorera nos marges. Le raisonnement est impeccable. A moins que...

Une grande société met en place un référencement de ses fournisseurs. Objectif : réduire le nombre de ceux-ci. En effet, avec le stockage des informations, la tenue à jour du fichier et le traitement des factures, la gestion d'un fournisseur coûte de l'argent. D'où l'idée lumineuse : moins de fournisseurs égale moins de frais de gestion, donc plus de marges.

Voyons cela d'un peu plus près ! Le client a besoin d'une prestation.

Supposons qu'il ait deux offres, l'une d'une SSII référencée et l'autre d'une non-référencée. Comme il est pressé et qu'il veut appliquer la politique "corporate", il va choisir le référencé, même si celui-ci est plus cher. Et comme le référencé le sait, il sera évidemment en situation de force pour négocier.

Autre situation, le client ne trouve de solution acceptable que chez un non-référencé. La technique est alors toute simple (elle est parfois suggérée par les acheteurs eux-mêmes). Il suffit pour l'entreprise non référencée d'aller voir une société référencée qui servira d'in-

termédiaire. Le mark-up, c'est-à-dire la dîme prélevée au passage, varie (entre une dizaine de pour cent et cent pour cent du prix initial, paraît-il). Et elle se répercute, c'est bien normal, sur le prix de vente au client.

Essayez de faire une simulation sur la prestation de quelques mois-homme d'un consultant de bon niveau. Juste pour voir si, par hasard, le surcoût ne dépasserait pas largement l'économie !

Jean-Marc Berlioux

A suivre...

Les e-mails sont-ils des correspondances privées ?

Un e-mail émis depuis un poste professionnel constitue-t-il une correspondance privée ? Le débat fait rage depuis un moment. Or un jugement important est intervenu le 2 novembre 2000. Le tribunal correctionnel de Paris a condamné trois cadres de l'Ecole Supérieure de Physique Chimie Industrielle de Paris. Ils avaient pris connaissance des e-mails d'un étudiant à son insu. Un commentaire de ce jugement paraîtra dans notre prochain numéro.

Portage de .NET sur Linux

Corel a déposé auprès de la SEC (U.S. Securities and Exchange Commission) des documents prévoyant la possibilité pour Microsoft de lui confier le portage partiel ou total de la plate-forme .NET de Microsoft sur Linux. L'intention apparente de Microsoft de porter sa future plate-forme .NET sur Linux n'est pas surprenante, compte tenu des efforts fait pour positionner .NET comme une solution équivalente à Java. Le portage de .NET sur Linux renforcera également les efforts de Microsoft visant à soumettre des éléments de .NET (à savoir, la Common Language Infrastructure et le langage de programmation

C#) à l'organisme de normalisation European Computer Manufacturers Association (ECMA).

Les méthodes traditionnelles en vigueur au sein de l'ECMA exigent généralement au moins deux mises en œuvre indépendantes pour que l'adoption d'une technologie soit envisagée. Lors des réunions des groupes de travail de l'ECMA, les représentants de Microsoft ont déjà indiqué que la société travaille sur deux mises en œuvre, dont l'une est basée sur les plates-formes non-Wintel.

La pénurie de compétences ne se réduit pas à une pénurie de personnes

En fait, l'accord donne à Microsoft le droit de faire payer une redevance (ou tout autre mode de rémunération) aux personnes qui souhaitent distribuer des copies de .NET pour Linux. Les entreprises qui prévoient de tirer parti de la plate-forme .NET doivent voir cet accord comme une étape positive, mais seulement comme une première étape, dans la longue marche visant à positionner .NET comme un concurrent sérieux de Java.

USA : doublement des visas high-tech

Le Sénat américain a décidé de doubler quasiment le nombre de visas H1-B temporaires réservés aux travailleurs étrangers hautement qualifiés, en le faisant passer de 115 000 en 2000 à 195 000 pour chacune des trois années qui suivront. Les visas H1-B, qui sont demandés par les employeurs, sont valables trois ans et peuvent être renouvelés pour trois ans. Cette mesure a été votée à la quasi-unanimité au Sénat par quatre-vingt-seize voix contre une. Elle a ensuite été examinée par la Chambre des représentants, qui l'a adoptée. En principe, le président Bill Clinton devrait approuver le projet de loi, dernier passage obligé avant que cette mesure devienne une loi.

Même si l'octroi de visas H1-B supplémentaires est indubitablement une grande victoire pour les lobbyistes de la Silicon Valley qui ont fait de fortes pressions pour faire passer cette mesure, il ne résout pas la grave pénurie de compétences informatiques. La pénurie de compétences ne se réduit pas à une pénurie de personnes et cette augmentation du nombre de travailleurs étrangers disponibles, quoique non négligeable, n'est pas une réponse au problème.

Aux Etats-Unis, le nombre de personnes employées dans l'informatique excède 1,5 million. Etant donné l'importance de cette main d'œuvre, la plupart des entreprises ne sentira pas l'impact de ces 80 000 travailleurs supplémentaires. Les plus touchées seront celles qu'on appelle les body shops, c'est-à-dire les sociétés spécialisées dans la fourniture de personnel informatique pour les projets, et les sociétés high-tech, qui sont les plus gros utilisateurs de visas H1-B. Même avec l'accroissement du nombre de visas, la demande en matière de personnel informatique restera supérieure de 20 % à l'offre jusqu'en 2003.

Oracle9i : Marketing et poudre aux yeux

À grand renfort de trompettes, Oracle a lancé Oracle9i, dernière version de son système de gestion de base de données (SGBD) – Oracle9i Database – et son serveur d'applications – Oracle9i Application Server. Oracle9i Application Server est disponible immédiatement ; Oracle9i Database sera mis sur le marché à la fin du premier semestre 2001.

La vision exprimée par Oracle concernant Oracle9i consiste à proposer un ensemble intégré de technologies (Oracle9i Database et Oracle9i Application Server) comme étant "les deux seules pièces" nécessaires pour une plateforme informatique internet offrant un haut niveau d'évolutivité, de disponibilité et de capacité à être pilotée. Cependant, Oracle9i n'est pas "magique", comme l'a proclamé le directeur général d'Oracle. C'est une amélioration

logique du SGBD de la société et un solide lien marketing avec son serveur d'applications remanié. Les solutions d'Oracle concernant les fonctionnalités d'autoparamétrage, les améliorations destinées aux OEM et la technologie d'envoi de fichiers journaux et de mise en veille (Data Guard) peuvent traiter certains des problèmes concernant la capacité à être piloté et la disponibilité qui ont longtemps gêné les administrateurs de bases de données Oracle, ce qui a poussé nombre d'entreprises vers les solutions de fournisseurs de logiciels

Les analystes Gartner recommandent la prudence en ce qui concerne Oracle9i

indépendants. Les entreprises qui mettent en œuvre des infrastructures de datawarehouse obtiendront enfin en partie l'intégration promise depuis si longtemps des cubes multidimensionnels (basés sur Oracle Express), de la transformation des extractions, du chargement (partiel) et du datamining.

Les améliorations de la fonction cluster d'Oracle (Real Application Clusters) visent également à offrir une plus grande capacité à monter en charge "scalabilité" pour faire face à un nombre important de transactions, ce qui renforce l'intérêt des clusters pour des systèmes transactionnels. En fait, les analystes Gartner estiment qu'à court terme, la plupart des entreprises continueront de faire évoluer leurs systèmes vers des serveurs multi-traitement symétriques mono-

image (single-image symmetric multiprocessing) plus importants. Les analystes Gartner recommandent la prudence en ce qui concerne Oracle9i. Sur le papier, la base de données Oracle9i semble être une amélioration solide du SGBD Oracle8i, susceptible de régler un grand nombre des problèmes pour lesquels les clients Oracle réclament des solutions depuis longtemps. Cependant, les entreprises doivent traiter les annonces d'Oracle concernant les performances de la base de données Oracle9i comme n'importe quelle annonce concernant un produit non encore diffusé. Les entreprises doivent prendre des décisions d'achat basées sur les produits mis sur le marché et sur les références de production en environnement réel. Les analystes Gartner pensent que la plupart des clients Oracle envisageront d'utiliser Oracle9i Application Server pour renforcer les applications internet et intranet multi-niveaux.

Batteries Dell : au feu !

Dell Computer a annoncé le rappel volontaire d'environ 27 000 batteries utilisées dans les portables Dell Latitude. Lorsque certaines conditions sont réunies, ces batteries peuvent créer un court-circuit et s'enflammer. Les clients Dell qui ont acheté les portables incriminés doivent contacter leur gestionnaire de compte Dell, télécharger un utilitaire de diagnostic de batteries (<http://www.support.dell.com>) ou prendre les mesures suggérées dans le bulletin d'informations de Dell relatif au rappel (<http://support.dell.com/battery/press.asp>) pour déterminer si leurs batteries ont besoin d'être remplacées.

