

WEB

Programmer en XUL

LINUX

Créer son application sous KDE

.NET

Formulaires InfoPath sur le Web

JAVA 5

Paramétrer son interface avec Synth

TESTS

- ▶ VisualFox Pro 9
- ▶ Sun Java Studio Enterprise 7.0
- ▶ NetBeans 4.0

CODAGE

Automatique ou manuel ?

LINUX OPEN SOURCE

NOUVEAUX MAÎTRES DU MONDE ?

*Eclipse : bataille gagnée ?
Migrer vers OpenOffice
Les principaux outils*





Code libre et appel aux développeurs ! ENT-e : l'école numérique selon Microsoft

Microsoft marque un grand coup, avec ENT-E, l'Environnement Numérique de Travail pour l'Éducation, dont le code source est mis librement à disposition des développeurs, éditeurs et des responsables de l'éducation. Le projet pilote, conçu par Microsoft France, démarre

sur plusieurs académies. Potentiel : 26 académies, représentant 12 millions d'élèves et 1 million d'enseignants !

Microsoft ouvre ce campus numérique aux développeurs pour créer de nouveaux modules.

Le projet, extrêmement ambitieux, est une galaxie de portails web, personnalisés pour chaque rectorat et chaque établissement, et destinés à accéder et échanger l'information entre tous les publics concernés :



Vincent Bastid

élèves, enseignants, parents d'élèves et administration. Chaque portail est un véritable point d'entrée unique pour un établissement. Ce vaste chantier fait suite à un accord cadre signé entre

Microsoft et le ministère, fin 2003. En effet, après l'équipement en ordinateurs et l'accès à Internet, il manquait aux établissements les outils de communication, de partage en ligne des ressources pédagogiques, et d'applications sur la vie scolaire.

La plate-forme ENT-E a été conçue par Microsoft sur un socle de technologies permettant de gérer simultanément un grand nombre d'établissements, comprenant chacun un grand nombre d'utilisateurs. Ceux-ci accèdent au site et à leur espace personnel, via



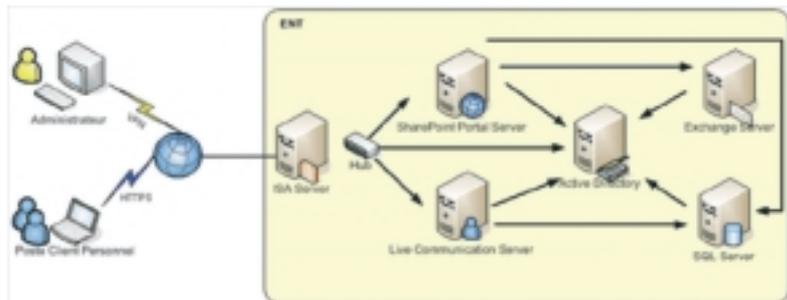
une identification unique et personnalisée. L'outil d'administration SSO est sécurisé SSL.

Partiellement en Open Source

Ce socle (voir encadré) repose sur Windows Server 2003, La licence, vendue 100 euros en moyenne, coûte moins cher que certaines distributions Linux, souligne Vincent Bastid, Marketing Manager Education de Microsoft. La subvention du ministère et de la Caisse des Dépôts s'élève à 350 000 euros, par académie retenue dans l'appel à projet. Le coût par élève "de quelques euros", est environ 5 fois moins cher que les solutions hébergées dans les établissements.

L'interface est personnalisable pour chaque Académie, tant au niveau des fonctionnalités modulaires (webparts) que du design. L'Académie de Montpellier a ainsi fait développer son propre carnet de texte électronique et l'a intégré dans son portail.

"MCS (Microsoft Consulting services) a réalisé le code spécifique de ENTE-e, mis librement à la disposition de tous, et conçu pour intégrer les services à venir", développe par les académies ou des éditeurs indépendants, explique Vincent Bastid. Il souligne : "Microsoft a mis son code spécifiquement développé pour l'éducation en open source ". Dit autrement, l'éditeur "met à disposition le code qu'il a développé en interne pour le portail, en libre accès et usage, modifiable et diffusable. Donc commercialisable ", précise-t-il, à l'usage des développeurs qui sont bienvenus pour réaliser des modules, au sein du portail, à l'intention des académies.



MS-Office, application officielle

L'application maîtresse est bien sûr Microsoft Office. L'éditeur rappelle la présence des correcteurs, orthographique et grammatical, l'utilisation universelle de l'outil, pour le PC comme le Mac. L'accord avec le ministère prévoit une copie gratuite - à usage professionnel - pour les enseignants, à leur domicile, dans la limite des licences déployées dans l'établissement. Un certain nombre de facilités sont accordées à l'Education Nationale. Par exemple, les établissements situés en ZEP bénéficient d'un tarif de 3,60 euros HT par an et par poste, pour une licence d'Office XP. Pro.

Un enjeu de société

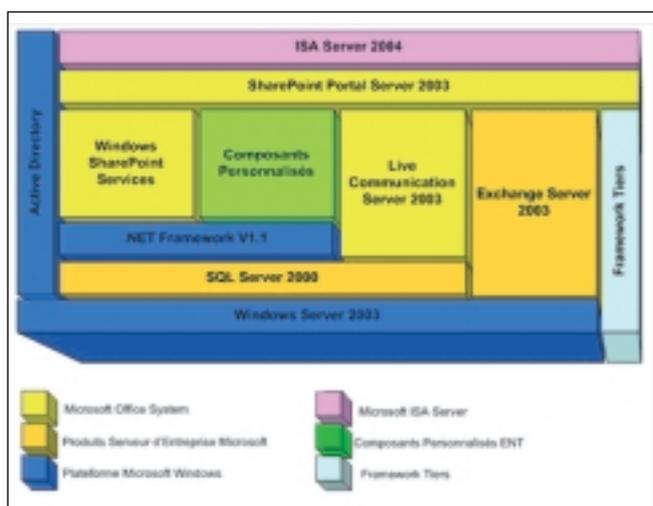
"Le message fort est un enjeu de société", commente Vincent Bastid : "c'est l'implication et le partage de tous à la vie scolaire, y compris des parents. Ils pourront voir l'emploi du temps, les notes, le contenu des cours, et communiquer avec les enseignants."

L'enjeu est de taille pour ce méga projet, initiative de Microsoft France. Après les premières académies, dont Versailles, Créteil, Poitiers, Montpellier, l'éditeur mise sur une montée en puissance au niveau national sur 5 ans. Potentiel : 12 millions d'élèves et 1 million d'enseignants. Sans compter les parents

d'élèves ! Enjeu économique : 9 millions d'euros environ, qui profiteront à l'emploi local, souligne l'éditeur. Et immense bénéfice d'image pour Microsoft qui fournit le socle. Mais il s'agit aussi d'un pari de la filiale française, la plus exposée dans le monde à la vague de l'open source dans l'administration, avec le risque d'exemplarité qu'elle représente. Ce pari est de répondre à l'open source avec un modèle qui mixe le super-discount et le quasi libre. Mais Microsoft a-t-il encore le choix ?

Pour en savoir plus : www.microsoft.com/france/education/ente

■ Jean Kaminsky



Une solution 100% Microsoft

- .Net Framework 1.1
- Windows Server 2003
- SQL Server 2000
- Exchange Server 2003
- Live Communication Server 2003
- Sharepoint Portal Server 2003 (travail collaboratif)
- ISA Server 2004

Exemples de fonctionnalités

- Pour l'élève : accès aux documents de cours, messagerie, carnet d'adresses (incluant professeurs et autres élèves), agenda, devoirs à rendre. Dans le cadre de groupes de travail : chat, visio-conférence et document partagé ("tableau noir"), forums, listes de diffusion ;
- Pour l'enseignant : partage de documents, avec ses élèves ou des collègues, communication avec les parents ;
- Pour le chef d'établissement : communication des emplois du temps et des congés, suivi des absences, suivi de l'occupation des salles et des matériels pédagogiques (développements envisagés, à réaliser par des éditeurs indépendants et développeurs tiers) ;
- Pour les parents d'élèves : accès aux emplois du temps, aux notes de l'enfant, connaissance de la vie de l'établissement.

Sun Java Studio Enterprise 7.0 : du costaud

L'éditeur continue à remettre au goût du jour sa gamme d'outils de développement. Après Java Studio Creator, la version Enterprise arrive en version 7 avec des nouveautés sans doute très attendues, telles que la modélisation et le travail en équipe. La v7 réserve même de bonnes surprises.

Disponible depuis quelques semaines, la version 7 de Java Studio Enterprise booste l'offre de développement Java orienté entreprise, en hissant son outil au niveau de la concurrence, avec parfois, une petite avance. Dans cette version, le focus a été mis sur plusieurs éléments : le refactoring de code, les applications J2EE (simplification de la création), la modélisation avec UML 2, la qualité des performances et des tests (via Junit), et le travail collaboratif. Il fonctionne sous Solaris et Windows.

UML et Wizards

La modélisation UML 2.0 est disponible en standard. Le modèleur se veut pratique et complet. Pour faciliter l'approche, il propose différents assistants pour concevoir les diagrammes de bases. Bien entendu, à partir du diagramme UML, on génère automatiquement le code adéquat. Pour aider encore un peu plus, on peut utiliser des Design Pattern (via des assistants). On peut aussi définir les scénarios d'utilisation de son projet. En complément, l'outil dispose d'un générateur de rapport (menu Modeling). Le modèleur à l'usage se révèle plaisant et fonctionnel.

Créer des applications J2EE n'est jamais une chose facile. Java Studio Enterprise 7 tente d'en simplifier le processus, via le Web

Application Framework. Il s'agit de disposer d'une interface de création visuelle (avec notion de réutilisation des composants). Par défaut, l'outil implémente différents templates, afin de faciliter la création selon un contexte donné. On peut même convertir des applications J2EE en Web Application Framework.

Aidons le développeur

L'éditeur, malgré une certaine complexité, possède une panoplie d'aides au codage très pratique. Outre le fameux Code Folder (qui se généralise dans tous les IDE), on peut définir sa propre liste de TODO, très pratique, quand on doit revenir à différents endroits du code (la liste prend en compte le fichier affiché et ceux indiqués par l'utilisateur). L'implémentation du refactoring aidera aussi le développeur lorsqu'il doit, par exemple, renommer ou extraire des méthodes. Sur une machine peu véloce, on constate tout de même une certaine lenteur de la vérification à la volée du code.

Applications Web / J2EE obligent, l'environnement de Sun introduit avec la v7, un puissant profiler. Il couvre différents besoins : les transactions, les statistiques EJB, trace et analyse des méthodes et de la mémoire. On peut ainsi réaliser des tests de charge et analy-

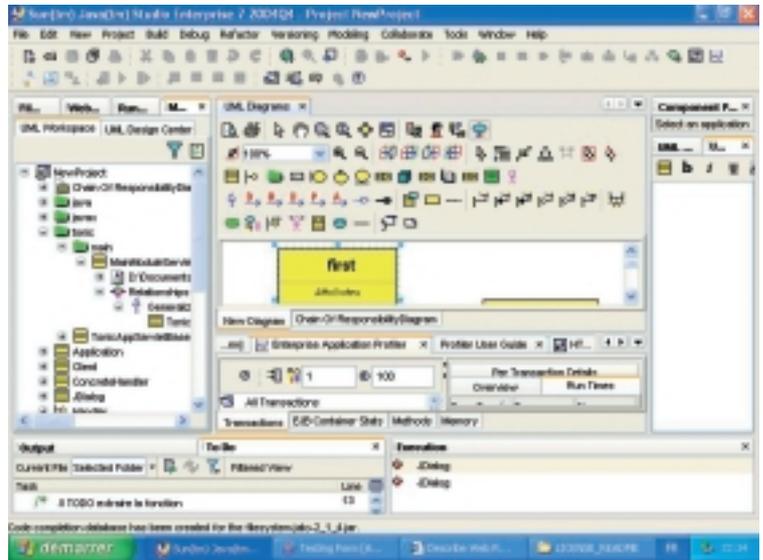
ser en temps réel le comportement http ou des composants J2EE. On peut donc assez facilement traquer les pertes de performances et où optimiser son application.

Une collaboration active !

Un des points forts de cette v7 concerne la collaboration. L'outil intègre un module Collaborate. On définit très simplement les membres de l'équipe. La connexion se fait via une interface particulièrement réussie, qui permet de dialoguer rapidement avec d'autres développeurs. Grâce à une messagerie instantanée, un dossier partagé, une liste de contacts, on évite ainsi de sortir de l'environnement de travail.

Une ergonomie pas toujours heureuse

Les nombreuses fonctions n'aident pas à travailler dans un environnement entièrement ergonomique. On ne retrouve pas la nouvelle organisation de NetBeans 4.x, Java Studio Enterprise 7 étant encore basé sur NetBeans 3.6. La prise en main de certains éléments n'est pas toujours évidente



notamment pour l'éditeur et le modèleur UML et la collaboration raviront bien des développeurs. Cette profusion nécessite l'emploi d'un grand écran pour pouvoir travailler confortablement. En tout cas, l'achat d'un tel environnement se justifie pour les équipes de développement J2EE exigeantes. Sun dispose là d'un concurrent viable face à Eclipse / IBM Atlantic (dans une certaine proportion) et à JBuilder Enterprise.

■ F.T.

Fiche technique

Produit :
Java Studio Enterprise 7.0

Éditeur : Sun

Prix : 1 895 \$ par développeur et par an ou 5 \$ par employé (licence / an) si au moins 1 000 employés. Version d'évaluation disponible (30 jours)

Les + : travail collaboratif, support d'UML 2, le refactoring, profiler, éditeur

Les - : basé sur NetBeans 3.6, prise en main

Cegid : La chaîne du développement plutôt que le développement à la chaîne

Suite à sa récente fusion avec CCMX, Cegid est devenu le premier éditeur français de progiciels. Avant la fusion, l'équipe de R&D comptait 265 personnes. Après, elle en comptera 430. Cegid a opté pour un Agl, développé par cinq personnes et ensuite utilisé par tous les développeurs. Il peut en cela illustrer des processus de développement qui tendent à se généraliser.



Christophe Raymond, Directeur technique.

Rappel du contexte : Cegid est un éditeur de progiciels de gestion, d'origine lyonnaise. Historiquement, il développait des produits pour les experts comptables (Profession Comptables Libérale ou PCL). Il intervient désormais sur trois marchés : la PCL, les entreprises en général, avec son ERP modulaire, et s'est focalisé sur certains secteurs d'activité (mode, hôtels-restaurants, BTP) avec des offres dédiées.

Le choix d'un AGL

En matière de développement, la révolution chez Cegid date de 1999, date à laquelle Jean-Michel Aulas, président de Cegid a pris la décision de réécrire l'ensemble de la gamme. En termes de technologies, en passant d'un mode caractère à Windows et SQL, et en passant d'un développement parcellaire à un développement ERP (cf. Schéma 1). L'éditeur a donc fait le choix de bâtir un Atelier de Génie Logiciel (AGL) qui lui est propre. Un travail considérable souligne Christophe Raymond, directeur technique de Cegid, qui avait pour objectif de disposer d'un seul et même outil pour toute la gamme, ce qui permet " des développements rapides et normalisés ". L'AGL

est orienté objet et 100 % natif SQL. Il inclut notamment un générateur de documents, un générateur de statistiques, un cube décisionnel, utilisables pour tous les produits.

La base de l'AGL était Delphi, langage auquel toutes les équipes de développement ont été formées, indique Emmanuel Pliez, ingénieur méthodes. Ce n'est pas pour autant l'unique langage utilisé. Il l'est essentiellement sur la partie applicatifs. Java a été retenu pour les kits d'installation, et ce qui est sur le serveur au niveau smart client a été développé en C++. L'homogénéité du développement est assurée par l'utilisation de l'AGL, une ligne de code de l'Agl pouvant être utilisée par plusieurs dizaines de développeurs. Hors AGL, les sources comptent quelque 2 millions de lignes de code, avec de nombreux sources communs et/ou partagés, précise Paul Chapuis, architecte PGI. Le modèle conceptuel de données est centralisé. La Socref est une base de référence

contenant description logique des données, tablettes, modèles d'états, écrans, paramétrage des listes et paramétrage société. Elle sert à mettre à jour les données clients.

Développés par une équipe de cinq personnes, les composants constituant l'AGL sont ensuite utilisés par les différents pôles de développement, soit quelque 200 personnes : ceux dédiés aux développements de produits génériques (comptabilité, gestion commerciale, paye, gestion de production), ensuite verticalisés par ceux qui se consacrent aux développements métiers.

Cegid ayant depuis peu développé un réseau de distribution indirect, l'éditeur fournit également à ses partenaires distributeurs un gestionnaire d'extension progiciel, afin que ceux-ci puissent adapter certaines fonctionnalités.

Multi-Bdd, multi systèmes

Cegid a toujours proclamé son agnosticisme en matière de bases de données (Oracle ou SQL Server, et depuis peu, DB2) et de systèmes : Windows, Unix et depuis peu, Linux et AS/400.

L'utilisation de l'AGL permet de rendre transparent pour les développeurs du progiciel le

portage sur une nouvelle base de données ou un nouveau système, celui-ci étant effectué en amont par l'équipe en charge de l'AGL. " La personne qui implémente les spécificités des SGBD dans l'AGL doit la rendre transparente pour le développeur final ", souligne Emmanuel Pliez.

Le choix de Linux illustre, selon Christophe Raymond cet agnosticisme. Une solution choisie par plus d'une quarantaine de clients en un an (sur 67 000 clients...). Lors de la signature en 2003 d'un accord de partenariat avec Microsoft autour de Microsoft business framework (MBF), Cegid avait parfois été accusé de s'allier au plus fort...ou de se jeter dans la gueule du loup. L'annonce de la disponibilité sous Linux et iSeries fait contrepoids...

Pour assurer les tests dans ces multiples configurations, Cegid a récemment mis en place des serveurs virtuels afin de simuler les cas clients de façon exhaustive, et utilise pour cela Virtual Server de Microsoft.

En matière d'architecture, la politique de Cegid est également d'être ouvert : client/serveur ou client léger. L'éditeur s'est toutefois fortement orienté, notamment avec la V.6 du PGI lancé en octobre 2004 sur le " smart client ", avec Cegid Web access, une solution qui, selon Christophe Raymond, tend à " devenir un standard de fait et allie le meilleur des deux mondes ".

Le respect des standards est une préoccupation permanente, avec une ouverture là aussi, la plus large possible. Ainsi, l'exportation des données doit se faire dans tous les formats (PDF, XLS, XML). L'intégration bureautique est

The screenshot shows a software interface titled "CEGID - Calculateur IAS - Analyse". It features a navigation menu on the left with options like "Cubier décisionnel", "Analyse statistique", and "Analyse Multi-axes". The main area displays a "Cubier décisionnel des ratios" with a table of financial data. The table has columns for "Cubier", "Tableaux", "Max en ligne", "Filtres", and "Graphique". The data is organized into sections for "Lyon" and "Paris", with various sub-categories like "A. Investir", "Achats - module GREGCOM", and "OP. D. de régularisation".

Le module IAS/IFRS de CEGID

également possible à partir du PGI, vers Microsoft Office comme vers StarOffice. Les liens OLE permettent " l'intégration transparente et directe de données PGI dans les outils bureautiques ". Enfin, Le SGBD/R est interrogeable hors applicatif et compatible avec les outils décisionnels du marché, bien que Cegid possède sa propre gamme décisionnelle, via sa filiale Cegid Business Intelligence.

Une chaîne de développement

L'organisation du processus de développement découle de l'adoption de cet AGL, devenu l'unique outil pour développer tous les

modules de l'ERP et leurs déclinaisons par métier.

Sur l'équipe de plus de 250 personnes, cinq personnes sont dédiées à temps plein à cet AGL, 180 font du développement pur et 50 sont chefs de produit, chargés qualité et responsables d'analyse (cf. schéma 2).

La division recherches et méthodes compte six personnes, chacune ayant son domaine de prédilection, sur lequel il est référent explique Emmanuel Pliez : validation des produits, SGBD, sécurité, solution télécoms. Cela permet aux équipes de développement, et au service d'assistance téléphonique, d'avoir un interlocuteur identifié pour un problème donné. Ces équipes (AGL et plates-formes & méthodes) ont également un rôle de veille technologique, indique Emmanuel Pliez. Elles étudient les nouvelles normes, standards, outils et leur impact éventuel sur la gamme Cegid. La gamme des sujets étudiés est large, de XAML aux web services. Il leur incombe d'avoir les compétences techniques nécessaires pour fournir un nouvel outil, au moment où le besoin fonctionnel existera.

Concrètement, explique Christophe Raymond, chaque chef de produit générique, en relation avec les clients et le comité technique, est responsable de l'analyse du plan de développement. Il dispose d'une équipe de 1 à 20 développeurs et d'une équipe qualité. Les développements métier sont sous la responsabilité du

Exemple de fonctionnement

Par exemple, En tant que responsable de projet comptabilité Trésorerie, Régis Rohault centralise les demandes d'évolution et d'améliorations et juge de la pertinence de ces demandes, et de la nécessité d'y répondre immédiatement, pour une version intermédiaire, ou de les prévoir pour la version majeure suivante. Il en vérifie la faisabilité, les priorise et une fois la demande acceptée, la planifie et la découpe pour la confier à ses équipes. Ces demandes viennent essentiellement du SAT, mais également des clubs utilisateurs.

Toutes les informations et le suivi sont saisis et consultables dans l'outil eQualité, qui produit des fiches qualités qui servent à planifier le développement. Y sont également saisies les anomalies remontées par le service qualité, lesquelles sont affectées à l'un ou l'autre des développeurs, qui intégrera la correction dans la source. Chaque vendredi est générée une exe à tester, qui rectifie une liste d'anomalies. Les chargés de qualité vérifient et valident les corrections. Toutes les corrections sont suivies dans l'outil de travail collaboratif Teamlink : ce qui a été fait, dans quelle source, rattaché à quelle fiche qualité. Objectif, souligne Régis Rohault : " suivre l'ensemble des sources et ne rien perdre des fonctions implémentées "

chef de produit. Il y a environ un chargé qualité pour deux développeurs. Ces chargés qualité sont en charge des tests, de la documentation, de l'élaboration des bases modèles et de démonstrations et enfin de la communication et de la formation interne.

Au sein de la direction technique a été mise en place une direction qualité, garant des méthodes et de la qualité de développement. Sa mission, explique Christophe Raymond est d'assurer le suivi des projets, la gestion des fiches de bug et des codes sources et d'assurer une cohérence en termes d'ergonomie. Ce souci de la qualité est le corollaire d'un souci d'économie : les 250 hot liners de l'éditeur reçoivent entre 30.000 et 40.000 appels par mois. " Ecrire correctement une fonctionnalité, en termes ergonomiques, et sans bug, c'est diminuer le nombre d'appels à la hot line ", constate Christophe Raymond.

Sur des projets à long terme, comme la sortie d'une nouvelle version, un des aspects essentiels, précise Paul Chapuis, est la synchronisation des équipes et le partage de savoir-faire, donc la communication entre les équipes. Selon lui, il s'agit de trouver un équilibre entre respect des méthodes et évolution, respect des échéances et réactivité, respect des normes et innovation et enfin, industrialisation des processus et gestion des ressources humaines. Un des outils développés pour ce faire est Ctechweb, un intranet, mais aussi un outil de travail collaboratif, Tealink, développé en interne et enfin eQualité, un outil également développé en interne permettant le suivi des corrections et du plan de développement, qui inclut également une base de connaissance et les plans de scénarios et de tests.

le cycle de vie d'une version

Cegid sort une nouvelle version de son ERP chaque année. Paul Chapuis explique le cycle de vie d'un tel projet.

- Plan de développement : élaboré par les chefs de produits et chefs de projet, il prend en compte les engagements clients, les remarques des clients, les orientations issues des clubs utilisateurs et comités techniques.
- Développement : analyse, validation de l'analyse, développement en dehors des sources communs, test et validation par la qualité, intégration dans le projet commun, validation finale après intégration
- Qualité : les chargés qualité assurent les tests avant et après intégration des développements, les plans de test, les scénarios de test (non-régression, nouvelles fonctionnalités) et la documentation.
- Plan qualité : il prévoit des réunions de coordinations entre chefs de produits, développeurs et chargés qualité, et entre toutes les équipes de l'ERP. le plan d'assurance qualité définit organisation, méthodes et procédures, processus et moyens mis en oeuvre pour atteindre cet objectif de qualité.
- Dernière étape, le comité feu vert, qui valide la mise en diffusion.

Rester au contact client

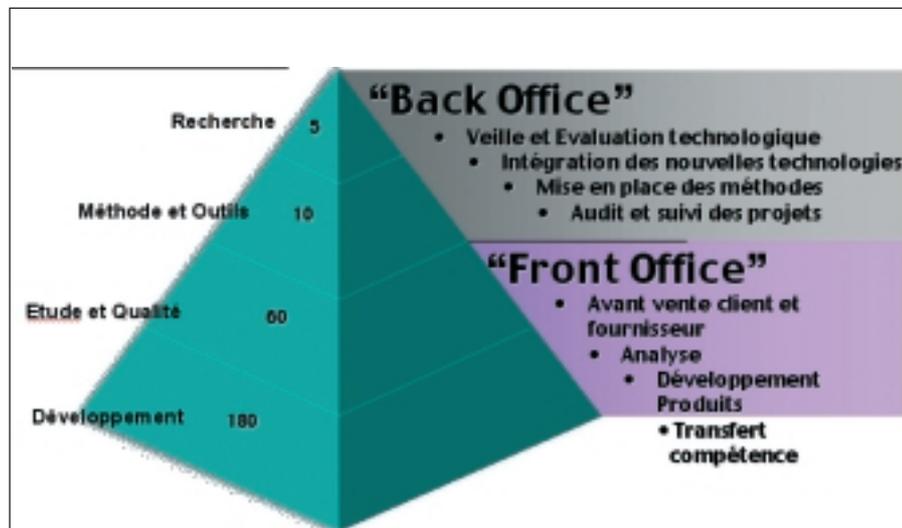
Les développeurs des produits métiers ont pour mission de sortir une fois tous les deux mois pour faire une installation, explique Christophe Raymond. Ils participent également aux clubs utilisateurs pour présenter aux clients les nouveautés. Les clients sont étroitement associés aux plans de développement, particulièrement sur certains secteurs, comme les experts comptables. " Le point commun à tout développeur Cegid est la culture du service - il n'est pas déconnecté de ce qui se passe - et l'apprentissage des aspects techniques ", affirme Pascal Guillemain, DRH de Cegid. Certains développeurs, ajoute Paul Chapuis, travaillaient en C et mode caractère et ont dû réapprendre un langage et la méthodologie objet.

L'équipe est très panachée, avec des origines

très diverses souligne Christophe Raymond.

" Cette ouverture d'esprit fait la richesse de la R& D de Cegid ". Ceci s'explique par les multiples rachats effectués par l'entreprise, dont en 1998 Apalatys, une société d'ingénierie dirigée par Christophe Raymond qui développait pour Saari, Sage ou CCMX, et en 1999 Servant Soft, éditeur de progiciels de gestion. Les équipes de R&D demeurent d'ailleurs dispersées sur sept sites : majoritairement à Lyon bien sûr (131 personnes), mais également à Roanne (33), Paris (18), Aix-en-Provence (15) Dijon, Annecy et Lorient. Avec le rachat de CCMX vont s'y ajouter des équipes sur Lille, Nancy, Orléans, Saint-Brieuc.

Pour faire fonctionner l'ensemble, les outils et les méthodes sont donc industrialisés. Les développeurs sont fortement incités à travailler en équipe et à demander ce qu'ils ne savent pas plutôt que d'essayer d'aller chercher quelque chose sur Internet dans leur coin " sur 250 personnes, il y a forcément la compétence ailleurs ", souligne Christophe Raymond. Un travail de formalisation des compétences génériques a d'ailleurs été récemment effectué pour intégrer dans une base de connaissance commune. ■ Carole Pitras



Cegid en chiffres

Date de création : 1983

Chiffre d'affaires 2003 : 130 M € (220 M € en y ajoutant CCMX, acquis en novembre 2004)

Effectif : 1 400 au 31/12/2003, 2 200 avec CCMX

67 000 clients (87 000 avec CCMX).

3 TECHNOLOGIES : VERS LA MATURITÉ

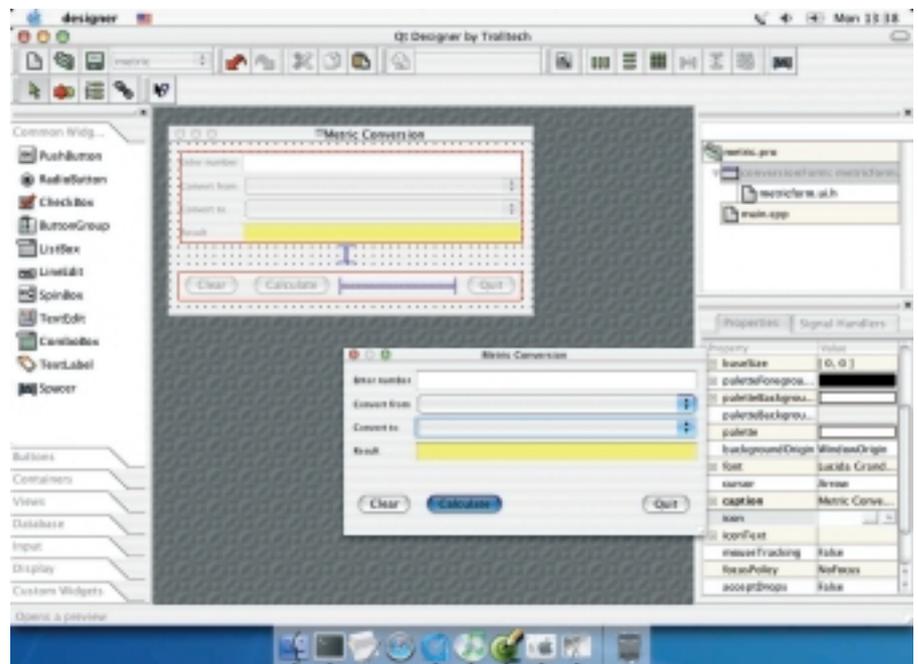
Outils : Une nouvelle façon de développer

L'utilisation de composants ou d'IDE ouverts se banalise et devient naturelle que ce soit sous .NET, Windows, MacOS X, Linux, Java, etc. Le développeur doit déterminer les outils "ouverts" à prendre et comment les intégrer dans ses projets. Désormais, on peut estimer que le tout propriétaire n'a plus grand sens, excepté dans des domaines très spécifiques, et que le tout Open Source n'a pas de sens non plus, sauf exception. Les deux mondes sont complémentaires. Sachez tirer parti du meilleur des deux univers. Bien utilisé, vous y gagnerez forcément ! Misez donc sur le mixage.

L'usage technique des outils ouverts est une chose, mais cela cache en réalité une nouvelle manière de développer. Les outils et composants ouverts permettent de se rendre tout ou partie, indépendant des éditeurs commerciaux – propriétaires. Le développeur n'est donc plus dépendant de la seule volonté d'une société. Cela peut être un avantage pour la réactivité et quand on souhaite changer d'environnement. De plus, les outils ouverts prennent souvent mieux en compte les standards (dits ouverts) du marché. Point important pour l'interopérabilité et quand on change d'outils. Les éditeurs commerciaux s'appuient aussi sur les standards, les outils et composants ouverts pour compléter leurs propres environnements. Ainsi, dans les frameworks, le test unitaire, le compilateur... on retrouve souvent les mêmes composants ouverts d'un éditeur à un autre.

Linux et multi-plate-forme

Le déploiement de GNU Linux sur le poste client ouvre de nouvelles perspectives aux développeurs, même si la tendance demeure encore faible au regard des projets sous Windows. L'offre de développement entre Windows et Linux est très proche, à quelques domaines près. En .NET, les outils ouverts sont de plus en plus nombreux sous Windows. Et Mono offre une belle solution pour créer des applications multi plate forme Windows – MacOS X – Linux. Le côté multi plate forme est important. Les



Qt : un puissant RAD C++ multi-plate-forme.

outils open source dans ce domaine sont pléthoriques, surtout autour de Java, J2EE. Eclipse, Struts, JSF, NetBeans, voilà quelques noms bien connus du développement Java, ils sont tous ouverts. Si les éditeurs commerciaux ont compris qu'il n'est plus réellement utile de vendre des IDE Java, ils proposent alors des plug-ins. Le marché des plug-ins Eclipse est très actif. De nombreux composants commerciaux existent afin d'apporter les fonctions manquantes à Eclipse. Par exemple, NitroX, petit nouveau sur le marché français (distribué

par IDEO) est un plug-in graphique, permettant de créer visuellement, de gérer et de debugger facilement des JSP et du code Struts. Les outils ouverts ne peuvent entièrement remplacer les outils commerciaux, notamment dans les environnements intégrés, les tests de montée en charge, l'automatisation des tests, l'orchestration de Web Services, la modélisation MDA / UML, etc. L'un des avantages des IDE libres est que l'on peut entièrement les configurer, les adapter à ces besoins réels. Eclipse est sans aucun

doute le meilleur exemple. IBM avec la gamme Atlantic illustre une nouvelle forme d'IDE modulaire, en s'appuyant sur Eclipse et en proposant uniquement des modules complémentaires. Revenons sur les aspects multi-plateformes, à part Java qui facilite la vie, d'autres solutions plus intégrées existent. Il y a Mono qui permet de développer des applications C# pour plusieurs systèmes.

Éditeurs et cross platform

Depuis plusieurs années, un vaste mouvement de portage d'outils et de composants de développements commerciaux s'est fait sur GNU Linux. Tous les principaux éditeurs, hormis Microsoft, ont tout ou partie de leurs offres de développement sur ce système. Cela évite donc bien souvent de se poser la question de savoir sur quelle plate-forme développer. Tout ce qui est Java / J2EE fonctionne, en général, dans les deux systèmes. Il est intéressant de noter que des éditeurs comme PC Soft portent sur Linux leur environnement. Le fait que PC Soft ait porté WinDev sur Linux prouve qu'une demande existe auprès des entreprises. Ainsi, il devient simplissime de faire une application Windows - Linux. Gupta, avec Team Developer propose aussi un environnement unique pour les deux environnements. Borland, avec le couple Delphi - Kylix reposait sur la même idée. Si vous avez à travailler sur les deux systèmes, privilégiez donc des outils/composants fonctionnant dessus. L'un des défis de demain concerne la migration des applications métiers. Sur ce point, on peut sans doute définir deux tendances :

Quand Adam Kolawa (Parasoft) juge le marché : "Tous les développeurs devraient s'inspirer de l'esprit Open Source"

Nous avons rencontré Adam Kolawa durant le salon Solutions Linux. Ses commentaires sur le marché et ses tendances valent le détour.

"50 % des développeurs utilisent Eclipse, 50 % non. Dans le futur, Eclipse progressera encore. Je pense que JBuilder mourra. NetBeans est aussi condamné à disparaître. Dans le modèle de développement des outils Open Source, seuls les meilleurs développeurs demeurent. Ils doivent prouver leur valeur. C'est un monde brutal. C'est en quelque sorte un darwinisme. Mais, les développeurs (de tout bord) devraient s'inspirer de l'esprit Open Source et de sa communauté pour leur propre code. Dans le futur, le code sera soit développé sous Linux, soit sous Windows, et il deviendra multi-plate-forme. Nous avons cette possibilité avec Java. Beaucoup de développeurs codent en Java et il n'y en a pas tant que ça sous Linux. Eclipse fonctionne très bien sous Windows. Sur la comparaison Linux, Windows, Linux n'a pas tous les problèmes de Windows, car la conception de Linux est mieux organisée en interne et le code mieux vérifié. Le noyau est contrôlé par un nombre restreint de personnes, et ils connaissent le code qu'ils doivent vérifier. Notre philosophie est d'avoir de petites équipes de développement, pas plus de 12, idéalement de 5 à 7. Les cerveaux ne s'ajoutent pas ! "

indépendance de l'application par rapport aux systèmes, code multi-plate-forme. Dans le premier cas, l'application Web (portail, J2EE...) représente la meilleure solution. Et là, au niveau client, on est indépendant du système. Dans le second cas, le problème devient un peu plus épineux. Cela peut dépendre du code existant (s'il y en a). Dans le cas d'un VB, il faudra entièrement réécrire en utilisant un langage indépendant ou un IDE multi-plate-forme. Une solution comme Gupta, PC Soft, Mono, voire Kylix - Delphi, répondra à vos

attentes. N'oubliez pas non plus qu'un environnement comme Eclipse sait aussi coder dans d'autres langages que Java. Il existe de petits éditeurs proposant de la cross compilation (ex. : RealBasic fonctionnant sur Windows, MacOS X, Linux). Si vous cherchez la performance et le fonctionnement natif, le C/C++ constitue le choix idéal. De nombreuses solutions libres ou commerciales existent. Parmi les plus connues, on citera : WxWindows, Swig, FLUID/FLTK, GTK+, Qt. Le C/C++ représente aussi une bonne solution pour prendre en compte les terminaux mobiles, bien que toutes les bibliothèques d'interface ne sont pas disponibles sur ces terminaux.

En résumé

	Pour	Contre
Développement web : site	Langages ouverts (PHP, Python) Possibilité de faire de l'ASP.NET (Mono, WebMatrix) Nombreux IDE pour le dynamique Couplage PHP-MySQL idéal dans bien des cas	Pas d'outils à la hauteur d'un GoLive ou d'un Dreamweaver Certains IDE PHP ou Python payants Peu d'outils réellement visuels
Développement web : J2EE / EJB	Offre pléthorique dans les IDE Nombreux frameworks et composants Outils visuels Communauté très active	Qualité très variable des composants Apprentissage parfois long
Développement Java	Nombreux outils dans tous les domaines	Qualité variable Richesse fonctionnelle parfois limitée Pas toujours adapté aux projets complexes et lourds Outils de tests encore limités
Développement .NET	Choix de plus en plus étoffé Possibilité faire du C# multi-plate-forme	Offre encore jeune Limitations de Mono
Développement mobile	Quelques frameworks et outils supportant J2ME	Offre limitée

Mobilité

Si dans l'univers de l'Embarqué, les outils et bibliothèques ouvertes ont déjà largement envahi le domaine (notamment Eclipse), les applications mobiles (TabletPC, PDA, Smartphone), la percée est bien plus faible. Si Java, avec J2ME, domine le marché, notamment sur les téléphones, les outils visuels libres ne sont pas légion. Avec l'arrivée prochaine de NetBeans 4.1, on bénéficie d'un RAD dédié à J2ME ! Une grande avancée. Dans ce domaine, l'Open Source a encore du travail pour combler le retard avec l'offre de développement commercial.

■ **François Tonic**

Eclipse : bataille gagnée ?

Nous l'écrivons depuis longtemps, dans Programmez ! les dernières études, notamment réalisées aux USA, montrent la croissance irrémédiable d'Eclipse par rapport à la concurrence. Les éditeurs commerciaux d'IDE Java, et on pense immédiatement à JBuilder, devront faire des choix dans les prochains mois. Chez IBM, on ne cache pas que la bataille est gagnée ou quasiment. Le marché aurait-il déjà choisi ?



L'avantage d'être un IDE modulaire et une plate-forme

Eclipse n'est pas uniquement un IDE. IBM et la fondation Eclipse mettent en avant la plate-forme Eclipse. Il s'agit d'un socle technique que tout développeur ou éditeur peut reprendre afin de bâtir sa propre solution. Déjà, plusieurs éditeurs, notamment dans le monde de l'Embarqué (QNX et WindRiver) utilisent Eclipse pour concevoir au-dessus leurs outils de développements. IBM migre au fur et à mesure ses solutions de développement sous Eclipse. Eclipse constitue donc un socle technique commun et tente ainsi d'unifier le modèle de développement Java.

La modularité a été l'un des atouts et une raison de son succès. Car, finalement, Eclipse demeure, à la base, un IDE assez frustré, avec les fonctions basiques. Pour tout le reste, il faut ajouter des plug-ins. Et ce marché a véritablement explosé, plusieurs centaines sont disponibles. Des éditeurs commerciaux ont bien compris l'intérêt. Ils proposent des composants payants. Borland propose lui aussi une solution pour Eclipse... IBM fait de même avec la gamme Atlantic, mais l'éditeur précise immédiatement que l'on peut mixer des plug-ins IBM et des plug-ins extérieurs. Il devient alors facile d'adapter l'environnement selon ses besoins réels, sans pour autant acheter un package complet parfois mal adapté aux besoins réels. Et là, on peut donc minimiser le budget. Cependant, le revers de la médaille est le temps perdu à chercher le bon composant et à le tester, la qualité étant variable. Quand un éditeur sort un plug-in fonctionnel, on peut être certain dans 99 % des cas, qu'Eclipse sera choisi. Dernièrement, il y a eu les annonces de Borland, Sybase ou encore BEA avec Beehive et le projet Polinate. De plus,



Eclipse ne se limite pas à Java / J2EE. On peut coder en C/C++, Python, voire même en C# !

Du client riche aux logos certifiants

Depuis la version 3.0, Eclipse met en avant la notion de client riche. Cette appellation, un peu trompeuse, désigne les éléments de l'Eclipse Platform, utilisés pour concevoir des applications basées sur Eclipse. Il est possible de mélanger le client riche Eclipse et l'IDE Eclipse. Désormais, Eclipse s'appuie sur les couches suivantes : Rich Client Platform (le bas niveau), au-dessus le Tools Platform, les frameworks et enfin tous les outils de développements (Java, C/C++, tests et performances, le développement Web...). Le client riche à la sauce Eclipse c'est : une ouverture totale, les standards, la mise à jour automatique, une indépendance par rapport aux systèmes.

Autre aspect qui démontre la maturité d'Eclipse, la création d'une labellisation, sanctionnée par des logos, attribués par la Fondation Eclipse. On aura le choix entre : Built on Eclipse (construit sur Eclipse), Eclipse Ready (prêt pour Eclipse) et enfin Eclipse Foundation Member. Ces logos n'ont rien d'anodin. Il s'agit pour les développeurs de voir immédiatement si tel ou tel outil utilise Eclipse ou s'utilise avec Eclipse. On devrait

donc rapidement observer une réelle dynamique de marché autour de ces logos.

NetBeans : à quoi bon ?

Seul le marché décidera du vainqueur. Pour le moment, Eclipse s'impose en tête des classements. Cependant, les autres solutions existent toujours. NetBeans, l'autre concurrent libre, promu par Sun, tente d'apporter une réponse alternative. Si les premières versions manquaient de puissance et de richesses fonctionnelles, avec la version 4.0, et la future 4.1, NetBeans dépasse Eclipse sur plusieurs points : ergonomie, prise en main, développement J2ME. Il dispose d'un RAD, (qui manque à Eclipse) et supporte dès maintenant Java 5. Si Eclipse demeure encore un peu trop complexe pour le débutant, NetBeans au contraire bénéficie d'une bonne ergonomie. Malheureusement, l'outil peine à sortir de son carcan Sun, l'unique éditeur majeur à utiliser NetBeans comme socle technique pour ses outils (Java Studio Creator et Enterprise). On peut tout de même espérer une progression de NetBeans auprès des développeurs. Les lacunes dans l'ergonomie et la prise en main pouvant partiellement se réduire avec des plug-ins de type MyEclipse, qui permettent de développer rapidement en J2EE (notamment grâce à un RAD) et pour moins de 30 dollars US !

Une autre réponse à Eclipse de la part de Sun a été le Java Tools Community. Une initiative qui visait à proposer un socle commun aux outils de développement, afin d'obtenir une meilleure interopérabilité entre les outils. En juin 2004 (soit 6 mois après l'annonce de l'initiative), Borland annonçait son ralliement. IBM refusa dès le départ d'y prendre part. Résultat, l'initiative est dans un profond coma !

■ François Tonic

Panorama des serveurs d'application J2EE Open Source

Longtemps cantonnés aux petits projets ou à des fins de formation, voire d'usage personnel, les serveurs d'applications Open Source constituent aujourd'hui une alternative vraiment sérieuse aux offres commerciales du marché.

En effet, la diversité de l'offre est désormais réelle, et la maturité et la complétude des solutions proposées font des serveurs d'applications J2EE Open Source des acteurs crédibles lorsqu'il s'agit de mettre en place une telle solution. Les outils couvrent désormais toute la palette des attentes utilisateurs en terme de services (JMS, JMX, JTA, etc.), et sont bien souvent le fruit d'agrégation d'autres solutions Open Source spécialisées tout aussi réputées et reconnues. L'offre est très diverse. On peut citer les principaux : JBoss, Jonas, Geronimo ou encore Red Hat Application Server.

Les différentes solutions de l'offre Open Source

JBoss

JBoss (www.jboss.org) est le serveur d'applications J2EE Open Source bénéficiant de la plus grande notoriété à ce jour. Sa maturité et sa popularité (plusieurs millions de copies téléchargées à ce jour) en font un sérieux concurrent des serveurs d'applications commerciaux tels que IBM Websphere, BEA Weblogic, et Oracle AS, etc.

JBoss (actuellement dans sa version 4) est le fruit du travail de la société éponyme JBoss Inc qui développe des solutions Open Source de qualité, mais sur le même modèle que celui adopté par un éditeur de logiciels commercial :

- Publication d'une roadmap et d'un plan produit,
- Implication de la communauté d'utilisateurs,
- Réalisation d'un guide de développement et diffusion du code source,
- Politique d'assurance qualité et suivi des anomalies,
- Documentation produit,
- Support professionnel, consulting et formation.

JBoss ne développe pas son propre conteneur de Servlets/JSP, cependant il s'intègre facilement avec Tomcat ou Jetty. D'ailleurs, les dis-

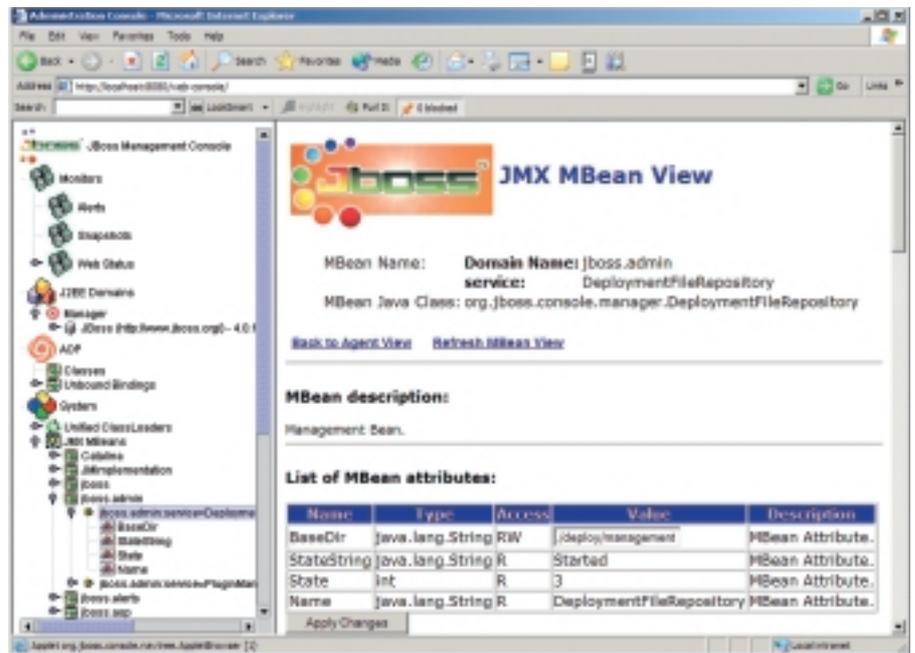


Fig 1 : JBoss Management Console.

tributions de JBoss qui sont proposées intègrent généralement ces serveurs d'applications Web par défaut. D'autre part, JBoss s'est longtemps vu reprocher son manque de documentation. Aujourd'hui, ce vide est comblé et le travail mené sur ce point par le groupe chargé du projet a porté ses fruits. JBoss a été retenu dernièrement par la Direction Générale des Impôts. La composition technique de JBoss repose sur les éléments techniques suivants, qui lui assurent une couverture fonctionnelle globale :

- JBoss Server, le conteneur EJB
- Une infrastructure reposant sur un micro-noyau basé sur JMX (Java Management eXtension),
- JMS pour l'échange de messages,
- JavaMail pour le courrier,
- JTA/JTS (Java Transaction API et Java Transaction Service) pour les échanges transactionnels,

- JCA (Java Connector Architecture) pour la connectivité,
- EJB,
- Hibernate pour la gestion de la persistance,
- JBossCache pour la gestion du cache,
- Gestion du clustering applicatif,
- JBoss AOP pour la programmation par aspects.

JBoss dans sa version 4 est certifié J2EE 1.4. (Fig : 1)

Apache Geronimo



La fondation Apache est largement reconnue pour la qualité de son serveur HTTP, pour ses initiatives dans le monde Java au sein du projet Jakarta, et pour son non moins célèbre serveur d'applications Web (conteneur de Servlets/JSP) Tomcat. Cependant, la fondation

3» TECHNOLOGIES : VERS LA MATURITÉ

Apache est longtemps restée absente sur le terrain des serveurs d'applications J2EE. Ce manque est comblé depuis la publication du projet Apache Geronimo (<http://geronimo.apache.org/>). Actuellement disponible en version 1.0-M3, la version finale devrait être délivrée dans les mois prochains. Du fait de la réputation du projet Apache et de l'ASF, Apache Geronimo viendra inéluctablement se positionner face aux deux autres serveurs d'applications Open Source que sont JBoss et JOnAS, ainsi que tous les autres serveurs d'applications J2EE du marché.

Le projet Geronimo a pour ambition de favoriser la création des solutions basées sur la technologie J2EE de Sun Microsystems.

A la différence de JBoss et JOnAS, Geronimo est annoncé par l'Apache Software Foundation comme certifiable par Sun "rapidement". En effet, l'ASF ayant collaboré avec Sun sur les "Technology Compatibility Kits" (TCK), suites et outils de tests qui rentrent dans le cadre d'une certification J2EE par Sun, cette dernière est en position favorable pour y parvenir.

Le modèle de développement proposé par l'Apache Software Foundation pour Geronimo se veut rapide.

Le serveur d'applications est développé sur la base de nombreux composants Java, sur lesquels le projet Apache a déjà travaillé. Ainsi, le conteneur de servlets/JSP Tomcat sera pleinement exploité et la solution s'appuiera sur JMX (Java Management Extensions). Les fonctionnalités proposées à ce jour pour le projet Geronimo sont les suivantes :

- Support des servlets 2.4 et des JSP 2.0,
- Support des EJB 2.1 et J2EE 1.4,
- J2EE Connectors 1.5,
- Support des J2EE 1.4 Application Clients et Application Archives,
- JDBC 3.0 pour la connexion à la base de données,
- JMS 1.1 pour l'échange de messages,
- JACC 1.0 pour la sécurité,
- JTA 1.0 pour la gestion des transactions,
- Support de SOAP, XML et des services Web au travers des APIs JAXP, JAX-RPC, JAXR et SAAJ-APIs.

On pourra toutefois regretter l'absence de documentation complète à cet état d'avancement du projet.

On notera cependant que Geronimo a passé avec succès les tests Sun TCK pour J2EE 1.4.

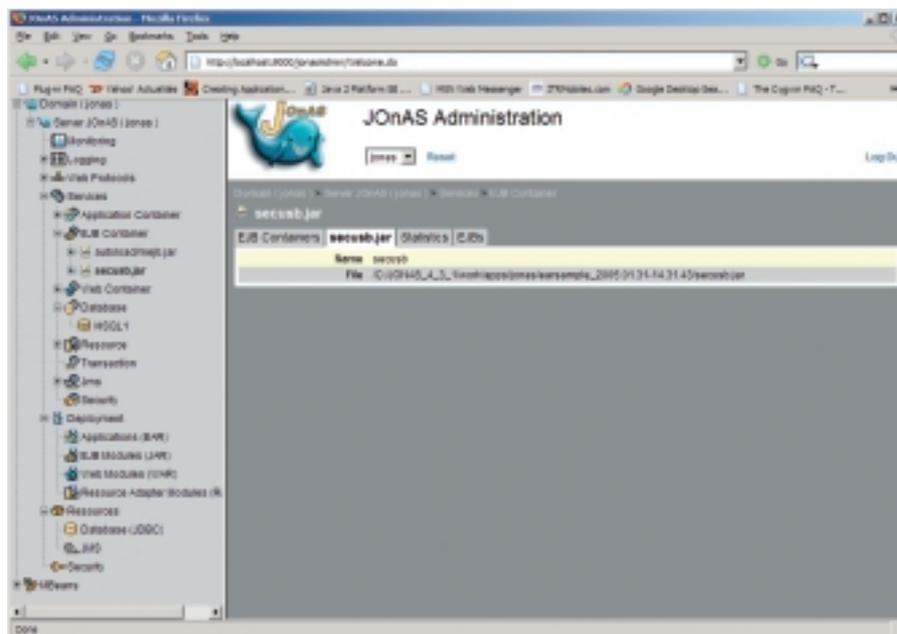


Fig 2 : Interface d'administration de JOnAS.

JOnAS

Le serveur d'applications J2EE JOnAS (<http://jonas.objectweb.org/>), a originellement été développé par la société française Evidian. Aujourd'hui, JOnAS poursuit son développement au sein d'ObjectWeb.

ObjectWeb est un consortium Open Source sans but lucratif, basé sur le même modèle que la " Apache Software Foundation ". Son objectif est de fournir un ensemble de logiciels middleware en Open Source. Il regroupe notamment l'INRIA, Bull et France Télécom R&D. JOnAS implémente l'ensemble des spécifications J2EE 1.4. A l'instar de JBoss, il intègre Tomcat et Jetty en tant que serveurs d'applications Web, ce qui lui permet de rester conforme aux spécifications Servlet 2.4 et JSP 2.0. Parmi les principales fonctionnalités de JOnAS, nous pouvons citer :

- Administration au travers d'une interface basée sur Struts et reposant sur JMX,
- Scalabilité : pool de beans sans états, pool de Message Driven Beans, etc.
- Clustering permettant d'assurer haute disponibilité, répartition de charge et support du fail-over, tant au niveau des EJB, que des applications Web.
- Support des services web via l'intégration d'Axis.
- Support de JDO.
- Support de JMS : trois implémentations sont supportées actuellement (JORAM également

issu du consortium ObjectWeb, SwiftMQ, ou WebSphere MQ). Elles s'exécutent en tant que services au sein de la même JVM ou dans une JVM spécifique.

Depuis début février 2005, JOnAS, dans sa version 4.3.1, a passé avec succès le Sun J2EE 1.4 CTS (Certification Test Suite).

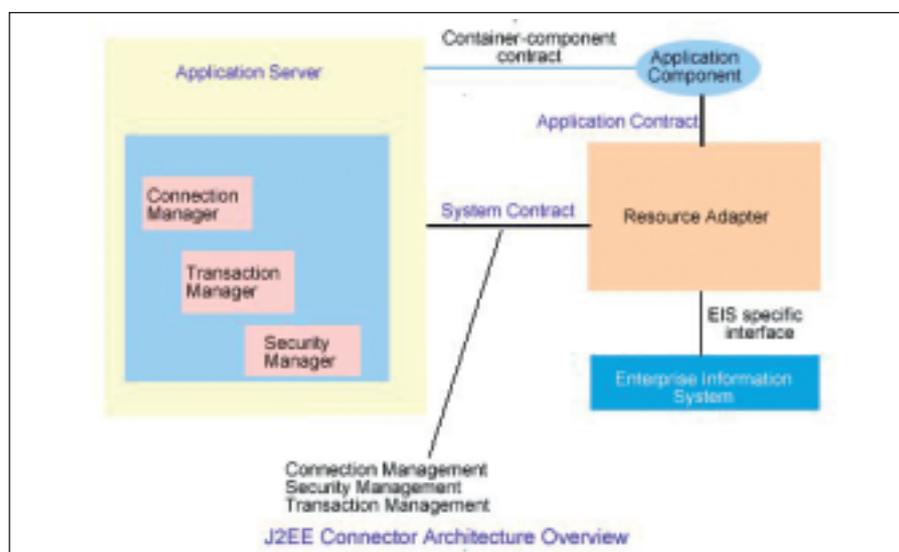
RedHat Application Server

En août dernier, Red Hat a annoncé la disponibilité de RedHat Application Server (<http://www.redhat.com/software/rha/appserver/>). L'objectif de RedHat en proposant cette solution consiste à apporter une réponse à la demande de certains clients face à un besoin de serveurs d'applications J2EE à bas prix et bénéficiant d'un support. RedHat garantit l'interopérabilité de son serveur d'applications avec les autres serveurs d'applications J2EE de BEA, IBM et Oracle. De plus, Redhat Application Server est par ailleurs testé et supporté sur les principales JVM du marché, à savoir : Sun SDK, BEA JRockit et IBM ; il en va de même avec les serveurs de bases de données relationnelles Oracle, DB2 d'IBM ou bien Sybase.

Le serveur d'applications J2EE de RedHat n'est pas issu de développements spécifiques ; il repose sur des briques déjà existantes et inclut les caractéristiques et les fonctionnalités les plus utilisées et les plus courantes des serveurs d'applications commerciaux J2EE, à savoir :

Produit	Version	JDK	EJB	Servlet s./jsp	JMS	JTA	Certification J2EE	Plateformes supportées
JBoss AS	4.0	1.4 et supérieurs	2.1	2.4/2.0	1.1	1.3	1.4	Windows, Linux, Unix, HP-UX, etc. (toutes plateformes supportant les JVM 1.4 et supérieures)
JBoss AS	4.1	1.4 et supérieurs	2.1	2.4/2.0	1.1	1.3	1.4	Windows, Linux, Solaris, HP-UX, Netware
Genosiris	1.0-N3	1.4/1.5	2.1	2.4/2.0		1.5	1.4 (Sun TCK)	Windows, Linux, Mac OS X
RedHat Application Server		1.4 et supérieurs	2.1	2.4/2.0	1.1	1.3	Idem JBoss	Redhat Linux

Fig. 3 : Matrice récapitulative des serveurs d'applications J2EE Open Source



- Serveur d'applications d'entreprise (JonAS du projet ObjectWeb) pour le support des EJB,
- Serveur d'Application Web (Apache Tomcat) pour le support des JSP et Servlets,
- Support des services web avec AXIS du projet Jakarta,
- Gestion et administration via JMX (sur les couches JonAS et Tomcat),
- Messagerie et support pour les transactions (JMS, JTA, JavaMail),
- Clustering et haute disponibilité,
- Support des architectures processeurs: IA-32, Itanium et PPC. (Fig 3)

Des serveurs commerciaux aux serveurs Open Source...

Une appropriation progressive

L'évolution des utilisateurs et clients finaux vis-à-vis des offres de serveurs d'applications J2EE Open Source est progressive. L'intérêt porté à ces dernières solutions s'installe progressivement. Leur utilisation commence souvent au sein de l'entreprise par les développeurs qui l'utilisent pour réaliser le développement des projets sur leur station de travail avant de réaliser l'intégration sur le serveur d'application

cible. Ils sont ensuite généralement utilisés sur des projets pilotes d'importance stratégique moindre (Intranet, etc.) avant que, une fois leur valeur démontrée, leur usage ne soit envisagé, sur des applications plus critiques.

Le support et la formation

La certification des serveurs J2EE Open Source n'est pas neutre et incite les entreprises à utiliser ces outils de façon plus conséquente. Toutefois, cela n'est pas tout. Là où le développeur tolère une documentation réduite à sa plus simple expression, un décideur informatique sera plus attentif à l'ensemble des services associés à l'outil choisi.

Aujourd'hui, la majorité des projets Open Source reconnus disposent d'une documentation complète, enrichie et corrigée de façon plus réactive. La richesse du support n'a plus rien à envier aux solutions commerciales. En effet, il est aujourd'hui possible pour une entreprise de faire former ses équipes techniques auprès de professionnels de la formation, et d'avoir recours à des prestataires de conseil et de services pour l'assister sur ses projets. D'autre part, la communauté Open Source offre un support et une réactivité exceptionnels au travers de mailing-lists ou de forums spécialisés.

Le développement et l'administration

Les autres éléments favorisant l'appropriation des serveurs d'applications J2EE Open Source par les entreprises sont d'ordre technique. Ainsi, l'intégration avec les outils de développements (Eclipse, Borland JBuilder, etc.) se fait sans encombre, ce qui facilite grandement la mise au point du code, son débogage et son packaging afin de le déployer. De même, les serveurs d'applications Open Source sont aujourd'hui capables de s'inscrire dans les contraintes d'exploitation et de production des entreprises. Pendant longtemps, la haute disponibilité, le clustering et le session fail-over furent le domaine réservé des solutions commerciales. Ce retard est désormais comblé et ces fonctionnalités couvertes à des degrés plus ou moins importants, par les serveurs d'applications J2EE Open Source.

Enfin, les outils d'administration de ces serveurs étaient souvent réduits à leur plus simple expression, c'est-à-dire à la manipulation de fichiers properties ou XML, ce qui n'était pas forcément très séduisant en comparaison des solutions commerciales. Les serveurs d'applications Open Source ont enfin corrigé ce défaut, facilitant ainsi leur prise en main par les équipes d'exploitation et de production des entreprises.

Conclusion

L'offre actuelle des serveurs d'applications Open Source est réellement intéressante et représente une véritable alternative aux solutions commerciales équivalentes sur le marché. D'un point de vue technique, on retrouve dans les solutions Open Source, tous les éléments qui constituent le minimum attendu d'un progiciel du marché. Les points sur lesquels les solutions Open Source ont souvent fait l'objet de reproches, qu'il s'agisse de la documentation, de la certification, de la formation, du support et, enfin, de la haute disponibilité applicative, sont désormais à un niveau de qualité similaire à celui de leurs équivalents commerciaux. Dès lors, les gains financiers engendrés par de telles solutions sont réels dans la mesure où il devient possible, pour l'entreprise, d'en faire un usage à grande échelle sur ses environnements de production.

■ Mickael Maindron - mickael.maindron@neoxia.com architecte chez NEOXIA, intégrateur de systèmes spécialisé dans les nouvelles technologies de l'information.



3» TECHNOLOGIES : VERS LA MATURITÉ

La sélection de l'Open Source

IdealX propose son guide des outils Open Source. Nous publions ici des extraits de la dernière édition. Guide complet sur www.idealx-opensource-guide.org

Outils de déplacements

Légende : **R** : Richesse - **E** : Exploitation - **T** : Tendence



Type	Nom	Version	R	E	T	Commentaire
Langages	Erlang	R10B-2	→	Langage fonctionnel industriel
	Perl	5.8.6	→	Puissant et doté de riches bibliothèques et documentations
	PHP	4.3.8	→	Parfaitement adapté au web. La version 5 semble trop jeune
	Python	2.4	→	Efficace mais encore peu répandu
Gestion de projet	Planner	0.12.1	→	Anciennement MrProject. Ergonomique.
	Double Choco Latte	0.9.5	→	Développement ralenti
Éditeur	Emacs	21.2	→	La référence
	NEdit	5.5	→	Ergonomique, puissant, multi-plate-forme
IDE	Eclipse	3.0.1	↑	En Java, un examen s'impose
	JEdit	4.2	→	Simple, ergonomique, extensible
	NetBeans	4.0	→	Menacé par Eclipse, il s'améliore vite
	KDevelop	3.1.2	→	Environnement intégré de développement multi-langage (KDE)
	JBoss IDE	1.4	↑	Java. Plugin pour Eclipse facilitant le développement pour JBoss (EJB, Servlet, WebServices) (Atos Origin)
	Anjuta	1.2.2	↑	Environnement de développement (Gnome)
Gestion de configuration	CVS	1.12.11	→	Éprouvé
	Subversion	1.1.2	↑	Ne souffre pas des limitations de CVS. suivre
MDA	openMDX	1.5.3	↑	Gestion intégré (modèle, développement, déploiement) et grande liberté quant la construction des générateurs (Atos Origin)
	AndroMDA	3.0M2	→	N'offre que les générateurs. Prometteur (Atos Origin)
Gestion d'évolution	Bugzilla	2.16.7	→	Rapport et suivi de bogue via une interface web
	TRAC	0.8	↑	Participe du gestionnaire de bugs de l'outil de gestion de projet. Bonne interface entre le développeur et le chef de projet. Projet dynamique
	GForge	4.0	→	Environnement de développement collaboratif en mode Web la SourceForge
Modélisation	ArgoUML	0.16.1	↑	UML 1.3 seulement. Peu riche mais prometteur. Bon outil d'initiation
Portabilité	autoconf, automake, libtool	NA	→	Rendent les codes (famille C) portables.
	Ant	1.6.2	↑	LE standard du monde Java J2EE
Portabilité GUI	wxWidgets	2.4.2	↑	
Exploration de source	lxr	0.9.3	→	Présentation Web
	GNU GLOBAL	4.8.2	→	
	Source Navigator	5.1.4	→	IDE orienté exploration de code. Développement manifestement arrêté
Documentation	doxygen	1.4	→	
Débogage	Valgrind	2.2.0	↑	Aide l'analyse des erreurs (alloc. mémoire et threads)
	KCachegrind	0.4.5	→	Profilage C et C++
	DDD	3.3.10	→	Gère l'essentiel des débogueurs symboliques et langages. Excellent frontal graphique pour examiner les structures de données
Test	DejaGnu	1.4.3	→	Gestion d'une suite de test
	JUnit	3.8.1	→	Java, test unitaire
	Cactus	1.6.1	↑	Code serveur : test fonctionnel et intégration. Mise en oeuvre délicate
Données	CloverETL	1.1.3	→	Transformation de données structures (ETL)
	SQLite	3.0.8	→	Moteur SQL autonome
Outils	Gengetopt	2.3	→	Gestion des arguments de la ligne de commande
	gettext	0.10.40	↑	Gestion de l'internationalisation et localisation
	nana	2.5	→	Assertions améliorées

Poste de travail

Légende : **R** : Richesse - **E** : Exploitation - **T** : Tendance

Type	Nom	Version	R	E	T	Commentaire
Interface	KDE	3.3.2	↑	Riche, cohérent, solide. Architecture évolue sous une apparence classique
	GPE	0.7	↑	Environnement graphique pour assistant personnel (PDA). Fond sur GTK
	GNOME	2.8	↑	évolution remarquable. Novateur et ergonomique
	OPIE	1.1.7	↑	Environnement graphique pour assistant personnel (PDA). Par Trolltech (QT)
Navigateur	Firefox	1.0	↑	Incontournable. Facilite la migration vers l'Open Source
Suite bureautique	OpenOffice.org	1.1.4	↑	Très riche et puissant, disponible en français
	Koffice	1.3.5	↓	Dépassé par OpenOffice.org mais moins exigeant en ressources
Tableur	Gnumeric	1.4.1rc5	↓	Dépassé par OpenOffice.org mais moins exigeant en ressources
Traitement de texte	AbiWord	2.2	↓	Dépassé par OpenOffice.org mais moins exigeant en ressources
Messagerie	Outport	1.1.25	→	Outil d'exportation de données MS-Outlook
	Mozilla Thunderbird	1.0	→	Riche et plaisant
Classement	POPFile	0.22.2	→	
Courrier et nouvelles	Pan	0.14.2.91	↑	Riche et plaisant. Semble abandonné
Dessin bitmap	The GIMP	2.2.1	→	Satisfera de nombreux professionnels (moins puissant que Photoshop CS ou Elements, ndlr)
	imgSeek	0.8.4	→	Gestionnaire d'images
Dessin vectoriel	Skencil	0.6.16	→	Anciennement Sketch. Manque d'objets (galerie)
	Sodipodi	0.34	→	Ergonomique
Transformation bitmap vers vectoriel	potrace	1.5	→	Efficacité surprenante
Modeleur 3D	Blender	2.36	↑	Professionnel, multi-plate-forme, extensible
PAO	Scribus	1.2.1	↑	évolution intéressante
Editeur HTML	Bluefish	0.13	→	
	Nvu	0.6	→	Prometteur
	Quanta+	3.3	→	Puissant
Fax	hylafax	4.2	→	Riche et adaptable
Emulation MS-Windows	WINE	20041201	→	évolution très soutenue. Une brique essentielle pour la migration. Riche et adaptable
	CrossOver	3.0.1	→	Non open source
Gestion de l'information	Zilverline	1.0.1	→	Moteur d'indexation des documents personnels
	Uniform Server	3.2	↑	Plate-forme Web libre légère et facile déployer
	Rekall	2.2.3	→	Frontal de gestion de données
	mySiteMaker	4.2	→	Publication de données MySQL par CGI en Perl

Exploitation

Légende : **R** : Richesse - **E** : Exploitation - **T** : Tendance

Type	Nom	Version	R	E	T	Commentaire
Prise de contrôle	TightVNC	1.3dev6	→	Prise de contrôle à distance. Très efficace
	NX	0.1	..	.	→	Très prometteur
	rdesktop	1.3.1	↓	Client pour serveurs TSE (protocole RDP). Fonctionnement erratique avec les MS-Windows contemporains (RDP 5)
Gestion d'incidents	Request Tracker	3.2.2	↑	Reconnu et populaire
	ZenTrack	2.5	→	Couverture fonctionnelle réduite mais prometteur (Atos Origin)
	Mantis	0.19.2	→	Discret mais efficace
	OTRS	1.3	→	
Analyse traces Web	AWStats	6.2	↑	Riche. Surtout utile en mode CGI (dynamique)
	Analog	5.32	→	Complet et fiable
	Webalizer	2.01-10	↓	Traite bien plusieurs mois d'activité
	Relax	2.80	→	Analyse des référents (REFERRER)
Administration SGBDR	phpMyAdmin	2.6	↑	Existe également pour PostgreSQL, Oracle 8 et Sybase
	TOra	1.3.14.1a	↑	Boîte à outils de l'administrateur et du développeur Oracle

3» TECHNOLOGIES : VERS LA MATURITÉ

Serveurs d'applications et frameworks

Légende : **R** : Richesse - **E** : Exploitation - **T** : Tendance

Type	Nom	Version	R	E	T	Commentaire
Plateforme et framework	Cocoon	2.1.6	↑	Framework Java de développement d'applications Web par assemblage de chaînes de transformations XML, incluant de nombreux connecteurs
	JBoss AOP	1.0	↑	Java. Simplifie la programmation de fonctionnalités transverses (sécurité, accès distant ...) (Atos Origin)
	Apache AxKit	1.6.2	→	Framework XML (Perl) de développement d'applications Web
	Zope	2.7.3	↑	Python. Environnement riche
.NET	Mono	1.0.5	→	développement rapide appuyé par Novell
	dotGNU Portable .Net	0.6.10	→	
	IKVM	0.8.0.0	..	.	↑	JVM pour .Net : permet d'exécuter des applications Java dans un CLR. Progression impressionnante.
Frameworks Java	OpenSymphony		→	Composants J2EE
	Struts		→	JSP et servlets. La référence, en constante amélioration
	jPOS	1.4.8	→	Framework métier de mise en oeuvre de transactions financières. Respecte la norme ISO-8583. Simple, stable et performant (Atos Origin)
	Hibernate	2.1.7c	↑	Java, mappeur relationnel / objet. La version 3, annoncée, inclura toutes les fonctionnalités EJB3 (Atos Origin)
	Spring	1.1.3	↑	Raccordement aisé des tiers, AOP (gestion des sessions et transactions)...
CORBA	ORBit	2.11	→	ORB conforme CORBA 2.4. Mature
Serveur d'application	JBoss AS	4.0.1	↑	Bonne qualité. Certifié J2EE 1.4, framework AOP, prêt pour les EJB3 (Atos Origin)
	JOnAS	4.2.2	→	Novateur. Profite de la richesse des projets du consortium ObjectWeb
Services Web	Axis	1.4	..	.	→	Conteneur d'applications pour services Web

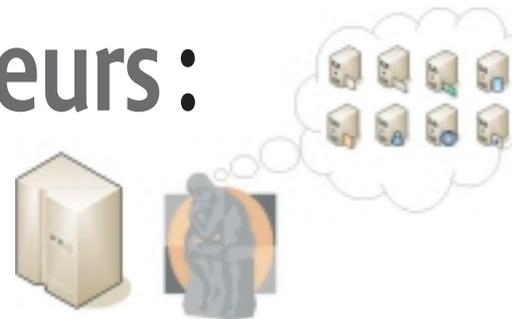
Sécurité

Légende : **R** : Richesse - **E** : Exploitation - **T** : Tendance

Type	Nom	Version	R	E	T	Commentaire
Chiffrement	GnuPG	1.4.0	↑	
	Cryptokit	0.9.3	↑	Confidentialité des documents et messages
Connexion	OpenSSH	3.9	↑	L'outil de référence. Sous MS-Windows : PuTTY
Contrôle	sysrtrace	1.4	↑	Limite les possibilités offertes aux programmes
Interface d'administration/filtrage	Firewall Builder	2.0.5	↑	Interface de configuration multi-plate-forme
Authentification	IDX-PKI	1.9	↑	Complète, cohérente et modulaire
	FreeRADIUS	1.0.1	↑	Stable et complet
Intégrité	AIDE	0.10	→	Simple
NIDS et intégrité	Prelude	0.8.10	→	Hybride (réseau, analyse de traces, vérification d'intégrité) et sondes hétérogènes, donc complet. Distribué. Extensible. Manque de documentation
NIDS passif	snort	2.3.0RC2	↑	Détecteur d'intrusions. Domaine prometteur, en cours d'exploration. Immature
Protection passive	Bastille Linux	2.1.3	→	
	COPS	1.04	→	
	Nessus	2.0.12	↑	Détection de vulnérabilités. L'outil le plus réputé
	SARA	5.1.1a	→	
	grsecurity	2.1.0	→	Patch Linux
VPN	FreeSWan	2.06	↓	Le développement est arrêté. Préférer l'IPSec d'OpenBSD
	IPSec (OpenBSD)	3.6	↑	Interopérable, souple et robuste
Scanner	nmap	3.75	↑	Fiable, fonctionnel et prouvé
Transport	stunnel	4.0.7	→	Attention aux failles d'OpenSSL
Firewall	Packet filter OpenBSD	3.6	↑	Puissant, simple d'administration, fonctionnel et stable
	Netfilter	1.2.11	→	Parfois difficile à mettre en oeuvre
Apache	ModSecurity	1.8.6	→	Module de détection et prévention d'intrusion

Document publié selon les termes de la licence Creative Commons by-nc-sa
Copyright © IDEALX S.A.S 2000-2005 - Tous droits réservés

Du poste client aux serveurs : les enjeux de la virtualisation



L'idée de base sur laquelle repose la virtualisation consiste à découpler les ressources physiques de leur représentation logique. Jusque là, rien de très neuf. Cette démarche est à l'œuvre depuis longtemps dans les infrastructures réseaux avec les VPNs (Virtual Private Network) et les VLAN (Virtual Local Area Network), mais aussi au niveau des applications, avec la mise en oeuvre de machines virtuelles. En effet, les deux principales plates-formes d'exécution récentes (JVM Java et CLR .NET) utilisent un mécanisme de virtualisation des processeurs pour s'affranchir des spécificités des plates-formes matérielles. Ce qui est nouveau, c'est que les configurations matérielles actuelles offrent une puissance telle que l'on peut désormais envisager la virtualisation, non plus au niveau du processeur ou d'une ressource réseau, mais directement au niveau du système d'exploitation. Cela ouvre la voie à l'exécution simultanée de plusieurs instances de systèmes d'exploitation sur une machine donnée.

Comment ça marche ?

La virtualisation repose sur la mise en place d'émulateurs capables de simuler les fonctionnalités de bas-niveau d'une configuration matérielle. Généralement cela se présente sous la forme d'un super BIOS logiciel qui reproduit le comportement des différents composants du système (CPU, RAM, disque dur, lecteur de CD-ROM, carte réseau, ...). En fait, un logiciel de virtualisation efficace doit permettre, soit de simuler une ressource physique (par exemple considérer un fichier comme un disque dur virtuel), soit d'associer une ressource physique à une machine virtuelle (par exemple dédier une carte réseau à une instance particulière). Ce procédé est assez ancien, mais il était auparavant l'apanage de configurations matérielles surpuissantes de type mainframe.

Virtualiser pour quoi faire ?

La virtualisation des postes ouvre de nouvelles perspectives dans l'utilisation des ressources

matérielles. On peut d'ores et déjà identifier trois usages possibles :

- La virtualisation du poste utilisateur qui facilite le déploiement des postes par l'utilisation de " masters " prêts à l'emploi et déployables sous la forme de disques virtuels sur des configurations matérielles hétérogènes. Cette technique est également intéressante pour faire fonctionner de logiciels basés sur des systèmes d'exploitation anciens (MS-DOS, Windows 3.x, Windows 9x, ...) sur des configurations récentes non compatibles.
- La virtualisation du poste de développement offre les mêmes avantages que la virtualisation des postes utilisateurs avec, en plus, la capacité de lancer simultanément plusieurs instances de systèmes d'exploitation (sur les configurations matérielles qui le permettent). Les postes de développement ayant tendance à voir leur stabilité se dégrader plus rapidement que les autres postes (surtout sous Windows), la possibilité de remasteriser la machine en quelques clics est également un atout important. Un autre point fort de ce type de solution concerne les tests de compatibilité des applications avec des systèmes d'exploitation différents, qui peuvent se faire sans avoir à reconfigurer sans cesse les machines de test. Cette dernière fonctionnalité intéressera particulièrement les éditeurs de logiciels.
- Enfin, last but not least, la virtualisation des serveurs permet le lancement simultané de plusieurs OS serveurs sur une même machine physique. Dans ce cas, le système de virtualisation doit être capable de partitionner les ressources (CPU et mémoire) et de les allouer aux différentes instances virtuelles qu'il héberge. Cette technique permet d'une part de découpler totalement les dimensions hardware et software et, d'autre part, de différencier l'utilisation d'une machine virtuelle de la possession de la machine physique, et de faciliter ainsi les démarches des hébergeurs pour offrir des solutions de

qualité. La virtualisation des serveurs est le prolongement logique d'une démarche de consolidation des ressources matérielles.

Les outils disponibles

Le marché des solutions de virtualisation est relativement restreint. Voici un aperçu des solutions disponibles sur le marché :

- IBM est un des plus anciens acteurs dans ce domaine avec sa gamme de systèmes basés sur Z/OS, son système d'exploitation pour mainframe. A noter qu'il s'agit ici uniquement de virtualisation des serveurs.
- VMWare est un des acteurs historiques du marché de la virtualisation. Ces produits adressent à la fois les postes clients (VMWare Workstation) et les serveurs (VMWare ESX/GSX Server).
- Microsoft est un nouvel entrant sur ce marché. Il a racheté en 2003 la solution de Connectix, désormais commercialisée sous le nom Microsoft Virtual PC 2004. Une version serveur baptisée Virtual Server 2005 est également disponible.
- Xen est une solution de virtualisation Open-Source dédiée à la plate-forme Linux.

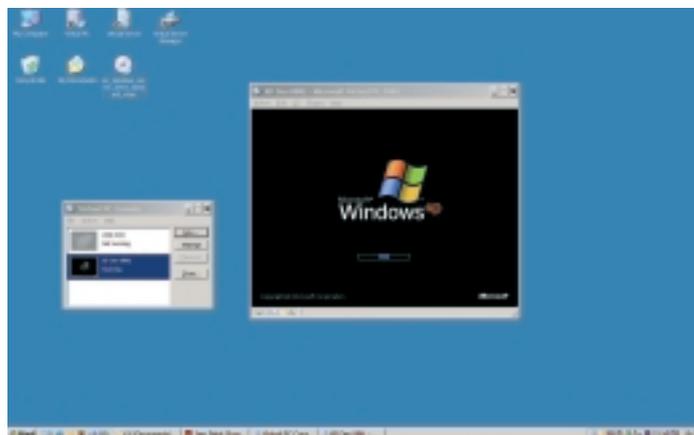
Conclusion

La virtualisation des postes clients et des serveurs est en route. Elle devrait se généraliser aux plates-formes de production dans les prochaines années. En attendant, elle est devenue incontournable pour les équipes de développement confrontées à des problématiques de tests de compatibilité sur des systèmes hétérogènes. Elle est aussi un atout appréciable pour remasteriser des postes, notamment dans le cadre de tests de non-régression. Attention toutefois aux performances des entrées-sorties avec les disques virtuels. Elles sont très loin d'égaliser celles des disques physiques !

■ **Médéric Morel** - mederic.morel@neoxia.com
Directeur technique de NEOXIA, intégrateur de systèmes spécialisé dans les nouvelles technologies de l'information.

Virtual PC 2004 et Virtual Server 2005 : La virtualisation selon Microsoft

VirtualPC est fortement axé poste client et n'est en aucun cas prévu pour faire de la consolidation de serveurs. Pour ce type de scénario, il est préférable de se diriger vers des solutions telles que Microsoft Virtual Server ou VMWare ESX/GSX. Par ailleurs, VPC2004 n'est pas administrable comme peuvent l'être ces produits, ce qui implique que la gestion des machines virtuelles est déportée directement vers l'utilisateur. Avec une population d'utilisateurs desktop (développeur, webmestre...), quels sont les scénarios envisageables pour ce type de produit ?



Considérons tout d'abord la population des programmeurs, fortement consommatrice de machines de développement. Les développeurs ont souvent besoin d'avoir à leur disposition un environnement de travail propre dans lequel ils effectuent des tests sur les applications. Avoir en permanence à sa disposition une machine virtuelle qu'il est possible de " stresser fortement " durant une phase de recette, présente un avantage considérable : elle supprime le risque d'avoir à remasteriser la machine après un déploiement infructueux. De plus, avec suffisamment de mémoire et un bon processeur, il est possible de faire fonctionner sur une même machine plusieurs environnements virtualisés, ce qui facilite ainsi la mise au point d'applications distribuées. Simuler des indisponibilités ou changer des configurations de déploiement se fait très facilement, sans avoir à mettre à sa disposition une plate-forme complète de tests. Du point de vue de l'administration des postes, la virtualisation des environnements soulage le département IT en terme, d'une part d'investissements, de temps de maintenance et de déploiement et d'autre part, de réduction des coûts.

Au-delà des développeurs, les équipes qui ont besoin d'un OS spécifique sont également concernées. Sur une même machine, il est possible

d'avoir un Windows et un Unix installés, ce qui permet d'utiliser une base d'applications beaucoup plus importante. On peut notamment utiliser une configuration de ce type pour les tests d'interopérabilités entre systèmes différents.

La faiblesse principale de Virtual PC 2004 est qu'il est impossible d'interagir de manière programmatique. En effet, la seule option disponible consiste à utiliser une ligne de commande afin de démarrer une machine virtuelle donnée... Cela nous conduit rapidement à penser qu'il ne répond pas aux besoins spécifiques tels que, par exemple, l'automatisation des tests de non régression.

Virtual Server 2005

Virtual Server 2005 est le pendant du côté serveur de Virtual PC 2004. Les avantages apportés par une telle solution sont multiples :

- Réduction du nombre de machines
- Meilleure utilisation de la capacité des machines
- Mise à disposition d'un environnement de tests

La notion de virtualisation n'est pas récente et de nombreuses implémentations existent sur des matériels propriétaires (comme par exemple les machines IBM). Mais, dans l'environnement PC, il existe assez peu de solutions. On citera notamment VMWare ou BOCHS (pour

processeurs x86) qui sont disponibles depuis quelques années.

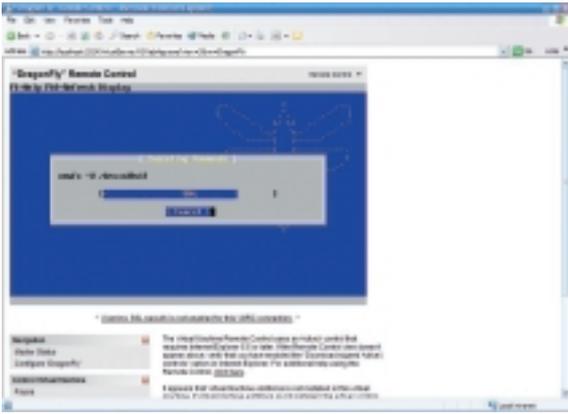
Virtual Server 2005 apporte son lot d'améliorations par rapport à Virtual PC 2004 :

- Processus de migration d'instances physiques en instances virtuelles.
- Gestion centralisée des instances virtuelles.
- Administration à distance au moyen d'un protocole dédié (VMRC)
- Fonctionnement sous un service NT disponible dès le démarrage de la machine hôte.
- Gestion des ressources processeur,
- Nouveau support de disques virtuels,
- Programmation au moyen d'une interface COM.

Virtual Server 2005 est disponible en deux versions : la Standard supporte jusqu'à quatre processeurs, tandis que la version Enterprise en supporte jusqu'à trente deux. Il faut toutefois être réaliste quant aux capacités du produit : une machine virtuelle ne pourra bénéficier que d'un seul processeur virtuel.

Processus de migration

VS2005 supporte quatre processus de migration pour la consolidation de serveurs. Le processus de migration le plus intéressant pour une entreprise est celui offert par le toolkit de migration, téléchargeable sur le site de Microsoft. Il permet en effet de migrer auto-



matiquement et très facilement un serveur physique vers une machine virtuelle. Les autres moyens d'installer une machine virtuelle sont l'installation "from scratch" (ISO ou CD) ou bien l'utilisation d'un serveur préalablement préparé au moyen de l'utilitaire SysPrep. La dernière façon de construire une machine virtuelle est de partir d'un disque virtuel déjà préparé et d'y appliquer des différences. Le disque original n'est donc jamais modifié et permet de remasteriser une nouvelle machine virtuelle très rapidement. Le support de disques différentiels constitue également un apport majeur de cette version.

Gestion centralisée

Une fois la machine virtuelle installée, on peut administrer les droits d'accès, attribuer la consommation CPU, partager la mémoire ou encore consulter les logs. Tout cela se fait au moyen d'une interface web assez complète. Il faut noter que l'interface n'est pas franchement conviviale, un outil de configuration fenêtré comme peut le proposer Virtual PC 2004 aurait certainement été bien plus pratique.

L'un des points le plus intéressant de VS2005 est qu'il est maintenant possible de configurer finement la consommation CPU d'une instance virtuelle (minimum et maximum), ce qui permet de gérer les performances dans un scénario de consolidation sur une même machine.

VMRC

Puisque l'instance virtuelle fonctionne sur un serveur à part, interagir avec elle devient beaucoup plus difficile. Avec un système Windows installé, il est toujours possible d'utiliser

Remote Desktop Protocol (RDP), mais avec d'autres OS, cela n'est pas toujours le cas. Reste des solutions de type VNC qui demandent, quoiqu'il en soit, l'installation d'un client sur la machine virtuelle. Microsoft a donc intégré un protocole, Virtual Machine Remote Control (VMRC), sorte de VNC amélioré et sécurisé qui permettra de se connecter à distance. Il faut noter que ce protocole ne rem-

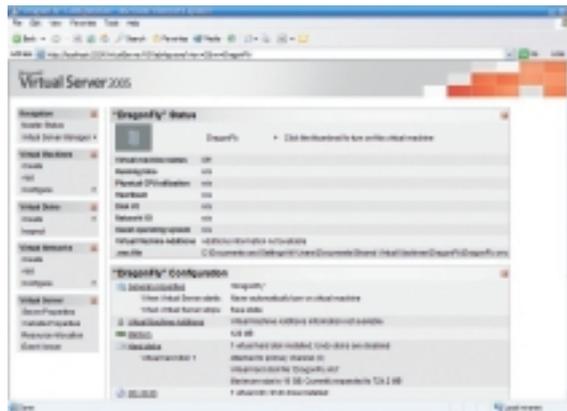
place pas RDP dans le cas où des sessions distinctes sont nécessaires.

Service

Virtual Server 2005 est un produit destiné principalement à des fins de consolidation. Les machines virtuelles hébergées doivent donc être disponibles dès le démarrage de la machine hôte. Il fonctionne dorénavant sous la forme d'un service (i.e. : plus besoin de se loguer pour lancer une application), ce qui rend les machines virtuelles disponibles très rapidement après le redémarrage de la machine hôte.

VS2005 API

Les développeurs ne seront pas déçus : Microsoft a intégré une API COM pour exploiter les moindres recoins de VS2005 de manière programmatique ! Voici un exemple pour



attacher une image ISO à un lecteur DVD et mettre en marche la machine virtuelle :

```

1 connexion au serveur
dim vmVirtualServer
set vmVirtualServer = CreateObject("Virtual
Server.Application")

```

```

1 recherche d'une machine virtuelle

```

```

dim vmVirtualMachine
set vmVirtualMachine = vmVirtualServer.Find
VirtualMachine("DragonFly")

if vmVirtualMachine.State = 5 then ' vmVM
State_Running
    vmVirtualMachine.TurnOff
end if

' mise en place d'une image ISO
dim vmDrive
set vmDrive = vmVirtualMachine.DVDROM
Drives(1)

vmDrive.ReleasImage
vmDrive.AttachImage("d:\Articles\image.iso")

' démarrage de la machine virtuelle
vmVirtualMachine.StartUp

```

Conclusion

Virtual Server 2005 est l'évolution logique de Virtual PC 2004 qui permet de faciliter la consolidation de serveurs. Ce type d'outil permet aux équipes de développement de reconstruire un environnement de test à peu de frais et très rapidement. Il ne faut toutefois pas sous-estimer la capacité machine physique : c'est en effet celle-ci qui assurera la fiabilité de l'hébergement des machines virtuelles. Il est donc essentiel de ne pas négliger CPU, mémoire, alimentation redondante, mise à jour hardware à chaud... Pour ce qui est des performances de Virtual Server 2005, on pourra toutefois regretter l'impossibilité d'assigner plusieurs processeurs virtuels à une machine, ce qui est d'autant plus dommage que la version Enterprise supporte jusqu'à trente deux processeurs. En conclusion, en dépit de quelques limitations Virtual PC 2004 et Virtual Server 2005 sont de bons produits, faciles à administrer, peu onéreux et qui peuvent rendre de nombreux services aux équipes de développement.

Ressources

- Virtual Server 2005 Migration Toolkit : <http://www.microsoft.com/windowsserver/system/virtualserver/evaluation/vsmt.msp>
- Automated Deployment Services : <http://www.microsoft.com/windowsserver2003/technologies/management/ads/default.msp>

■ Pierre Chalamet - pierre.chalamet@neoxia.com
Architecte chez NEOXIA, intégrateur de systèmes spécialisé dans les nouvelles technologies de l'information.



Coder plus vite donne-t-il un meilleur résultat ?

RAD et coding from scratch...

Le Rapid Application Development permet de travailler de manière efficace et rapide pour créer des applications graphiques complexes. Delphi/Kylix, .NetStudio, JBuilder, Qt-designer, Glade... ne sont qu'un petit échantillon de ce qui existe. Ce qui importe avec ces outils, est la vitesse de développement, mais qu'en est-il du résultat généré ? Pouvons-nous nous passer de nos claviers ? Quel prix devons-nous payer en terme de performances ?

Notre exemple

J'ai choisi de vérifier cela en générant une même application, tantôt avec un outil IHM, tantôt à la main. Je travaillerai avec GTK# via l'implémentation libre de .Net : go-mono (<http://www.monoproject.com/about/index.html>).

Dans le précédent numéro de Programmez, la version Windows vous a été distribuée. Mes outils RAD seront : Monodevelop (<http://monodevelop.org/>) pour l'environnement et Glade (<http://glade.gnome.org>) en éditeur IHM. Outils utilisables sous Linux et Windows. Sachez que pour ceux qui ont installé mono sur Windows, Glade est inclus dans la distribution. En ce qui concerne les linuxiens sous Mandrake, il suffit de taper la commande " urpmi monodevelop " pour tout installer sans difficulté, à condition d'avoir installé les médias internet (rendez-vous sur le site <http://www.urpmi.org>). Sous Debian et Fedora, passez par le dépôt " dag " pour apt-get ou Yum: <http://dag.wieers.com/home-made/apt/FAQ.php#B>

Le principe de fonctionnement

Glade va générer un fichier au format XML qui définira la forme de l'interface (placement des éléments, taille, signaux...) Nous instancierons celle-ci, via libglade qui générera l'affichage et l'interaction signal->callback en mémoire. C'est une pratique de plus en plus courante de générer un " code template " instancié durant l'exécution. Cela permet de ne pas générer trop code, et de faciliter les modifications de l'interface sans avoir à recompiler l'application. Certains autres systèmes ne fonctionnent pas de cette façon et génèrent du code en dur. C'est le cas de JBuilder, par exemple, qui modifie directement le source.

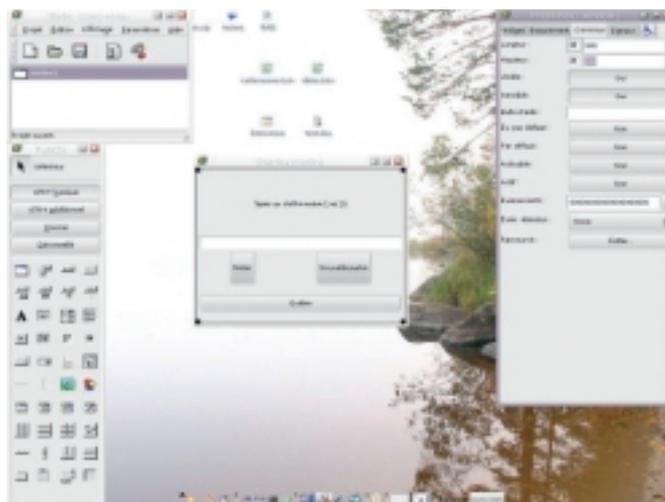
Voyons donc notre exemple en GTK# (GTK+ avec C#). Nous allons faire un jeu de recherche de chiffre.

L'application

Ouvrons monodevelop et créons un nouveau projet Glade# nommé "seekNumber".

Nous avons un fichier Main.cs qui est créé, et un fichier gui.glade qui est enregistré. Ouvrons ce fichier sous Gladez et modifions l'interface : 1 bouton pour vérifier notre chiffre, 1 bouton pour quitter, un bouton pour une nouvelle partie.

Il faudra un label pour afficher les messages et une entrée de texte de type " bouton de réglage " pour taper notre chiffre.



Les premiers moments sous Glade sont difficiles pour la plupart des novices en IHM de type " boîte ". Il faut se mettre en tête que vous ne placerez pas vos éléments " à la volée " mais dans des conteneurs. La solution ici étant de mettre une boîte verticale (vbox) de 4 lignes, et dans la troisième ligne une boîte horizontale de 2 colonnes (hbox). La première ligne contient le label, la seconde l'entrée de texte, la troisième ligne scindée en deux contient les deux boutons, et enfin la troisième un bouton " quitter ".

Nommons nos objets : validButton, newgameButton et quitButton. Le bouton de réglage de chiffre (spinbutton) se nommera numberEntry et le label se nommera infoLabel.

Si nous revenons dans monodevelop et que nous exécutons le projet, nous voyons notre application se lancer sans problème, certes elle ne fait rien de spécial, mais elle s'exécute...

Ajoutons les événements à chaque bouton, laissez les noms par défaut. Pour l'entrée de nombre, il faudra activer l'évènement " activate ", les boutons nécessitent le signal " clicked " (voir dans l'onglet " signaux "). Il s'agit maintenant de coder l'application. Tout d'abord en faisant une fonction qui enregistre un nombre entre 1 et 10. Ensuite, il faudra tester le nombre donné par le joueur, et dire si il est plus grand, moins grand ou s'il a gagné.

La première méthode sera appelée " generateRand " et utilise

System.Random.Next

Ici, nous réalisons que sans développeur, il est presque impossible de créer l'application. Il faut connaître l'API .Net, et savoir coller les signaux à leurs callbacks.

Il faudra aussi déclarer avant tout les éléments graphiques, via des meta-attributes :

```
[Widget] SpinButton numberEntry;
[Widget] Button validButton;
```

...

Le compilateur comprendra qu'il faut récupérer automatiquement les éléments nommés depuis le fichier glade vers des widgets GTK.

Je vous laisse apprécier le code du listing 1 pour voir les différentes méthodes de callbacks. Le prototype des callbacks est de la forme :

public void nom_de_callback(object o, EventArgs args)

Voici les méthodes que j'ai ajoutées :

```
public void generateRand(){
    this.rand = new Random();
    this.randnumber = rand.Next(0,11);
    this.testcount=0;
}

public void test(){
    if(numberEntry.ValueAsInt>randnumber){
        infoLabel.Text="Votre nombre est trop grand";
        testcount++;
    }
    else if(numberEntry.ValueAsInt<randnumber){
        infoLabel.Text="Votre nombre est trop petit";
        testcount++;
    }
    else{
        infoLabel.Text="Vous avez trouvé en " + testcount + " coups";
        this.generateRand();
    }
}

public void on_validButton_clicked(object o, EventArgs args){
    test();
}

public void on_numberEntry_activate(object o, EventArgs args){
    test();
}

public void on_quitButton_clicked(object o, EventArgs args){
    Application.Quit();
}

public void on_newgameButton_clicked(object o, EventArgs args){
    generateRand();
}
```

Et n'oublions pas, dans le constructeur, l'appel à la génération du nombre aléatoire :

```
public GladeApp (string[] args) {
```

```
Application.Init();
generateRand();
...
```

A la main...

Maintenant, fermons le projet et créons-en un nouveau, nommé seek-number2, de type GTK# (et non plus Glade#). Monodevelop nous prépare le travail et génère les fichiers Main.cs et MyWindow.cs qui contient la classe de notre fenêtre principale.

Copions les méthodes de notre ancien projet (pour ne pas les recoder) et voyons le listing 2 qui nous montre comment je m'y prends pour coder sans l'aide d'un éditeur IHM.

```
public void initWidow(){
    this.validButton = new Button("Tester");
    this.quitButton = new Button ("Quitter");
    this.newgameButton= new Button("Nouvelle partie");
    this.vbox= new VBox();
    this.hbox=new HBox();
    this.infoLabel = new Label("Entrez un nombre entre 1 et 10");
    this.numberEntry = new SpinButton(1,10,1);
    //placement
    this.vbox.PackStart(infoLabel);
    this.vbox.PackStart(numberEntry);
    [...]
```

Tous les évènements sont liés de cette manière :
this.validButton.Clicked += on_validButton_clicked;

Bilan

Deux librairies liées contre quatre pour l'application générée depuis l'éditeur IHM et pratiquement 50% d'espace d'exécutable en moins!

Certes, il m'a fallu plus de temps pour coder l'application numéro 2, mais le résultat est un nombre de dépendances amoindri, un code plus léger à la compilation et des ressources systèmes moins sollicitées. Pour des applications de type GUI cela n'est pas " si grave " mais dès lors que nous codons pour le Web, si le site est très sollicité, les temps de réponse vont augmenter considérablement et votre site sera comparé à un somnifère technologique surpuissant.

Le rôle du développeur est essentiel. Il faudra faire une analyse et des tests afin de trouver le meilleur compromis entre le temps de coder et le résultat optimal. On oublie très souvent ce genre de problématique alors que toute une génération d'outils RAD vient emplir nos barrettes de RAM. Pensons-y !

RAD : 50% d'exécutable en plus

Executable	taille	librairies liées
RAD	12ko	2 : mono(mscorlib) = 1.0.5000.0 , mono(gtk-sharp) = 1.0.0.0
Manuellement	7,5ko	4: mono(mscorlib) = 1.0.5000.0,mono(gtk-sharp) = 1.0.0.0,mono(glade-sharp) = 1.0.0.0,mono(glib-sharp) = 1.0.0.0

Deux librairies liées contre quatre pour l'application générée depuis l'éditeur IHM et pratiquement 50% d'espace d'exécutable en moins. En mémoire, c'est tout aussi optimal.

■ Patrice Ferlet - Aston

SSII de 270 personnes, implantée à Paris, Rhône-Alpes et Sud Ouest. www.aston.fr



WEB CONNECTOR : Publier un formulaire électronique en Open Source avec .Net

Si ces dernières années les sites des collectivités locales se sont multipliés, surtout dans les grandes agglomérations, la plupart d'entre eux restent basiques, ne proposant pas de services à destination de leurs administrés ou de procédures à distance.

Administration électronique

Un outil permettant de publier sur Internet des formulaires administratifs au format électronique a dans ce contexte un intérêt évident.

Qui plus est, un tel outil s'inscrit pleinement dans la logique des projets ADELE (ADMinistration ELEctronique 2004-2007) lancés par le gouvernement français (<http://www.adae.gouv.fr/adele/>), et plus particulièrement le projet de dématérialisation des demandes d'état civil (http://www.adae.gouv.fr/article.php3?id_article=298).

Pourtant, le déploiement de ce type d'outil est freiné d'une part, par le tarif prohibitif des solutions proposées pour de " petites " collectivités locales et d'autre part parce que les outils ne sont pas adaptés à un usage par du personnel non informaticien.

Afin de dépasser ces contraintes, l'Association pour le Développement des e-Procédures (ADeP) a sollicité Aston pour réaliser, sous licence Open Source, un outil baptisé WebConnector, permettant de concevoir et de publier un formulaire électronique sophistiqué sans avoir recours à du développement.

Quels sont les services fournis par WebConnector ?

WebConnector permet donc de mettre en ligne des formulaires électroniques sur l'Internet ou sur un intranet et peut également être utilisé comme une application autonome pour gérer des formulaires électroniques sur un serveur Web. En effet, WebConnector transforme les formulaires au format Microsoft Office InfoPath, pour les rendre disponibles sous forme de formulaires web dynamiques. Les formulaires ainsi transformés peuvent ensuite être remplis avec n'importe quel navigateur Web, sans aucun plug-in, et le résultat est stocké dans un document XML. Quand l'utilisateur valide le formulaire, le service WebConnector envoie, sous forme de document XML, les données saisies par mail à une adresse de réception paramétrable.

Il devient ainsi extrêmement simple de mettre à la disposition des administrés, via Internet, des formulaires administratifs tels qu'une demande d'inscription en crèche, une demande d'inscription à la bibliothèque, ...

Moteur de conversion de formulaire et site web de publication

WebConnector permet aux utilisateurs de la suite bureautique Office de transformer sans développement des formulaires InfoPath en pages web ASP.NET. Il permet de collecter rapidement et simplement des informations qui peuvent ensuite être traitées grâce à InfoPath, ou aller alimenter un processus métier plus complexe (en s'interfaçant par exemple avec Microsoft BizTalk Server).

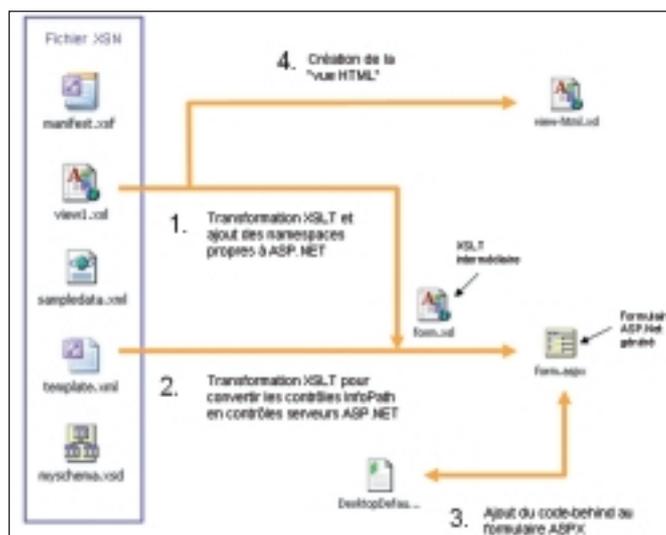


Figure 1 : les grandes étapes de la transformation

Pour chaque formulaire électronique, il est possible d'indiquer :

- le nom du formulaire
- deux catégories de classification
- l'adresse email à laquelle les données sont transmises
- un en-tête et un pied de page qui sont ajoutés au formulaire

Les formulaires publiés sont présentés par défaut dans une page web qui les regroupe par catégorie. Cette page web peut être personnalisée et intégrée dans un site déjà existant.

Une console web d'administration permet de publier un formulaire InfoPath, de configurer (nom, catégories...) les propriétés d'un formulaire ou de supprimer un formulaire.

Comment ça marche ?

WebConnector ne remplace pas Microsoft InfoPath, il le complète en permettant aux utilisateurs de créer facilement des formulaires électroniques qui peuvent être utilisés par tous avec un simple navigateur web. Le moteur de transformation de WebConnector est basé sur des feuilles de styles XSLT. Le principe de la transformation consiste à réaliser un mapping entre les contrôles utilisateurs d'InfoPath et les contrôles serveurs ASP.NET.

Pour comprendre de façon macroscopique comment fonctionne le processus de transformation, distinguons ses grandes étapes :

1. Lorsque vous créez un formulaire, InfoPath crée de son côté un fichier

.xsn qui équivaut à un fichier .cab doté de fichiers XML standard, tels qu'un fichier XSD (définissant la structure du document) et des fichiers XSLT (définissant la mise en forme du formulaire). WebConnector commence donc par transformer l'un des fichiers XSLT définis dans le fichier .xsn en une vue utilisant des contrôles ASP.NET

2. Cette " vue ASP.NET " est ensuite appliquée au template du formulaire InfoPath afin de générer le formulaire ASPX final
3. Un code-behind " générique " (c'est-à-dire couvrant les fonctionnalités communes à la majorité des formulaires) est associé au formulaire ASPX généré, afin de gérer les événements intervenants sur les contrôles ASP.NET.
4. Enfin, pour permettre une visualisation sans utilisation d'InfoPath des formulaires saisis, on génère une feuille XSLT permettant de présenter au format HTML les données saisies par un utilisateur.

Quelles sont les fonctionnalités d'InfoPath couvertes par WebConnector ?

WebConnector ne prend pas en charge l'ensemble des fonctionnalités d'InfoPath. La mise en page et l'aspect visuel du formulaire InfoPath sont conservés ; les contrôles graphiques les plus courants sont supportés : zone de texte, calendrier, case à cocher, case d'option, liste déroulante, section.

Les sections ou tableaux extensibles ou optionnels sont transformés en sections ou tableaux statiques.

.NET et l'Open Source font bon ménage !

Certaines idées reçues laissent croire que les technologies Microsoft ne sont pas adaptées au mode de diffusion Open Source, c'est oublier un peu vite les initiatives de Microsoft, telles que les Applications Blocks ou les Starter Kits, dont les codes sources sont diffusés gratuitement et librement. En réalité, il existe un nombre croissant de projets Open Source utilisant les technologies Microsoft.

WebConnector illustre justement cet état de fait. Ce logiciel, entièrement basé sur des technologies Microsoft, a été développé par Aston à l'initiative de l'ADeP, Association pour le Développement des e-Procédures, sous licence CeCILL. La licence CeCILL, soumise au droit français et respectant les principes de diffusion des logiciels libres, est soutenue par le ministre de la fonction publique (pour plus d'informations sur la licence CeCILL, consultez le site <http://www.cecill.info/>).

Le logiciel WebConnector est téléchargeable gratuitement sur :

- <http://www.adep-france.fr/webconnector/index.html>
- <http://www.aston.fr/metiers/offres/e-business/WebConnector.asp>

Pré-requis

Pour utiliser WebConnector sur un serveur web :

- Windows 2000, Windows XP, ou Windows Server 2003.
- Microsoft .NET Framework v1.1 (disponible gratuitement en téléchargement).

Pour créer les formulaires :

- Microsoft InfoPath 2003 SP1

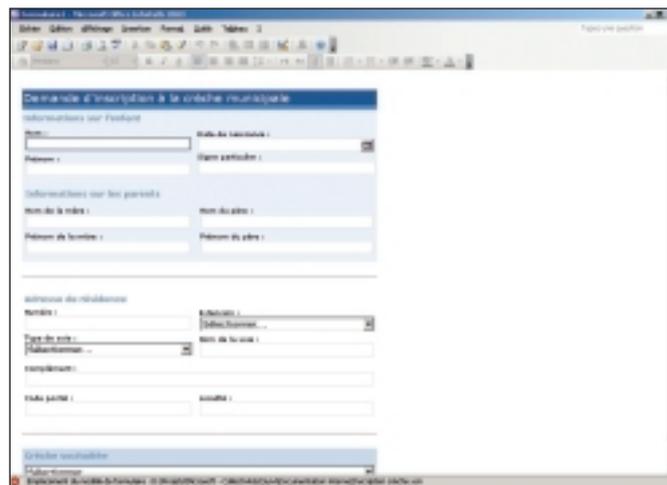
Pour utiliser les formulaires :

- Navigateur web (Firefox, Safari 1.2, Internet Explorer 6, ...)

■ Olivier Lepeltier – Consultant ASTON

Pour toute information complémentaire : webconnector@aston.fr

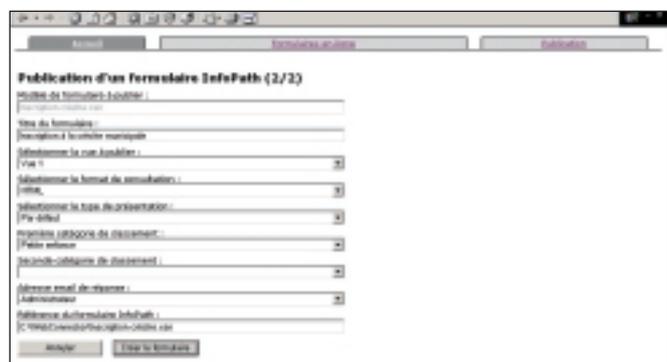
Un exemple de création et de publication d'un formulaire



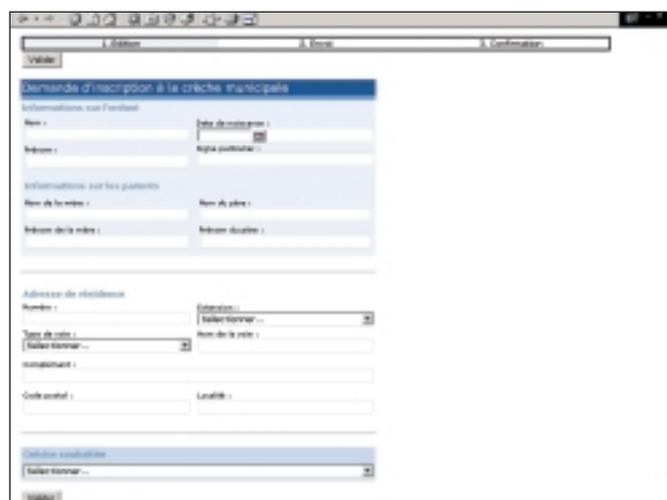
Création du formulaire dans InfoPath



Téléchargement du fichier InfoPath avec WebConnector



Saisie des paramètres de publication sur Internet du formulaire



Le formulaire (en cours de saisie) dans sa version Internet

WinDev : les ardoises magiques

Dans un langage de cinquième génération, la frontière de l'intelligence artificielle est atteinte. Pour en utiliser le potentiel, le principe des ardoises magiques pourrait bien constituer une solution simple.

La fonction Compile de WinDev transforme des lignes de texte en fonctions exécutables. Dès lors, des blocs de texte, sauvegardés en tant que tels, sont intégrables à la demande aux programmes qui les appellent. Ils deviennent des bibliothèques, au même titre que des modules. L'avantage cependant est clair. Si on les charge dans une zone de saisie, il est rapide de les modifier, de les adapter, de les tester, puis de les exécuter et d'en sauvegarder la version modifiée. On atteint ainsi une souplesse d'utilisation sans égale dans les autres langages, puisqu'on peut générer du code à partir de l'exécutable lui-même. Cerise sur le gâteau, les clients exigeants n'auront plus besoin de réclamer une version spécifique d'un logiciel standard. Leurs moutons à cinq pattes seront gérés selon le même principe, à l'aide de code stocké dans des fichiers texte.

Les ardoises magiques

Une ardoise magique se compose de :

- une zone de saisie du code à tester, baptisée ici Code_à_tester
- une zone de réception du résultat, baptisée ici Description_de_l_erreur parce que, outre les résultats, elle est supposée afficher les erreurs en phase de mise au point.
- un ou plusieurs boutons contenant du code répétitif à paramétrer.
- L'ardoise magique peut bien sûr à partir d'elle-même charger un fichier texte contenant du code, à l'aide de l'instruction Code_tester = fChargeTexte("C:\MonFichier.txt")

Dans notre cas, les applications les plus courantes d'une ardoise magique sont :

- Recherche et résolution de bugs dans les codes délicats, sans aller-retour développement-tests. Si l'endroit problématique exige cinq minutes de manipulations avant d'être atteint, il reste ici accessible de façon instantanée.
- Analyse fine et rapide de dysfonctionnements éventuels du langage ou de doutes quant à l'application d'une syntaxe.
- Listage des champs de saisie d'une application. Plus de documentation à consulter.
- Listage des fenêtres d'une application.
- Test en direct des valeurs renvoyées par une fonction.
- Passage de paramètres.
- Mise à jour manuelle d'une base de données maître, à partir d'une base esclave, avec vérification fiche à fiche.
- Création et modification rapide d'un fichier paramètres ne comportant qu'une seule fiche.
- Mise à jour d'un fichier lié B après ajout d'une rubrique dans le fichier A.

Dans la mesure où la fonction Compile affiche la plupart des erreurs de syntaxe, la mise au point s'en trouve guidée sans aucun inconvénient lié à la méthode. On pourrait, grâce à ces principes, bâtir un système expert dont les articulations pourraient être affinées par l'utilisateur, en combinant les fichiers de paramètres et les fichiers de code également au format texte. Si l'on ajoute que WinDev peut charger également des

fichiers texte en dehors de la fonction Compile, et les considérer comme partie intégrante du programme d'origine, il suffit dès lors de gérer les articulations entre ces blocs. Nous sommes bien, au niveau conceptuel, à la frontière des principes de l'intelligence artificielle, à savoir la récursivité et l'autoconstruction des phrases, la souplesse en plus.

Lister les champs de l'application

Le code suivant dans une ardoise magique énumère les champs d'une fenêtre ouverte :

```
i = 1
tantque pas EnumèreChamp(fiche_lots, i) = ""
Description_de_l_erreur += EnumèreChamp(fiche_lots, i) + cr
i = i + 1
fin
```

Pour sophistiquer la présentation à raison de trois entrées par ligne, on sélectionne une police non proportionnelle et on écrit :

```
i est un entier = 1
j est un entier
babouin est une chaîne
tantque pas EnumèreChamp(fiche_lots, i) = ""
babouin = EnumèreChamp(fiche_lots, i)
babouin = complète(babouin,30)
Description_de_l_erreur += babouin
j += 1
i += 1
si j = 3 alors
    Description_de_l_erreur += cr
    j = 0
fin
fin
```

En cours de test, on dispose ainsi d'un aide-mémoire, et d'un outil complet, puisqu'on ne subit pas de limitation dans la taille du code à tester.

Lister les fenêtres de l'application

Pour réaliser une attribution de variable d'une fenêtre à une autre, comment connaître le nom exact de chaque fenêtre ?

```
// énumération des fenêtres d'un projet
sNomFenêtre est une chaîne = EnumèreElément(enumFenêtre)
// on parcourt toutes les fenêtres
TANTQUE sNomFenêtre <> ""
    // on passe à la fenêtre suivante
```

```
sNomFenêtre = EnumèreElément()
Description_de_l_erreur += sNomFenêtre + cr
Fin
```

Le passage de valeur d'une fenêtre à l'autre peut s'effectuer ainsi :

Stocker un code dans un bouton de commande

L'ardoise magique utilise, si nécessaire, un code répétitif affecté à un bouton de commande. C'est dans le bouton Réplica que se trouve le code suivant, destiné à la mise à jour manuelle d'une base de données maître par rapport à une base esclave (base collaborateurs ou simple sauvegarde) :

```
Code_à_tester += "Mon_chemin2 est une chaîne = " + Caract(34) + "\\
ZEUS\Efco_Dossiers_Clients\Fichiers" + Caract(34) + CR
Code_à_tester += "Mon_chemin1 est une chaîne = " + Caract(34) + "c:
\Fichiers_10082004" + Caract(34) + CR
Code_à_tester += "HChangeRep(" + Caract(34) + "*" + Caract(34) + ",
Mon_chemin1)" + CR
Code_à_tester += "HLitPremier(Mon_fichier, Ma_clé_composée)" + CR
Code_à_tester += "informations" + CR
Code_à_tester += "Entrée_sur_l_écran" + CR
Code_à_tester += "Menu.Libellé = " + Caract(34) + "Répertoire de travail
en cours : " + Caract(34) + " + Mon_chemin1" + CR
Code_à_tester += "Description_de_l_erreur = " + Caract(34) + "OK, " +
Caract(34) + " + ma_clé_composée" + CR
```

En cliquant sur le bouton Réplica, le code suivant s'affiche, et il fonctionne :

```
Mon_chemin2 est une chaîne = "\\ZEUS\Efco_Dossiers_Clients\Fichiers"
Mon_chemin1 est une chaîne = "c:\Fichiers_10082004"
HChangeRep("*", Mon_chemin1)
HLitPremier(Mon_fichier, Ma_clé_composée)
Informations // procédure locale
```

Ce code lit des fichiers de sauvegarde du 10/08/2004. En activant alternativement Mon_chemin1 et Mon_chemin2, on lit sur A puis si nécessaire on écrit sur B. Une fois la mise au point terminée, on transférera le code dans des boutons de commande dédiés. En attendant, les boutons de navigation propres à l'application fonctionnent parfaitement sur le serveur, comme en local.

Des fichiers paramètres en mode texte

Le code suivant, dans l'ardoise magique, crée un fichier de paramètres en mode texte, puis vérifie sa validité en extrayant les éléments entrés :

```
Mon_texte est une chaîne
Mon_texte = "Couleur_des_entrées " + tab + "8997532" + tab
Mon_texte += "Couleur_des_sorties " + tab + "255" + tab
Mon_texte += "Couleur_des_infos " + tab + "64732" + tab
description_de_l_erreur = fsauvetexte("C:\Mes_paramètres.txt", Mon_texte)
description_de_l_erreur += Mon_texte + cr
ch est une chaîne = fcharsetexte("C:\Mes_paramètres.txt")
description_de_l_erreur += cr + extraitchaine(ch,2) + cr + extraitchaine
(ch,4) + cr + extraitchaine(ch,6)
```

La supériorité de cette méthode réside dans la souplesse de extrait chaîne, qui n'exige pas de connaître la position de chaque élément, lorsque ceux-ci sont de taille variable.

Lister les fiches détruites d'un fichier

Le code suivant liste toutes les fiches, même détruites, du fichier spécifié :

```
//Lit tous les enregistrements (y compris les détruits)
i est un entier = HNbEnr("Lots",HEtatTous)
description_de_l_erreur = i

POUR i = 1 A HNbEnr("Lots",HEtatTous)
  HLit("Lots",i)
  description_de_l_erreur += Lots.Projet + " " + Lots.Numéro + " " + hetat
  ("Lots") + cr
FIN
```

Note : Description_de_l_erreur est la zone de réception de l'ardoise magique. Elle signale les éventuelles erreurs et sinon, reçoit les résultats.



Saisie du code

Mise à jour d'un fichier lié après ajout d'une rubrique

Une rubrique à rapatrier vers le fichier B est rajoutée dans le fichier A. Pour rapatrier dans B toutes les valeurs concernées, voici un exemple de moulinette réalisé dans une ardoise magique (voir ce mot) :

```
babouin est une chaîne
hlitpremier(efco_dossier_client)
entrée_sur_l_écran
Tantque pas HenDehors(Efco_Dossier_Client)
Description_de_l_erreur += "ok"
babouin = Complète(dossier_bancaire.Nom_client,50) + complète(dossier
_bancaire.Prénoms_client,50)
Description_de_l_erreur += babouin + cr + taille(babouin)
hlitrecherche(Efco_clients, "nom_prénom", babouin)
efco_dossier_client.id_client = efco_clients.idefco_clients
dossier_bancaire.id_client = efco_clients.idefco_clients
Description_de_l_erreur += babouin + efco_dossier_client.id_client + " "
+ efco_clients.idefco_clients + cr
Executetraitement("Fiche_suivante",trtcl)
Fin
```

■ Jacques De Schryver

Découvrir XUL

Sources de l'article
www.programmez.com

*XUL est la réponse que le monde de l'Open Source fait au XAML de Microsoft.
Découvrons le ensemble*

Microsoft nous le dit fièrement, XAML sera le standard pour la création d'interface utilisateur client et il fera partie de Longhorn, la future version de Windows. Standard ou pas, XAML n'aura pas le mérite de la nouveauté, car XUL, pour XML-based User interface Language, est quelque chose de semblable à XAML, que le monde de l'Open Source met d'ores et déjà à notre disposition. Mieux que cela, Firefox, le nouveau navigateur est entièrement écrit en XUL. Découvrons cet intéressant langage ensemble.

Qu'est-ce que XUL ?

XUL est un langage XML dédié à la création d'interfaces utilisateur. Cette création est uniquement descriptive. Un moteur de rendu se chargeant ensuite de transformer cette description en réalité. Bien entendu ce moteur de rendu existe, il s'agit de Gecko qui est au coeur de Mozilla, le célèbre navigateur Open Source. Gecko peut être intégré dans d'autres applications. Le couple Gecko/XUL constitue le coeur de Mozilla et permet à celui-ci d'être étendu très simplement, uniquement au moyen d'ajout de fichiers XUL. Pour en savoir plus sur Mozilla et Gecko consultez <http://www.Mozilla.org>. Un atout majeur de XUL est le découplage de la présentation et de la logique d'une application. XML étant tout ce qu'il y a de portable, XUL l'est de facto. XUL permet de définir des interfaces utilisateurs très élaborées :

- Contrôles de saisie tels que zones de textes et cases à cocher.
- Barres d'outils avec boutons ou autre contenu.
- Barres de menus et menu surgissants.
- Dialogue avec onglets.
- Arbres pour représenter les données hiérarchies.
- Raccourcis claviers.

Un peu de pratique

Rien de tel qu'un peu de pratique pour se fixer les idées. Nous allons donc créer en 4 étapes un document XUL définissant une fenêtre avec labels et boutons fonctionnels. Ce document pourra être visionné en l'ouvrant directement dans Mozilla ou Firefox (ou tout autre application intégrant Gecko), ou servi par un serveur Web, ou encore "invoqué" par une ligne de JavaScript depuis un fichier HTML. Nous utilisons Apache comme serveur Web.

1 Pour se faire la main

Nous commençons par écrire un document XUL décrivant une fenêtre désespérément vide :

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>

<window
  id="programmez-window"
  title="Programmez!"
  orient="horizontal"
```

```
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
</window>
```

Vous trouverez le fichier progwin.xul correspondant à ce document sur le site [programmez.com](http://www.programmez.com). Examinons ce code ligne par ligne. Vient, en premier lieu, une ligne déclarant que nous avons un document XML, ce qui ne surprend personne. Evidemment cette ligne est indispensable. La ligne suivante définit le style du document. L'URL mérite que l'on s'y attarde. Si vous allez sous le répertoire d'installation de votre Mozilla ou Firefox, vous y verrez un répertoire du nom de chrome. Ce répertoire contient moult archives jar qui sont autant de packages. Un package consiste en un certain nombre de fichiers XUL, groupés dans une archive, plus un fichier RDF décrivant le contenu de celle-ci. Tout l'ensemble de ces archives constitue le code de votre Mozilla. Ajouter ici une archive, revient à ajouter une extension au navigateur. Un tel code a des droits supérieurs au code XUL de la page Web que nous créons. En ce qui concerne notre ligne de code, nous voyons que nous ne spécifions pas un style particulier. Les styles par défaut seront donc utilisés. Finalement un code XUL début toujours par ces deux lignes.

Vient ensuite la balise 'window' qui déclare une fenêtre. Les attributs de balises servent tout naturellement à affiner la définition. Le dernier attribut (xmlns) définit un espace de nom. Cet espace de nom est là pour indiquer que tout ce que contient notre fenêtre est du XUL. Le document pointé par l'URL n'est en fait pas chargé, car Mozilla le conserve en interne.

Nous pouvons ouvrir le document dans le navigateur, mais c'est encore sans intérêt :-). Notre fenêtre s'insère dans le navigateur, donc nous ne voyons absolument rien. Nous pouvons aussi charger le document comme s'il était distant. Pour cela, il faut le déposer dans un répertoire que sert Apache (ou votre serveur web). Pour Apache il convient en outre d'ajouter cette ligne à votre fichier httpd.conf:

```
AddType application/vnd.mozilla.xul+xml .xul
```

Après quoi, le démon ayant été relancé, saisir <http://localhost/progwin.xul> dans votre navigateur, vous fait accéder au document. Oui nous ne voyons toujours rien :-)

2 XUL et JavaScript

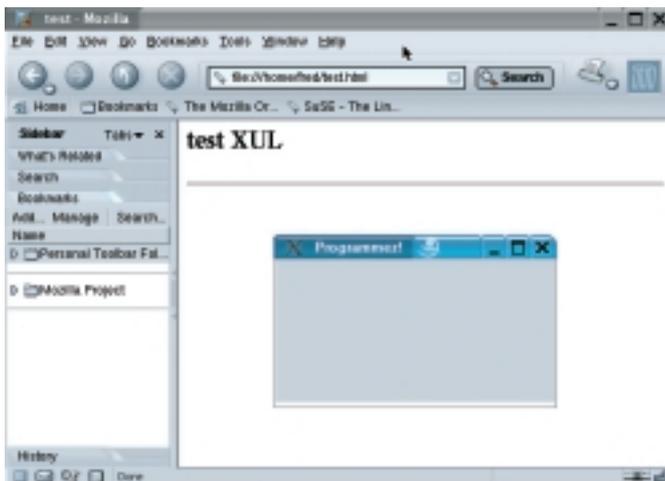
JavaScript est utilisé pour programmer le comportement de l'interface graphique et effectuer les traitements en réponse aux actions sur les composants de l'interface. Nous allons utiliser JavaScript pour rendre notre fenêtre plus vivante, c'est à dire l'ouvrir en dehors du navigateur. Nous écrivons donc un petit document HTML qui servira en outre pour tester nos autres documents.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
```

```
<head>
  <title>test</title>
</head>

<script>
window.open("./progwin.xul", "mafenetre", "chrome,width=250,height=100");
</script>

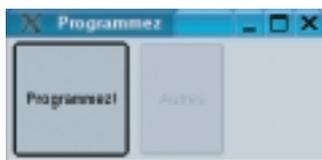
<body>
  <h1>test XUL</h1>
  <hr>
</body>
</html>
```



Notre fenêtre en XUL.

On donne comme URL notre document XUL et surtout, on note que "chrome" apparaît dans les caractéristiques de la fenêtre. Notre document test.html une fois déposé dans le répertoire du serveur, il suffit de saisir dans le navigateur <http://localhost/test> pour admirer le résultat.

3 Poussée de boutons



Deux balises et voilà notre fenêtre équipée de boutons.

Ajoutons des boutons à notre fenêtre. Rien de plus simple. Quelques balises HTML et le tour est joué. Une balise pour Programmez! et une autre pour les autres revues (fichier progwin1.xul).

```
<window
  id="programmez-window"
  title="Programmez"
  orient="horizontal"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">

  <button id="prog-button" label="Programmez!" default="true"/>
  <button id="autre-button" label="Autres" disabled="true"/>
</window>
```

On peut apprécier ici le découplage logique/présentation. Nous n'avons

pas touché au "code" du fichier HTML et rien de plus simple que d'incorporer les balises dans le document XUL. Les boutons sont évolués. Nous obtenons un bouton par défaut et un bouton désactivé au moyen des attributs de balises.

4 Mise en forme et événements

A l'instar des meilleures boîtes à outils graphiques, XUL offre des gestionnaires de mise en forme (layout). Vous l'avez deviné, ceux-ci se déclarent au moyen de balises :-)



L'organisation des widgets s'obtient au moyen de balises de mise en forme.

Le document progwin3.xul enrichit le contenu de notre fenêtre et surtout organise ce contenu au moyen des gestionnaires de mise en forme; Voyons maintenant comment réagir à un clic de souris. Nous pouvons insérer du JavaScript dans la balise de fenêtre (progwin3.xul). Mais ce faisant, on perd en découplage:

```
<window
  id="programmez-window"
  title="Programmez!"
  orient="horizontal"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">

  <script>
function conseil() {
  alert("Abonnez vous!");
}
</script>

<!-- etc -->
</window>
```

Le mieux est d'écrire un fichier JavaScript séparé (conseil.js) et de l'insérer dans le fichier XML comme ceci (progwin4.xul):

```
<script src="./conseil.js"/>
```

Le découplage est ainsi préservé. Notre bouton est opérationnel.



Notre bouton est désormais opérationnel.

En guise de conclusion

Nous n'avons fait qu'effleurer XUL qui est bien plus que ce que nous en avons dit ici. XUL est riche et puissant. Nous vous encourageons vivement à le découvrir davantage.

■ Frédéric Mazué - fmazue@programmez.com

Synth : créer son look & feel sous Java 5

Java 5 propose Synth, un nouveau Look and Feel, paramétrable depuis un fichier XML que nous découvrons aujourd'hui.

Sources de l'article
www.programmez.com

Une des originalités de Java 2 était d'offrir plusieurs Look and Feel. Ce terme recouvre à la fois l'aspect et le comportement des applications graphiques Java. Le Look and Feel vedette étant Metal. Disponible dans toutes les implémentations de la JVM, Metal permet d'assurer un comportement et un aspect identiques quelle que soit la plate-forme hôte. Dans Java 1.5 Metal a été ... relooké. Et s'il a toujours été possible de créer son propre Look and Feel, il est maintenant possible d'en créer un uniquement avec un fichier XML. C'est dans l'air du temps, comme en témoignent le XAML à venir dans Windows Longhorn, ou le XUL de Mozilla/FireFox. Ce Look and Feel à géométrie variable s'appelle Synth. Son atout majeur est son découplage d'avec le code d'une application. Un nouveau fichier XML, et celle-ci se voit complètement relookée, sans qu'il soit besoin d'en modifier la moindre ligne de code.

Etat des lieux

Tous les Look and Feel (LAF) préfabriqués ne sont pas disponibles partout. Ainsi, vous ne trouverez pas le LAF Windows sur une JVM tournant sous Linux pour des questions de droits. Pour connaître les LAFs disponibles utilisez ce code :

```
UIManager.LookAndFeelInfo[] lafi =
    UIManager.getInstalledLookAndFeels();

for(UIManager.LookAndFeelInfo laf : lafi) {
    System.out.println(laf);
}
```

Remarquez au passage la nouvelle syntaxe pour l'itération dans un tableau. Le nom d'un LAF obtenu, il est possible de le faire apparaître à la volée. (encadré 1). Les LAF décrits par une chaîne dans le tableau look_and_feel proviennent d'une JVM 1.5 tournant sur Windows XP. Metal est automatiquement Ocean et reste Metal sur une JVM inférieure. Windows est le LAF natif de Windows XP. WindowsClassic est le LAF natif des Windows antérieures. Vous trouverez le code complet, ainsi que les autres exemples et les fichiers XML sur le site programmez.com.

Synth

Apprenons maintenant à manipuler 4 aspects de Synth, ce qui est globalement aisé, pourvu que l'on possède quelques notions de XML. Comme les autres, LAF Synth est chargé et activé dynamiquement. Charger Synth c'est d'abord lire un fichier XML comme nous l'avons mentionné. Selon toute vraisemblance la localisation exacte d'un tel fichier est inconnue de l'application. Ici notre application est DemoSynth1. Nous supposons simplement que le fichier est dans le même répertoire ou archive que le fichier DemoSynth1.class. L'extrait de code montre comment charger le fichier XML. Si nous ne savons pas où se trouve notre classe DemoSynth1.class, le ClassLoader, lui, le sait puis-

qu'il a chargé la classe. On commence donc par obtenir une instance de l'objet Class représentant la classe et par cette instance nous obtenons un flux d'entrée sur le fichier XML. Il ne reste plus qu'à charger celui-ci avec la méthode Load de SynthLookAndFeel. Après quoi on installe le LAF dynamiquement, exactement comme on le fait pour un autre LAF prédéfini. Le lecteur remarquera que notre code revient ensuite au LAF Metal pour les composants contenus dans un panneau. La raison est que nos fichiers XML sont très rudimentaires. Nous ne nous occupons pas de tous les widgets et du coup, certains de ceux-ci disparaissent presque avec le nouveau LAF. Alors nous les faisons revenir à la vie avec cette astuce.



Fig.1 : demosynth1.xml définit un style pour tous les labels

1 Un Look pour label

Notre fichier XML doit être valide. Sa DTD se trouve dans le répertoire `api\javax\swing\plaf\synth\doc-files` de votre Javadoc. Vous y trouverez également le fichier

synthFile Format.html qui explique cette DTD. Nous y apprenons que nous pouvons définir un style pour une catégorie de composants ou bien au coup par coup. Commençons par définir un style qui impose un fond blanc, un texte rouge et une fonte italique de 48 points pour tous les composants JLabel (Fig.1) :

```
<synth>
<style id="monstylelabel">
  <font name="Serif" size="48" style="BOLD ITALIC"/>
  <opaque value="true"/>
  <state>
    <color value="RED" type="FOREGROUND"/>
    <color value="WHITE" type="BACKGROUND"/>
  </state>
</style>
<bind style="monstylelabel" type="region" key="Label"/>
</synth>
```

La classe DemoSynth1 (encadré 2) chargera ce LAF. Une fois défini, un style doit être attaché (bind) à un ou une catégorie de composants. Lorsque l'attribut type de la balise bind vaut region, la catégorie de composants concernée est décrite par l'attribut key, dont la valeur est une expression régulière.

Les valeurs possibles de key ne sont pas les noms de classes des composants mais sont à prendre dans la classe `javax.swing.plaf.synth.Region`. (Le lecteur voudra bien se reporter à la Javadoc)

2 Label a du style

Le deuxième exemple `demosynth2.xml` définit un style particulier à chacun des deux labels. Pour que notre exemple fonctionne vous devez le programmer avant de charger le fichier XML, sinon les styles génériques de l'exemple d'avant vont rester prépondérants. Intéressons nous au `bind` de ce nouvel exemple :

```
<bind style="monstyleprogrammezname" type="name" key="programmez"/>
```

Ici la valeur `programmez` de l'attribut `key` correspond à la propriété ainsi définie dans le code de l'application :

```
jLabel1.setName("programmez");
```

3 Label avec classe embarquée

A la base, les possibilités de `synth` ne sont pas illimitées, mais elles peuvent le devenir, car il est possible d'embarquer dans les fichiers XML des classes dérivant de `SynthPainter`. C'est ce que fait notre troisième exemple `demosynth2.xml`. Les deux balises clé sont d'abord :

```
<object id="gradient" class="programmez.fred.lafdemos.GradientPainter"/>
```

`GradientPainter` est une classe dérivant de `SynthPainter`. Vous la trouverez sur le Cd-Rom. Attention de bien spécifier le package complet. Et ensuite

```
<painter method="labelBackground" idref="gradient"/>
```

Le lecteur remarquera que la valeur de l'attribut `method` diffère de nom réel de la méthode, celui-ci étant reconstitué par `Synth` au chargement.

4 Quand label réagit aux événements

Un LAF gère non seulement l'aspect, mais aussi le comportement. Notre dernier exemple modifie la police et la couleur de texte des boutons lorsque ceux-ci sont survolés par le pointeur de la souris (exemple `DemoSynth2`).

```
<style id="button">
<font name="Serif" size="12" style="BOLD"/>
<state value="MOUSE_OVER">
<font name="Serif" size="12" style="ITALIC BOLD"/>
<color value="RED" type="TEXT_FOREGROUND"/>
</state>
</style>
```

Bonne programmation avec `Synth`, A bientôt pour de nouvelles aventures dans l'univers de Java.

Encadré 1

Modification dynamique d'un Look and Feel (Extrait)

```
public class UIManagerDemo extends JFrame {
```

```
    static final String[] look_and_feel = {
        "javax.swing.plaf.metal.MetalLookAndFeel",
        "com.sun.java.swing.plaf.motif.MotifLookAndFeel",
        "com.sun.java.swing.plaf.windows.WindowsLookAndFeel",
        "com.sun.java.swing.plaf.windows.WindowsClassicLookAndFeel",
    };
```

```
    private void setLAF(int i) {
```

```
    try {
        UIManager.setLookAndFeel(look_and_feel[i]);
        SwingUtilities.updateComponentTreeUI(getContentPane());
    }
    catch (/* etc */) {
    }
}
```

Encadré 2

// Programme de démonstration de Synth (extrait)

```
import javax.swing.plaf.synth.*;
```

```
import java.io.*;
```

```
public class DemoSynth1 extends JFrame {
```

```
    public DemoSynth1() {
```

```
        try {
            jblnit(); // Méthode générée par JBuilder
                    // Pour l'installation des composants
                    // cf. DemoSynth1.java
            jLabel1.setName("programmez");
            jLabel2.setName("abonnement");
        }
```

```
        catch (Exception ex) {
            ex.printStackTrace();
        }
    }
```

```
        SynthLookAndFeel synth = new SynthLookAndFeel();
```

```
        JButton jButton1 = new JButton();
```

```
        JLabel jLabel1 = new JLabel();
```

```
        JLabel jLabel2 = new JLabel();
```

```
        JPanel jPanel1 = new JPanel();
```

```
        JTextField jTextField1 = new JTextField();
```

```
// Gestionnaire d'événement du bouton
```

```
public void jButton2_actionPerformed(ActionEvent e) {
```

```
    try {
        Class cetteClass = DemoSynth1.class;
        InputStream is =
        cetteClass.getResourceAsStream(jTextField1.getText());
        synth.load(is, cetteClass);
        UIManager.setLookAndFeel(synth);
        SwingUtilities.updateComponentTreeUI(getContentPane());
        pack();
```

```
        UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
        SwingUtilities.updateComponentTreeUI(jPanel1);
        pack();
    }
    catch (/* toutes les exceptions */) {
    }
}
```

■ Frédéric Mazué - fmazue@programmez.com

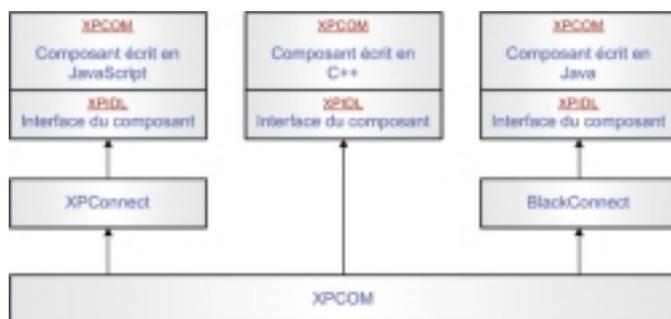
Embarquez votre navigateur Web avec Gecko

Gecko implémente HTML 4.01, XHTML 1.1, XML 1.0, CSS 2.1, RDF et le langage de descriptions d'interfaces XUL... Malgré toute cette panoplie technologique, Gecko est léger et facilement incorporable à vos applications C# ou autres.

Gecko (anciennement NGLayout, alias Raptor) est un moteur d'affichage de pages Web multi plate-forme, dont le développement a débuté en 1998 dans le cadre du projet Mozilla. Il est aujourd'hui employé par de nombreux navigateurs tournant sous Linux, Windows ou Mac OS X (dont le plus connu est Firefox, mais vous avez aussi Salamander, Galeon, Camino, etc.). Il s'agit d'un moteur de rendu, dans le sens où il n'est pas livré avec une interface utilisateur : son boulot consiste à décoder un flux Web et à l'afficher. En associant Gecko avec XUL (eXtensible User-interface Language) vous obtenez l'affichage d'une interface utilisateur multi plate-forme, basée sur XML (avec un mélange d'HTML, de CSS et d'ECMAScript). FireFox utilise XUL, qui fait appel à son tour à Gecko pour afficher les menus et les fenêtres de dialogues.

1 Se procurer la trousse à outils de Mozilla

Ceci dit Gecko, n'est qu'un des constituants de la boîte à outils de Mozilla. En tant que développeur, vous bénéficierez directement ou indirectement des autres éléments pour enrichir vos applications. Il y a d'abord Necko, qui est une bibliothèque réseau fournissant une API indépendante de la plate-forme. Ensuite, XPCOM qui est l'acronyme de Cross Platform Component Object Model, autrement dit, modèle d'objets composants multi-plates-formes. Il s'agit d'un framework dont le but est de réaliser des applications multi-plates-formes. Ces objets XPCOM bénéficient de nombreux "wrapper" avec en tête ECMAScript (le JavaScript d'ECMA), mais aussi Python (PyXPCOM), Perl (plXPCOM), Ruby (rbXPCOM) et Java (BlackConnect). Entre parenthèses, pour développer un composant XPCOM vous devez commencer par vous procurer le SDK de Gecko, que l'on peut télécharger sur les serveurs FTP de Mozilla. XPCOM, comme son nom semble l'indiquer, est une technologie similaire à Microsoft COM. Cependant, ces deux technologies n'ont pas été construites de la même manière et ne sont pas compatibles entre elles. XPCOM est Open Source, portable, et peut être utilisé indé-



Système d'exploitation : Windows XP

Langage : C#

Besoin : Nécessite Mozilla Control 1.7.1

(<http://www.iol.ie/~locka/mozilla/MozillaControl171.exe>)

pendamment de Mozilla. Gecko est portable, grâce à l'utilisation de l'architecture XPCOM.

Dans cette trousse à outils multi-plate-forme pour programmeur, on trouve également ImageLib. Il s'agit d'une bibliothèque de rendu d'images qui se charge de décoder divers formats comme le JPEG, le PNG, le GIF et le PPM (Portable Pixel Map). Enfin, vous disposez d'outils comme Venkman, un puissant débogueur de JavaScript, d'un inspecteur de DOM et d'une console Java et JavaScript.

Ne perdons pas de vue trois points fondamentaux : premièrement, une application compatible Mozilla, s'exécutera sur toute plate-forme compatible Mozilla, deuxièmement, Mozilla est entièrement Open source et libre, troisièmement, en développant une application basée sur Mozilla, vous éliminerez la frontière existant entre une application classique et une page Web.

2 XAML ou XUL ?

Si vous devez effectuer un choix technologique, nous vous conseillons de vous pencher sérieusement sur la trousse à outils de Mozilla. D'abord cette solution existe d'ores et déjà, tandis que du côté de Microsoft la parade ne sera opérationnelle qu'à l'horizon 2007 avec XAML (eXtensible Application Markup Language). Tout comme XUL, XAML est basé sur le langage XML et sera capable de réaliser des interfaces visuelles riches et unifiées mais dédiées aux applications Windows. Le XAML ne reposera pas sur Gecko (...) comme XUL, mais sur un sous-système graphique qui porte le nom d'Avalon. Au final, l'utilisateur n'ouvrira plus un navigateur en tant que tel : tout sera intégré à la même interface. Autrement dit, le navigateur intégré au futur Windows sera complètement transparent pour l'utilisateur final. Malheureusement, XAML est un format propriétaire non reconnu par le W3C. Sans compter le risque sécuritaire que cela comporte : à vouloir intégrer en profondeur le navigateur au sein du système, on risque de transformer très vite un bogue applicatif en faille système.

Concrètement, pour dessiner un bouton avec XAML (source Microsoft MSDN) on procédera ainsi :

```
<Button Background="Green" FontSize="12pt">
  DeclencheLeTraitement
</Button>
```

Avec XUL, il existe deux possibilités. D'abord une première syntaxe proche de celle de XAML :

```
<button label="DeclencheLeTraitement" style="background:Green;
font-size: 12pt"/>
```

Ensuite, une syntaxe qui permet de séparer la structure de la présentation. Ce qui donne pour la structure :

```
<button label="DeclencheLeTraitement" id="Bouton1"/>
```

Et pour le design sa définition figurera dans une feuille de style CSS :

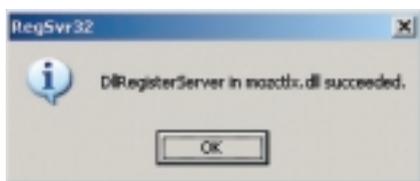
```
#Bouton1 { background:Green; font-size: 12pt }
```

XAML n'est pas encore une technologie à la portée du grand public, mais on peut déjà être certain d'une chose : XUL et XAML seront bien incompatibles.

3 Consultez un code source existant

Gecko peut être intégré dans n'importe quelle application, ce qui dépasse le cadre du simple domaine de la navigation Web. Par exemple l'environnement de développement Komodo de l'éditeur ActiveState est basé sur Gecko. C'est le cas aussi de l'éditeur HTML NVU et de bien d'autres projets. Une démarche logique pour comprendre les arcanes du moteur serait donc de consulter un code source public existant qui l'encapsule.

Une deuxième démarche serait d'aller consulter sur le site de Mozilla la description du jeu d'API. Cependant, au lieu d'employer directement l'API, un développeur peut s'appuyer sur des classes déjà écrites (ce qui masque un certain niveau de difficulté). Sous Windows, il est ainsi possible d'utiliser un contrôle ActiveX (Mozilla ActiveX Control) dans une application. Le contrôle Active X a été écrit en C++ avec ATL. La bibliothèque ATL (Active Template Library) est un ensemble de classes C++ basées sur les modèles et qui permettent de créer facilement des objets COM (Component Object Model) petits et rapides. Le résultat comporte ici plusieurs avantages : une distribution minimale de quelques Mo (alors que le composant similaire pour piloter IE occupe plus de 10 Mo), et la non dépendance de la distribution de la DLL MFC (si le composant avait été créé avec les MFC (Microsoft Foundation Class Library).



Autre avantage de taille : ce contrôle implémente les mêmes interfaces IWebBrowser et DWebBrowserEvents définies pour Internet Explorer ! Un programmeur peut

passer du composant IE à celui de Mozilla sans changer son code source. Tout au plus devra-t-il définir une référence à `_MozillaBrowser` plutôt que d'utiliser la référence `_WebBrowser`. Le principal atout de cette démarche est que du coup, tout devient plus rapide (tout en employant, nous le répétons, exactement les mêmes méthodes, propriétés et événements que Webbrowser).

4 Télécharger le package du contrôle

Comment procéder pour intégrer ce contrôle ActiveX à une application .Net sous Windows (et ici plus spécifiquement en C#) ? La première étape consiste à télécharger le package du contrôle (MozillaControl171.exe). Une fois la procédure d'installation terminée, vous devez commencer par enregistrer le composant dans la base des registres.

```
regsvr32.exe mozctlx.dll
```

5 Générer les assemblages interops

Ensuite, nous allons devoir générer les assemblages interops pour obtenir un contrôle Windows Forms opérationnel (ce qui nous permettra de l'ajouter dans la boîte à outils de Visual Studio). Sans cette opération, vous ne pourrez faire glisser le contrôle Mozilla de la boîte à outils vers un formulaire. Microsoft a développé l'outil axlmp dans le but de convertir les définitions de types d'une bibliothèque de types COM associées à un contrôle ActiveX en un contrôle Windows Forms. Il s'agit bien d'une opération d'importation de contrôles ActiveX dans Visual Studio. Pour bien expliquer, Windows Forms ne peut héberger que des classes dérivées de `System.Windows.Forms.Control`. L'importateur Aximp se charge de générer une classe wrapper qui elle, pourra être incluse dans un Windows Form.

Pour importer le contrôle ActiveX, il est nécessaire de générer ce contrôle wrapper dérivant de `System.Windows.Forms.AxHost`. Ce contrôle wrapper se présente comme un contrôle Windows Forms, mais est capable de communiquer avec le contrôle ActiveX sous-jacent. Ouvrez une boîte de commande Visual Studio et générez les fichiers d'assemblages :

```
C:\Program Files\Mozilla ActiveX Control v1.7.1>dir mo*
Volume in drive C has no label.
Volume Serial Number is CC2C-DFD6
```

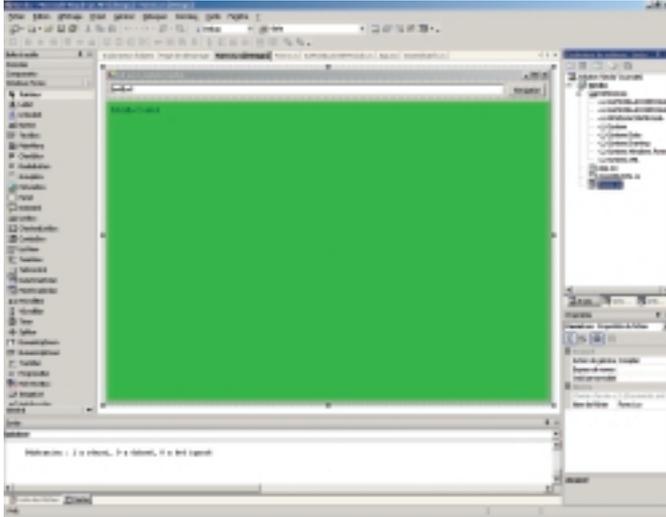
```
Directory of C:\Program Files\Mozilla ActiveX Control v1.7.1
```

```
09/07/2004 12:12      200.704 mozctl.dll
09/07/2004 12:12         8.192 mozctlx.dll
09/07/2004 11:37      10.752 mozilla-ipc.dll
09/07/2004 11:24      37.888 mozz.dll
                4 File(s)      257.536 bytes
                0 Dir(s)  98.747.027.456 bytes free
```

```
C:\Program Files\Mozilla ActiveX Control v1.7.1>axlmp /source mozctl.dll
Generated Source: C:\Program Files\Mozilla ActiveX Control v1.7.1\AxMOZILLACONTROLLib.cs
Generated Assembly: C:\Program Files\Mozilla ActiveX Control v1.7.1\MozillaControlLib.dll
Generated Assembly: C:\Program Files\Mozilla ActiveX Control v1.7.1\AxMOZILLACONTROLLib.dll
```

6 Les tests

Testons maintenant. Créons un nouveau projet C#, et ajoutons-y la référence à `mozctl.dll` (COM). Pour ajouter le contrôle ActiveX à la boîte à outils dans le menu Outils, sélectionnez Ajouter/Supprimer des éléments de la boîte à outils. La boîte de dialogue Personnaliser la boîte



à outils s'affiche. Il suffit de cliquer sur l'onglet Composants COM et d'activer la case à cocher située en regard du contrôle ActiveX Mozilla pour que vous puissiez l'utiliser.

Faisons glisser le composant Mozilla vers le formulaire vierge. Ajoutons-y un bouton de commande `button1`, un contrôle `textBox1` et une barre d'état `statusBar1`. Éditez les propriétés du contrôle Mozilla pour y ajouter la reconnaissance de l'évènement `StatusTextChanged`.

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace Gecko
{
    /// <summary>
    /// Description résumée de Form1.
    /// </summary>
    public class Form1 : System.Windows.Forms.Form
    {
        private AxMOZILLACONTROLLib.AxMozillaBrowser axMozillaBrowser1;
        private System.Windows.Forms.TextBox textBox1;
        private System.Windows.Forms.Button button1;
        ...
    }
}
```

Le code ajouté est très simple :

```
private void Form1_Load(object sender, System.EventArgs e)
{
    this.textBox1.Text = "http://www.programmez.com";
    object n = null;
    axMozillaBrowser1.Navigate(this.textBox1.Text, ref n, ref n,
ref n, ref n);
}

private void button1_Click(object sender, System.EventArgs e)
{
```

```
    object n = null;
    axMozillaBrowser1.Navigate(this.textBox1.Text, ref n, ref n,
ref n, ref n);
}

private void axMozillaBrowser1_StatusTextChanged(object sender,
AxMOZILLACONTROLLib.DWebBrowserEvents2_StatusTextChangedEvent e)
{
    statusBar1.Text = e.text;
}
```

Attention : lorsque nous agrandissons la fenêtre il y a lieu de redéfinir le layout. Cliquez sur les propriétés du formulaire pour y activer l'évènement `Layout`. A chaque fois que l'utilisateur modifiera la taille de la fenêtre de l'application nous recalculerons automatiquement les proportions.

```
private void Form1_Layout(object sender, System.Windows.Forms.
LayoutEventArgs e)
{
    textBox1.Location = new Point(
        this.ClientRectangle.Left,0);
    button1.Location = new Point(this.ClientRectangle.Right -
button1.Width,0);
    textBox1.Size = new Size(button1.Left - textBox1.Left, button1
.Height);

    statusBar1.Location = new Point(
        this.ClientRectangle.Left,
        this.ClientRectangle.Bottom - statusBar1.Height);

    axMozillaBrowser1.Location = new Point(
        this.ClientRectangle.Left, textBox1.Bottom);
    axMozillaBrowser1.Size = new Size(
        this.ClientSize.Width,
        statusBar1.Top - axMozillaBrowser1.Top);
}
```

Enfin, nous vous laissons le soin de définir une barre d'outils pour contrôler la navigation arrière et avant, l'arrêt et le rafraîchissement. Le code nécessaire à implémenter sera d'une facilité déconcertante :

```
// En arrière
axMozillaBrowser1.GoBack();
// En avant
axMozillaBrowser1.GoForward();
// Arrêt
axMozillaBrowser1.Stop();
// Rafraîchissement
axMozillaBrowser1.Refresh();
```

Et sous Linux ?

Sous Linux, il existe le projet `wxMozilla` qui expose sous la forme d'un composant `wxWindows` des fonctionnalités sous-jacentes de `gecko`.

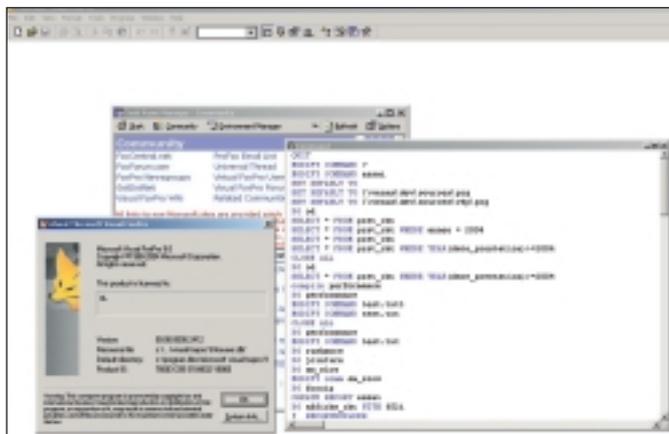
■ **Xavier Leclercq** - Xavier.Leclercq@programmez.com

Visual FoxPro Europa : la meilleure version depuis Visual FoxPro 3

Europa est le nom de code de la dernière mouture de Visual FoxPro proposée par Microsoft. Cette version 9 apporte de nombreux changements, comme un nouveau générateur de rapports ultra puissant supportant GDI+.

Les principales caractéristiques de VFP

La caractéristique principale de Visual Foxpro est son interface de développement. Si vous lancez le logiciel vous vous retrouvez en mode ligne de commande avec, à votre disposition, un moteur de base de données embarqué. VFP est à la base un interpréteur et cette approche permet de développer des applications très rapidement. Il s'agit à notre avis d'un des meilleurs outils de développement pour petites entreprises voulant élaborer des applications verticales de gestion (comme l'encodage de prestations, un module de facturation, la génération de rapports divers, la création de fiches de paies, etc.). VFP 9 (build 2412) n'est pas intégré à Visual Studio 2005 et c'est une bonne chose : en effet le runtime d'EUROPA occupe moins de 5 Mo et une installation de base ne nécessite pas 220 Mo d'espace disque. Nous sommes loin des besoins gargantuesques de Visual Studio ! Cependant, l'interaction entre Visual Studio et VFP a été améliorée : amélioration d'une part du support XML qui existait déjà avec VFP 7 et VFP 8 (encadré 1), mais aussi introduction de nouveaux types de données pour une meilleure compatibilité avec Microsoft SQL Server 2000 et SQL Server 2005 (Varchar, Varbinary et Blob).



1 Les améliorations de l'IDE

Chaque nouvelle version de VFP a toujours été accompagnée d'amélioration de l'IDE et cette version 9 ne fait pas exception à cette règle.

L'IDE de VFP est très comparable à celui de VB. On y retrouve la coloration de syntaxe, l'intellisense, y compris sur les objets COM, des fenêtres ancrables, un gestionnaire de projet arborescent, un débogueur avec trace, call, watch, locals, points d'arrêts par click, une liste des tâches, une intégration avec Visual Source Safe, la mémorisation de

Système d'exploitation : Windows XP recommandé, exécution sous Linux via l'émulateur Wine possible

Langages : SQL et langage propre à Visual Foxpro

Besoins : 250 Mo d'espace disque, 256 Mo de RAM, machine cadencée à au moins 1 Ghz

Encadré 1

Sous VFP le XML est omniprésent. Vous pouvez convertir une table en un format XML et inversement :

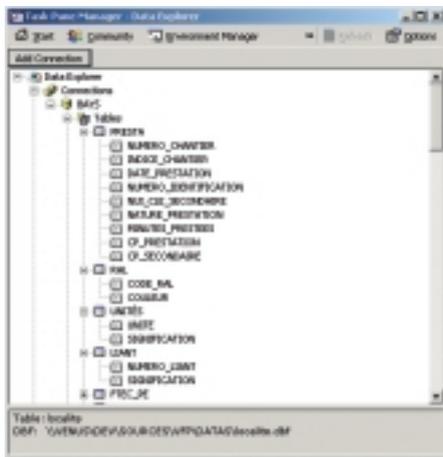
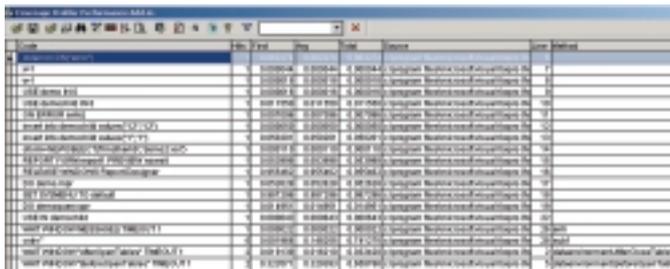
```
CLOSE DATABASE ALL
USE HOME(2)+"\data\customer"
CURSOR TO XML("customer", "lcXML", 1, 32)
STR TO FILE(STRCONV(lcXML, 9), "customer.xml")
XML TO CURSOR("customer.xml", "curCustomer", 512)
RETURN
```

On peut aussi associer à chaque fichier XML un schéma (XML Schema Definition (XSD)), et ainsi créer ou manipuler celui-ci par programmation (à travers l'API XML Schema Object Model (SOM)). Ce qui donne ceci, par exemple, pour créer un curseur à partir d'un schéma :

```
adapter = CREATEOBJECT("XMLAdapter")
cFile = "Votre_Chemin\StructureXML.xml"
adapter.LoadXML(cFile, .T.)
adapter.Tables(1).Fields(1).DataType = "C"
adapter.Tables(1).Fields(1).MaxLength = 10
adapter.Tables(1).ToCursor()
```

la taille, de la position, et de la présentation des fenêtres. L'intellisense y est adaptable (introduction de vos propres fonctions et procédures à l'aide de l'intellisense manager), la fenêtre Document View présente un sommaire complet des modules existants dans un fichier programme, une classe ou un écran. Enfin, il y a la fenêtre des commandes qui a forgé une grande partie de la réputation de Foxpro. Dans celle-ci, vous pouvez y exécuter toute commande ou groupe de commandes pour test avant son intégration dans un programme.

Avec cette nouvelle version, vous pouvez laisser le soin à l'IDE de compiler votre code à l'arrière plan (il suffit d'activer la coloration syntaxique), ce qui est vraiment un plus concernant la détection d'erreurs. Pour l'optimisation du code, un outil de profilage a été ajouté (add-in coverage profiler).



En outre, VFP 9 est livré avec un nouvel outil le DATA EXPLORER que l'on peut activer du Task Pane Manager. Il sera utilisé pour parcourir des tables de bases de données locales Foxpro ou externes (comme SQL Server), créer des requêtes et en visualiser le résultat, etc.

2 Le SQL revu et corrigé

Avant VFP 9, la syntaxe du SQL de Foxpro datait de la version 2.6 pour DOS... Ce dialecte SQL a toujours été non compatible avec le standard ANSI. VFP 9 améliore enfin ce point crucial, tout en conservant une compatibilité avec l'existant si nécessaire. La syntaxe proposée est maintenant fort proche des produits SQL Server de Microsoft. Une table VFP peut comporter jusqu'à 255 champs et 1 milliard d'enregistrements. Chaque champ peut revêtir un des 12 types correspondants aux standards SQL. On peut associer à une table un fichier index comportant des clés en nombre illimité.

Malheureusement, la limite de 2 gigas pour une table n'a pas été franchie. C'est dommage, mais compréhensible dans le sens où intrinsèquement VFP 9 est un produit 32 bits (il pourra tourner sous une architecture 64 bits mais en mode compatibilité 32 bits). Et puis, et surtout, Microsoft ne désire pas empiéter sur son moteur de base de données haut de gamme SQL Server (remarquez que le produit, pourtant de base, SQL Server 2005 Express Edition, peut atteindre la limite de 4 gigas par table !). Ceci dit, dans d'autres domaines certaines limites n'existent plus, comme le nombre de jointures et de sous-requêtes, ou le nombre d'UNIONS au sein d'un SELECT. Une requête peut faire référence à un nombre illimité de tables, et vous pouvez avoir plus de 24 arguments (note 1) dans un SQL IN (select * where variable in (1,2,3,4,...,25,26,27,...)). Les performances ont été améliorées dans plusieurs domaines. Nous avons réalisé des tests. Au niveau de la vitesse d'interprétation pure VFP 9 est plus lent que ses prédécesseurs. Cette constatation peut s'expliquer : plus de mots clés (tokens) sont reconnus par la boucle d'interprétation. Au niveau des requêtes SQL, il y a par contre une nette amélioration si on tient compte de ce qui a été optimisé, comme l'utilisation des mots clés LIKE ou TOP dans un SELECT, ou encore de l'emploi du nouveau type d'index dit "binaire".

Cet index encore appelé index bitmap, convient pour des index d'expressions logiques, comme par exemple "l'enregistrement est effacé ou non". Il tient beaucoup moins de place et sera par conséquent (beaucoup) plus rapide.

Enfin, VFP 9 permet de créer une application à partir d'une base de données VFP et, si nécessaire, la transférer sur une base SQL sans changer le code de l'interface utilisateur.

Note 1

La limite du nombre d'arguments est fixée par un appel à SYS(3055) SYS(3055 [, nComplexity])

Par défaut ce niveau de complexité est fixé à 320 mais peut varier jusqu'à 2040.

3 Un gestionnaire de rapports plus élaboré

Ici aussi les concepteurs ont décidé de ne pas entraver la compatibilité descendante : vous pouvez toujours éditer ou utiliser vos rapports conçus avec une version antérieure : c'est à vous de décider de basculer ou non d'un générateur à l'autre.

*- Utilise le nouveau générateur
_REPORTBUILDER = HOME() + 'ReportBuilder.app'

*- Utilise l'ancien générateur
_REPORTBUILDER = "

Le nouveau Report Writer peut maintenant partager avec d'autres rapports un environnement de données. Et si vous le désirez, cet environnement de données (bases de données, tables et curseurs qui sont employés par le générateur), sera à son tour sauvegardé en une classe pouvant être rechargée ultérieurement. Le gain de temps est appréciable, d'autant plus que vous pouvez, de la même manière, créer un rapport en vous basant sur l'environnement de données d'un autre. Nouvelle fonction : chaque objet défini au sein de votre layout pourra être protégé : par exemple, tel champ ne pourra être déplacé, ou effacé, ou sera bloqué dans ses proportions sur le layout. Cet aspect des choses est important pour développer rapidement, car on n'est jamais à l'abri d'une mauvaise manipulation qui effacerait ou changerait de position un groupe de champs (le rapport complet pourra être protégé ; ou vous déciderez de ne protéger que des éléments individuels).

Un des plus grands changements se situe dans la création de bandes de détails multiples. Cette possibilité assez révolutionnaire pour VFP, autorise un développeur à traiter plusieurs tables enfants pour un enregistrement parent.

*- Ouvertures des tables enfants
USE Membres IN 0 ORDER ClientsClePrimaire
USE Vehicules IN 0 ORDER ClientsClePrimaire
USE Habitations IN 0 ORDER ClientsClePrimaire

*- Ouverture de la table parent
SELECT 0
USE clients ORDER ClientsClePrimaire

*- Mise en place des relations

```
SET RELATION TO ClientsClePrimaire INTO Membres
SET RELATION TO ClientsClePrimaire INTO Vehicules ADDITIVE
SET RELATION TO ClientsClePrimaire INTO Habitations ADDITIVE
```

* - Génère le rapport
REPORT FORM ASSURANCE PREVIEW

Ainsi pour chaque client vous pouvez obtenir plusieurs bandes détails :

[Première bande détail]

Client : Monsieur Jean DUPONT

Rue du paradis

Membres :	Nom	Date naissance	
	Jean Dupont	01/02/1963	
	Céline Dupont	28/03/1965	
Véhicules :	Année	Type	Valeur
	1983	Ford	xx Euros
	1990	Opel	xx Euros
	1995	Peugot	xx Euros

Habitation

Année construction	Rue du paradis	xx Euros
1954		

[Deuxième bande détail]

Client : Monsieur Charles Petit

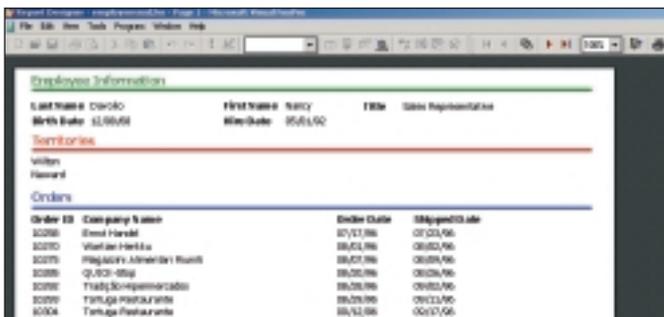
Rue de l'enfer

Membres :	Nom	Date naissance	
	Charles Petit	01/04/1958	
Véhicules :	Année	Type	Valeur
	2000	Ford	xx Euros
	2001	Renaud	xx Euros

Habitation

Année construction	Rue de la résistance	xx Euros
1978		
1985	Rue de l'enfer	xx Euros

etc.



4 VFP 9 et .NET

Il est facile de créer un service Web avec Visual Studio puis de le consommer sous VFP 9. Le code .Net en C# :

```
[WebMethod(Description="Un exemple simpliste")]
public string Bonjour(string Nom)
{
    return "Bonjour, " + Nom + "! Il est maintenant : " +
```

```
    DateTime.Now.ToString();
}
```

Le code VFP 9 :

```
o = CREATEOBJECT("MSSoap.SoapClient30")
? o.MSSoapInIt("http://localhost/VFPWebService/" +
    "VFPInteropService.asmx?WSDL")
? o.Bonjour("Xavier")
```

Le contraire est aussi envisageable, quoique plus compliqué : vous pouvez créer une application Web avec ASP.NET qui interrogera un service Web construit en VFP. Par exemple vous pouvez interroger à distance votre base de données Foxpro, et ce mécanisme d'échange retournera le résultat de vos requêtes SQL. Remarquez que vous pouvez déjà interroger vos tables DBF en ASP.NET via OLEDB, comme le montre cet extrait de code :

```
...
private bool GetCustomerList(string lcCursor)
{
    this.oConn = new OleDbConnection("Provider=vfpoledb.1;" +
        @"Data Source=C:\Programs\vpf\Samples\Data\testdata.dbc;" +
        "Exclusive=false;Nulls=false");
    try {
        this.oConn.Open();
    }
    catch(Exception ex)
    {
        this.lblErrorMsg.Text = ex.Message;
        return false;
    }
    string lcSQL = "select company, contact, city from customer";
    OleDbDataAdapter oAdapter = new OleDbDataAdapter(lcSQL,this.oConn);
    if (this.oDS == null)
    {
        this.oDS = new DataSet();
    }
    try
    {
        oAdapter.Fill(this.oDS,lcCursor);
    }
    catch(Exception ex)
    {
        this.lblErrorMsg.Text = ex.Message;
        return false;
    }
    finally
    {
        this.oConn.Close();
    }
    return true;
}
...
```

Enfin, vous pouvez employer des composants .Net en VFP 9 via COM, ou automatiser des applications depuis VFP 9 via COM.

```
...
* - Extrait de code de l'automation d'Open Office
PUBLIC goOOoDesktop
goOOoDesktop = OOoServiceManager_Createlnstance( "com.sun.star.frame.Desktop" )
COMARRAY( goOOoDesktop, 10 )
...
```

5 Migrer plus facilement vers .NET avec l'assemblage VFPToolkitNET

Depuis 2002, il existe un assemblage du nom de VFPToolkitNET.DLL qui comprend plus de 250 fonctions FoxPro, permettant à un développeur .Net d'utiliser en code managé des fonctions VFP comme CreateObject() ou DOW(). Ce qui donne par exemple en C# :

```
DateTime tDateTime = DateTime.Now;
int nDow = VFPToolkit.dates.DOW(tDateTime);
```

Cet assemblage a été placé dans le domaine public. Son code source est lui aussi disponible (l'implémentation des fonctions VFP, soit en Visual Basic.NET soit en C#). Ainsi, l'implémentation de la fonction (simple) DOW en C# est la suivante : public static int DOW(System.DateTime dDate) {return (int)dDate.DayOfWeek;}

6 Faire appel à GDI+

Nous avons gardé le meilleur pour la fin. Microsoft Visual FoxPro 9 utilise GDI+ en interne. Avec les précédentes versions de VFP, pour afficher un graphique dans un formulaire vous deviez obtenir au préalable un "contexte" via les API Windows, et cette possibilité était complètement masquée avec le générateur de rapports. Avec VFP 9 est livrée une nouvelle propriété GDIPPlusGraphics de la classe ReportListener, qui contient un handle utilisable par votre code VFP.

```
...
if not .IsSuccessor
    .SharedGDIPPlusGraphics = .GDIPPlusGraphics
endif not .IsSuccessor
.oGDIPGraphics.SetHandle(.SharedGDIPPlusGraphics)
...
```

En clair, vous pouvez dorénavant afficher par programmation un graphique dans vos rapports ! Les possibilités sont ici fort nombreuses.

EXEMPLE

Affiche la liste des polices de caractères disponibles dans un rapport :

```
CREATE CURSOR crsrFontList ( ;
    cFont char(32))

llGotFonts = AFONT(laFonts)
FOR i = 1 TO ALEN(laFonts)
    INSERT INTO crsrFontList VALUES (laFonts(i))
ENDFOR

CREATE REPORT fontList FROM DBF("crsrFontList")
```

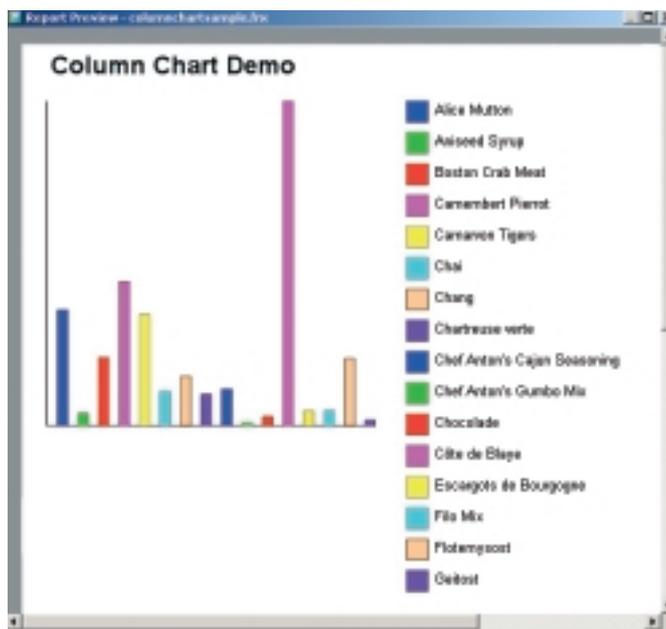
```
loListen = CREATEOBJECT("FontListener")

REPORT FORM fontList PREVIEW OBJECT loListen

DEFINE CLASS FontListener as ReportListener
    ListenerType = 1

PROCEDURE EvaluateContents
    LPARAMETERS nFrXRecno, oObjProperties

    oObjProperties.FontName = ALLTRIM(cFont)
    oObjProperties.Reload = .T.
ENDPROC
```



Pour conclure

En ce qui concerne l'avenir de Foxpro il serait facile de déclarer "il n'y a actuellement aucune assurance pour que VFP 10 soit un jour publié, le marketing de Microsoft poussant essentiellement la branche SQL Server et en particulier les derniers nés de la famille Express". C'est juste, mais cette même litanie se répète depuis Visual Foxpro 3 !

De l'aveu même de l'éditeur 100 000 développeurs utiliseraient encore VFP dans le monde, essentiellement pour le compte de petites entreprises. Il est clair que ceux-ci ont eu des années pour changer d'outil. Et s'ils ne l'ont pas fait c'est qu'il existe de très bonnes raisons pour cela. VFP n'est pas gratuit, mais il est très abordable. En outre les programmes développés avec VFP (exécutables) sont distribuables avec le runtime VFP gratuitement. En plus d'être abordables, tout est compris dedans : un générateur de report, une panoplie d'assistants, une puissante base de données et un runtime. VFP 9 nous permet aujourd'hui de nous connecter à toutes les bases de données du marché par oledb, odbc, xml, etc. VFP n'est pas Access, ni SQL Server. Il s'agit bel et bien d'un outil unique en son genre pour développer très rapidement une application de gestion. Microsoft nous livre ici un excellent cru. L'essayer c'est l'adopter.

■ **Xavier Leclercq** - xavier.Leclercq@programmez.com

Ecrivez vos applications pour Linux/Kde

Moyennant une connaissance de base en C++ et l'outil KDevelop, vous pouvez écrire facilement des applications pour KDE, le bureau de Linux.

Sources de l'article
www.programmez.com

Le monde des logiciels libres est d'une richesse telle que l'on peut toujours trouver l'outil qui convient à un projet. Aujourd'hui, nous nous proposons d'aborder la programmation d'applications graphiques en C++ pour Linux. Et déjà, nous avons l'embarras du choix. Quel outil de développement utiliserons nous ? Quelle librairie graphique (Toolkit) sera le support de notre application ? Travaillerons nous pour le bureau KDE ou le bureau GNOME ? A moins qu'il ne soit possible de faire en sorte qu'un même programme puisse tourner sous les deux bureaux ?

1 Environnements et outils

Par préférence, votre serveur choisit le bureau KDE. Ceci dit, sans vouloir faire renaître les discussions enflammées entre partisans de Gnome et de KDE :) Nous utiliserons l'outil de développement KDevelop, par définition parfaitement intégré à KDE (Fig.1), mais qui fonctionne aussi très bien sous Gnome, ce qui contentera tout le monde. KDE et ses applications sont construits à partir de la librairie graphique Qt. Nous verrons donc comment écrire une application Qt pure, ce qui présente de nombreux intérêts. Une application Qt pure fonctionnera parfaitement sous Gnome. Et même sous... Windows

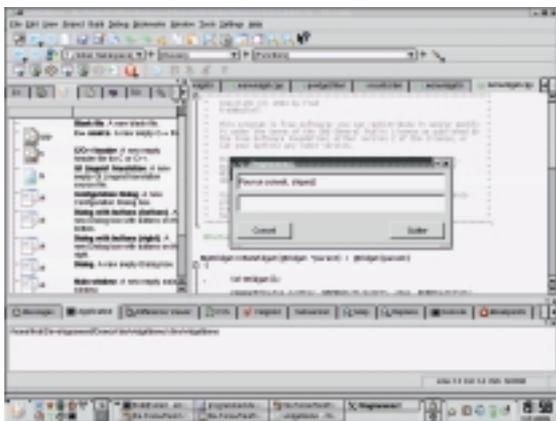


Fig.1 : KDevelop en action dans KDE.

moyennant une recompilation des sources. Pour les applications Qt pures, KDevelop génère un fichier .pro. Sous Windows ce fichier sera repris par Visual C++ pour générer un projet Visual à base de Qt. Ceci bien sûr, pour peu que vous ayez installé au moins la version personnelle de Qt dans votre Windows. C'est très intéressant que de mêmes sources puissent construire les mêmes interfaces sous Linux (ou même UNIX) et Windows. Qt sait le faire, comme wxWidget, autre toolkit remarquable, alors sachons en profiter. Nous verrons aussi comment écrire une application KDE simple. Une telle application devrait aussi tourner sous Gnome, moyennant l'invocation automatique de Kbuildsysoca. Mais il est évident qu'une application KDE telle qu'une extension du bureau ne tournera pas convenablement. Nos objectifs sont donc modestes pour aujourd'hui, mais nous avons quand même pas mal de grain à moudre car nous avons beaucoup à découvrir.

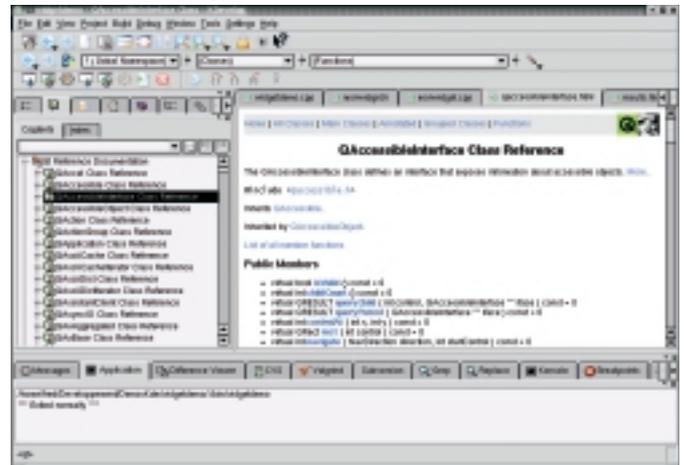


Fig.2 : Bien configuré KDevelop intègre remarquablement bien les documentations.

2 Prendre en main KDevelop

Selon toute vraisemblance, vous en disposez déjà, car il fait partie de votre distribution Linux. A condition que vous ayez installé les paquetages de développement. Si ce n'est pas le cas vous devez le faire maintenant, MÊME si KDevelop est déjà présent sur votre système. En effet, vous aurez besoin d'en-têtes .h et de documentation et ceux-ci font partie des paquetages de développement. Si tout va bien à ce point vous pouvez, si vous êtes pressés, passer au paragraphe suivant. Il n'est pas rare que KDevelop fonctionne mal au niveau de la documentation avec les installations standard des distributions. C'est très dommage, car ses fonctions de recherche et de visualisation de la documentation et de navigation dans les sources et les en-têtes de Qt et KDE sont très puissants (Fig.2). Si vous voulez profiter vraiment pleinement d'un KDevelop configuré aux petits oignons, le mieux est de le télécharger sous la forme de sources et de le compiler. Faisant ainsi vous serez sûr que tout ira bien. Voici dans ce cas la marche à suivre. D'abord désinstaller le paquetage rpm de KDevelop. Ensuite télécharger les sources au plus haut niveau de version compatible avec votre version de KDE à <http://www.kdevelop.org>. Pour information j'utilise un KDE 3.1.4 et un Kdevelop 3.0.4. Décompressez l'archive de sources quelque part. /usr/src est un endroit de bon aloi. Puis positionnez vous dans ce répertoire. Par exemple:

```
cd /usr/src/kdevelop-3.0.4
```

Vous devez ensuite préparer la compilation en exportant quelques variables d'environnement décrivant l'installation de votre librairie Qt et de KDE lui même. Par exemple, sur ma distribution Suse, Qt se situe en /usr/lib/qt3 et KDE en /opt/kde3. Si vous n'arrivez pas à localiser ces répertoires, vous pouvez, par exemple, utiliser kpackage qui vous permettra de visualiser où sont situés les fichiers. Voici maintenant la for-

mule magique pour exporter les variables. Naturellement vous adaptez cette formule à votre configuration.

```
export KDEDIR=/opt/kde3
export QTDIR=/usr/lib/qt3
export LD_LIBRARY_PATH=$QTDIR/lib:$KDEDIR/lib:$LD_LIBRARY_PATH
export LIBRARY_PATH=$QTDIR/lib:$KDEDIR/lib:$LIBRARY_PATH
export PATH=$QTDIR/bin:$KDEDIR/bin:$PATH
```

Nous pouvons maintenant lancer le fameux configure, mais pas n'importe comment. C'est le moment de dire à KDevelop où se situe la documentation pour en profiter pleinement ensuite. Là aussi vous devez chercher le répertoire kdelibs-apidocs sur votre système. Vous le trouverez en principe sous le répertoire d'installation de votre KDE. Voici pour mon système la formule magique. Attention à bien taper la totalité sur une seule ligne de commande

```
./configure --prefix=$KDEDIR --with-kdelibsdoxy-dir=$KDEDIR/share/doc/HTML/en/kdelibs-apidocs
```

Il est possible que l'opération de configuration vous signale l'absence de htdig, l'utilitaire pour indexer la documentation. Installez le depuis les Cd-Roms de votre distribution. Il n'est pas exclu non plus que cette installation dépose le programme htsearch dans un endroit incongru que KDevelop ne saura pas trouver tout seul. Dans ce cas, vous donnerez manuellement cet emplacement depuis la boîte de configuration de KDevelop. Il est également courant que des distributions (c'est le cas de ma Suse 9.0) ne donnent pas l'outil graphviz que KDevelop utilise dans le navigateur de classe. Le cas échéant, téléchargez graphviz à www.graphviz.org. Il est maintenant temps de lancer la compilation :

```
make
```

Vous pouvez aller prendre un café, ou même deux, si votre machine n'est pas très puissante. Prévoyez une promenade en forêt si votre bécane date un peu. Une fois la compilation effectuée, installez KDevelop avec cette commande lancée sous root :

```
make install
```

3 Une première application

Bouillants d'impatience, nous allons construire ensemble notre première application KDE. Le lecteur voudra bien garder à l'esprit que KDevelop évolue très vite, à l'instar d'autres projets Open Source. Aussi peut il y avoir des différences entre les indications que nous donnons et la réalité sur votre machine. Toutefois dans l'esprit l'essentiel doit y être. Vous trouverez tous nos exemples sur le site programmez.com. KDevelop vient avec une ribambelle de wizards que l'on atteint par le menu projectnew project. Choisissez une application C++ puis une application KDE. S'offrent alors à vous de nombreuses possibilités ;

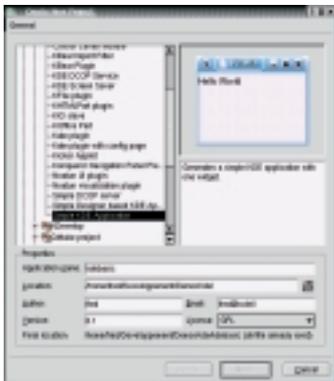


Fig.3 : Notre première application KDE doit être aussi simple que possible.

parmi lesquelles une " grosse " application KDE avec architecture Document/View à la manière des MFC de Microsoft, une extension au bureau et même une application toute simple. Optez pour ce choix (Fig.3). Donnez un nom au projet, puis parcourez tous les dialogues qui vont suivre en cliquant sur le bouton 'suivant' jusqu'à terminer. KDE génère un projet complet pour votre système et emploie pour cela le fameux configure ainsi que vous pourrez le voir. Une fois le projet généré, compilez puis exécutez l'application depuis le menu ou la barre d'outils.

4 Distribuer une application

Notre application est certes tout ce qu'il y a de modeste (Fig.4), mais dès que vous saurez l'enrichir, rien ne vous retiendra plus pour écrire le programme le plus génial de tout l'univers connu et inconnu, et bien légitimement, ce programme vous voudrez le distribuer afin que tout le monde en profite. Il y a deux façons de procéder : sous la forme d'une archive de sources, ou sous la forme de paquetage RPM. L'archive de source est ce qu'il y a de plus universel dans le monde d'UNIX. Voici comment procéder. Depuis le menu Build lancez d'abord Project Clean. Puis Dist Clean. Ce faisant, vous avez retiré du projet tout ce qui est spécifique à votre système. Depuis le menu du Project, basculez maintenant sur la configuration de construction (Build Configuration) par défaut, car vous êtes sans doute sur la configuration Debug. Reconstituez votre projet et, éventuellement relancez l'exécutable si vous voulez être certain que tout va bien. Puis allez dans le menu Tools/Project Distribution & Publishing. Cliquez simplement sur le bouton Create Source Archive du premier volet et c'est tout (Fig.5). Vous trouverez l'archive toute prête à la racine de votre projet. Il s'agit d'une archive UNIX dans les règles de l'art avec macros M4 et tout et tout. Les macros M4 sont des scripts futés, capables de vérifier que gcc, qt, ou autres outils nécessaires sont bien présents sur le système hôte. Cette recherche se fera lorsque la personne qui veut compiler votre application fera la fameuse invocation ./configure, suivie des non moins fameux make et make install.



Fig. 4 : Notre application est tout simple mais ce n'est qu'un début ;-)

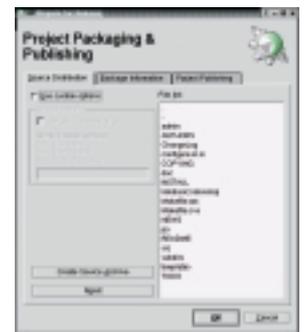


Fig.5 : Un simple clic sur un bouton et la distribution de votre application est prête.

5 La librairie Qt et l'organisation d'une application Qt

Nous laissons pour un temps KDE, car nous devons faire connaissance avec la librairie Qt qui est à la base de tout. Qt c'est une vaste panoplie de widgets (composants graphiques) et un mécanisme signal/slot aussi original que puissant. Une fois familiarisé avec ce mécanisme, le programmeur peut se lancer dans les écritures des applications Qt ou KDE les plus folles. Bien entendu, il est bon de bien connaître tous les widgets mais ceci demande du temps et de la pra-

tique et notre propos n'est pas de les passer tous en revue ici :-)

Créer un nouveau projet C++ et demandez cette fois une application Qt aussi simple que possible soit 'Qmake project' et 'HelloWorld Application' Eviter l'application complète, car le wizard générerait un code complexe et de nombreux fichiers. Une telle application a une architecture Document/View à la MFC Microsoft et n'est pas simple à appréhender. Remplacez le code généré par le wizard par le code ci-contre (encadré 1). Compilez et essayez. Le résultat est modeste mais déjà intéressant (Fig.6). Examinons ce code. Nous utilisons deux classes Qt. La première, QApplication, est fondamentale puisqu'elle gère tout ce qui concerne l'application au bas niveau. C'est notamment cet objet qui lance une pompe à message dans un thread, afin que tout événement tel que clic de bouton de souris ou frappe de touche du clavier parvienne au widget concerné. La deuxième classe est QPushButton, qui encapsule le comportement d'un bouton poussoir. Il s'agit donc d'un widget spécialisé qui étend, comme tout widget spécialisé, la classe QWidget. Le code d'exemple utilise l'instance de ce widget pour en faire le widget principal de l'application, aussi bizarre que cela paraisse de la part d'un simple bouton. C'est un comportement classique sous X-Window qu'un widget quelconque puisse remplir une fenêtre. Bien sûr nous aurions pu y mettre une instance de QWidget en lieu et place, mais notre fenêtre aurait eu l'air encore plus vide :-). Les deux lignes de code suivantes sont anodines. Examinons les deux dernières :



Fig.6 : Notre application est tout simple mais ce n'est qu'un début :-)

```
pb.show();
```

Comment son nom ne l'indique pas, cet appel ne fait rien apparaître à l'écran, mais, en jargon X11, réalise les widgets, c'est à dire les construit, non d'un point de vue C++ mais d'un point de vue X11. Les widgets n'apparaissent à l'écran que quand l'application lance la pompe à message par:

```
return App.exec();
```

Quand le widget principal est détruit, ou bien quand l'application s'arrête, suite à une invocation de sa méthode quit, la méthode exec rend la main. Juste avant cela, tous les widgets sont automatiquement détruits, y compris tous les widget enfants. En fait, ainsi que nous allons le voir, tout widget peut servir de conteneur à d'autres widgets qui deviennent ses enfants.

6 Le mécanisme signal/slot

Signaux et slots sont des méthodes de classes déclarées dans des sections signals : et public slots : Signaux et slots peuvent être connectés les uns aux autres par la suite. Un signal d'une classe X peut être connecté à de multiples slot de classe X, Y, Z, etc. Et un slot d'une classe X peut être connecté à de multiples signaux de classe X, Y, Z etc. Quand une classe émet un signal, toutes les méthodes slots auxquelles celui-ci est connecté sont exécutées, dans un ordre arbitraire.

Modifiez notre application de départ en y ajoutant une ligne, pour

connecter un signal que le bouton émettra lorsqu'il sera cliqué à la méthode quit de l'objet QApplication, cette méthode étant un slot.

```
pb.setCaption("Programmez!");
QObject::connect(&pb, SIGNAL(clicked()), &App, SLOT(quit()));
pb.show()
```

Recompilez et testez. Un clic sur le bouton ferme l'application. Nous avons donc à faire à des extensions au langage C++. Pour que le compilateur gobe cette chose, toute classe déclarant des signaux ou des slots doit comporter la macro Q_OBJECT au début de déclaration. Quand le préprocesseur rencontre et étend la macro, les mots clés signal slot et emit sont retirés. Parallèlement, l'utilitaire de Qt Qmake (ou l'utilitaire de construction de KDevelop pour une application KDE) va mettre à jour votre makefile afin de générer, via le compilateur moc de Qt, un fichier auxiliaire C++ préfixé moc_. A partir de quoi, le compilateur C++ et l'éditeur de liens peuvent construire l'application sans s'apercevoir de rien :-). Ce mécanisme est complexe mais heureusement transparent. Toutefois, c'est une belle aventure que d'examiner le contenu des moc_ qui contiennent l'implémentation des extensions :-). On notera que les méthodes signal sont déclarées mais jamais implémentées par le programmeur. C'est le compilateur moc qui s'en charge.

7 Une application plus évoluée

Nous allons enrichir notre application Qt (demowidget sur le Cd-Rom) (Fig.7). Ceci fait, le code sera très aisément transposable à une application KDE (cf. widgetkde sur le Site) Nous partons d'un widget QWidget qui sert de conteneur à deux QLineEdit et deux QPushButton. Nous créons un slot qui sera connecté au signal du premier bouton. Dans le corps de ce slot, nous modifions le texte du premier QLineEdit puis nous émettons un signal de notre cru, c'est à dire déclaré dans notre classe. Ce signal est connecté à un slot du second QLineEdit, afin que son texte soit lui aussi mis à jour. Nous procédons ainsi pour les besoins de la démonstration, car dans ce cas il eut été plus direct d'invoquer la méthode setText du second QLineEdit :-). Un dernier mot pour vous éviter des déboires : toujours bien faire attention aux concordances de types entre paramètres de méthodes signal et paramètres de méthodes slot. Bonne programmation avec Qt/KDE.

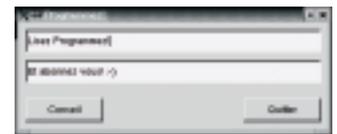


Fig.7 : Notre application Qt est maintenant enrichie et fort avisée :-)

encadré 1

Code d'une application QT de base

```
#ifndef HAVE_CONFIG_H
#include <config.h>
#endif

#include <qapplication.h>
#include <qwidget.h>
#include <qpushbutton.h>

int main(int argc, char **argv)
```

```
{
    QApplication App(argc, argv);
    QPushButton pb("Programmez!", NULL);

    App.setMainWidget(&pb);
    pb.setFixedSize(300, 200);
    pb.setCaption("Programmez!");
    pb.show();

    return App.exec();
}
```

encadré 2

Définition de notre widget personnalisé

```
#ifndef MONWIDGET_H
#define MONWIDGET_H

#include <qapplication.h>
#include <qwidget.h>
#include <qpushbutton.h>
#include <qlineedit.h>

class MonWidget : public QWidget
{
    Q_OBJECT

public:
    MonWidget(QWidget *parent=0);
    ~MonWidget();

signals:
    void updateInfo(const QString&);

public slots:
    void conseiller();

protected:
    void initWidget();
    QLineEdit *qle1;
    QLineEdit *qle2;
    QPushButton *bouton_conseil;
    QPushButton *bouton_quitter;

private:
};

#endif
```

encadré 3

Implémentation de notre widget personnalisé

```
#include <monwidget.h>

MonWidget::MonWidget(QWidget *parent) : QWidget(parent)
```

```
{
    initWidget();

    connect(bouton_quitter, SIGNAL(clicked()), qApp, SLOT(quit()));
    connect(bouton_conseil, SIGNAL(clicked()), this, SLOT(conseiller()));
    // Attention aux types
    connect(this, SIGNAL(updateInfo(const QString&)),
            qle2, SLOT(setText(const QString&)));
}

MonWidget::~MonWidget()
{}

void MonWidget::conseiller()
{
    qle1->setText("Lisez Programmez!");
    emit updateInfo("Et abonnez vous! :)");
}

void MonWidget::initWidget()
{
    this->setGeometry(0, 0, 400, 140);
    this->setMinimumSize(0, 0);
    this->setFixedSize(400, 140);

    qle1 = new QLineEdit(this);
    qle1->setGeometry(10, 10, 380, 30);
    qle1->setMinimumSize(0, 0);
    qle1->setText("Pour un conseil, cliquez");

    qle2 = new QLineEdit(this);
    qle2->setGeometry(10,50, 380, 30);
    qle2->setMinimumSize(0, 0);
    qle2->setText("");

    bouton_conseil = new QPushButton(this);
    bouton_conseil->setGeometry(10,100, 100, 30);
    bouton_conseil->setMinimumSize(0, 0);
    bouton_conseil->setText("Conseil");

    bouton_quitter = new QPushButton(this);
    bouton_quitter->setGeometry(290,100, 100, 30);
    bouton_quitter->setMinimumSize(0, 0);
    bouton_quitter->setText("Quitter");
}
```

■ Frédéric Mazué - fmazue@programmez.com