

Programmez!

Mensuel - Janvier 2005 - N°71 • 5,95 €

INTÉGRATION

Nouvelle génération

DÉVELOPPER GRATUITEMENT

C'est possible !

"PHP reste le langage le plus accessible"

Rencontre avec Rasmus Lerdof et Zeev Suraski, les "pères fondateurs" de PHP



Votre CARTE DE VŒUX est aussi un JEU !



STUDIO S&P/BOC

M 04319 - 71 - F : 5,95 €
Printed in France - Imprimé en France
REGISTRÉ 648 € - SUISSE 12 175
LUXEMBOURG 648 € - Canada 6,99 \$ CAN

TECHNO
XUL-XAML le futur de l'interface
Spring et Hibernate : alternative aux EJB CMP

C++
Interfaces graphiques avec FLTK
Maîtrisez la barre des tâches de Windows

JAVA
Manipulation de fichiers
PHP
Programmez en PHP 5 APPLICATION
Delphi 2005

Un Delphi relevé pour .Net

Découvrons ensemble Delphi 2005, la dernière mouture du produit mascotte de Borland, dans son imposante édition Architecte.

2005, cela représente dix années de Delphi. En 1995 Delphi était le meilleur. Son seul concurrent était Visual Basic. Aujourd'hui, le paysage informatique s'est considérablement enrichi et Delphi trouve un concurrent redoutable en Visual Studio .Net. Alors nous nous sommes demandés si dans un tel contexte Delphi avait une chance de s'imposer face à un adversaire sur son propre terrain, ou même simplement s'il avait encore une chance d'exister. Et c'est avec cette idée en tête que nous avons examiné le produit.

Un IDE à multiples facettes

Delphi 2005 reste Delphi. Il est toujours possible de développer de "simples" applications natives Win32 avec lui et avec la fameuse librairie de composants graphiques, la VCL. Mais Delphi 2005, c'est aussi du développement Delphi pour .Net et ASP et même du développement C# et ASP. Développement visuel dans tous les cas, avec une palette de composants sensibles au contexte et des concepteurs graphiques en pagaille, y compris pour HTML, JavaScript, etc. (Fig.1)

Un éditeur ultra productif

Comme avec JBuilder 2005, que nous vous avons présenté récemment, l'éditeur de code de Delphi 2005 a été considérablement remanié pour accroître encore le confort de travail. Nous avons particulièrement remarqué deux fonctionnalités baptisées SyncEdit et HelpInsight. La première consiste en ceci : vous sélectionnez un bloc de code d'un coup de souris. Si le bloc contient plusieurs instances d'un même sym-

bole, un bouton apparaît dans la gouttière. Un clic sur le bouton enclenche la fonctionnalité : la modification d'une instance d'un symbole se répercute sur toutes les autres instances. Quel développeur n'appréciera pas ? En ce qui concerne HelpInsight, laissez traîner le pointeur de souris sur un symbole. Apparaît alors une bulle d'aide décrivant ce dernier : paramètre reçu, type retourné, appartenance à un espace de nom, etc. (Fig.2). La plupart des infos sont des liens vers la documentation. L'éditeur est expert en refactoring. L'extraction de chaînes de ressource mérite d'être notée, quoique valable pour Delphi seulement, puisque C# ne connaît pas le bloc resourcestrng. Une autre grosse nouveauté de l'éditeur est son gestionnaire d'historique, capable de travailler en local ou avec StarTeam. Par les descriptions de ces quelques détails nous espérons avoir fait sentir que malgré son look très .Net, Delphi 2005 se démarque par une quantité de petits plus qui, pris globalement, le rendent supérieur au quotidien

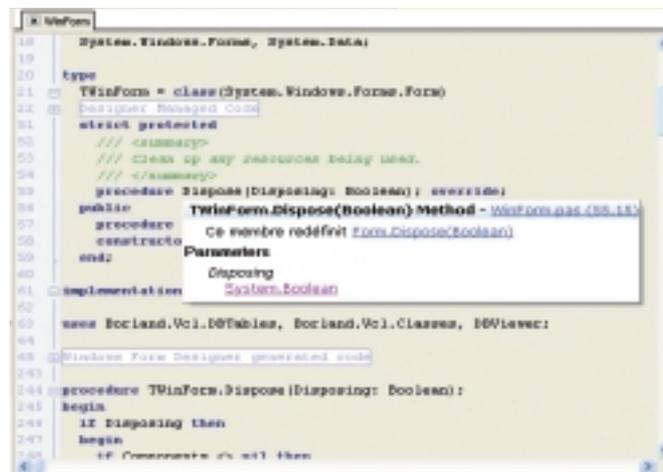


Fig. 2 : Des bulles d'aides fûtées.

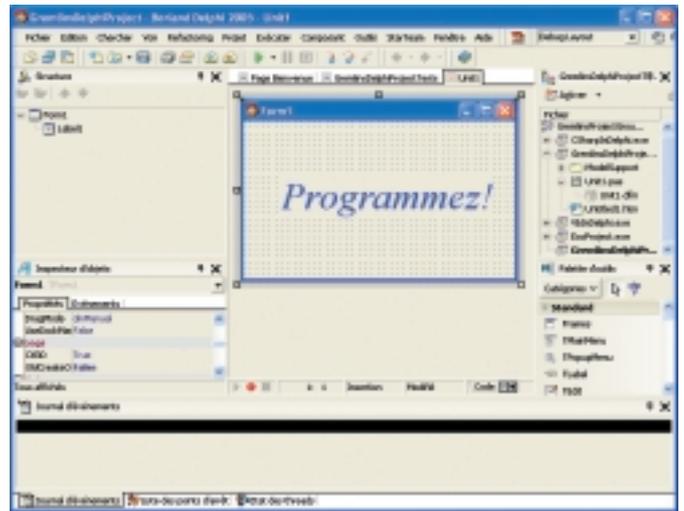


Fig. 1 : Vue d'ensemble de Delphi 2005

et sans manifester la lourdeur qui est toujours à craindre avec de tels poids lourds.

Une pléthore de composants logiciels

Ecrire du code c'est bien, ne pas en écrire, c'est souvent mieux. Dans cette optique, ce qui frappe est l'incroyable richesse de la palette de composants, qui permet de mettre en place la charpente de n'importe quel type d'application en quelques clics de souris. Mention spéciale pour les bases de données qui sont très nombreuses à être supportées, dont MySQL (composants

dbExpress seulement) et pour les composants DataSync et DataHub qui assouplissent pertinemment le travail avec ADO. DataHub et DataSync se positionnent entre le DataSet et les pourvoyeurs de données, la base de données étant distante ou non. Mention spéciale encore pour l'intégration d'IntraWeb.

Conclusion

Delphi 2005 sait faire tout ce que fait Visual Studio .Net, mais il sait faire souvent beaucoup plus, ou avec plus de brio, et apporte des outils de haut niveau tels que : composants logiciels, connexion aux bases de données facilitées, outils de déploiement, modélisation d'applications ECO, contrôle du cycle de vie, intégration d'outils tiers, dont IntraWeb et les outils de tests Dunit et Nunit, tout en conservant ce qui a bâti sa réputation des débuts. Cela donne à penser que c'est Delphi 2005 qui intègre avec habileté la plate-forme .Net plus que le contraire. Et cela fait de lui un outil de développement remarquable qui a encore de l'avenir.

■ Frédéric Mazué

fmazu@programmez.com

PowerBuilder 10 : XML et modélisation

Sybase n'édite pas seulement des bases de données. Les dernières sorties produits nous rappellent que l'éditeur possède un IDE performant ainsi qu'un outil de modélisation. Aujourd'hui, nous allons lever le voile sur PowerBuilder 10.

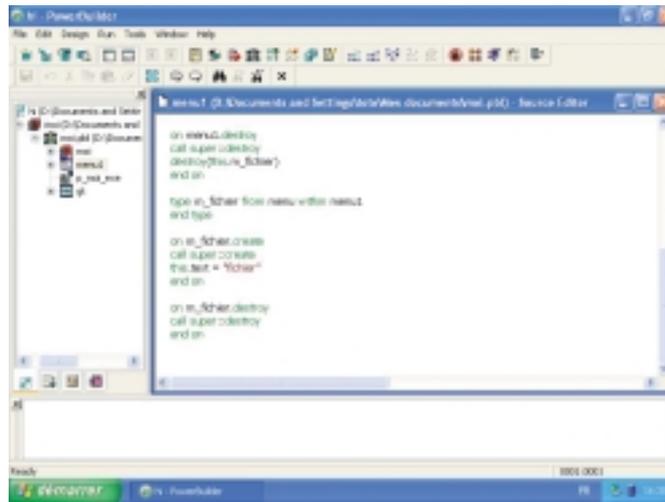
Dédié à l'origine au développement d'applications client / serveur, PowerBuilder a su prendre en compte les applications Web (et le client riche) ainsi que l'architecture 3 tiers, même si l'application C/S demeure le point fort de l'outil. Pour un développeur VB ou Jbuilder, l'ergonomie générale rappelle quelque peu les IDE connus. Cependant, PowerBuilder diffère sur la structure des projets. On définit tout d'abord un workspace, puis, on crée un projet PB, puis enfin, on ajoute les objets spécifiques voulus. Le choix est varié et répond bien aux orientations de l'outil (Web, SGBD, applications, etc.). Pour les non-experts de PowerBuilder, il faudra un petit temps d'adaptation, il faut acquiescer les réflexes.

Un RAD simple, mais efficace

On dispose d'un RAD intégré. Les objets standard sont assez peu nombreux (les principaux sont présents : label, grid, barres de déplacement, etc.). Pour tester rapidement une interface, on dispose d'une fonction Preview (menu Design). À partir de l'onglet Layout (la partie RAD), on accède à différents onglets liés aux codes par événement, les fonctions utilisées et les variables. Si on va dans l'onglet suivant (à droite de Layout), on accèdera aux différents objets et événements à coder. On dispose d'une syntaxe colorisée, mais l'aide à la volée demeure limitée : pas de complétion, pas de vérifica-

tion à la volée. La vérification se fait lorsque l'on sort de l'événement.

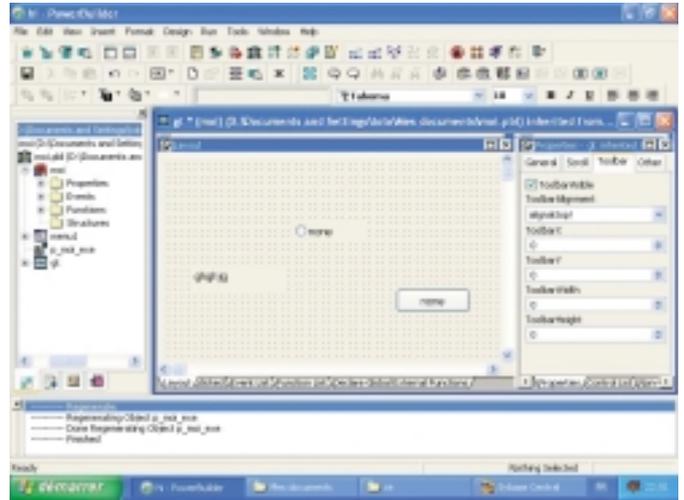
Un petit plus indéniable, la fonction To Do List. PowerBuilder indique les tâches restant à faire avant de builder et de déployer une application. Très utile pour savoir où on est. Sur les objets d'interface, il est possible d'importer des ActiveX et objets OLE. On ne dispose pas de fonctions de travail en équipe ou de versionning. Cependant, PowerBuilder s'intègre avec des outils tiers (ex. : StarTeam de Borland). On dispose tout de même d'une fonction source



control permettant de "partager" les sources d'un projet.

XML et Web Service

Initié depuis la v9, PowerBuilder intègre encore mieux tout ce qui est XML et Web Services, notamment via les projets. On peut maintenant exposer rapidement un



code PowerBuilder sous forme d'un service Web. De plus, avec l'intégration du UDDI, on peut directement dans l'environnement chercher un service (dans la version précédente, il fallait sortir de PowerBuilder pour rechercher la référence du service).

Parmi les autres nouveautés, on

MobilLink. PowerBuilder autorise le contrôle de MobilLink et la création de script de synchronisation. Si Sybase n'a pas intégré de module de modélisation, on peut ajouter un plug-in PowerAMC. Il est ainsi possible de faire générer le modèle UML à partir de son code. Bien entendu, on dispose d'une interaction : si on touche au modèle, le code se modifie et vice versa. Bref, une version attrayante pour les développeurs PowerBuilder pour faire évoluer leur application. Pour les autres, la migration vers l'outil Sybase à partir d'un JBuilder, Eclipse ou Visual Studio, sera justifiée pour des projets spécifiques.

■ F.T.

Fiche technique

Produit : PowerBuilder 10
Éditeur : Sybase
Les versions : Desktop à 279 euros par utilisateur, Professional à 1225 euros par utilisateur, Enterprise à 2833 euros par utilisateur

Points forts :
 ergonomie générale, performances, UDDI natif, ADO.NET, refactoring, plug-in PowerAMC

Points faibles :
 aide au codage parfois limitée, pas de véritable versionning, travail en équipe

Fnac Music mise sur l'intégration et la mutualisation des flux

Lancé en septembre 2004, le nouveau site de téléchargement musical de la Fnac, se distingue par une plate-forme orientée services. Une architecture sophistiquée, mise en place en quelques mois seulement.

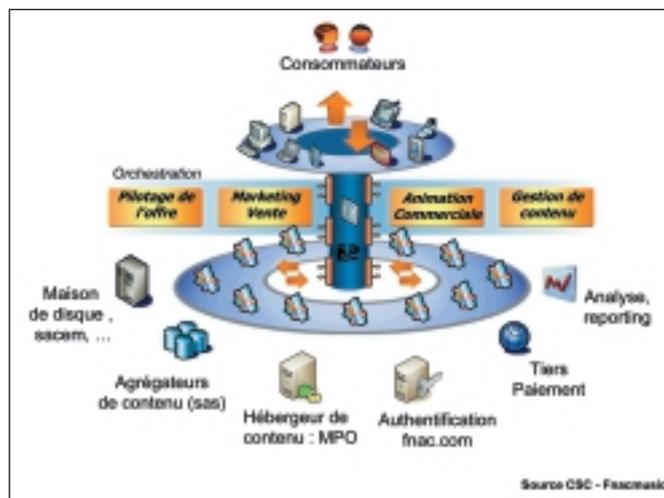
Pour faire face à la déferlante de sites de téléchargements musicaux, la Fnac a décidé d'être aussi un acteur de poids avec le projet Fnacmusic. « Depuis longtemps, nous proposons des services de téléchargements de musique à travers l'option DigiFnac du site Fnac.com » déclare Alexandre Blaizot, directeur de projet, à la Fnac, et de poursuivre : « Mais, notre souci, avec DigiFnac, est que nous travaillions en marque blanche et que nous ne maîtrisions pas complètement cette activité. Il fallait que nous possédions notre propre site de téléchargement. » Fin 2003, la direction de la Fnac lance un appel d'offres, à l'issue duquel, elle retient CSC, pour l'intégration du projet, Axance en tant que Web Agency pour le design du site et MPO On line pour la numérisation des titres fournis par les distributeurs et la gestion des droits d'auteur. « L'objectif de cette plate-forme est de proposer aux internautes un site musical sécurisé, où ils peuvent télécharger le titre de leur choix en toute légalité, en achetant un droit de licence à partir d'un tarif unique » résume Alexandre Blaizot. Le chantier est de taille. CSC participe en tant que maître d'œuvre à la définition du périmètre fonctionnel du projet. « Il fallait envisager toutes les possibilités de l'acquisition et la numérisation de la musique, en passant par les formats de fichiers musicaux, jusqu'aux métadonnées accompagnant chaque fichier

musical, le tout dans une approche totalement sécurisée » résume Gilles le Caro, chef de projet chez CSC. De son côté, la Fnac se chargeait en parallèle des aspects juridiques et des premières négociations avec les distributeurs de musique, majors, et labels indépendants.

Personnalisation, Intégration et Interopérabilité

La plate-forme est orientée SOA, avec comme bus d'échanges de données centrales, l'EAI Biztalk Server 2004, et en middle office Commerce Server pour la gestion

Visual Studio ont été nécessaires pour répondre à nos attentes, en particulier pour la gestion des catalogues musicaux » remarque Alexandre Blaizot. Le front office intègre pour sa part Microsoft IIS. « Un besoin pour la FNAC, explique Gilles le Caro, concernait la modularité et le support à terme des différents terminaux pour la consultation du site. Une révision de l'architecture du front office, effectuée conjointement avec les experts de Microsoft a conduit au développement d'une solution basée sur les templated controls d'ASP.NET. L'optimisation



des catalogues et le CRM (profil des internautes, play list, recommandations des adhérents, etc.). « Malgré les avantages d'un progiciel (compacité et facilité d'intégration de la solution), Commerce Server ne répondait pas intégralement à nos besoins métier. Des développements spécifiques en

de la granularité et des fonctionnalités de ces contrôles s'est avérée délicate, mais le résultat fut satisfaisant. » Autre difficulté : l'acquisition des titres provenant de MPO on line. « Nous recevions des métadonnées, avec des formats très hétérogènes et différents selon les maisons de



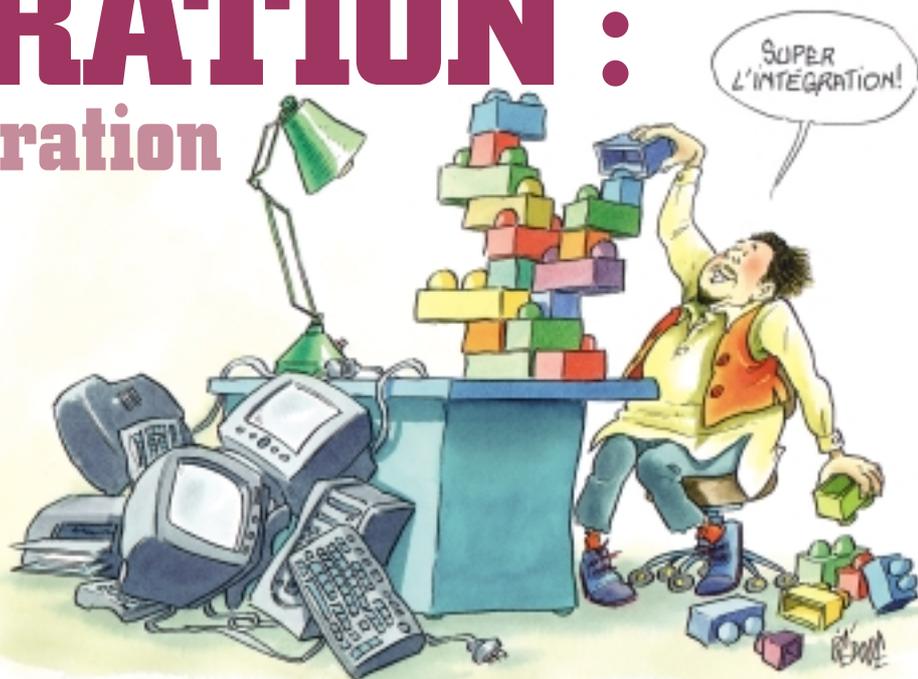
disques (auteur, titre, genre, mais aussi les jaquettes, les extraits, la classification, etc). Ces métadonnées étaient essentielles pour s'assurer de la parfaite identification des morceaux choisis et de l'homogénéité de leur présentation vers la bonne nomenclature » explique Alexandre Blaizot. CSC a dû développer en . Net « un sas » de réception de données pour filtrer les métadonnées, avant de les envoyer en préproduction. Autre point important : l'authentification des internautes. Pour leur éviter de saisir plusieurs fois leur mot de passe, mais aussi pour des raisons de trafic, de visibilité et de marque, Fnacmusic s'appuie sur le site de Fnac.com. « Cette interopérabilité avec les deux sites, permet de capitaliser sur les process d'authentification de Fnac.com, mais aussi de mutualiser les informations » note Alexandre Blaizot. Après neuf mois de développement, le site propose désormais plus de 300 000 titres et a été inauguré à la date prévue. « Ce projet a été un enrichissement intellectuel à tous les niveaux, qu'il s'agisse du domaine juridique, des technologies utilisées, ou des relations avec les majors. Tous nos partenaires ont travaillé avec beaucoup d'enthousiasme, une motivation, qui a été un facteur déterminant pour la réussite de ce site » conclut Alexandre Blaizot.

■ Annie Lichtner

INTEGRATION :

nouvelle génération

Pour ce dossier, nous avons voulu privilégier plusieurs approches : l'intégration avec l'Open Source, l'intégration des données, les nouvelles formes d'intégration, le délicat choix entre intégration, migration, mixage, la suite du combat EAI - Web Services.



Intégration, migration, mixage : le choix cornélien

L'intégration, la migration, le mixage, chacun de ces choix possède des avantages et des inconvénients. Tout dépend de la stratégie. Dans tous les cas, le coût n'est jamais nul en terme de développement, de management et d'outils et ne croyez pas échapper aux coûts avec des outils Open Source. Dès que l'on souhaite intégrer des systèmes hétérogènes provenant de plusieurs sites / départe-

ment, avec de multiples applications et données, mieux vaut utiliser des solutions dédiées. Cela facilite le développement et la mise en place de l'intégration, de plus, vous bénéficierez de la gestion de processus. Gestion indispensable dans une mise en œuvre complexe de Web Service ou orienté service. Plus votre intégration est complexe, plus votre architecture le sera, avec une véritable plomberie pour faire

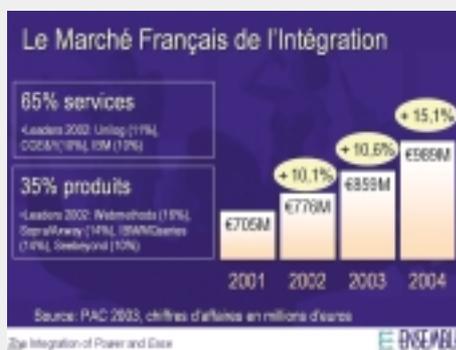
tout cela. Si grâce à XML et consorts, on utilise des standards, ce n'est pas pour autant que l'on simplifie forcément l'intégration.

L'intégration lâche : le concept d'avenir ?

Aujourd'hui, l'EAI ne constitue plus une solution miracle, tant s'en faut. Trop lourd, trop complexe, trop cher, l'EAI voulait intégrer les appli-

Un marché en pleine évolution et consolidation

Selon une étude portant sur les années 2004 à 2009, les éditeurs historiques du marché devraient voir leurs positions confortées, voire augmentées : IBM, Tibco, WebMethods, BEA, SeeBeyond. Cependant, le marché du BPM croît très rapidement, tendance que l'on observe depuis 2003. Les analystes prévoient aussi une évolution de l'offre, avec un nouveau niveau de fonctionnalités, l'EAI propriétaire est désormais bel et bien enterré. Une étude d'IDC, datant de 2003, prévoit pour 2005 un marché de l'EAI de 116 milliards de dollars. La croissance devrait être assurée par le marché financier, l'eBusiness ou encore le supply chain. Pour la



France, Pierre Audoin Consultant prévoyait pour 2004 un marché d'environ 989 millions d'euros pour presque 300 environnements d'EAI. Si on suit les conclusions du Syntec d'octobre 2004, le marché de l'intégration aurait augmenté de 2 % (projet et intégration, croissance 1er semestre), avec une tendance de croissance pour le futur. Avec la structuration des solutions Open Source / Libre d'EAI, BPM et ESB, on peut s'attendre dans les prochaines années à un bouleversement du marché, et chez les éditeurs "traditionnels". L'arrivée de nouveaux acteurs dans certains marchés prometteurs (XML, Web Services, SOA, BPM) sera aussi un autre facteur d'évolution important.

“Notre Credo, c’est le standard”

Pierre Olivier Chotard – BEA France

Programmez ! : Existe-t-il de nouvelles formes d’intégration ? SOA est-elle la solution miracle ?

P.O. Chotard : Non, je n’en vois pas. Chez BEA, il s’agit de pouvoir avec du SOA mettre les applications sous forme de services, avec des échanges de données dans un format standard, dans un moteur de processus. Le SOA est une tendance que l’on voit chez nos clients, via les appels d’offres. Il y avait de gros projets avec beaucoup d’argent il y a quelques années (avec les solutions EAI, ndlr). Ils n’ont pas abouti, ou alors le résultat n’est pas celui escompté. Les clients sont devenus plus pragmatiques, ils demandent à voir si la solution fonctionne. Les éditeurs d’EAI ont dû tout re-développer, ce qui n’est jamais simple. Notre credo c’est le standard. Le choix du standard est important. BEA n’a jamais fait d’EAI. Avec le SOA, on cible les 200-300 plus grandes entreprises françaises. Même dans une grosse PME, l’approche SOA est difficile à mettre en œuvre.

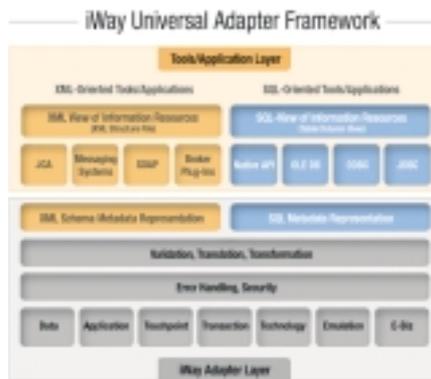
Programmez ! : Ne vaut-il pas mieux migrer qu’intégrer ?

P.O. Chotard : Encore faut-il en avoir les moyens. C’est très complexe. Le patrimoine applicatif est très varié. L’intégration permet d’isoler certains (aspects) métiers et d’appliquer des standards.

cations à partir d’une tour de contrôle. Dans des environnements très hétérogènes de plusieurs filiales, l’EAI constitue un frein. Désormais, il faut de la souplesse pour une meilleure productivité de son SI et avoir un découplage suffisamment lâche entre toutes les couches et applications (et données) pour pouvoir adapter rapidement son nouveau SI. Et avec un dogme écrit en gros : le métier de l’entreprise avant la technique.

Intégrer aujourd’hui, c’est faire communiquer plusieurs pôles techniques/business : le front office (portail, applications web, mobile, etc.), le back-office (serveurs en tout genre avec les applications), les données. À cela peuvent s’ajouter les applications et données distribuées des autres services et filiales distribuées géographiquement ou non. Donc en fait, on a une tour de contrôle d’intégration lâche qui est là pour synchroniser et gérer l’ensemble, ensuite, l’intégration elle-même, ce sont des composants services, des flux XML, éventuellement des Web Services, du middleware technique/métier, du client riche ou léger selon le terminal cible. La notion d’intégration lâche est importante. Il faut impérativement garder le maximum de souplesse et d’indépendance en optant pour des standards reconnus du marché, sans même parler obligatoirement de SOA, bien trop lourd à implémenter dans les petites structures.

Ainsi, vous gardez votre legacy, vos données, vos applications VB, Java, etc. vous pouvez exposer le tout dans des applications web, des services sans aucun problème, et vos flux reposent sur du standard XML et des protocoles reconnus (IP, SOAP, etc.). Finalement, les frontières entre intégration, migration et mixage sont désormais de plus en plus floues, car, l’in-



tégration lâche est un compromis entre les 3 concepts. Côté données pures, l’intégration lâche de données se fera, via de la fédération et/ou de la virtualisation, mais là, attention, il faut disposer de SGBD critiques, hors de question, de faire cela avec du MySQL ! Malgré tout, l’intégration lâche est sans doute valable pour des architectures peu complexes.

Les nouvelles formes d’intégration

SOA, l’architecture orientée service

L’architecture service est le nouveau terme à la mode chez les éditeurs et les media. L’intérêt du SOA est de mettre le métier, le business, le service avant la technique (et non le contraire). En théorie, SOA constitue donc une réelle avancée, et sa stratégie d’intégration peut tendre vers cette architecture. En pratique, les freins sont nombreux : compétences XML / Web Services pas toujours adaptés, maîtrise et connaissance des services et des processus métiers de l’entreprise, outils compatibles. SOA demeure réservée aux grands comptes et aux projets conséquents. Cependant, pour une petite structure maîtrisant son métier, pourquoi pas, à condition

Minipanorama des outils

Voici une petite liste ci-contre (non exhaustive) des outils disponibles sur le marché. Le choix est très large, et ce, dans tous les domaines de l’intégration.

Les solutions d’EAI

E*Gate SeeBeyond	Nouvelle version orientée XML – Web Service
iBolt Magic	Nouvelle version. Propose un IDE supportant J2EE / .NET. Inclut divers outils (portail, serveur, moniteur).
Tibco	Un des acteurs historiques, inclut une centaine de connecteurs et un kit de développement
WebSphere Business Integrator IBM	Très large panoplie de connecteurs, suite logicielle très variée
Biztalk Server Microsoft	" tout-en-un ", support XML – Web Service, plus de 200 connecteurs
Sterling Integrator Sterling	Support XML
EntireX Mediator Software AG	EAI XML
WebLogic Integration BEA	Web Services, nombreux connecteurs en standard
Enterprise Integrator Iona	Solution très appropriée pour intégration des applications Corba, support web service
Ensemble InterSystems	Inclut un serveur, un SGBD et le développement. Supporte J2EE, XML, .NET
DataExchanger Server DataExchanger	EAI basé sur la conception de scénario (données, BPM, XML, messaging, etc.)
Nosica OpenWide	Solution Open Source, support XML. Fonctionnalités limitées
Integration Server WebMethods	Implémente outil de BPM, un kit de développement

Les solutions ESB / BPM

Sonic ESB Sonic Software	Anciennement SonicXQ. Un acteur historique de l’ESB. Propose aussi une approche SOA
Artix Extensible ESB Iona	Tout comme Sonic, Iona propose un ESB extensible avec une approche SOA, Web Service.
CapeClear 6 Cape Clear Software	Environnement ESB avec support BPEL, Web Service, possibilité de SOA
JetStream (WebSphere 6) IBM	Outil ESB dans la version 6 de WebSphere
Oracle BPEL Process Manager Oracle	Rachat de l’outil BPM de l’éditeur Collaxa
Mega Process Mega	Export des processus vers Microsoft, webMethods, Iona...
W4 (avec rachat de Akazi)	Solutions de gestion de BPM, avec notamment l’outil FlowMind

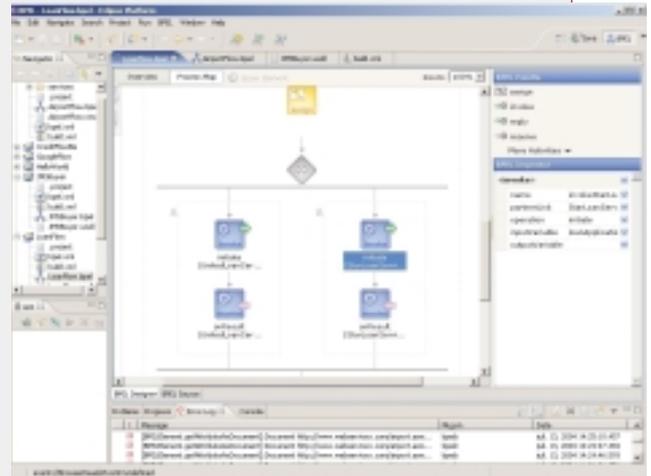
Migration

La migration est la solution la plus radicale et définitive, car difficile de revenir en arrière, une fois le processus enclenché. La migration permet de remettre à neuf données et applications. Vous pouvez, lors d'une réécriture des applications et de la structure des données, en profiter pour remettre à plat l'ensemble de votre SI, votre architecture.

La migration permettra dans ce cas présent de : redéfinir l'architecture serveur (logiciel + matériel), d'homogénéiser les langages / les interfaces / les outils de développement, découpler le contenu du contenant (idéal dans une politique multi-terminal), de s'appuyer sur des standards du marché, appliquer de l'orienté métier et non technique, mettre en place du couplage mou afin d'avoir une flexibilité dans les applications, les couches et de pouvoir changer rapidement des règles métiers ou de middleware. Cependant, attention. La migration nécessite une maîtrise de son applicatif, de ses données. Vous devez aussi, dans le cadre d'un couplage mou et d'un orienté métier, parfaitement maîtriser votre environnement et les aspects métiers de chaque département, de chaque service. D'autre part, la migration nécessite une compétence technique forte dans l'ancienne technologie et dans la nouvelle, afin de minimiser les coûts de transition, de développement et surtout de pouvoir mettre un planning "acceptable" sans perte fonctionnelle et business.

Oracle s'investit dans l'intégration

L'éditeur Oracle s'est récemment investi sur le marché de l'intégration.

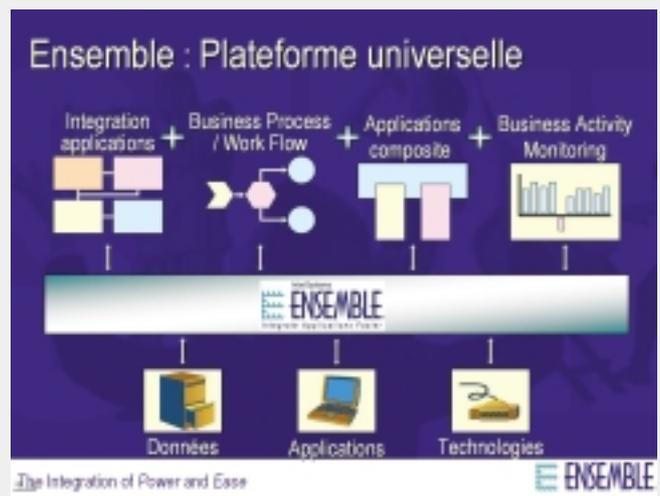


Depuis quelques mois, il possède une offre de BPM avec Oracle BPEL (grâce au rachat de Collaxa, spécialiste du domaine). Il s'agit de pouvoir orchestrer et lancer des Web Services et des processus métiers. Oracle Integration est une plateforme d'intégration de données, d'applications d'entreprise, de collaboration B2B, d'applications composites et de services web. Oracle Integration repose sur un serveur d'applications auquel s'ajoutent diverses fonctions. Il implémente aussi l'architecture SOA (qui est présentée comme le successeur de l'EAI par Oracle). Dans cette solution d'intégration, on dis-

pose aussi de EDI, RosettaNet, ebXML, HIPAA, HL7. Une console BAM est présente. Oracle est disponible seul ou dans le package Oracle Application Server Enterprise Edition. Enfin, Oracle Data Hub s'occupe de la synchronisation de l'information à partir d'un point central de l'ensemble des systèmes de l'entreprise. Oracle parle d'une vue à 360° des données d'entreprise. Il s'agit de manipuler une seule et unique source de données (cohérente et de qualité). Actuellement, seule la version Customer Data Hub est disponible, les autres déclinaisons seront disponibles courant 2005.

InterSystems & L'intégration

L'éditeur InterSystems connu pour son SGBD Caché possède aussi une solution d'intégration universelle lancée il y a peine un an (novembre 2003). L'offre répond à trois besoins : une intégration de données et d'applications avec orchestration de processus métiers, un environnement de développement (J2EE, .NET, Web Services, etc.), des fonctions de disponibilité et de montée en charge. Ensemble propose une intégration de bout en bout avec des fonctions de serveurs, de données, de BPM et ce, en prenant en compte les principales plates-formes du marché et les standards actuels. Il possède même depuis quelques mois d'un module BAM. Ce BAM permet d'avoir une vue globale de l'environnement d'intégration au niveau des flux et processus. Ensemble s'accroche aussi à la base Caché. Comme tout bon EAI, Ensemble vient avec une multitude de connecteurs pour les données, les progiciels, les systèmes transactionnels, les terminaux, les protocoles et les technologies / langages. Il fonctionne principalement sous Windows, Linux et Solaris. Ensemble est la réponse d'InterSystems à la réduction des coûts, en disposant d'un EAI global couvrant les besoins. Cette globalité de fonctionnalité dans un



unique environnement est un avantage certain sur des offres plus classiques, disposant de plusieurs outils non intégrés, pouvant multiplier ainsi les coûts, les procédures de contrôles et de développements et synchronisation entre les outils dédiés.

Site : <http://www.intersystems.fr/AccueilEnsemble.htm>

Passer d'un langage événementiel à de l'orienté objet (ex. : de VB6 à C# ou VB.NET), ne va pas forcément de soi. Dans le cadre d'applications web critiques, la migration sera d'autant plus sensible que l'on touchera au cœur de l'activité. Si vous changez de plate-forme (d'ASP / JSP à PHP, de Java à .NET, etc.), là aussi, la réécriture n'ira pas forcément de soi. Un audit de l'existant permettra d'ajuster le projet de migration. La migration est une solution radicale, difficilement réversible. Si vous avez un lourd existant ou un existant dont vous n'avez plus le code source ou un manque de données techniques, la migration est un avantage et un inconvénient. Avantage, car vous aurez une base fonctionnelle neuve, un inconvénient, car la migration des applications et du code risque d'être plus laborieuse. La migration est possible, uniquement dans le cadre d'une maîtrise parfaite de vos applications et données. Dans certains cas, la migration sera difficile, voire impossible. Par exemple, dans les SGBD, il n'y a pas d'équivalent à Oracle dans le monde libre pour les fonctions avancées ou les très grosses bases... Il faut être pragmatique dans les choix et éviter absolument toute précipitation. Encore une fois : audit, maquettage réel, étude d'impacts, etc.

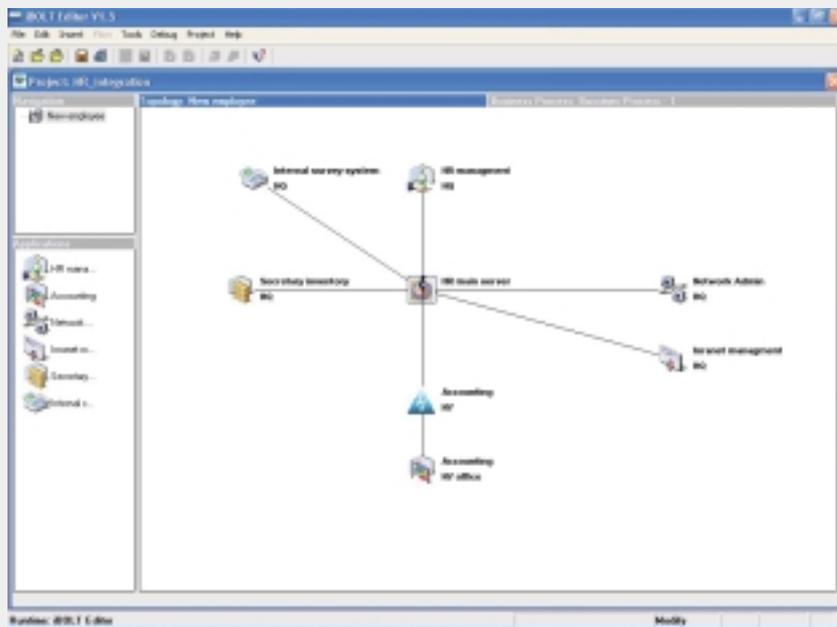
Mixer intégration et migration

Le mixage d'intégration et de migration répond à deux objectifs. Préserver un cœur applicatif ou de données difficilement migrable en un bloc pour diverses raisons (manque du code, de compétence, aspects critiques, coûts de la migration, etc.). Le mixage concerne aussi bien le legacy que les applications/données plus récentes, mais souvent monolithiques, ayant un couplage fort et offrant une souplesse quasi impossible. L'avantage du mixage est de ne pas tout casser. Il permet d'introduire une souplesse de développement et d'utilisation appréciable. Dans ce cas, souvent, on aura un front office neuf avec un back-office peu modifié, avec au milieu un middleware en couple mou, tout en profitant d'un environnement plus homogène. Très pratique dans le cadre d'applications web, d'exposition en services, d'une architecture SOA ou tout simplement pour préparer une future migration totale. Cependant, il faut définir l'environnement à migrer, l'environnement à garder. Le choix de la nouvelle architecture demande une profonde réflexion de fond (Java, .NET...).

■ François Tonic

Magic Software monte en puissance sur l'EAI

Lancée il y a moins de deux ans, l'offre EAI de Magic Software s'enrichit au pas de course. Pour la version 2 d'iBolt annoncée en novembre 2004, l'éditeur soigne tout particulièrement les développeurs.



Intégré à l'IDE iBolt Studio, le nouveau Data Mapper Service d'iBolt 2 permet de relier les bases sources aux bases ou applications cibles dans une interface complètement graphique. Il simplifie également la configuration des process de l'outil d'EAI. " Notre objectif est de permettre aux développeurs de poser des composants et de les assembler sans programmation, explique Alain Dubois, directeur marketing France de Magic Software, et ainsi de faciliter et accélérer la mise en œuvre d'infrastructures EAI dans les entreprises ".

Pour les fonctions avancées ou personnalisées, le développeur dispose désormais d'interfaces XML sur les composants qui facilitent leur contrôle et leur paramétrage. Ils ont également la possibilité de concevoir leurs propres composants en utilisant les IDE J2EE du marché, Visual Studio.NET de Microsoft ou même eDeveloper, l'environnement RAD de Magic Software, iBolt étant compatible avec les trois technologies. Côté administration et supervision de l'exécution des processus, iBolt a également enrichi ses fonctionnalités de méca-

nismes de roll back. Critiques dans le cadre de l'EAI lorsqu'une transaction échoue, ces mécanismes ont gagné en précision, iBolt étant par exemple capable de repérer à quel moment un processus a échoué dans une transaction globale et de ne le reprendre qu'à partir du point défectueux. Enfin, l'éditeur a également renforcé les performances de son outil BAM (Business Activity Monitoring) qui enregistre les activités de l'EAI, afin d'offrir ensuite des vues métiers sur les processus. Combinées aux outils de reporting intégrés à iBolt, ces fonctions regroupées au sein d'une console permettent, par exemple, de savoir combien de transactions ont été réalisées dans le cadre des ventes sur un produit donné.

■ Marie Varandat

iBolt 2

Prix : à partir de 65 000 euros ht.

Concurrents : plus de 600 acteurs sur le marché de l'EAI/BPM/BAM dont : WebMethods, Seebeyond, Tibco, Iona, IBM, Microsoft, etc.

Intégration de données

ETL et EII : de faux concurrents

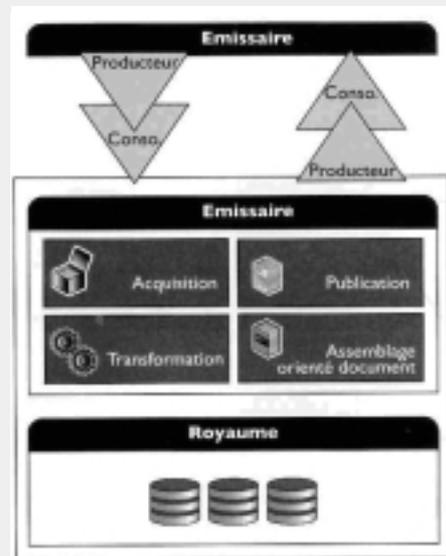
Souvent présentés comme des concurrents, ETL et EII partagent certes des mécanismes similaires, mais n'ont pas la même vocation. Tandis que le premier déplace des données pour les consolider dans une nouvelle base en monde asynchrone, le second facilite l'accès à des sources hétérogènes en temps réel. Mais pas plus l'un que l'autre ne prend en charge la problématique sous-jacente à tout projet d'intégration de données : la qualité.

Projet Web, décisionnels, évolutions d'applications existantes, création d'une vraie plate-forme de gestion de la relation client, mais également mises en conformité avec les nouvelles réglementations telles que les normes IAS et Bâle II... les problématiques d'intégration de données se multiplient. Isolées dans des applications existantes disposant de leurs propres bases et surtout de modèles de données spécifiques, les informations des entreprises sont en effet éparpillées dans le système d'information. Or, la plupart des nouvelles applications agrègent des informations provenant de sources hétérogènes pour constituer des vues métiers telles qu'un dossier client, une fiche produit ou encore, un historique de commandes. Il existe aujourd'hui deux principales catégories de solutions sur le marché qui répondent à cette problématique d'intégration : les ETL (Extraction, Transformation and Loading) et les EII (Enterprise Information Integration).

Les premiers sont issus de l'univers du décisionnel et alimentent en mode asynchrone les datawarehouses. Les seconds, plus récents, ont des mécanismes similaires à ceux des ETL, mais fonctionnent en temps réel. Leur vocation est moins de déplacer des données vers une nouvelle base, bien qu'ils puissent le faire, que de fédérer les accès au travers de vues qui agrègent des sources de données hétérogènes. Bien que partageant des mécanismes d'autant plus similaires que la plupart des ETL sont aujourd'hui dotés de fonctions d'EII, ETL et EII répondent donc à des problématiques d'intégration des données différentes.

Partage d'un vocabulaire commun en entrée/sortie

L'intégration de données doit reposer sur une véritable politique de gouvernance des données pour être efficace et rentable sur le long terme. En élaborant un modèle de données unique, l'entreprise réduit en effet ses besoins en développement d'interfaces spécifiques, tout en augmentant la qualité de ses données. Ce modèle de données unique ne signifie pas que l'en-



entreprise doive faire table rase sur son existant mais simplement mettre en place des mécanismes au sein de chaque application (royaume) pour assembler, transformer et publier les données dans un format compris de l'ensemble du système d'information (émissaire).

(source Octo Technology – Livre blanc sur «Architecture de système d'information – Gouvernance de la donnée»)

ETL, l'intégration en mode batch

Le mécanisme d'un ETL est très simple : il extrait des données d'une source, les transforme pour les adapter au modèle de données de la base cible et les charge ensuite dans la nouvelle base, selon un «job» (ou ensemble de processus) généralement modélisé dans une interface graphique. La transformation peut comprendre plusieurs étapes, dont notamment la vérification d'intégrité, la génération d'alertes si la donnée n'est pas cohérente par rapport à des critères préalablement définis, etc. Au passage, l'ETL stocke des informations sur les données (ou métadonnées) dans un référentiel : provenance, transformation effectuée, etc. Certains ETL peuvent également enrichir leur référentiel en interrogeant les dictionnaires de métadonnées des applications sources et ainsi agréger un historique complet de la donnée. Historique particulièrement important pour certaines applications telles que mise en œuvre des normes comptables IAS, pour lesquelles l'entreprise doit justifier de la provenance de ses données. Enfin, l'ETL ne peut être dissocié de ses connecteurs qui lui permettent de se greffer sur les sources de données. Longtemps propriétaires, ils sont aujourd'hui complétés par une technologie d'intégration plus généraliste : les Web Services. Ascential et Informatica restent aujourd'hui les deux principaux acteurs indépendants du marché de l'ETL, mais on en trouve également dans les offres d'éditeurs du décisionnel, dont Cognos (Decision Stream) et Business Object (Actaworks).

	Sécurité	Accès	Services	Données	Processus
Analyste	Spécifications de la sécurité : identification, habilitations...	Spécifications de l'interface homme machine	Spécifications des services externes, des services internes selon la complexité	Spécifications des données publiques et des données privées	Cas d'usage internes et externes
Développeur	Fondations techniques sécurité : annuaire, module d'authentification ...	Fondations techniques IHM : gestion des formulaire, des enchaînements, des contrôles...	Fondations techniques des traitements : concurrence, gestion des erreurs...	Fondations techniques de la persistance : lecture, écriture, concurrence...	Fondations techniques des échanges : emballage, émission, réception...
Exploitant	Déploiement physique du module de sécurité	Environnement d'exécution client (PC...)	Environnement d'exécution serveur (Unix...)	Serveur de stockage (cluster + SAN...)	Serveurs d'échange

Le développeur au coeur du projet d'intégration de données
(source : livre blanc « Architecture de système d'information – Gouvernance de la donnée » d'Octo Technology, téléchargeable gratuitement sur http://www.octo.com/fr/techno/wp_gouv.html)

Conflit d'identité	Le même objet possède des identités différentes selon les sources de données. Le client X est identifié par la clé X1 dans un référentiel et X2 dans un autre.
Conflit de schémas	Le même concept est modélisé de manière différente. - Noms de champs différents - Constructions différentes Le client est modélisé par le champ « customer » d'un côté, et par les deux champs « nom » et « prénom » par ailleurs.
Conflit sémantique	Le même concept est interprété de manières différentes. Un « compte » est un compte courant bancaire dans une application et un identifiant utilisateur permettant de gérer la sécurité des accès pour une autre.
Conflit de valeur	Le même objet a des valeurs différentes selon les sources. La balance d'un compte courant apparaît différente suivant les bases dans lesquelles on le consulte.

Classification des principaux conflits dans un projet d'intégration de données liés à l'interopérabilité et à la qualité des données. (source Octo Technology)

Les EII tentent de faire en temps réel ce que l'ETL fait en asynchrone

L'EII, pour sa part, se comporte plus comme un hub de données qui centralise et expose des points d'accès homogènes en temps réel dans une vue, en effectuant un mapping similaire à celui de l'ETL. Il emprunte en quelque sorte à l'EAI sa faculté d'extraire des informations en temps réel et à l'ETL sa capacité à consolider ces données. Son domaine de prédilection :

l'optimisation des requêtes distribuées en environnement hétérogène. Très proche du concept d'API, la vue accepte en effet différents formats de requête : SQL, XQuery, JCA, Soap, etc. IBM avec DB2 II (Information Integrator) et BEA avec Liquid Data, sont aujourd'hui les grands leaders de ce marché. Là encore, le mécanisme est simple : il suffit de définir les sources d'information et de récupérer les champs de tables ou autres éléments XML, l'ensemble des paramétrages étant effectué au sein d'interfaces graphiques.

L'EII peine sur les performances

Plus adapté aux requêtes multi-sources qu'à la constitution de datawarehouses, l'EII répond à des problématiques de mapping en temps réel, mais également de performances pures. Ainsi, un grand laboratoire français utilise DB2 II pour accéder à des bases Oracle réparties sur la planète. Objectif : fédérer les informations sur les travaux de recherche réalisés au sein de ses différentes entités afin d'avoir une vue globale. Faire des accès à plusieurs sources de données avec une seule requête SQL tout en préservant les performances, nécessite un tuning très fin. En s'appuyant sur DB2 II, ce laboratoire utilise une des particularités du produit qui est de faire faire une grosse partie du travail au moteur de la source. DB2 II se comporte en effet comme un client Oracle, fait exécuter la requête à la base et, plutôt que de rapatrier des tables, transfère un résultat directement exploitable par l'application.

Reste qu'en voulant réaliser en temps réel des opérations souvent complexes, l'EII peine. Masquée, sa mécanique s'appuie en effet sur un moteur SQL/XML qui optimise la distribution et la parallélisation des requêtes SQL et XQuery et calcule le meilleur plan d'exécution en fonction des informations techniques présentes dans la description de chaque source de données stockée. La génération d'outils actuelle a encore du mal à optimiser efficacement l'ordre des requêtes, la création de jointures par la source ou par l'outil, etc. Et c'est précisément pour monter en puissance que la dernière version de DB2 II intègre deux évolutions majeures : parallélisation et surveillance des sources de données.

Un mixte du meilleur des deux mondes

Une problématique à laquelle l'ETL n'est pas confrontée, en raison de son fonctionnement en mode batch, et à laquelle l'éditeur français Sunopsis répond par un stockage intermédiaire dans un format pivot. À mi-chemin entre l'ETL, l'EAI et l'EII, son offre s'appuie sur un bus orienté données, doté de fonctions de nettoyage et de transformation des données. Comme DB2 II, Sunopsis fait faire les transformations au moteur de la source, mais stocke ensuite le résultat dans une base (DB2, Oracle, SQL Server, etc.) baptisée DataStore, sous forme de vue accessible par une application.

Principaux acteurs du marché de l'EII et de l'ETL

Editeur	Produit	Catégorie	Caractéristiques
Actuate	Nimble Integration Suite (rachat Nimble)	EII	Désormais intégrée à l'offre de reporting d'Actuate, Nimble Integration Suite s'articule autour d'un référentiel de données XML et s'appuie sur les Web Services pour le requêtage.
Ascential	DataStage	ETL	Acteur traditionnel de l'ETL, Ascential est désormais capable de faire l'intégration en temps réel et dispose d'une offre en amont pour gérer la qualité des données et enrichir les dictionnaires de méta-données. Plus de 40 connecteurs natifs vers des sources de données, dont IBM/Informix, Oracle, Sybase, Teradata et IBM DB2. Package complet dédié à SAP et à la collection de modules MySAP.
BEA	BEA Liquid Data for WebLogic	EII	Interfaces : JDBC (Java Database Connectivity), l'architecture de connexion JCA (Java Connector Architecture) ou encore les Web Services (interfaces SOAP/WSDL).
IBM	DB2 Information Integrator	EII	JDBC, JCA et Web Services. Requêtage XQuery, parallélisation et surveillance de la source des données dans la dernière version.
Informatica	PowerCenter	ETL	Offre très complète pour cet éditeur, qui au cours de cette dernière année, s'est complètement refocalisé sur l'intégration de données, avec en amont une offre très riche pour gérer la qualité et les méta-données. Gamme très vaste de connecteurs spécifiques aux sources de données pour consolider tous les principaux entrepôts de données.
iWay	iWay XML Transformation Engine (iXTE)	EII	
Hummingbird	Genio Suite MetaBase Sunopsis V4	ETL	Moins connue que certains de ses concurrents, l'offre d'Hummingbird repose sur le rachat de l'ETL de Genio. Connecteurs pour Oracle, Sybase, Teradata, Hyperion Essbase, MS SQL Server et IBM DB2. Prise en charge nouvelle des formats de données : XML, mainframe, SAP en natif, binaires, etc. En EAI: Siebel, SAP, support de MQ Series.
Metamatrix		EII	Système de gestion de requêtes multisources (basé sur des connecteurs). Repose sur un socle Java avec un outil de développement adapté à la création des meta-modèles de données (ou référentiels) chargés de structurer les accès vers les bases sous-jacentes.
Sunopsis		EII/ETL	Plate-forme complète pour intégrer et gérer la qualité des données. Dans la version 4, l'interface a été améliorée, ainsi que la gestion des versions (jobs). Une version d'évaluation gratuite peut être téléchargée sur le site de l'éditeur.

Au passage, tout DataStore peut être explosé sous forme de Web services, afin de faciliter l'intégration avec d'autres applications.

L'intégration de données vise la qualité

Quel que soit l'outil adopté, ETL et EII ne résolvent toutefois qu'une partie du problème de l'intégration de données. Au fil des années, l'entreprise s'est en effet laissée dicter des modèles de données par des applications commerciales ou spécifiques. Résultat, entre un progiciel de gestion intégré, une solution de gestion de la relation client et un développement sur mesure, elle dispose de trois définitions de ce que peut être un chiffre d'affaires : avant ou après amortissement, brut, net, etc. La première phase d'un projet d'intégration des données est donc souvent consacrée à la gestion de la « qualité » des données. Qualité, qui a un impact direct sur la performance d'une entreprise : selon une étude commanditée par SAS à l'institut d'analyse KRC Research, les trois quarts des entreprises françaises estiment en effet que leur rentabilité est affectée significativement par des données fausses, incomplètes ou incohérentes.

La prise de conscience du phénomène chez Pierre Fabre a donné naissance à une remise à plat des processus et concepts utilisés par le groupe pharmaceutique avec, pour seuls outils, le papier et le crayon. Les concepts ont ensuite été référencés dans une base Oracle à laquelle toutes les applications de l'entreprise sont asservies. En d'autres termes, quelle que soit l'application, elle partage la même définition de ce qu'est un client, une facture, une commande, etc. Pour Pierre Pezziardi, directeur technique d'Octo Technology et co-auteur du livre blanc « Architecture de système d'information – Gouvernance de la donnée », ce modèle unique de données sous forme de charte dans laquelle l'entreprise arrête tous les concepts, est indispensable pour mener à bien et sur le long terme un projet d'intégration de données : « L'idée est de l'utiliser pour remplacer des interfaces existantes ou en créer de nouvelles. À terme, on imagine que chaque bloc du système gèrera sa production et sa consommation de données « publiques » dans un seul module, que nous appellerons émissaire. Le royaume, derrière l'émissaire, exploitera pour longtemps des données privatives et hors gouvernance : l'objectif n'est pas de refondre les modèles de données des applications, mais seulement d'exiger d'elles un vocabulaire commun en entrée/sortie ».

■ Marie Varandat

HP passe de 1500 datawarehouses à une dizaine

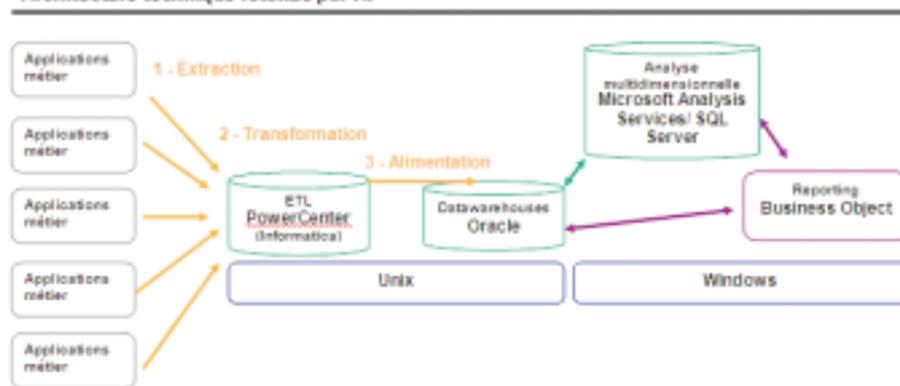
Le géant mondial de l'informatique a décidé de rationaliser et uniformiser sa plate-forme décisionnelle et, pour l'alimenter, vient de créer un infrastructure mondiale afin d'intégrer ses données. Un projet colossal et critique pour cette société qui attribue autant d'importance à son reporting qu'à son progiciel de gestion intégrée.

Il y a 4 ans, le géant informatique HP arrivait à un constat malheureusement très fréquent dans le domaine du décisionnel : les projets échouaient. Le groupe a donc entrepris une réflexion au niveau mondial pour comprendre la raison de ces échecs et est parvenu à la conclusion suivante : « pour réussir, nous, techniques, devons balayer devant notre porte et offrir aux fonctionnels une vision claire des données, explique Yves Cabanac, membre de la direction du système européen d'HP. Or, nous avons un peu de tout dans tous les sens, les données étaient éparpillées dans 1500 datawarehouses et autres ODS, etc. Nous avons donc cherché à rationaliser tout ce qui pouvait l'être, tout en uniformisant notre plate-forme technique. Aujourd'hui, nous n'avons plus que 50 datawarehouses et notre objectif est de parvenir à une dizaine en 2007, avec un seul hub de données pour toute l'Europe. C'est un projet capital, car cette infrastructure de données sert de socle à tout le reporting réalisé au sein d'HP. Or, le reporting est devenu une fonction aussi critique pour notre activité qu'un SAP, sans compter que la rationalisation nous a permis de réaliser des économies significatives ».

Remplacer au fur et à mesure des besoins

Dans un premier temps, HP a arrêté une plate-forme technique en optant pour l'ETL PowerCenter d'Informatica, le moteur OLAP de Microsoft sur des bases de données Oracle/Unix et des outils de reporting de Business Object ainsi qu'un modèle de données commun, s'appuyant en partie sur ce qui existait déjà, a été arrêté pour l'ensemble du groupe. « C'est probablement une des parties les plus complexes d'un projet d'intégration de données, précise Yves Cabanac. Les données sont utiles aux fonctionnels, elles doivent donc être organisées en adéquation avec leurs

Architecture technique retenue par HP



besoins». HP a opté pour une approche pragmatique en remplaçant les applications existantes au fur et à mesure des besoins d'évolution. Chaque nouveau besoin donne naissance à la création d'un job dans PowerCenter. « Nous gérons à peu près tous les types de flux de données qui existent : CSV, XML, fichiers plats, etc., précise Yves Cabanac. Aujourd'hui, nous disposons déjà de la plus grosse infrastructure d'intégration de données jamais créée de par le monde et elle ne cesse de s'enrichir ».

Trois millions de dollars tous les six mois pour faire évoluer l'architecture

Dès le début du projet, HP a mis en place un référentiel destiné à la documentation sur les données et les processus afin de faciliter la compréhension et la réutilisation des développements. Critique pour la qualité des données, ce référentiel a posé des problèmes : « Nous avons un peu buté sur le versionning et travaillé avec Informatica sur PowerCenter pour améliorer ces fonctions capitales pour un projet de la taille du notre, explique Yves Cabanac, qui reconnaît que le travail réalisé au cours de ces trois dernières années est colossal « mais il est également très payant : notre

performance technique était misérable. Pour 100 dollars dépensés autrefois en développement, nous n'en dépensons plus que 20 aujourd'hui. Ce qui ne nous empêche pas d'investir trois millions de dollars tous les six mois en développement pour parvenir aux objectifs que nous nous sommes fixés ».

En rationalisant son infrastructure d'intégration de ses données, HP a également réduit ses besoins en compétences diverses pour les ramener aux seuls outils du groupe. L'opération s'est également concrétisée par 2,5 millions de dollars d'économies sur le support dans les deux dernières années. Enfin, l'expérience d'HP a donné naissance à une initiative conjointe avec Cap Gemini. Baptisée BI Factory, elle comprend un framework, ainsi qu'un ensemble de méthodologies pour mener à bien les projets de Business Intelligence.

■ Marie Varandat

Fiche société HP

Activité : solutions technologiques destinées au grand public, aux collectivités et aux entreprises.

Nombre d'employés : 142 000

CA : 73,1 milliards de dollars en 2003

Site : <http://www.hp.com/>

Skandia adopte le modèle de données unifié

CAS D'ENTREPRISE

La filiale française de l'assureur suédois a mis en œuvre un bus orienté données avant de déployer ses applications. Grâce à cette approche qui préserve par ailleurs sa capacité d'évolution, elle dispose d'un système ouvert, capable d'intégrer facilement les données sans problème de qualité majeur.

Premier assureur suédois, Skandia dispose d'une structure en France depuis juin 2002 qui a eu la chance de partir de zéro pour bâtir son système d'information. Chance dont la société a profité pour mettre en place une politique d'intégration des données, basée sur la qualité. « Nous avons posé la route avant de mettre les villes, souligne Bertrand d'Argentré, Directeur Informatique et gestion clients de Skandia France, ou en d'autres termes, avant de poser la première brique applicative, nous avons souhaité mettre en place un bus d'intégration, et ce, afin d'éviter par la suite tout problème lié à la qualité des données. Concrètement, nous avons pensé notre système d'information en termes d'urbanisation, afin de pouvoir évoluer beaucoup plus vite ou même changer une brique sans que cela remette en cause l'infrastructure générale ».



Bertrand d'Argentré, Directeur Informatique et gestion clients de Skandia France

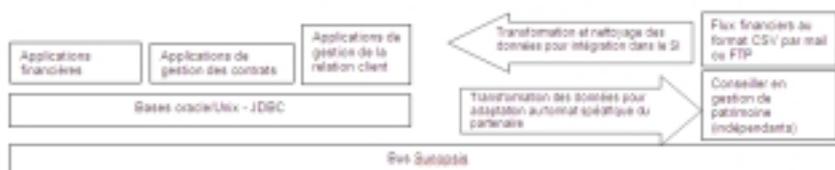
Composé de trois principaux volets — applications financières, gestion des contrats et animation du réseau de partenaires —, le système d'information s'appuie sur le bus de l'éditeur Sunopsis pour l'intégration des données. Plutôt que d'arrêter un modèle de données unifié et d'y assujettir toutes les applications de son système, Skandia s'est basé sur les modèles de données des applications. Ainsi, dès le départ la société a adopté le

Chaque type de donnée existe en version primaire dans une seule application

« Un objet existe en version primaire dans une seule application, explique Régis Portugal, ingénieur développement chez Skandia. Nous évitons ainsi les problèmes d'incohérence et de qualité des données ».



Régis Portugal, ingénieur développement chez Skandia



Un seul bus pour l'urbanisation interne et les échanges avec l'extérieur

même identifiant pour chaque pan de son système. La définition du concept client, par exemple, est fournie par le logiciel de gestion de la relation client, Selligent, et est imposée à toutes les autres applications. Le système de gestion de contrats, impose pour sa part la définition du concept contrat. « Un objet existe en version primaire dans une seule application, explique Régis Portugal, ingénieur développement chez Skandia. Nous évitons ainsi les problèmes d'incohérence et de qualité des données ».

Une vue centralisée sur tous les échanges de données

En adoptant un bus orienté données, Skandia a voulu éviter le « syndrome spaghetti ». Grâce à la plate-forme Sunopsis, la société dispose en effet d'une vue centralisée de tous les échanges de données et ils sont nombreux. Outre la communication applicative au sein de son système d'information, la société récupère tous les jours des flux financiers par mail ou FTP (fichiers plats de type CSV) qu'elle traite (vérification et transformation pour adapter au modèle interne) pour ensuite les intégrer à ses bases de données sous Oracle/Unix. Elle extrait également de ses bases des flux destinés à ses partenaires qui sont transformés automatiquement dans le format adhoc, spécifique à chaque conseiller en gestion de patrimoine indépendant, par Sunopsis.

Trois mois de mise en œuvre

« La mise en œuvre d'une telle approche de son système d'information implique de sérieuses compétences en développement et en bases de données, explique Régis Portugal,

sans compter une bonne appréhension de la philosophie du produit de Skandia. Ceci étant, à part les requêtes SQL, tous les mécanismes d'intégration sont masqués dans Sunopsis et la réalisation de scripts en Java, langage utilisé par Sunopsis, n'est nécessaire que lorsqu'on veut personnaliser les jobs. Pour le reste, tout peut être modélisé dans des interfaces graphiques ». Entreprise en février 2003, la mise en œuvre du bus d'intégration était terminée trois mois plus tard. Choisie pour son approche originale, à cheval sur les concepts d'EAI, d'EII et d'ETL, la plate-forme de Sunopsis alimente également un datawarehouse (Discoverer d'Oracle) qui sert de socle aux applications décisionnelles de la société. «Sunopsis est un outil très complet qui permet de faire de l'EAI orienté données. Il s'appuie en outre sur la puissance du système d'information existant pour faire exécuter les traitements aux applications sources, approche qui permet d'optimiser les performances. Enfin, c'est une solution ouverte, et ce d'autant plus qu'elle s'appuie sur java, donc sur des standards qui sont également une garantie de pérennité» conclut Bertrand d'Argentré et Régis Portugal.

■ Marie Varandat

Fiche société Skandia

Activité : assureur spécialisé dans le domaine de gestion de patrimoine (assurance vie)

Nombre d'employés : 33 personnes en France, 5 500 personnes dans le monde

CA France : 50 millions d'euros en 2003 (9,89 milliards d'Euros monde en 2003)

Site : <http://www.skandia.fr>

Web Services vs EAI



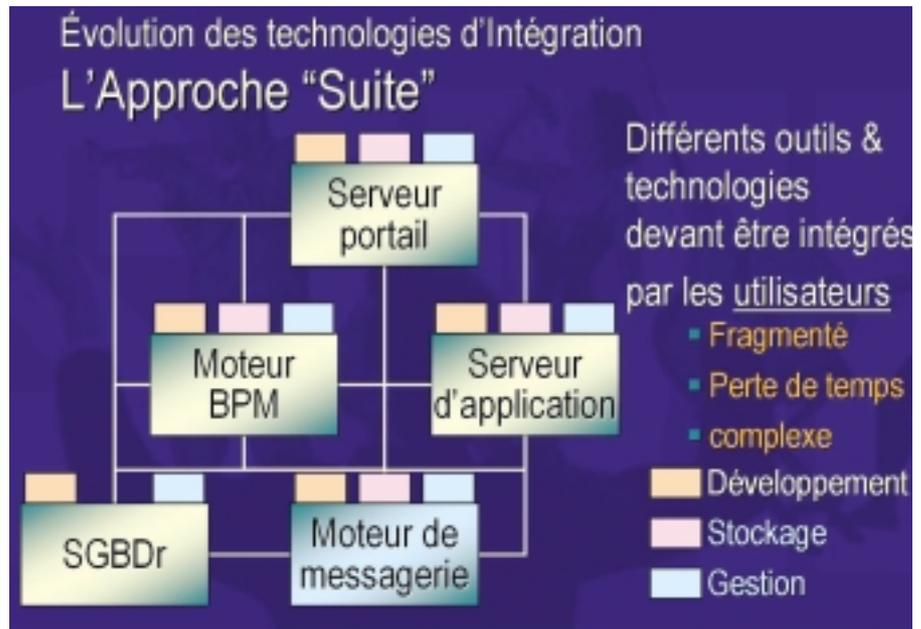
L'EAI (Enterprise Application Integration) est un concept apparu à la fin des années 90 qui consiste à interconnecter les applications qui composent le système d'information. Jusqu'alors, les échanges entre applications s'effectuaient par l'intermédiaire de batchs exécutés la nuit et qui se chargeaient de garantir la cohérence des informations entre les applications, par exemple, recopier dans l'application comptable les nouveaux clients saisis du jour, ou bien synchroniser l'ERP avec la gestion des stocks.

A l'époque, les applications étaient monolithiques et assez difficilement inter opérables. Les solutions EAI offraient (sur le papier) tous les mécanismes nécessaires pour effectuer ces échanges en continu, au fil de l'eau, plutôt que dans un mode batch. Parmi les services offerts par l'EAI on peut noter, la capacité à traiter les principaux formats d'échange de données, des transformateurs de documents (XML vers TXT, TXT vers EDI, XML vers XML, ...) et des moteurs de workflow capables d'appliquer une logique métier aux échanges inter-applicatifs.

Le moins que l'on puisse dire est que le bilan de la première génération d'outils EAI est assez mitigé. Les attentes déraisonnables qu'ils ont suscitées, combinées à des tarifs prohibitifs, ont eu assez rapidement raison des "super projets" à l'échelle de l'entreprise. Ce sont finalement les petits projets réalisés au niveau départemental, ou bien encore ceux destinés à remplacer des échanges EDI simples qui ont eu les meilleurs taux de réussite. Une des raisons de ce relatif échec est que les applications en place n'étaient tout simplement pas prêtes pour être "branchées" sur ces super moteurs d'échanges. Les causes en sont multiples, mais on retiendra surtout leur conception monolithique et des barrières technologiques coûteuses à franchir.

Changer notre conception des applications

L'apparition des architectures orientées services (SOA) et la convergence des protocoles d'échanges vers SOAP laissent penser que ces obstacles sont en train d'être levés. En effet, la mise en place d'applications utilisant des protocoles standardisés pour la mise à disposition de services, avec une granularité plus fine qu'auparavant, ouvre la voie à la création de véritables processus métier s'exécutant dans

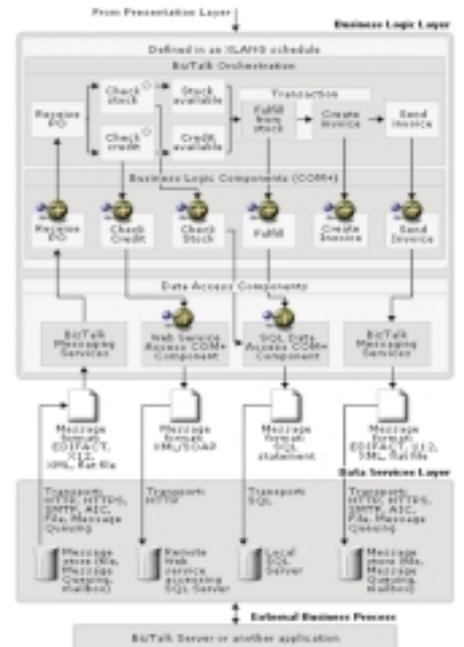


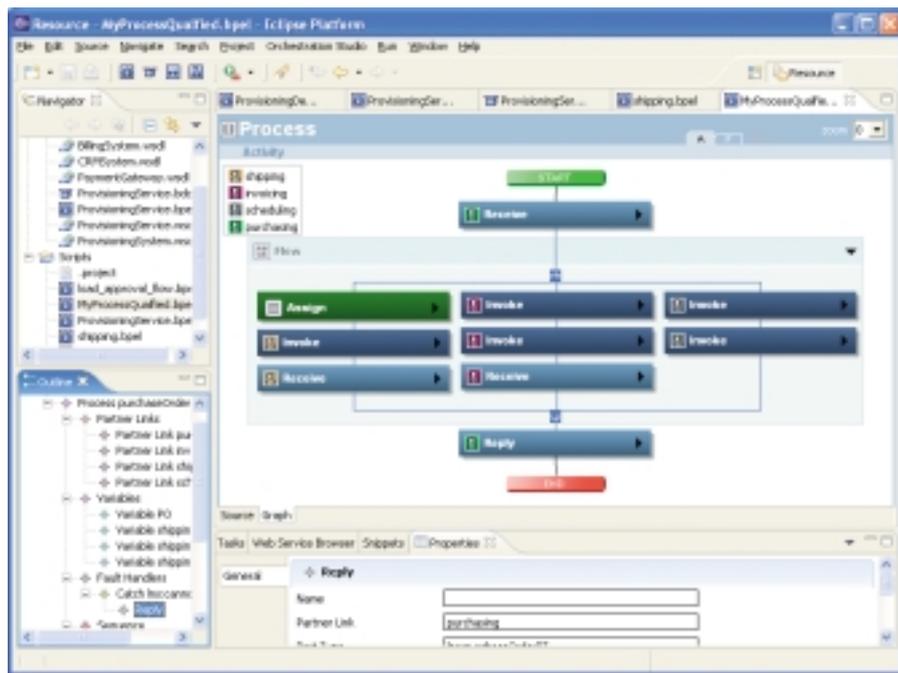
The Integration of Power and Ease



plusieurs applications. Pour en arriver là, il faut commencer par changer la façon dont nous concevons les applications et passer d'un monde où l'application est autonome par rapport au reste du SI, à un monde où elle doit en permanence collaborer avec d'autres applicatifs, soit en consommant leurs services soit en leur en proposant. Cette phase est délicate car il faut trouver la bonne granularité de découpage : trop grande et on retombe dans les travers de l'EAI, trop petite et les problèmes de performance sont certains. Car il ne faut pas l'oublier, on est ici en environnement distribué et les accès réseau ont un coût, dont il faut tenir compte.

Ce qui différencie les Web Services des EAI, c'est essentiellement l'utilisation du protocole de transport HTTP/HTTPS, considéré comme un standard, capable de passer les firewalls et utilisable sur tous les types de réseaux, y com-





L'orchestration selon Capesience

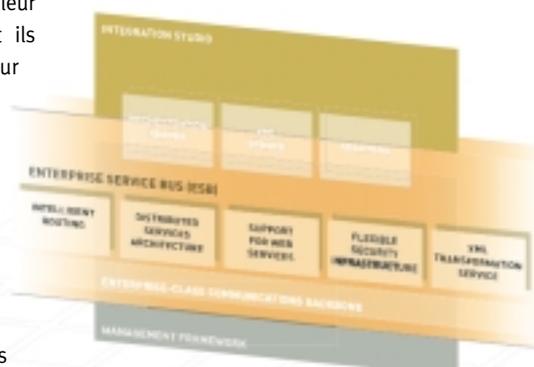
pris WAN. Les solutions d'EAI quant à elles imposent généralement un middleware orienté message (MOM) pour le transport des informations. Ceux-ci disposent de fonctionnalités évoluées telles que la persistance des messages, ou la gestion des transactions qui font toujours défaut aux Web Services. Mais leur utilisation est limitée au réseau local et ils peuvent difficilement être branchés sur Internet pour des raisons de sécurité. Les solutions d'EAI conservent néanmoins tout leur intérêt dans les environnements informatiques où les applications " legacy " occupent une place importante. On désigne par ce terme les applications anciennes qui font partie du " patrimoine applicatif " de l'entreprise. Généralement développées avec des technologies " démodées " de type Cobol ou PowerBuilder qui ne se prêtent pas à un découpage en service, elles s'intégreront beaucoup plus facilement avec le reste du SI à l'aide d'une solution EAI qu'avec des Web Services.

Les Web Services n'apportent pas toutes les réponses

Alors, quelle est la place des Web Services dans ce nouvel ensemble ? Ils ont assurément un important rôle à jouer de liant entre les différents composants utilisés par un processus. C'est la seule technologie qui puisse aujourd'hui

garantir une totale interopérabilité, que ce soit en terme de plates-formes ou de langages de programmation.

Mais est-ce, pour autant, que les Web Services répondent à tous les besoins pour l'intégration des applications ? Sûrement pas, les Web



Architecture du bus ESB de Sonic

Services, tels que nous les connaissons aujourd'hui, ne répondent qu'à la problématique du transport des messages entre applications. Ils n'apportent aucune réponse dans des domaines tels que l'orchestration des échanges (les aspects transactionnels), la sécurité applicative, où encore la gestion des processus longs (problématique de sérialisation de processus). Pour tous ces aspects il est nécessaire de faire appel à des solutions basées sur des technologies propriétaires, même si des standards commencent à émerger, on peut citer par exemples BPEL dans le

domaine de l'orchestration des processus. La standardisation des services additionnels sous les appellations WS-XXXX va dans le sens de l'ouverture. Mais on touche là à des solutions dont les implémentations sont dans le meilleur des cas en phase de bêta. Le produit Indigo de Microsoft étant probablement celui le plus abouti à ce jour.

A plus court terme, ce sont surtout les solutions de BPM (Business Process Management) qui vont bénéficier de l'émergence des premières solutions SOA à l'échelle du SI. Elles vont (enfin) trouver des applications découpées dès le départ en services et donc pouvoir donner toute leur mesure.

Industrialiser les infrastructures logicielles

Enfin, il ne faut pas oublier que la diffusion de solutions basées sur les Web Services et des orchestrateurs BPM va accroître fortement les dépendances entre applications. Elle renforcera le besoin d'industrialiser les infrastructures logicielles (Middlewares, SGBDR, moteurs BPM, annuaires, solutions de clustering ...) pour assurer la qualité de service attendue par les utilisateurs en terme de disponibilité et de capacité de montée en charge des applications. Alors, quelle technologie choisir pour conduire l'intégration du SI ? De toute évidence, les

Web Services répondent surtout aujourd'hui à la problématique du transport de l'information, alors que les solutions EAI sont performantes dans les domaines de la transformation de données, du routage de messages et de l'orchestration des appels. Celles-ci intègrent toutes, aujourd'hui, des connecteurs capables de consommer ou de mettre des Web Services à disposition des autres applications. Ceux qui recherchent uniquement un moyen de communication inter-applicatif se tourneront vers les Web Services, les autres les utiliseront pour le transport et orchestreront les échanges avec une solution EAI. Loin d'être en concurrence, les deux solutions tendent à se rapprocher avec l'intégration progressive des Web Services dans les offres EAI.

■ Médéric Morel

Directeur technique de Neoxia, cabinet de conseil en architecture des systèmes d'information.

mederic.morel@neoxia.com





Adam Kolawa

Dans le cas d'une intégration dans un service web, on a souvent affaire à l'un de ces deux scénarios :

1/ Le système d'information de l'entreprise est constitué d'anciens systèmes : Pour être à même de proposer des services et partager des données avec des partenaires, consommateurs ou d'autres systèmes d'informations, la société doit actualiser ses anciens systèmes à l'aide des technologies disponibles et les publier sous forme de services web.

2/ Des middlewares sont déjà en place dans le système d'information de l'entreprise, mais ils doivent être publiés sous forme de services web.

Dans les deux cas, les développeurs ont le choix entre deux approches pour la publication : J2EE et .NET. Bien que les deux architectures proposent un grand nombre de technologies et standards, les points forts de Java semblent, à première vue, avoir plus de poids que ceux de .NET (à relativiser, ndlr). Les développeurs rencontreront toujours un certain nombre d'obstacles lorsqu'ils s'attacheront à la construction et au déploiement de services web, et ce, quelle que soit l'architecture concernée. Dans cet article, nous aborderons les différences entre les deux architectures, puis nous nous concentrerons sur les pièges que les développeurs doivent éviter lorsqu'ils publient des systèmes anciens et des middlewares.

Publier des systèmes existants

Les grandes entreprises possèdent souvent du code COBOL et C++. Elles ont beaucoup dépensé et accumulé des quantités importantes de données. Il est essentiel de préserver et réutiliser ces systèmes, les intégrer et les publier sous forme de services web. L'intégration de

Les services web : intégrer les systèmes existants

Dans un contexte où les entreprises s'orientent vers des stratégies e-business, la communication et l'intégration des systèmes d'information existants devient un enjeu majeur.

Les systèmes existants est souvent une tâche ardue lors de la construction d'un service web. Côté Java, JCA (J2EE Connector Architecture) peut être utilisée pour intégrer les anciens systèmes. Il s'agit d'une spécification permettant de créer des « adaptateurs de ressource » qui savent communiquer avec les systèmes existants. Ces adaptateurs de ressources sont réutilisables dans tous les conteneurs qui supportent JCA et actuellement, un grand nombre d'éditeurs connus sur le marché en proposent. .NET fournit une solution pour l'intégration des existants avec son Host Integration Server 2000, mais la connectivité qu'il offre avec ces systèmes est limitée, parce qu'il existe (encore) peu de solutions qui le supportent, mis à part Microsoft.

La portabilité de Java joue un rôle important dans le développement des services web. Les programmes écrits en Java sont portables sur un réseau et peuvent tourner sur différentes plates-formes et OS. C'est un avantage de taille, puisque la plupart des e-business ont des clients et consommateurs sur toutes sortes de plates-formes. La portabilité des langages .NET est, elle, au mieux limitée parce qu'ils ne tournent que sur des machines Windows. Et même si le projet Mono crée une implémentation libre de .NET, l'avenir du projet est encore incertain. (cependant, dans un web service « .NET », la portabilité est en principe bonne, ndlr).

Publier des middlewares

Une autre situation que l'on peut rencontrer, est l'intégration de middlewares sous forme de services web. La combinaison Java – services web est naturelle puisque les services communiquent avec d'autres, via des applications middlewares, et la plupart des systèmes d'entreprise d'aujourd'hui ont des middlewares écrits en Java. Si c'est le cas, publier un middleware est relativement facile en utilisant n'importe quelle librairie Java.

Des librairies tierces, comme Axis d'Apache,

sont disponibles et peuvent aider à accélérer le développement des services web. Ces librairies contiennent des routines orientées objet et des définitions de classes qui peuvent être appelées, ce qui permet d'éviter l'ajout de nouveau code ou les modifications sur l'environnement du système d'origine – un processus qui risque d'amener de nombreuses complications.

Les dangers à éviter

Malgré l'apparence simple du processus de conversion des classes Java en XML pour les services web, il n'en est pas moins plus complexe que ce que les développeurs peuvent penser. Quelle que soit l'architecture, il y a toujours un certain nombre de détails à prendre en considération lorsque l'on construit et déploie des services web.

Structure du message SOAP : Si vous essayez d'intégrer des systèmes avec des backgrounds hétérogènes, vous devez gommer les différences de structure de message SOAP. Ainsi, si vous intégrez une application Java avec une application .NET, l'implémentation Java utilisera probablement le modèle de communication RPC avec encodage SOAP, alors que l'implémentation .NET va utiliser le modèle document avec encodage littéral.

Cette différence doit être aplanie pour assurer l'interopérabilité des systèmes. Il peut être plus intéressant d'utiliser le XML littéral (donc sans encodage) que l'encodage SOAP, parce que l'encodage SOAP interfère avec la validation de schéma. On peut d'ailleurs voir dans son omission du WS-I basic profile, une indication que l'industrie est en train d'abandonner l'encodage SOAP.

La question style RPC ou style document n'est pas aussi claire. Microsoft .NET préfère le style document alors que de façon générale, les autres outils préfèrent le style RPC. Du point de vue de la validation de schéma, le style document est plus efficace, et avec l'abandon

de l'encodage SOAP, certains bénéfices du modèle RPC en sont diminués. Cependant, il est assez répandu pour que les projets d'intégration fassent souvent appel à des services RPC. Rappelez-vous que les messages de type RPC ont une « enveloppe » supplémentaire qui est le nom de l'opération.

De même, bien que les déclarations de messages en WSDL ne sont en théorie pas liées à un binding particulier et ne dépendent pas du mode choisi (RPC ou document), en pratique, il y a une forte corrélation entre la façon dont sont déclarés les messages et les binding utilisés. Le schéma habituel consiste à avoir des messages avec des parties déclarées comme éléments et prévues pour être utilisés avec le mode document, ou bien des messages avec des parties déclarées comme types et qui seront utilisés avec le mode RPC.

Sérialisation et désérialisation : C'est le processus par lequel des données sont converties d'une structure spécifique à un langage (par exemple des classes Java) vers XML qui supporte le protocole SOAP. Pour que différentes machines comprennent les traductions du sérialiseur, vous devez décider de la disposition des objets Java dans chaque machine. La signification d'un objet dans une machine peut être différente dans une autre machine. Traiter

des types primitifs est plutôt simple, mais traiter des types complexes rend l'étape de la sérialisation un peu plus délicate.

À titre d'exemple, une classe complexe peut très bien avoir plusieurs chaînes. Si deux de ces chaînes sont identiques, il faut décider de leur traitement, soit comme deux références à une chaîne, soit comme deux chaînes séparées. Or, cette décision affecte la façon dont les chaînes seront traduites de chaque côté du service web.

Notez que la distinction référence / valeur n'est disponible que dans l'encodage SOAP. Une pratique répandue en Java consiste à créer des classes bean où les méthodes accesseur suivent un schéma simple. Cela permet la réutilisation d'une paire sérialiseur/désérialiseur, réduisant ainsi la quantité de code supplémentaire à maintenir.

Mapping de types de données XML à partir de classes Java : Pour exposer des fonctions Java utilisées dans un service web, il faut les mapper. Cependant, il n'y a pas de mapping équivalent entre différentes implémentations. Une implémentation prendra une donnée Java et la transformera en date`Time` en XML. Une autre prendra cette donnée Java et en fera une autre valeur XML. Par exemple, voici des types primitifs définis en XML Schema :

```
dateTime, date, time, gYearMonth, gYear,
gMonthDay, gDay, gMonth.
```

N'importe lequel de ces types peut être mappé en classe Java `java.util.Date`. Il n'y a pas de réponse juste et des implémentations Java SOAP différentes ne mappent pas de la même façon. Autrement dit, il ne faut pas oublier que la librairie avec laquelle vous travaillez ne va pas mapper nécessairement en types XML et vice-versa.

Conclusion

L'avenir de l'e-business est sans aucun doute lié aux services web. Il est important que l'architecture des organisations qui construisent et développent des services web, ou qui s'y préparent, soit capable de supporter les services web. Java apporte ce support avec sa J2EE connector Architecture, avec sa portabilité et des librairies extensives –des éléments qui permettent tous, la traduction en service web, sans qu'il y ait d'efforts trop importants de réécriture à fournir.

■ Adam Kolawa et Marvin Gapultos

Adam Kolawa est Président et cofondateur de Parasoft. Docteur en Sciences Physiques, il est co-auteur du Bulletin proofing Web Applications (Hungry Minds 2001). Ak@parasoft.com

Intégration grand système, standards ouverts et Open Source.



Les SI sont au coeur de la réactivité des entreprises. Accompagnant les métiers de celles-ci, leurs évolutions sont rythmées par les changements de stratégies, de métiers ou d'organisations internes. Les modifications et les périodes de transition de ces systèmes sont donc fréquentes, pour ne pas dire permanentes, afin de permettre les prises en charge des nouvelles exigences. Les nombreuses évolutions de ces systèmes témoignent de l'activité informatique des dernières décennies : systèmes centralisés par lots, puis transactionnels, systèmes dépar-

tementalisés transactionnels, puis clients/serveurs, systèmes bureautiques en réseau local puis en réseaux étendus, web services en station, puis en nomade : autant de couches qui pour certaines ont disparu, mais qui pour d'autres se sont accumulées, les rendant de plus en plus complexes et hétérogènes.

La cartographie et l'Urbanisme tendent à dominer ces systèmes, et au-delà des problèmes purement techniques, à définir des blocs architecturaux, technico-fonctionnels, orientés "service". Elles impliquent une quête permanente de réutilisabilité, de flexibilité, de productivité, mais avant tout d'interopérabilité. Les mainframes, et en particulier les systèmes MVS, sont redevenus, malgré leur disparition annoncée, le centre névralgique de par leur fonction de serveurs de données, leur référence absolue en terme de stabilité, leur capacité de stockage et leur montée en charge. À contre-courant des systèmes centraux, fermés, et des solutions EAI traditionnelles, s'activent de plus en plus, de nouvelles approches, liées à l'effervescence des standards Web et en particulier des Web Services. Ces solutions ouvertes, souvent liées au monde Open Source et/ou aux technologies JAVA et XML, offrent des architectures à la fois plus légères et moins spécifiques.

Normes et interopérabilité

À l'opposé des logiciels propriétaires et fermés, les solutions Open Source et les systèmes ouverts tendent à garantir un respect plus important des normes d'interopérabilité en vigueur dans le monde de l'informatique. Cependant, ce qui semble dans un premier temps être un avantage ne l'est pas toujours. Le choix de ce type de solution passe par la

validation des normes sur lesquelles il repose.

Les organismes comme l'IETF (Internet Engineering Task Force) ou le W3C sont en charge de la réalisation et des évolutions de ces normes. De par le dynamisme des solutions libres, l'intégration de nouvelles normes est réalisée très rapidement au sein des nouvelles bibliothèques et logiciels, et ceci, parfois avant même que celles-ci ne soient figées. Le statut et la stabilité des normes utilisées, validées par ces organismes, sont fondamentaux pour éviter des problèmes majeurs.

Alors qu'il est encore difficile pour beaucoup d'intégrateurs de faire le choix des standards ouverts en milieu de production, il est pourtant impératif de s'y préparer. En effet, que ce soit pour les langages de programmation ou les systèmes d'exploitation tel GNU/Linux, il ne fait plus aucun doute, qu'aujourd'hui, poussés par des IBM, SUN et même Microsoft, les standards ouverts deviendront omniprésents et remplaceront peu à peu les systèmes fermés dont l'interopérabilité difficile ne répond plus au dynamisme exigé des S.I.

Apports à l'intégration

Les réductions de coûts font partie intégrante du métier d'intégration. Celles-ci reposent sur deux facteurs principaux : la réduction des

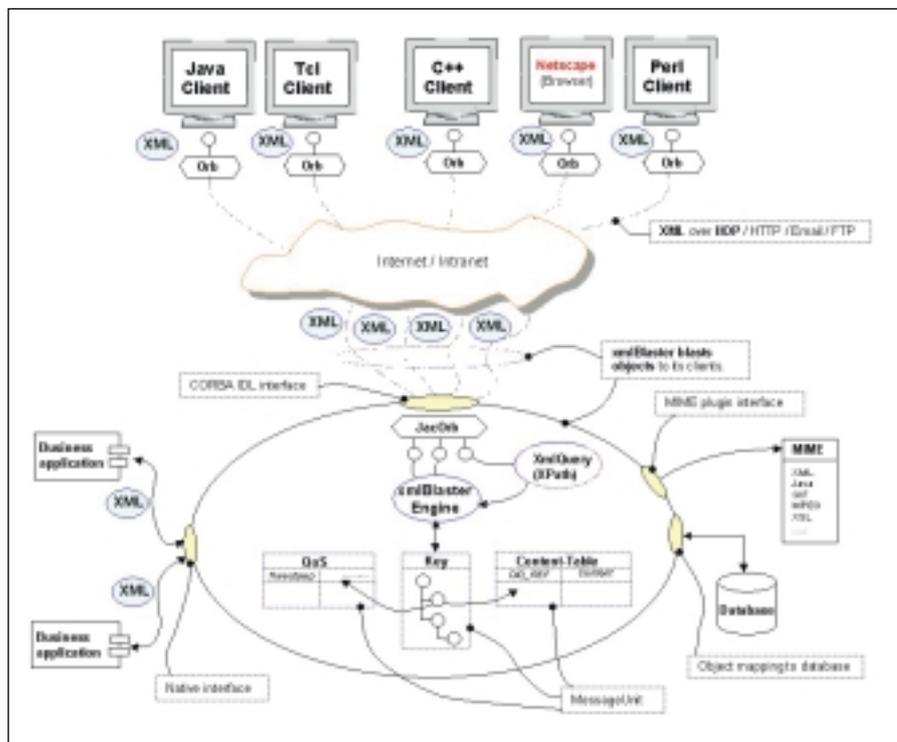


temps de développement et la réduction des frais de maintenance. Les bibliothèques libres, de norme LGPL, permettent une intégration peu dangereuse d'éléments fonctionnels souples, robustes, et maintenus par l'ensemble de la communauté libre. Les temps de développements sont donc réduits à leur minimum, et la maintenance de ces composants ne demande que peu d'investissement. Bien que le monde Open Source n'offre aucune garantie ("Use it at your own risk"), ni aucun formalisme quant au traitement d'éventuelles anomalies découvertes, le dynamisme des développeurs de la communauté Open Source assure, dans la plupart des cas, des réponses claires, techniquement fondées, et des corrections rapides sur tout problème découvert. La bibliothèque ZLib est un bon exemple des possibilités offertes en terme d'intégration par les logiciels libres, y compris en environnement grand système comme MVS. De plus, la maintenance de solutions libres est souvent facilitée par le fait qu'elles ont été éprouvées, avant intégration, par le monde Open Source. Ainsi, par exemple, le serveur HTTP libre Apache en est venu à dépasser les solutions propriétaires comme IIS (Microsoft),

avec plus de 67 % des parts de marché, soit plus de 30 millions de serveurs déployés en production. De même, dans le monde JAVA, les Frameworks Open Source font référence, avec par exemple des solutions telles Struts, log4j ou Hibernate et sont très novateurs, par exemple, avec JBoss AOP ou JAC pour l'AOP.

Une mise en œuvre complexe

La multiplicité des solutions et standards ouverts apporte une nouvelle complexité. Alors que les grandes références propriétaires sont peu nombreuses et identifiées, les solutions ouvertes se multiplient et sont souvent peu connues. Dans le cas des EAI par exemple, qui plus est, programmés en langage JAVA et disponibles en licences GPL ou LGPL, ce qui constitue des critères restrictifs conséquents, la liste de ces produits demeure importante. Des produits comme OpenEAI, jEngine, BIE (Business Integration Engine), SolAce, Proteus, XMLBlaster, Elemenope constituent une liste loin d'être exhaustive et répondant aux critères précités. La décision de l'intégration d'une de ces solutions implique donc un choix réfléchi et souvent complexe. Cependant, alors que cette abondance des solutions ouvertes représente une opportunité technique importante d'obtenir une solution au plus près des besoins, il pose un autre type de problème, que les intégrateurs doivent prendre en compte : le problème juridique de la gestion des



licences. Les logiciels ouverts, et plus particulièrement les logiciels libres, dépendent tous de licences parfois complexes, nombreuses et plus ou moins reconnues d'un pays à l'autre. Ces licences sont souvent difficilement abordables pour les juristes et encore plus pour les intégrateurs. Cependant, elles ne sont pas à négliger et ont valeur juridique (confère les

procès en cours entre SCO et IBM par exemple). Ainsi, ne serait-ce que pour celles qui sont compatibles avec la licence GPL, on constate la liste non exhaustive suivante : la licence Guild, la Gnu Ada, la X11, la Expat (ou MIT), la copyright ML standard du New Jersey, le domaine public, la cryptix, la modifiée BSD, la Intel Open Source Licence (publiée par l'OSI). La migration vers les standards ouverts et les logiciels libres a commencé. La démarche d'intégration, hier guidée par un panel restreint de solutions techniques fermées, s'ouvre aujourd'hui vers une démarche plus fonctionnelle, teintée d'urbanisme, et soutenue par la multiplication des solutions ouvertes. Aussi, il est du devoir des intégrateurs, dont la culture pour les logiciels propriétaires est prédominante, de prendre en compte cette nouvelle approche.

■ Eric Gueguen

Architecte Chef de Projet chez Kedros (filiale de Homsys Group)

À propos de Homsys Group :

Créé en 1991, Homsys Group est spécialisé autour de trois grands pôles : la Business Intelligence (Homsys), les Architectures Systèmes et Réseaux (Kedros) et les technologies Objet et Internet (Objet Direct). Le groupe est implanté à Bordeaux, Grenoble, Lyon, Marseille, Paris, Rennes et Toulouse.

homsys

www.homsysgroup.com

KEDROS

Exemple d'intégration de bibliothèques Open Source et de standards ouverts dans le cadre d'une production bancaire nationale.

Les protocoles d'échanges des composants intégrés au SI sont régulièrement enrichis pour répondre à de nouvelles exigences fonctionnelles. Différentes versions de composants communicants peuvent donc être amenées à cohabiter. Au sein de la production bancaire d'une grande banque nationale, l'utilisation du protocole ouvert XML a été recommandée par Kedros. Bien que plus verbeux que des solutions plus classiques, le protocole XML, accompagné de bibliothèques de parsing libres et portées en environnement MVS, assure la compatibilité ascendante des communications intercomposants et donc une souplesse sans égale des déploiements de production.

De plus, l'usage du XML a permis, à cette banque, de pouvoir faire communiquer aisément ses systèmes hétérogènes ouverts, comme les serveurs d'applications avec ses environnements Mainframe. Pour ce faire, deux solutions principales ont été mises en œuvre : des échanges de XML sur protocole TCP/IP et l'emploi du middleware orienté messages IBM MQSeries, couplé aux API Java JMS. La solution déployée depuis plusieurs années, montre toute satisfaction, aucune anomalie n'ayant été découverte dans les parties Open Source, ni sur les parties standards ouverts. Elle a donc permis une réduction des coûts de développement, de maintenance et de déploiement.

PHP face à ses concurrents

Avec PHP 5, on a pu découvrir la liste assez conséquente des nouveautés, à savoir la refonte complète du modèle objet, souvent qualifié par le passé de parent pauvre du langage, l'ajout d'une nouvelle librairie nommée Simple XML pour gérer très simplement les flux XML comme son nom l'indique (!), l'inclusion d'une base de données embarquée SQLite, le tout orchestré par le dernier moteur Zend Engine 2.0.

L'un des apports majeurs de PHP5 reste assurément le nouveau modèle objet, ou plus exactement l'ajout de tous les éléments qui faisaient défaut à la version 4 pour soutenir une comparaison digne de ce nom avec JAVA ou .NET. Là où PHP4 se rapprochait plus des technologies asp ou jsp de Microsoft et JAVA, PHP 5 va plus loin et offre l'opportunité d'implémenter des design patterns robustes et complets. Tout d'abord, on va dorénavant pouvoir définir au sein de sa classe un constructeur et un destructeur. Le nouveau modèle propose par ailleurs un niveau de sécurité plus satisfaisant pour les classes, en permettant de définir des variables et méthodes privées ou protégées (la distinction faisait cruellement défaut à la version précédente où tout était public par



Une partie des membres de l'AFUP avec Rasmus

défaut). Il devient possible de déréférencer un objet et la notion de classe abstraite fait son apparition. Enfin, gardons le meilleur pour la fin, la gestion des exceptions est intégrée nativement au sein du langage, au moyen d'instructions du type TRY ... CATCH, là aussi, c'est une demande récurrente à laquelle l'équipe de développement a répondu dans le cadre de cette version 5. Le pas qui vient d'être franchi est énorme, mais l'un des points faibles de PHP reste la faiblesse de l'offre en termes de serveur d'applications. Il n'existe pas à ce jour d'équivalent PHP de la plate-forme J2EE ou .NET, la première étape d'un gros projet PHP sera donc souvent la mise en place de briques techniques réutilisables au sein d'un framework. Les environnements de développements sont encore également peu nombreux et ne proposent pas encore en standard d'exploiter les évolutions majeures de la version 5 de

PHP. On peut citer l'environnement PHPedit développé par la société française Waterproof et l'IDE de Zend, très aboutie également. Concernant les autres nouveautés, nous invitons le lecteur à visiter le site officiel de PHP où figurent de nombreux exemples d'utilisation de Simple XML ou SQLite. Signalons juste qu'il devient trivial, grâce à Simple XML, de parser un contenu XML standard, sans avoir recours à la lourdeur du moteur DOM qui a d'ailleurs eu également droit à un petit lifting pour l'occasion. Quant à SQLite, ce petit SGBD reposant sur des fichiers plats propose une implémentation exhaustive de la norme SQL ANSI92 et se positionne comme une alternative à MySQL dans le cadre de petits projets. La sortie de PHP5 offre de nombreuses perspectives à moyen terme pour les différents projets basés sur ce langage. Il est très vraisemblable que l'on verra très rapidement

Historique de PHP et chiffres clés

Le langage PHP a été créé en 1995 par Rasmus Lerdorf. Ce dernier a décidé très tôt d'ouvrir le code de PHP à la communauté Open Source. Andy Gutmans et Zeev Surasky ont alors rejoint le projet et contribué à propulser PHP sur le devant de la scène. Au fil des versions, le nombre de serveurs et de sites l'utilisant a toujours connu une progression soutenue. On comptait en février 2004, 14 millions de sites utilisant PHP, ce qui représente 1 site Internet sur 3 dans le monde (plus de la moitié des serveurs Apache utilisés par 70 % des sites, source : Netcraft, février 2004). En France, son taux d'utilisation avoisine les 40 % (Source : Nexen Services). PHP est en version 5 depuis cet été. À l'aube de ses dix ans, il nous paraît intéressant de dresser un état des lieux sur la place de PHP en entreprise. Longtemps considéré comme un langage de script destiné à des sites personnels, PHP a, avec la version 5, acquis ses lettres de noblesse et apparaît désormais comme une alternative crédible et mature face aux environnements proprié-

taires Java ou .NET. Sur Internet, PHP est déjà omniprésent. Les plus gros sites français en termes de trafic et de charge comme Wanadoo, Free ou Voila reposent sur lui (80 % des 10 sites dont l'audience est la plus forte). Récemment, on a pu relever l'utilisation de PHP sur le site de campagne du candidat américain démocrate John Kerry qui a dû faire face à une montée en charge importante avant l'élection. Pour revenir en France, on le retrouve sur des sites d'information à forte fréquentation comme lemonde.fr, ou encore sur le site de France 3 ; Le cybermarchand Houra.fr utilise PHP depuis ses débuts, après un court démarrage sous vignette. Et plus récemment, le gouvernement français l'a élu support de sa plate-forme éditoriale au travers du projet SPIP AGORA. De nombreuses sociétés du CAC40 déclarent également une utilisation croissante de PHP pour leurs sites Web ou pour des projets plus stratégiques de type intranet (78 % d'après une récente étude menée par Globalis Media Systems).

apparaître, au sein de la communauté Open Source, des projets de framework s'appuyant sur le nouveau modèle objet. De même, on s'attend à ce que nombre de librairies soient portées sur la version 5, afin de tirer partie des nouveautés et on compte beaucoup sur leur intégration dans les quelques environnements de développement disponibles sur le marché. De plus, PHP5 se rapproche de plus en plus des langages JAVA ou .NET en terme de syntaxe, il sera facile pour les développeurs ayant des compétences dans ces autres langages de le prendre en main. Enfin, s'il était déjà possible de créer des applications de type client riche au moyen de la librairie PHP-GTK sur les postes de travail, l'ajout d'une base de données embarquée avec le langage, permettra de construire de petites applications clientes complètement autonomes.

Les projets phares autour de PHP

Parmi les projets matures s'appuyant sur le langage PHP, l'un des plus anciens et de surcroît, des plus utilisés est certainement phpMyAdmin. Ce projet Open Source, qui est apparu en même temps que le langage, permet de contrôler à travers une interface web conviviale le SGBD MySQL. La plupart des actions sont possibles en quelques clics : ajout d'une table, ajout d'un enregistrement, export de la base ... Le projet est très actif et a donné des idées à d'autres équipes qui ont tour à tour développé le même type d'outil pour les bases PostgreSQL ou ORACLE (respectivement phpPgAdmin et phpOraAdmin).

Un des autres champs d'applications populaires du langage PHP est la création de webmail. SquirrelMail propose une interface conviviale, simple et facile à déployer, très large-

ment utilisée sur le net. On peut également citer le produit Imp basé sur le projet Horde lui aussi open source, utilisé entre autres par free.fr.

La gestion collaborative, très en vogue actuellement, dispose également de son lot d'outils open source performants et matures basés sur PHP. C'est le cas, par exemple, de la solution eGroupware qui propose sous la forme de modules un certain nombre de fonctionnalités, comme la gestion d'agenda, de messagerie, ou de forum de discussion. Selon le contexte, on pourra aussi déployer le module de gestion de projets, ou celui de gestion de contenu. L'un des autres projets phares de la communauté sur ce domaine est PHProjekt qui lui aussi propose un nombre important de modules autour de la gestion collaborative dans plus de 25 langues. Dans la catégorie des projets de gestion de contenu, le français SPIIP s'est récemment illustré dans les médias : il a été utilisé et modifié pour les besoins du cabinet du premier ministre. La version ainsi modifiée est intitulée SPIIP AGORA et a été également mise à disposition de la communauté Open Source. De nombreux débats animent la communauté ces derniers temps afin de savoir si SPIIP AGORA doit rester un fork indépendant de SPIIP, ou s'il doit servir de base pour le futur SPIIP (solution qui n'a pas vraiment les faveurs des fondateurs du projet). Pour en savoir plus sur la saga SPIIP, nous vous invitons à visiter le site officiel.

Pour terminer le panorama des grands projets PHP, on ne peut passer à côté des divers frameworks et librairies mis à disposition par la grande communauté des développeurs PHP (plus de 500 000 personnes dans le monde à ce jour). Soutenu par le groupe PHP, la librairie

PEAR propose sous la forme de packages, une couche d'abstraction de base de données nommée PEAR::DB ou encore une couche permettant de gérer l'authentification sur un site, nommée PEAR::AUTH. Plus généralement, PEAR introduit une méthodologie et une norme de développement robuste et fiable qu'il est très facile de s'approprier au sein d'un projet PHP quel qu'il soit. Le projet SMARTY propose une gestion souple des formulaires et est de plus en plus plébiscité par la communauté. Des projets plus ambitieux comme phpMVC ou Copix ont vu le jour et proposent des frameworks complets, basés sur le traditionnel modèle MVC, largement répandu au sein de la communauté JAVA (au travers du projet STRUTS entre autres) néanmoins, leur utilisation reste encore marginale du fait de leur jeunesse. La capitalisation faite autour de ces outils montre là encore que le langage est arrivé à maturité. Il était souvent nécessaire hier, de construire ses propres briques de A à Z faute de solution stable au sein de la communauté, aujourd'hui, l'intégration de librairies tierces au sein des projets PHP est quasiment systématique (L'utilisation croissante de PEAR::DB en est une belle illustration).

PHP face à ses concurrents

Le tableau suivant présente un comparatif des trois technologies concurrentes.

Pour positionner PHP face à ses deux grandes technologies concurrentes que sont .NET et J2EE, il convient de considérer les points suivants :

- le choix de J2EE est principalement motivé par la robustesse des développements pour des architectures massivement transactionnelles et par le nombre de connecteurs pro-

Technologie	Aujourd'hui	En 2005
Java	J2EE est la technologie la plus mature et la plus versatile (disponible sur toutes les plates-formes et capable de communiquer avec tous les types de systèmes). Cependant, on lui reproche sa complexité en développement et la complexité des architectures mises en place	Java 1.5 apporte des améliorations syntaxiques importantes. Les environnements de développements se normalisent autour de WSAD et Eclipse. L'usage d'utilitaires d'aide aux tests, packaging, déploiement (JUNIT, ANT, MAVEN) devient incontournable. L'utilisation de la technologie s'industrialise, les plates-formes projet J2EE commencent à atteindre un bon niveau de maturité, tant sur le plan de la normalisation des architectures que sur le plan des méthodologies des cycles projet
.NET	.NET est une technologie mature, permettant le développement rapide (15% plus efficace que Java). Cependant, elle est limitée aux environnements Microsoft, et propose l'intégration principalement au travers de Web Services.	le framework 2.0 arrive avec un environnement de développement en équipe, Visual Studio Team System, améliorant encore la productivité. Les utilitaires se développent avec un peu de retard sur Java. L'offre globale couplée Biztalk / .NET présente un intérêt certain pour les projets EAI liés au commerce électronique, et a l'avantage d'être cohérente
PHP	PHP est la technologie « low cost » : apprentissage rapide, gratuité de la plate-forme LAMP (Linux, Apache MySQL, PHP), offre d'hébergement à faible coût. PHP offre une très grande performance. Par contre, il est très limité en terme d'intégration avec le SI.	PHPS devient un langage véritablement objet, capable d'affronter J2EE et .NET. Il apporte des outils plus simples pour traiter XML. Le mode open source reste un atout majeur sur cette technologie, garant de qualité. Cette différence est tellement nette que des acteurs comme IBM tentent de manœuvrer pour rendre Java open source.

posés par les éditeurs de solutions d'entreprise (SAP, IBM, outils EAI, etc.). Ce choix a souvent la faveur des directions informatiques dans un contexte applicatif devant s'interfacer avec un existant riche, pour des applications à taille d'entreprise.

- Le choix de .NET est mis en avant dans des infrastructures basées sur les technologies Microsoft, dans des architectures orientées services (l'intégration des Web services est excellente dans la plate-forme.NET), sur des projets à taille départementale. Le développement est également 15 % plus rapide qu'en environnement Java (libre de type Eclipse ou commercial)
- Le choix de PHP, notamment avec les évolutions de PHP 5 qui le rapprochent au niveau de la qualité des concepts de J2EE et .NET (héritage, polymorphisme, meilleure gestion des exceptions) sera pertinent sur des archi-

tectures 100 % Web (site de gestion de contenu, intranets) sans interactions fortes avec un existant. Les développements sont également très rapides, à l'instar de ceux sur la plate-forme de Microsoft. Enfin, la gratuité de l'investissement initial et la disponibilité de nombreux frameworks et composants techniques issus de la collectivité des développeurs sauront trouver un écho très favorable auprès des décideurs informatiques.

En conclusion : pourquoi PHP est un choix porteur

Lors de la phase préliminaire au lancement d'un projet, de plus en plus d'entreprises étudient l'opportunité de s'appuyer sur PHP dans le cadre de l'enrichissement de leur système d'information, ou pour les nouvelles versions de leur site web. Les décideurs voient en PHP un langage simple à mettre en place, aussi

bien en interne qu'en externe (la quasi-totalité des hébergeurs propose une offre PHP). Ils y voient une solution portable et pérenne : PHP est disponible sur tous les systèmes d'exploitation, contrairement à .NET et fait partie de la fondation APACHE, la communauté est nombreuse et la main d'œuvre bien meilleur marché que pour les compétences J2EE ou .NET. Enfin, PHP reste très souple d'utilisation ce qui écourte bien souvent les temps de développement, Il assure l'interopérabilité avec d'autres langages (de nombreux connecteurs sont disponibles, il est possible d'instancier une classe JAVA ou un objet COM+ depuis PHP ...) Il est robuste et tient très bien la charge, tout en restant complètement gratuit. Plus que jamais, PHP dans sa toute dernière mouture s'avère être un choix judicieux pour de nombreux projets, y compris les plus stratégiques.

■ Jérôme LAVANCIER – SQLI

INTERVIEW

“PHP reste le langage le plus accessible”

Entretien avec Rasmus Lerdorf et Zeev Suraski, « pères fondateurs » de PHP

PHP 5 bénéficie de la puissance de l'orientation objet, mais ne la rend pas obligatoire. Le langage reste simple et accessible, et un programmeur peut facilement utiliser XML ou SOAP sans connaissances préalables. Et si PHP devenait la référence pour s'interfacer avec Java ?

Lors de la 4^e édition du Forum PHP qui s'est tenue à Paris, les 18 et 19 novembre, Programmez a rencontré Rasmus Lerdorf et Zeev SURASKI, pères fondateurs de PHP (script et langage).

Comment est né le langage PHP ? À quels besoins répondait-il ?



Nexen services SA

Rasmus : En 1993-1994, j'écrivais des programmes Web en CGI et en langage C. Récrivant sans cesse les mêmes lignes de code, j'ai fini par constituer des 'bibliothèques' génériques

dont il suffirait d'appeler les fonctions, pour éviter de recompiler chaque fois toutes les fonctions. Cette motivation m'a poussé à créer une sorte de langage intégré dans du code HTML (HTML embedded language).

J'ai rapidement constaté la puissance que

pouvait apporter ce type de langage et plus encore, lorsqu'on y intégrait des instructions telles qu'une structure conditionnelle "if ... else" réutilisant ou faisant référence à d'autres parties de code ou d'autres fichiers.

Je n'ai pas voulu créer un langage. Pendant les deux premières années, il s'agissait d'une solution de "templating for the Web" simple et techniquement abordable. Ce que j'ai imaginé en mettant PHP en Open Source, était un moyen de relier les pages Web à du code en langage C, via mon API. Et j'imaginai que les gens écriraient la logique applicative en C, puis utiliseraient ensuite de simples templates (PHP) pour l'utiliser. Car personne ne désire réécrire sans cesse du C.

L'objectif de départ était de résoudre un problème du HTML qui ne permettait pas de disposer simplement de variables, ni de moyen pour stocker un résultat. Or, lorsque vous utilisez des variables et des structures conditionnelles, vous êtes en mesure d'interagir avec un

langage de programmation. Par exemple, je disposais de la boucle "while", ainsi que de l'intégration de fichier et de l'appel de fonctions. Cela ouvrait de nouvelles perspectives, car il était possible d'incrémenter et de décrémenter des variables statiques... Ces possibili-

“Je n'ai pas voulu créer un langage”

tés sont capitales pour la puissance de PHP, et limitent les échanges entre client et serveur Web en traitant des opérations complexes en une seule fois (par exemple : (2+3)/7 est traité en un seul échange et pas en deux fois).

PHP est devenu un langage, pour combler les lacunes du langage C à répondre simplement aux besoins du Web. Les enrichissements successifs de PHP l'ont transformé en langage,



Zeev Suraski

parce que les gens en avaient besoin.

Zeev : Ma réponse diffère légèrement, car notre histoire n'est pas la même. Je n'ai été impliqué dans PHP qu'à des phases ultérieures. Au départ, j'étais surtout un utilisateur. Ayant rencontré un problème plutôt complexe pour relier du code en langage C, nous nous sommes penchés sur les sources du PHP pour participer à son débogage et son évolution. Mais le problème semblait difficile à résoudre. J'ai alors pensé que nous pouvions le résoudre autrement : en transformant le PHP en langage. L'idée était de reprendre le concept d'"HTML embedded template" pour en faire un langage à part entière.

Comment expliquez-vous le succès de PHP ?

Zeev : Je vois deux raisons majeures au succès du PHP et à son explosion en 1998-1999. La première réside dans son extensibilité, la possibilité d'ajouter facilement de nouvelles fonctions. La seconde raison est sa facilité d'utilisation. Même avec les nouvelles technologies et nouveaux protocoles (XML, SOAP, etc.), PHP reste le langage le plus accessible. Il est très facile de commencer à programmer en PHP, y compris pour créer des applications très complexes. C'est mon avis, mais c'est aussi ce que j'entends de très nombreux utilisateurs...

Rasmus LERDORF

Où et quand êtes-vous né ?

Je suis né en 1968 à Qeqertarsuaq au Groenland [NDLR : une vraie torture à prononcer, cf www.qeqertarsuaq.gl]. (Zeev : "le seul mot avec 3 Q dans le monde...").

Où habitez-vous ?

Je vis près de San Francisco.

Quels sont vos hobbies ?

J'aime jouer au football (soccer) lorsque j'ai le temps. Passer du temps avec mon fils et l'emmener au parc, jouer au foot avec lui, lui lire des histoires.

Combien d'enfants avez-vous ?

J'ai un fils de 2 ans et demi.

Quel est votre travail actuellement ?

Je suis ingénieur chez Yahoo !

Que pensez-vous de la France ?

Je suis très souvent en France. Mon épouse

Zeev SURASKI

Où et quand êtes-vous né ?

A Haïfa en 1966.

Où habitez-vous ?

Actuellement j'habite à Tel-Aviv

Quels sont vos hobbies ?

Venir en France, la moto et les ordinateurs.

Avez-vous des enfants ?

Pas encore. Pas de projets immédiats, mais certainement dans le futur.

Quel est votre travail actuellement ?

Je suis CTO (directeur technique) de Zend Technologies. Je dirige toute la partie technique.

Comparé à d'autres langages, il reste plus simple que Visual Basic, ASP ou c#. Quant à JSP, il est beaucoup plus compliqué.

Rasmus : Avec, le boom d'Internet, les programmeurs se sont retrouvés confrontés aux mêmes problèmes avec le Web. Et nombre d'entre eux n'étaient pas expérimentés. PHP est arrivé au bon moment. De plus, de nom-

“ajouter facilement de nouvelles fonctions à PHP”

breuses petites entreprises ne pouvaient se payer les services de programmeurs confirmés. Avec PHP, elles pouvaient mettre en ligne des services qui fonctionnaient.

parle français et aime parler avec les Français. Nous aimons beaucoup ce pays. Si un jour nous revenons en Europe, nous nous installerons certainement en France. D'ailleurs, nous en avons déjà parlé plusieurs fois. Nous nous installerions alors dans le sud de la France, car la météo est plus clémente. Surtout lorsque l'on est habitué à la Californie. J'aime la France et je trouve les Français très intéressants.

Que faites-vous pendant vos vacances ?

Je ne prends pas réellement de vacances. J'essaie plutôt de profiter de mes déplacements professionnels pour intégrer un peu de détente. Par exemple, ma femme et mon fils me rejoignent à Paris cette semaine. Et parfois nous en profitons pour embarquer pour une croisière, etc.

Que pensez-vous de la France ?

La France est le second pays pour la popularité de PHP en Europe, après l'Allemagne. La fréquentation de cet événement PHP a augmenté avec près de 60 participants supplémentaires par rapport à l'an dernier. Un résultat très intéressant ! De plus, Zend enregistre une croissance de ses utilisateurs français. J'aime la France !

Que faites-vous pendant vos vacances ?

D'habitude, je voyage avec ma femme. Généralement en Europe. Il y a deux mois, nous étions en Italie.

Les développeurs ont fait entrer PHP dans les entreprises

Zeev : Aujourd'hui, la plupart des grandes entreprises s'intéressent à PHP. Mais cette pénétration n'est pas le fruit d'une décision stratégique de la DSI. En général, des développeurs ont apprécié PHP, comme meilleure réponse à leur besoin. Parfois, ils en ont démontré l'efficacité, sans avoir forcément demandé d'autorisation. Cela diffère largement des processus d'adoption pour Java/J2EE par exemple, où la décision émane de la direction. Souvent, des programmeurs nous demandent « comment puis-je les convaincre ? », « que dois-je leur dire pour continuer à utiliser PHP ? » ...

Comment placez-vous PHP par rapport à J2EE, .Net ou Mono ?

Zeev : Pour la plupart des utilisateurs, cela ne pose aucun problème. PHP seul leur suffit pour des sites modestes ou personnels. En revanche, les entreprises ne passeront pas intégralement à PHP. Car elles doivent composer avec des systèmes propriétaires, et des technologies déjà déployées avec J2EE ou .Net. C'est pourquoi PHP doit s'intégrer à cet existant. Aujourd'hui, les services Web autorisent la communication avec Java, .Net, et autres. Et PHP 5 dispose d'un excellent support des services Web.

À propos de la JSR 2.23, Zend discute avec Sun afin de les convaincre d'établir des spécifications d'intégration de Java avec tous les langages. Et PHP devrait devenir l'implémentation de référence en la matière. Alors, il sera très simple d'utiliser PHP pour communiquer avec des composants d'entreprise Java. Aujourd'hui, le moyen le plus répandu pour créer des applications, via des serveurs J2EE, consiste à utiliser JSP. Or, de nombreuses entreprises nous

confirmer (et s'en plaignent aussi auprès de Sun et d'IBM) que JSP demeure très complexe. Or, développer avec PHP est plus simple, et permet de communiquer avec tout.

Rasmus : JSP n'est finalement qu'un moyen d'accéder à du code Java. Ce serait donc magnifique de disposer d'un "super JSP", et PHP peut jouer ce rôle. Parfois, lancer une JVM (machine virtuelle Java) n'a aucun sens. Par exemple, est-il vraiment judicieux de mettre en route tout cela uniquement pour accéder à une base de données ? Les bases de données ne sont pas écrites en Java. Donc, mieux vaut accéder à leur moteur dans leur code natif.

Zeev : Il y a cinq à huit ans, les développeurs Web utilisaient les langages C ou C++, mais ne développaient pas tout avec ces langages. Pour publier du texte ou quelques pages HTML basiques, cela paraît totalement démesuré. Non seulement, les performances sont alors moins bonnes, mais surtout c'est très compliqué. De plus, on réduit fortement la productivité et cela augmente le time-to-market des applications Web. Et les investissements en développeurs confirmés et en temps, gonflent le coût total des développements.

La machine virtuelle Zend serait-elle plus performante que la JVM ?

Seev : Non. Le moteur PHP utilise du pcode [NDLR : il précompile les sources en code pré-compilé opcode et l'exécute ensuite.] tandis que la JVM est plus fermée et plus rapide. Mais, car il y a un grand "mais", vous devez d'abord démarrer la JVM. Puis, si elle démarre, vous êtes en code natif et devez lancer le processus. Or, tout ce processus est plutôt lent. Concernant la gestion de la mémoire, il est plus sûr de travailler en mode prédictible dans une application Java, parce que vous n'avez aucun contrôle et vous ne savez pas comment fonctionne la garbage collection [NDLR : la récupération des emplacements en mémoire qui ne sont plus utilisés par le système]. Ce qui n'est pas le cas avec PHP, cette gestion est beaucoup plus souple et peut être aisément réajustée.

Quelles sont les évolutions majeures des versions 5 et 5.1 de PHP ?

Zeev : Premièrement, l'orientation objet. Puis, la possibilité de mieux intégrer de nouvelles extensions comme les services Web, le XML, etc. De plus, l'utilisation du XML via PHP est simple, même sans l'avoir déjà fait. Jusqu'à

présent, pour utiliser ce langage, vous deviez assimiler et implémenter des technologies complexes comme Dom, Corba, etc. Avec PHP 5 vous accédez à la puissance du XML sans vous préoccuper de la plomberie. Idem pour les services Web, .Net, etc.

Rasmus : L'orientation objet est meilleure, et la complexité invisible la rend accessible à tous. Bien sûr, vous pouvez plonger dans un niveau de complexité plus avancé, mais la plupart des programmeurs s'en passent très bien.

L'orientation objet ne rend-elle pas PHP plus complexe pour certains programmeurs ?

Rasmus : Ils ne sont pas obligés de passer à la programmation-objet. Ils peuvent toujours continuer à programmer, sans rien changer à leurs habitudes.

Zeev : Avec PHP5, de nombreux aspects comme "simple XML" utilisent la programmation orientée objet. Toutefois, la complexité est masquée et prise en charge par le moteur. Donc, il est tout à fait possible de continuer à programmer en PHP avec ses nouvelles possibilités, sans recourir à la programmation-objet. Cependant, l'objet présente de nombreux avantages, comme la réutilisation de composants, par exemple, qui augmente la productivité.

Zeev : Certes, sur de grands projets d'entreprise, la programmation-objet représente de réels avantages. Toutefois, dans la majorité des situations, vous ne développez pas d'application enrôlant 50 programmeurs, ni des programmes en perpétuelle évolution. De plus, depuis les débuts de la programmation orientée objet, les entreprises qui recourent à cette programmation très structurée ne savent pas toujours comment en bénéficier pleinement. En fait, elles l'utilisent surtout pour structurer les développements et obtenir une excellente qualité de programmation.

Rasmus : La programmation orientée objet offre bien plus de puissance que la programmation procédurale. Cependant, elle est plus complexe. Si vos développeurs sont peu expérimentés, mieux vaut opter pour le mode procédural, souvent suffisant. De plus, la maintenance du code sera plus accessible. Si quelque chose correspond à une logique objet, cette orientation convient. Mais rien ne sert de forcer les choses à devenir absolument des objets. Par exemple, j'ai vu un objet HTML appelé "br" dont la méthode produisait une balise
. On se demande alors quelle raison

HISTORIQUE

10 ans d'âge pour un langage qui a bien mûri !

• Fin 1993-début 1994 :

Rasmus Lerdorf rédige des scripts pour compter les visiteurs de son site et stocker un résultat dans une base de données. Il assemble ses scripts sous forme de fonctions écrites en C dans une librairie.

• Février 1994 :

Publication de cette API, « Personal Home Page ».

• 1994-1995 :

Écriture de PHP/FI (Personal Home Page / Forms Interpreter) inclut les fonctions de base et un système d'interprétation automatique des variables de formulaires.

• Novembre 1997 :

Publication officielle de PHP 2.0 (PHP/FI en bêta depuis deux ans)

• 1997-1998 :

Andi Gutmans et Zeev Suraski réécrivent PHP/FI, et collaborent avec Rasmus.

• Juin 1998 :

Publication officielle de PHP 3.0 rebaptisé "PHP: Hypertext Preprocessor".

• 1998-1999 :

Andi Gutmans et Zeev Suraski réécrivent le moteur PHP, baptisé "Zend Engine" (Zeev & Andi)

• Mai 2000 :

Publication officielle de PHP 4.0

• Mi-juillet 2004 :

Publication officielle de PHP 5.0

peut pousser à introduire une syntaxe aussi compliquée pour ce type de résultat.

Sera-t-il possible un jour d'embarquer des applications php-gtk dans un navigateur Web ?

Rasmus : Nous n'avons aucune raison de le faire. Car ce développement sur la partie cliente nécessite une énergie considérable. En effet, il faut alors supporter une multitude de plates-formes différentes.

Zeev : Cela n'est programmé ni aujourd'hui ni dans le futur. Mais ce projet peut se concrétiser si la demande est forte, comme le prouve le mouvement Open Source. Encore faudrait-il que cela soit réellement utilisé.

Rasmus : Pour y parvenir, il faudrait d'abord nous convaincre de l'utilité de ce projet. Puis il faudrait également persuader tous les éditeurs de navigateurs de supporter gtk. Les chances sont alors très minces.

■ José Diz

Développez gratuitement : c'est possible !

Il est possible de coder pour pas cher, voire pour zéro euro. Que vous soyez un amateur du développement, un étudiant, une petite structure, un développeur indépendant, tout est bon pour réduire la facture... Vous aurez l'embaras du choix sur la plupart des plateformes et langages. Attention, la qualité et la richesse fonctionnelle varient énormément. Le gratuit n'a pas la prétention de remplacer tous les outils commerciaux.

Nous vous proposons ici un panorama assez large, mais non exhaustif des possibilités du développement gratuit en vous indiquant les outils disponibles.

Open Source, Libre, Shareware, versions « light »

Développer gratuitement n'est pas la chasse gardée du « libre » et de l'Open Source. Il existe de nombreux freewares, sharewares sur le marché. D'autre part, de plus en plus d'éditeurs commerciaux mettent à la disposition du développeur des versions allégées gratuites ou peu chères, voire des outils complets. Une des grandes tendances actuelles est de faire un mixte entre les outils payants (souvent propriétaires, mais pas uniquement) et les outils gratuits. Inutile aussi de passer sous Linux pour en profiter. MacOS X et Windows étoffent chaque jour leur logithèque ! Si vous possédez déjà un environnement commercial, ne «cassez» pas tout. Cherchez plutôt à réduire le coût d'achat de composants supplémentaires. Pour les tests, pour les unitaires, Junit / Nunit peut suffire ! L'usage d'un outil Open Source permettra aussi d'être indépendant d'un éditeur, ce qui n'est pas négligeable. C'est aussi un bon moyen d'introduire les standards du marché plus rapidement et de suivre leur évolution. Malgré tout, n'attendez pas des miracles de l'offre gratuite : dans certains cas, impossible de se passer de l'offre commerciale des éditeurs. Pour les versions limitées ou lights d'éditeurs, vérifiez le type de licence et d'applications que vous pouvez ou non développer. Il y a parfois (souvent ?) des restrictions.

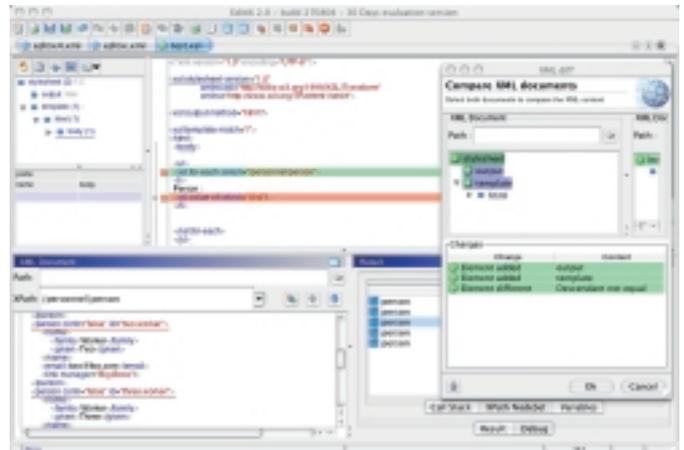
Quand ? : l'usage du gratuit (d'un éditeur, projet Open Source), peut se faire à tous les niveaux d'un projet, d'une application. Dans le cas d'un développement, mieux vaut étudier les outils gratuits avant le codage afin d'effectuer le choix. De plus en plus, les outils gratuits / libres se placent dans le middleware, les IDE et les outils serveurs. Cependant, l'usage d'un outil gratuit doit être pertinent dans le cadre

Aller chercher l'information

Question à Rodrigo Laurens, Directeur Technique, NetS

Existe-t-il une réelle différence fonctionnelle entre les outils libres et commerciaux ?

Rodrigo Laurens : Il existe des différences. Dans les outils de montées en charge, il n'existe pas d'outils comparables à ceux de Mercury. Pareillement pour les SGBD complexes avec Oracle. Mais cela va venir. Il existe aussi des offres libres qui n'existent pas chez les commerciaux. Pour la Factory Software, des outils libres le faisaient depuis pas mal de temps, IBM commence tout juste et Microsoft arrivera dans quelques mois...



L'IDE Editix

d'un projet, d'une application. N'hésitez jamais à tester les outils commerciaux via les versions limitées ou lights (de plus en plus courantes).

Comment ? : le plus difficile est de trouver la bonne information. Les grands projets Open Source, comme Apache, possèdent la plupart du temps une section dédiée aux outils. Les IDE gratuits (NetBeans, Eclipse) proposent un catalogue très complet des plug-ins tiers. Pour les projets Open Source, il y a l'incontournable SourceForge. Ou encore, Java-channel.org pour les outils et librairies Java.

Quoi ? : gratuit ne signifie pas choisir les yeux fermés, surtout si vous utilisez ces outils dans un projet. Dans le gratuit, orientez-vous avant tout vers les projets Open Source solides, fédérant une grande communauté. Montrez-vous vigilants vis-à-vis des projets exotiques et nouveaux. Leur existence est souvent brève (la remarque vaut aussi parfois pour les éditeurs commerciaux). Examinez attentivement la licence de l'outil. Autorise-t-elle le développement de logiciels commerciaux ? Dans le cas d'un outil libre, impose-t-il de mettre son logiciel sous une licence particulière ? Qui assure support et assistance ? L'outil est-il mis à jour régulièrement ?

Les IDE Java / J2EE

Pour coder en Java, les possibilités ne manquent pas. Trois IDE gratuits sont disponibles : Eclipse, NetBeans et Jbuilder Foundation. Au niveau fonctionnel, Eclipse et NetBeans sont sans doute supérieurs à Jbuilder Foundation. Ils bénéficient aussi de plusieurs centaines de plug-ins (souvent gratuits) afin de compléter l'environnement. Pour un débutant

en Java ou maîtrisant mal le codage pur et dur, NetBeans ou Jbuilder, équipés d'un RAD, constituent un bon choix. Parmi les IDE d'un faible coût, Sun Java Studio Creator est un excellent outil, même s'il se limite aux applications Web. Il dispose d'une version limitée dans le temps (30 jours). Côté indépendant, il y a aussi Jcreator (69 dollars une licence) ou IntelliJ (IDEA), un peu cher (499 dollars) mais performant.

Le pour : de plus en plus, les IDE gratuits sont utilisés par les éditeurs pour construire leur solution, les développeurs les utilisent largement. Modulaires, soutenus par une forte communauté, ils sont polyvalents. Notons l'existence d'IDE spécialisés (souvent payants).

Le contre : attention aux supports et assistance. Environnement fonctionnel parfois limité dans certains domaines, nécessite l'ajout de plug-ins (qualité variable).

Quelles applications ? : applications web, composants serveurs, applications desktop, web services

.NET

Le monde .NET a aussi droit à ces IDE gratuits ou allégés. En attendant la gamme Express (disponible gratuitement en préversion), Microsoft propose un RAD dédié à ASP.NET, Web Matrix.

Côté Borland, il faut attendre la version personnelle de Delphi 2005 (incluant désormais C# Builder qui n'est plus disponible séparément), prévu fin 2004-début 2005. Diverses solutions indépendantes existent : Emacs C# Editor Mode, #develop (sans doute le meilleur éditeur libre .NET) ou le shareware Zeus. Vous pouvez aussi opter pour un .NET « alternatif » : Mono (Novell). Il permet de coder en C# et ASP.NET sous Windows, Linux, MacOS X. Un outil de développeur est disponible (monodevelop).

Le pour : offre assez diverse. Marché encore jeune, communauté en pleine évolution

Le contre : aucun réel rival à Visual Studio et sa gamme Express (même si Delphi 2005 rééquilibre l'offre). Une offre plutôt spécialisée.

Quelles applications ? : applications web / desktop / client-serveur, intégration, web services, mobile...

Quelques outils Microsoft gratuits

Outils	Commentaires
SDK .NET Framework 1.1	Le kit de développement (avec ligne de commande et notepad !)
WebMatrix	Développement RAD ASP.NET
SQL Server 2000 Desktop Engine	
http://www.microsoft.com/sql/msde/default.asp	Moteur SGBD
CLR Profiler	Outil d'analyse des applications .NET
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag/html/scalenetowto13.asp	
Fx Cop	Librairies de développement
http://www.gotdotnet.com/team/fxcop/	
XML Tools	Divers outils Microsoft pour XML
http://www.gotdotnet.com/team/xmltools/	
Visual Basic .NET resource kit	Kit de ressources pour développeurs VB.NET
Visual Basic PowerPack	Ensemble de contrôles VB
http://www.gotdotnet.com/workspaces/workspace.aspx?id=167542e0-e435-4585-ae4f-cl11fe60ed58	
.NET Framework 2.0	Préversion gratuite
Starter Kits	Ensemble packagé et gratuit pour démarrer en ASP.NET
Windows Installer XML	Installeur Microsoft en format XML pour MSI et MSN
http://sourceforge.net/projects/wix/	

IDE = produit d'appel

Question à Erik Dasque, senior Product Manager, Novell

Comment positionner Mono par rapport à Microsoft et à la communauté libre ?

Erik Dasque : Mono est un logiciel entièrement libre, contrairement à d'autres outils libres qui limitent aux développements non commerciaux. La seule restriction concerne le compilateur C#. S'il est implémenté dans un logiciel, il faut que celui-ci soit GPL. Pour le développement, on dispose de Monodevelop. On dispose d'une chaîne de développement entièrement libre et gratuite. Notre plateforme de choix est Linux. On peut très bien avoir une plate-forme cible Windows et développer sous Linux.

L'avis d'Open Wide

“Des outils propriétaires pour les besoins pointus”.

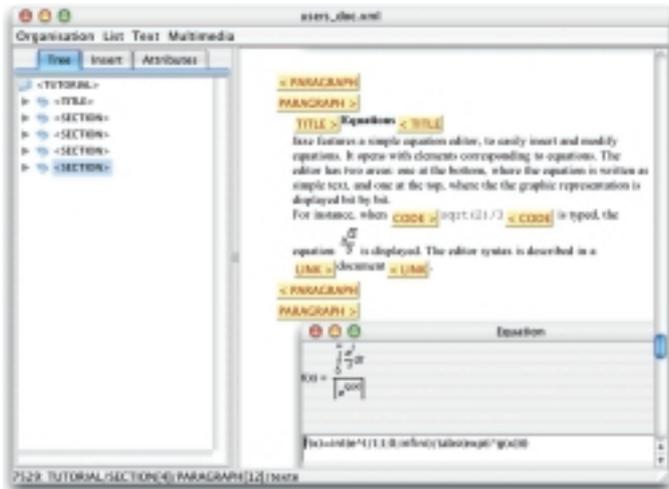


Fabrice Dewasmes
Responsable du
domaine système
d'information
Open Wide

Dans le monde du temps réel et de l'embarqué, les développements ne présentent aucune interface graphique. Sous Linux, de tels développements sont écrits en C, ce qui ne nécessite qu'un simple éditeur de texte. Un simple VI suffit donc, mais les développeurs lui préfèrent souvent emacs. Ces deux outils sont open source et fournis avec n'importe quelle distribution, tant ils sont utilisés. Le compilateur est également gratuit puisqu'il s'agit de GCC. Ce type de développement (pour ce qui est de Linux) peut donc entièrement être réalisé à l'aide d'outils gratuits.

Dans le monde de l'applicatif, la réponse est plus nuancée. Si plusieurs IDE gratuits tels que Eclipse et NetBeans couvrent largement le besoin d'outils, même évolués, pour le développement d'applications web, l'offre pour le RAD est bien plus pauvre, notamment du côté Eclipse où Visual Editor vient à peine de sortir une version 1. Il faut alors se tourner, par ordre de capacité de l'outil, vers des plug-ins Eclipse payants, vers NetBeans ou mieux, vers JBuilder de Borland. Le même type de constatation peut être fait pour ce qui est des outils UML (où un outil comme Poseïdon montre rapidement ses limites face à un Rose ou un Objecteering) ainsi que des outils de tests de performance où Borland Optimizelt reste le maître incontesté.

En résumé, même si le monde du libre couvre largement les outils de base du développement, il faudra encore se tourner vers les outils propriétaires (et souvent très onéreux) pour les besoins plus pointus, notamment pour la modélisation graphique, la modélisation UML ou les tests de performance.



Les outils XML se multiplient

PHP

PHP constitue un des « symboles » du monde libre. Il existe une vaste logithèque dédiée à ce langage dans tous les domaines et quel que soit le système. Les principaux IDE / éditeurs sont payants (Zend Studio, NuSphere PhpED, PHP Edit...). Mais bien d'autres sont gratuits, même si là on a souvent des outils rudimentaires : Emacs, BlueFish, K PHP Develop, PHP IDE, Dev-PHP, TruStudio Foundaton (existe en version commerciale), EasyPHP, HydraPHP, PHPEclipse, etc. Autour de PHP, on trouve aussi des frameworks, des compilateurs, des outils de tests, des outils serveurs, des composants, des connecteurs SGBD, etc. Avec la généralisation de PHP, de multiples plug-ins PHP existent. Ainsi, on peut coder en PHP sous Macromedia Dreamweaver ou encore sous Eclipse (voir projet PHPEclipse).

Le pour : large logithèque gratuite ou commerciale

Le contre : qualité et fonctions diverses. Les « grands » IDE sont payants.

Quelles applications ? : applications web, site dynamique

XML (et dérivés)

Les divers langages XML constituent désormais un incontournable. Les SGBD et éditeurs l'implémentent quasi systématiquement. Ici aussi le choix offert est vaste aussi bien en freeware, Open Source qu'en outils commerciaux. Les éditeurs XML sont les plus courants. Par contre, pour

les véritables IDE / RAD XML, l'offre Open Source est restreinte, heureusement, de nombreux éditeurs commerciaux proposant des versions d'essais ou « lights ».

Outils	Commentaires
Jaxe http://jaxe.sourceforge.net/	Editeur XML Open Source.
Morphon http://www.morphon.com/	Editeur wysiwyg pour créer documents XML.
XMLStarlet http://xmlstar.sourceforge.net/	Ensemble d'outils en ligne de commande pour les documents XML.
Oxygen http://www.oxygenxml.com/	Editeur XML, version limitée 30 jours disponible
Authentic http://www.altova.com/products_doc.html	Editeur de contenu XML gratuit de la société Altova
Bonfire Studio http://nzworks.com/bonfire/index.asp	IDE supportant XML, XSL, HTML. Gratuit
Conglomerate http://www.conglomerate.org/	Editeur Open Source XML.
Editix http://www.editix.com/	Editeur XML (et XSLT, XSL, DTD...), version 30 jours disponible

Pour aller plus loin, un excellent site en anglais : <http://www.xmlsoftware.com/editors.html>

Python

Python est un langage de plus en plus utilisé. Une logithèque se développe autour de lui. L'outil le plus connu de la communauté est Zope, orienté développement Web, téléchargeable sur le site officiel (zope.org). Outre Python, il existe aussi Jpython (technologies Java à la sauce Python).

Outils libres

- wxPython, librairie GUI <http://www.wxpython.org>
- PyQt, librairie GUI <http://www.activestate.com/Products/Komodo/>
- PyGtk, librairie GUI <http://www.pygtk.org/>
- Eric3, IDE <http://www.die-offenbachs.de/detlev/eric3.html>
- BOA-Constructor, IDE <http://boa-constructor.sourceforge.net/>
- Pythonwin, IDE, <http://www.python.org/windows/pythonwin/>
- Zope, Serveur Web, serveur d'applications

Outils commerciaux

- Wing IDE <http://wingware.com/>
- Visual Python, un plug-in pour développer en Python sous Visual

L'avis du développeur : Frédéric Mazué

“J'utilise volontiers ces outils non parce qu'ils sont gratuits, mais parce qu'ils sont meilleurs !”

Au cours de ces vingt dernières années, j'ai utilisé des quantités d'outils commerciaux et gratuits. Je les ai souvent utilisés conjointement, mais pas indifféremment. Ce qui peut me faire opter pour un produit, est avant tout ce que l'on peut en faire, et si on peut le faire rapidement et efficacement. La qualité technique d'abord. Par exemple, des Delphi et C++Builder sont efficaces et abordables financièrement en édition professionnelle. Pour moi la notion de "gratuit" est plutôt secondaire, même si certains produits commerciaux, dans la sphère Java notamment, affichent des tarifs rédhibitoires pour un indé-

pendant, tout en présentant un intérêt contestable. De même pour les droits des bases de données commerciales. Souvent trop chers pour les petites applications de gestion de stock que je peux avoir à développer. De plus en plus, les outils commerciaux tendent à pêcher par lourdeur, manque de sens de la simplicité et de l'efficacité. Dans le monde de l'Open Source c'est différent. J'utilise volontiers ces outils, non parce qu'ils sont gratuits, mais parce qu'ils sont meilleurs. Quelques exemples : pour moi le meilleur langage de script c'est Python. J'ai souvent avancé à écrire un script en Python avec mon

vieux XEmacs (autre outil gratuit) qu'en VB avec Visual Studio .Net. Pour moi, le meilleur serveur Web c'est Zope. J'utilise volontiers Jython, que les produits commerciaux Java ignorent superbement. La meilleure librairie GUI est certainement wxWidgets. Les IDE gratuits sont moins puissants que les commerciaux et cela impose parfois un peu de travail supplémentaire, quoique pas toujours, comme je le disais à propos de Python. En revanche la gratuité du support technique est cette fois un vrai argument. Une question dans une liste de diffusion et vous avez la solution à votre problème dans l'heure.

Studio .Net http://www.activestate.com/Products/Visual_Python/

- Komodo, IDE <http://www.activestate.com/Products/Komodo/>
- BlackAdder, IDE <http://www.thekompany.com/products/blackadder/>
- Zeus, IDE Shareware <http://www.zeusedit.com/python.html>
- Arachno, IDE http://www.scriptolutions.com/arachno_python.php

Le pour : logithèque assez étoffée gratuite ou commerciale, puissance du langage

Le contre : manque de reconnaissance en France

Quelles applications ? : applications web, site dynamique

API, frameworks, librairies

Aujourd'hui, difficile de ne pas utiliser des frameworks techniques, de développement, de mapping, de tests, métier, etc. Privilégiez les frameworks reconnus du marché. Vous aurez la sécurité et un fort retour d'expérience. Car, utiliser un framework que l'on ne connaît pas pose toujours un problème au début. L'utilisation d'un framework libre permet aussi de vous rendre indépendant d'un éditeur. Dans le cadre d'un projet, c'est important, et le framework, une fois maîtrisé, économise du codage... Il existe des framework pour les principaux langages (Java, .NET, PHP...). Les IDE implémentent souvent plusieurs framework Open Source en standard. Les principaux frameworks Java/J2EE ont aussi l'avantage d'être disponibles sous .NET.

Nom framework	Langage	Commentaires
Copix http://copix.aston.fr/	PHP4	Pour une organisation modulaire d'une application PHP
Hibernate / Nhibernate http://www.hibernate.org/	Java / .NET	Mapping O/R
Avalon http://avalon.apache.org/	Java	Développement de composant serveur
Struts http://struts.apache.org/	Java	Gérer les flux des applications. JSF de Sun le remplace
Cocoon http://cocoon.apache.org/	Java	Développement Web, implémentation XML
Aukyla PHP Framework http://aukyla.sourceforge.net/	PHP5	Modulaire et orientée objet. Basé sur le widget
Seagull Framework http://seagull.phpkitchen.com/	PHP	Orienté objet, couche de présentation, accès aux données, etc.
Apache AxKit http://axkit.org/	Perl	Développement applications web
Spring http://www.springframework.org/	Java / .NET	Orienté AOP
PEAR http://pear.php.net/	PHP	Framework technique
JAC http://jac.objectweb.org/	Java	Programmation aspect
Junit / Nunit	Java / .NET	Tests unitaires
Cactus http://jakarta.apache.org/cactus/	Java	Tests
Enhydra http://www.enhydra.org/	Java	Développement application web
Beehive http://incubator.apache.org/beehive/	Java	Développement d'applications J2EE. Fondation de BEA WebLogic Workshop
IEWeb Controls http://www.asp.net/IEWebControls/Download.aspx?tabindex=0&tabid=1	.NET	Contrôles ASP.NET
Morrison Schwartz http://www.ms-inc.net/	.NET	Contrôles .NET en freeware, sources disponibles.

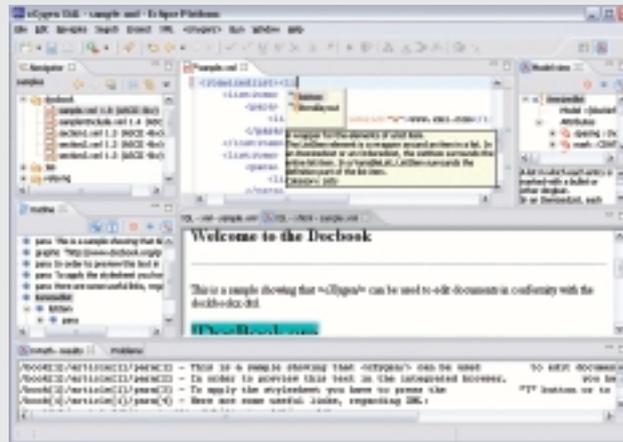
Le pour : diversité, indépendance des couches

Le contre : apprentissage au début des librairies, composants...

Quelles applications ? : majoritairement orienté application web et middleware.

■ François Tonic

Oxygen, éditeur XML shareware



Situé à Craiova, en Roumanie, SyncRO Soft a été créé en 1998 par un groupe d'étudiants en école d'ingénieurs en informatique. Utilisateurs de XML, ils ont commencé à développer début 2002, oXygen, un éditeur XML généraliste. Sylvain Perchaud a rencontré pour Programmer George Christina Buna, fondateur de SyncRO Soft.

D'où vient l'efficacité d'oXygen ?

Nous n'avons pas seulement implémenté des choses pour avoir une longue liste de fonctions. Nous nous sommes focalisés sur l'implémentation la meilleure, la plus propre et facile de chaque fonction offerte par oXygen. Chaque fonction est conçue avec attention, implémentée et paramétrée afin d'assurer qu'elle offre non seulement la fonctionnalité désirée, mais aussi qu'elle l'offre de façon efficace et aisée.

Quel langage, IDE, bibliothèques utilisez-vous pour oXygen ?

Depuis le début, nous avons voulu créer une application multi-plate-forme et c'est pourquoi nous avons décidé d'utiliser Java. Nous nous sommes aussi focalisés sur le support de quelques plates-formes spécifiques, par exemple oXygen sur Mac OS X ressemble vraiment à une application Mac native. Pour le développement nous utilisons principalement Eclipse, d'où la disponibilité d'oXygen en tant qu'application indépendante mais aussi en tant que module pour Eclipse. Nous utilisons de nombreuses bibliothèques, la plupart provenant de la fondation Apache, mais aussi Saxon qui est à la source de quelques importantes fonctions d'oXygen (XQuery, XPath 2.0, XSLT 2.0, débogage XSLT 2.0).

Comment gérez-vous le travail en équipe pour le développement de l'application ?

Nous utilisons des outils comme CVS et Bugzilla. Chaque bogue ou fonction a sa place sur Bugzilla où, avec la description et diverses autres données, nous enregistrons la version et l'estimation de temps restant, afin que nous ayons à tout moment une estimation pour la prochaine version.

■ Propos recueillis par Sylvain Perchaud

Europe Shareware - <http://www.europe-shareware.org>

Open Wide : Les pragmatiques du libre

Open Wide ne fait pas partie de la première génération de SSL, mais avec moins de fonds au départ, elle s'est construite sur un modèle sans doute un peu plus mature. Créée en 2001 par Patrick Bénichou, venu du monde des services informatique et non de la technologie pure et dure, elle est aujourd'hui bénéficiaire et touche de nouveaux types de clients.

L'activité d'Open Wide a démarré en pleine crise du secteur, en octobre 2001. Dès le départ, fort de son expérience dans le secteur des services " traditionnels " (voir encadré), Patrick Bénichou décide de se positionner auprès des grands comptes, jusque là peu intéressés par l'Open Source. Ayant travaillé dans le monde des services, et conscient qu'il y avait une " énorme déperdition de savoir-faire " du fait des affectations " drivées par l'activité et non le savoir-faire ", il se fixe comme but de " capitaliser sur la technologie et sur les hommes ". Concrètement : adopter un positionnement de niche, ne pas répondre à tout. Surfer sur la vague de l'Open Source, dans le sillage des grandes annonces d'IBM notamment, mais sur le marché traditionnel des sociétés de services, à savoir essentiellement les grands comptes. Les premières affaires sont davantage des missions d'évangélisation que des projets opérationnels, indique-t-il. L'année 2002 fut d'ailleurs difficile (500 KE de CA), la société comptait alors moins de 10 personnes. Le véritable développement intervient en 2003 (1,5 M€ de CA) et une deuxième étape fin 2003, avec une offre d'infogérance.

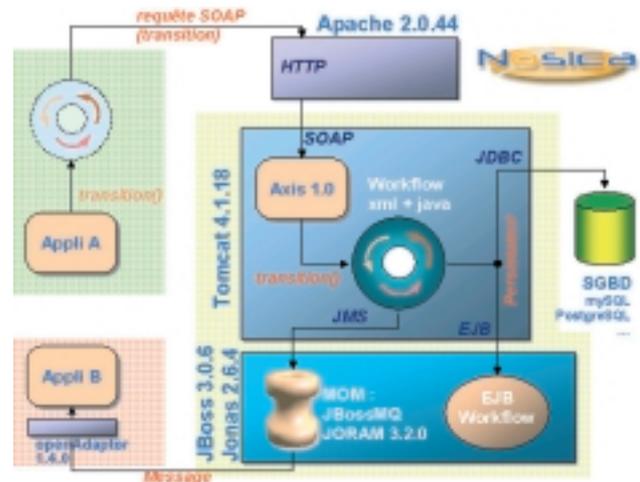
Jusqu'ici société d'architecture et d'intégration, OpenWide a racheté Accelance MSP pour répondre aux demandes de ses clients concernant l'hébergement, le support, l'administration et la supervision de plateformes internet. Bien que non issu du monde Open Source, Patrick Bénichou s'est entouré de figures reconnues de ce milieu, dont au poste de directeur technique, Pierre Ficheux, auteur d'un ouvrage sur Linux Embarqué et de nombreux articles autour de Linux.

Une offre structurée

Open Wide compte trois pôles d'activités, correspondant à trois types de clients, et trois types d'interlocuteurs au sein des sociétés clientes, explique Patrick Bénichou :

- Solutions industrielles et embarquées
- Urbanisation des systèmes d'information
- Portails collaboratifs et gestion de contenu.

Côté ressources humaines, open Wide est en train de prendre un virage, celui des compétences métiers. Après avoir bâti une équipe sur des critères technologiques et misé sur ce dif-



férentiel techno, Patrick Bénichou souhaite s'appuyer sur des arguments métiers. La société a ainsi récemment recruté un spécialiste du secteur de l'assurance. La société recrute assez peu, compte tenu de sa taille (15 personnes en 2004, 20 à 25 personnes en 2005), mais des profils pointus avec trois ans d'expérience minimum, et souvent diplômés d'écoles d'ingénieurs du groupe 1, mais aussi de l'Epita ou de l'UTC (Compiègne). Connaissant l'Open Source bien sûr, mais également des spécialistes Java.

La plupart des collaborateurs sont des architectes et intégrateurs Open Source, capables de conseiller le client sur les briques à utiliser, et donc d'évaluer ces briques, mais aussi de les installer et de concevoir des prototypes, ce qui suppose de savoir lire le code. Patrick Bénichou souligne recruter des " passionnés de technique, pas autistes " : ils doivent être capables de discuter avec le client. Un des profils recherchés est celui de spécialiste en ingénierie de processus.

La société ne compte en revanche pas de " bataillons de développeurs " et préfère sous-traiter en régie la partie du développement plus " banalisée ". ■ Carole Pitras

Patrick Bénichou, une carrière de services



Rien ne prédestinait Patrick Bénichou, ingénieur de formation, à tomber dans le bain du libre, souvent chasse gardée de " gourous " de l'Open Source. Il a réalisé toute sa carrière dans le monde des services informatiques, chez Cisi d'abord (ensuite racheté par CS), où, dit-il, il a touché à tous les métiers, puis chez Thales, où il a créé l'agence de Lyon. Lorsqu'on lui propose de prendre en charge le département industrie sur Paris, il refuse et décide de créer sa société, dans l'Open Source, car il souhaitait donner à sa société un positionnement technologique fort. Il cherche des business angels et trouvera finalement comme partenaire financier Thales, intéressé par l'Open Source dans le cadre de systèmes embarqués, où la maîtrise du code est jugée stratégique... Open Wide sera donc la spin-off du groupe Thales dédiée à l'Open Source. Le capital a été ouvert en avril 2002 à Schneider Electric Ventures.

Open Wide en bref

4,5 M€ de chiffre d'affaires prévu en 2004, contre 1,5 M€ en 2003.

Présente à Paris et Lyon

50 collaborateurs, 33 chez Open Wide,

17 chez Accelance, 60 % à Paris, 40 % à Lyon

XUL et XAML : la prochaine génération d'interface sera XML

Les technologies de l'Internet ont complètement transformé la façon de penser les IHM (Interface Homme Machine) et l'ergonomie des applications. Ainsi, les applications client/serveur basées sur des interfaces riches, intégrant de la logique métier ont laissé largement la place à des interfaces légères basées sur HTML. Cependant, avec le temps, les applications Web se sont complexifiées, et les contraintes des interfaces en HTML montrent de plus en plus fréquemment leurs limites.

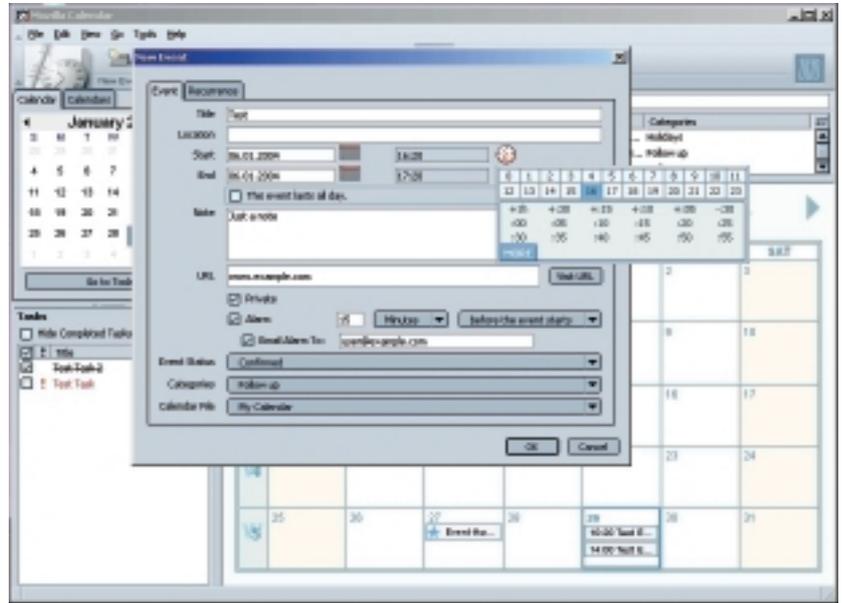


Figure 1 : Mozilla Calendar, une application XUL complète.

Dès lors, les attentes en terme d'interfaces homme-machine évoluent vers des applications dites " riches " et téléchargeables à la demande.

Pour répondre à cela, deux solutions s'appuyant sur XML se positionnent pour apporter une nouvelle topologie d'IHM :

- d'un côté, la fondation Mozilla propose le langage XUL (eXtensible User-interface Language), mis au point et utilisé depuis déjà quelques années pour le développement du navigateur éponyme,
- de l'autre côté, Microsoft met en avant XAML (eXtensible Application Markup Language), annoncé dans Longhorn, la prochaine version de Windows.

L'annonce faite par Microsoft, il y a plusieurs mois, de la technologie XAML a été ressentie comme un coup de pied dans la fourmière du monde open-source, qui n'arrivait pas à arrêter une technologie unique pour la définition d'interfaces graphiques.

La menace brandie par Microsoft et XAML a ainsi déclenché une vive réaction des acteurs de la scène open source, provoquant certaines

alliances (notamment, parmi les plus significatives, entre Mozilla et Gnome), ce qui laisse présager une vive bataille pour savoir laquelle de ces technologies s'imposera comme standard.

En effet, Microsoft est longtemps resté très discret sur le terrain des interfaces graphiques tout XML, mais ce n'est que pour mieux préparer son arrivée sur ce domaine, avec la future version de son système d'exploitation, Longhorn. Cette version du prochain Windows aura pour particularité de reposer sur une série de composants entièrement basés sur le Web. Le framework .Net dans sa version 2.0, associé à XAML a d'ailleurs été créé dans ce but.

La fondation Mozilla est désormais tournée vers son projet Mozilla 2. Ce projet n'est pas le nom de son prochain navigateur, mais celui d'une plate-forme de développement d'applications, orientée Web.

XUL

XUL a été mis au point par la fondation Mozilla il y a déjà plusieurs années. Le navigateur du même nom s'appuie complètement sur XUL et

se compose d'un ensemble de composants graphiques pré développés, et d'un interpréteur. A ce jour, l'ensemble de ces briques s'installe une seule fois sur le poste client, lorsque l'utilisateur installe le navigateur sur son poste de travail.

Par la suite, le déploiement d'une nouvelle application XUL nécessite simplement de cliquer sur un lien ou d'accéder à un URL.

La création d'un fichier XUL qui décrit une application peut s'effectuer indifféremment en statique, ou en dynamique côté serveur à partir de n'importe lequel des langages adaptés (JSP, PHP, Perl, et même ASP).

La réalisation d'applications XUL est souvent associée, à tort, à un usage contraint au sein même du navigateur Mozilla (ceci n'est en réalité valable que pour les applications Web).

Or, il n'en est rien, car à partir du moment où Mozilla est installé sur un ordinateur, l'environnement d'exécution est présent et peut être utilisé indépendamment.

Les composants XPCOM et XPConnect permettent d'utiliser Javascript pour accéder à des bibliothèques C/C++ ou Java depuis une appli-

cation XUL, ce qui autorise (comme avec le navigateur Mozilla) le développement d'applications basées sur XUL, totalement indépendantes du navigateur et surtout, du système d'exploitation et de la plate-forme sur laquelle elles s'exécutent.

Dès lors, à partir du moment où Mozilla est installé sur un système, les applications développées sur ce modèle (sous réserve de compatibilité du code C/C++ ou Java utilisé) pourront fonctionner de façon indépendante et portable.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
  id="findfile-window"
  title="Find Files"
  orient="horizontal"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">

  <button id="find-button" label="Find" default="true"/>
  <button id="cancel-button" label="Cancel"/>
</window>
```

Exemple 1 : définition d'une IHM avec XUL

XPCOM

XPCOM, est l'élément central de la technologie XUL. Le nom n'est pas sans rappeler la technologie COM de Microsoft.

XPCOM ne se limite pas à un langage précis, ni à une plate-forme particulière, comme le fait le modèle COM de Microsoft.

XPCOM se veut être une architecture open source de développement multi-plate-forme, et utilise un ensemble de bibliothèques internes pour gérer ses composants.

A l'origine, la technologie XPCOM a été développée par les équipes de Netscape, dans le cadre du projet Mozilla, avant d'être rendue indépendante du fait de l'étendue des possibilités qu'elle offrirait.

XPCOM permet ainsi au développeur de réaliser des développements allant bien au-delà de ce que peut proposer JavaScript, et permet, grâce aux composants existants, de réaliser des connexions HTTP ou FTP, de consommer des services Web avec SOAP, d'effectuer des accès vers une base de données relationnelle, etc.

XPCOM

XPCOM assure la liaison entre le Javascript, utilisé pour le développement des applications, et les composants XPCOM disponibles, et permet de les manipuler de manière transparente.

XBL

XBL (eXtensible Binding Language) est, comme son nom l'indique, un langage de liaison qui se veut extensible.

Il s'agit d'une grammaire XML qui décrit les relations susceptibles d'être attachées à d'autres documents, à l'aide d'une feuille de style CSS, ou bien de DOM.

Ainsi associé à une application XUL, il permet d'ajouter des comportements spécifiques à certains composants qui peuvent ensuite être transposés sous la forme d'éléments d'interface réutilisables

XBL peut être comparé à XSLT, à la différence près que XSLT ne peut travailler que sur un document statique, tandis que XBL peut agir à la volée sur les éléments du DOM. (Fig. 1)

XAML

XAML, ou eXtensible Application Markup Language est une spécification propriétaire, autour de laquelle Microsoft construit en grande partie Longhorn.

Le langage XAML est un élément de la couche de présentation Avalon de Longhorn.

Cette couche assure la gestion et l'intégration de l'interface graphique. Elle permet ainsi de supprimer, au sein du système d'exploitation de Redmond, la frontière technique existant entre les applications Web, et les applications Windows natives.

Un autre apport intéressant d'Avalon concerne l'utilisation beaucoup plus importante de DirectX, ce qui permettra au système d'exploitation de mieux profiter des cartes graphiques évoluées (notamment avec l'apparition de plus en plus fréquente des formats d'image vectoriels).

Les principes retenus pour la conception de XAML sont simples, puisqu'ils reprennent les nomenclatures déjà utilisées au sein du framework .NET :

- les noms des classes du framework .NET sont repris pour devenir des balises XML,
- les noms des propriétés des classes du framework .NET deviennent des attributs XML,
- la hiérarchie des composants visuels de Microsoft .NET est reprise et transcrite en XML par l'imbrication des balises.

Dès lors, grâce à XAML, il est possible de créer un fichier XML qui définit la structure visuelle de votre application (positionnement des boutons, des zones de texte, des images, des listes déroulantes).

Ensuite, en utilisant un langage de programmation (indifféremment CSharp ou VB.Net), il est possible de mettre en place toute la logique applicative.

XAML autorise aussi le fait de laisser le code événementiel au cœur du fichier XAML à l'intérieur d'un bloc CDATA; dès lors, un simple double clic sur le fichier lance l'application, et la compilation s'effectue alors en arrière-plan. La création d'une page XAML est relativement simple :

```
<Canvas
  xmlns="http://schemas.microsoft.com/2003/xaml"
  Background="LightCyan"
  Width="100%" Height="100%">

  <Image Source="lh.bmp" Canvas.Left="5"
  Canvas.Top="5" />
  <Text Canvas.Left="90" Canvas.Top="20"
  FontSize="36">Hello, Longhorn!
</Text>
</Canvas>
```

Exemple 2 : définition d'une IHM avec XAML

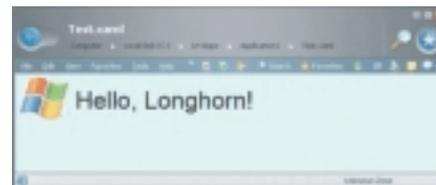


Figure 2 : exemple d'application XAML

De même, l'ajout d'un bouton dans une page est facile à réaliser :

```
<Canvas xmlns="http://schemas.microsoft.com/2003/xaml">
  <Button Canvas.Left="10"
  Canvas.Top="10"
  Width="90px"
  Height="32px">Click Me</Button>
</Canvas>
```

En C#, pour réaliser la même chose, il faudrait écrire les lignes de code suivantes :

```
Button btn = new Button();
btn.Width = new Length(90);
btn.Height = new Length(32);
Canvas.SetTop(btn, new Length(10));
```

```
Canvas.SetLeft(btn, new Length(10));
btn.Content = "Click Me";
```

L'apport de la technologie XAML dans le domaine du développement Windows est relativement important, puisqu'à l'image du développement Web, il autorise la réalisation de l'interface graphique par un designer, alors même que les développements de la partie applicative sont en cours, sous réserve bien évidemment de s'être préalablement entendu sur le nommage des contrôles et propriétés de cette dernière.

Un autre apport significatif de XAML dans le monde Windows, est qu'il permet de laisser à l'utilisateur la possibilité de personnaliser son interface graphique comme bon lui semble, soit en changeant les fichiers XAML qui contiennent uniquement la partie graphique, soit en autorisant, au sein de l'application, la possibilité d'éditer les contrôles, afin de pouvoir les déplacer, les redimensionner, en changer l'apparence, etc. Cette fonctionnalité nouvelle au sein du système d'exploitation évite d'avoir à réaliser des développements spécifiques et parfois complexes pour personnaliser une application.

Conclusion

Deux types d'interfaces vont s'affronter techniquement ces prochaines années, pour tenter de faire de l'une d'elle la référence dans le domaine de l'IHM tout XML : d'un côté, XUL portée par la communauté open source, de l'autre XAML portée par Microsoft. Ces deux langages vont s'affronter bien au-

Tableau Comparatif XUL/XAML

	XUL	XAML
Editeur	Mozilla Foundation (fondation Open Source)	Microsoft
Licence	MPL (Mozilla Public Licence)	Propriétaire Microsoft
Disponibilité	Depuis mars 1999	Courant 2006
Portabilité	Multi-plate-forme (Windows, Linux, Mac OS)	Longhorn et versions précédentes
Standard	Non	Non
Rendering	Mode points	Vectorel
Mode d'exécution	Interprété	Compilé
Exemples d'implémentation	Mozilla Firefox Thunderbird Calendar Nvu Etc.	Windows Longhorn

délà du Web, puisque chacun d'entre eux cherche à se positionner comme un futur standard universel pour la création d'un site Internet, d'une application locale ou d'une application Web.

XUL a l'avantage de l'âge et de la maturité; avec les alliances stratégiques annoncées récemment, on peut sans crainte, prédire que le langage sera adopté unanimement par la communauté open source.

XAML, quant à lui, aura pour avantage d'être complètement intégré à Longhorn, avec une fusion totale d'Internet Explorer dans le système d'exploitation, ce qui fera de XAML le langage des futures applications Windows.

La création d'interfaces à partir de XML est ainsi un des principaux enjeux de l'informatique des prochaines années. Ce n'est donc pas sans raison que Microsoft investit lourdement dans ce domaine. Les retombées sur le

développement rapide et la portabilité des applications sur les terminaux mobiles, ou sur la possibilité de décliner rapidement une application en fonction de ses cas d'utilisation (Web, Windows, ...) sont à la clé.

XUL est aujourd'hui opérationnel et a déjà bien fait ses preuves (Mozilla/Firefox sont en progression auprès des utilisateurs, et quelques entreprises s'intéressent à XUL pour leurs applications internes), alors que XAML n'arrivera probablement pas avant 2006 avec la nouvelle version de Windows.

La bataille s'annonce féroce, mais il est difficile de présager laquelle des deux solutions sera à même de s'imposer.

■ Mickael MAINDRON

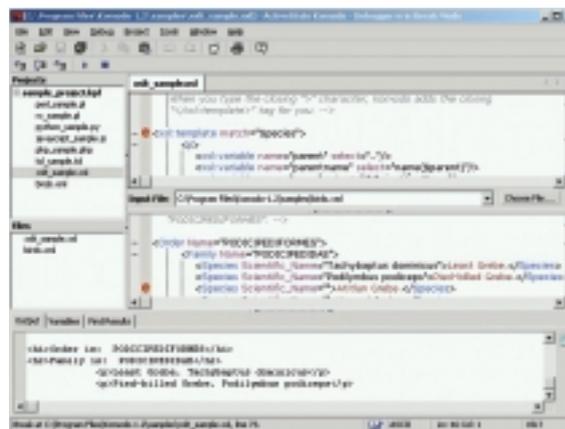
Architecte chez NEOXIA, intégrateur de systèmes spécialisé dans les nouvelles technologies de l'information.

mickael.maindron@neoxia.com



La guerre des standards

XUL, XAML, deux technologies basées sur XML pour la description de l'interface, deux conceptions différentes du monde. XUL appartient au monde libre, XAML gravite dans l'univers Microsoft et apparaît comme une technologie propriétaire, même si ce dernier point demeure incertain, XAML ne devant pas être disponible avant 2006. Si Avalon (utilisant XAML) devait être uniquement disponible sous Longhorn, Microsoft a décidé de le porter sous Windows XP et Server 2003. Si XUL a, en théorie, le temps pour s'imposer, il



Le projet Luxor

demeure encore peu utilisé (même s'il l'est fortement dans Mozilla) et il doit absolument convaincre les développeurs et les éditeurs de supporter la technologie. Cependant, XAML a

des chances de supplanter XUL si celui-ci n'arrive pas à conquérir le marché. XAML bénéficie d'une intégration dans Windows, .NET et les outils de développements. Une communauté XAML commence à se former et à sortir des outils Open Source ou commerciaux (ex. : XAMLON). XUL et XAML poursuivent le même but : établir un nouveau modèle descriptif de l'interface (avec XML) sans changer l'ergonomie utilisateur. Surtout, on bénéficie (du moins avec XUL) d'une interface multi-plate-forme découpant le contenant, du contenu (fonctionnel, données). ■ F.T.

Spring et Hibernate : une alternative aux EJB CMP

Spring Framework couplé à Hibernate se présente de plus en plus comme une alternative simple et efficace aux EJB. À travers cet article, nous allons comparer les mécanismes de persistance transactionnelle des deux solutions.

Spring

Spring se contente de fournir un socle de base, sur lequel il sera possible de greffer des composants à la demande. Spring se positionne comme un framework minimaliste avec un container, autour duquel va s'articuler un ensemble non exhaustif de briques logicielles. Le container de Spring s'appuie sur le design pattern d'inversion de contrôle :

- le container se charge d'instancier les composants
- il résout les dépendances des composants et leur transmet les instances d'éventuels autres composants qui seraient nécessaires à leur bon fonctionnement interne

Lors de son exécution, un composant n'a plus besoin d'effectuer des appels à une 'factory' ou un service de 'lookup' pour demander des instances dont il aurait besoin.

L'loC introduit un contexte favorable pour la séparation entre les interfaces et implémentations (principe de 'Separation of Concern'). Les composants présentent et manipulent uniquement des interfaces, le container fournit les implémentations adéquates, telles que définies dans les fichiers de configuration.

Spring s'appuie également sur son container pour introduire la notion de programmation orientée aspect. Les instances produites par le container peuvent être enrichies d'une couche logicielle qui sera capable d'intercepter les appels aux méthodes (mécanisme de proxy). Cette fonctionnalité va permettre à Spring d'introduire en toute transparence des mécanismes d'interception d'appel sophistiqués, capables de prendre en charge des notions telles que la propagation/démarcation transactionnelle avec niveau d'isolation.

Fort du potentiel de son container (qui intègre les mécanismes de loC, de SoC et AoP), Spring intègre, entre autres, dans sa version 1.1 les services suivants :

- Persistance avec les implémentations Hibernate, JDO et iBatis

- Gestion des DAO
- La transaction/isolation reposant sur un mécanisme d'interception (AoP) et de propagation de contexte
- MVC avec les frameworks web (Struts 1.1, Webwork2)

Une application sera libre d'utiliser Spring dans une configuration dépourvue de toutes fonctionnalités annexes, d'utiliser une grappe de services pour ses besoins, ou encore de développer ses propres composants services. Au cours de cet article, nous allons particulièrement nous intéresser à la gestion transactionnelle de la persistance avec Hibernate.

Container EJB / CMP

Un container d'EJB gère le cycle de vie des instances des objets dit 'Sessions' et 'Entités'. Les entités représentent une donnée métier sous forme d'objet persisté en général dans un SGBDR. Les objets entités se déclinent en deux familles, les BMP (Bean Managed Persistence) qui prennent en charge eux même leur persistance, et les CMP (Container Managed Persistence) qui délèguent au container la gestion de la persistance.

Dans le cas des Entités CMP, les liens entre les colonnes d'une table et les attributs d'un objet sont décrits à travers un fichier de mapping. Les relations entre objets pourront être reflétées à travers des jointures entre tables.

La norme EJB introduit une notion de schéma de données abstrait qui permet de définir une structure logique de base de données reposant sur plusieurs schémas réels de base. Cette fonctionnalité permet par exemple de distribuer en toute transparence les données entre plusieurs bases, ou encore de se calquer simplement sur des bases existantes en utilisant un schéma différent du modèle de données métier.

Le rôle du container sera d'assurer automatiquement la persistance et le chargement des objets sous sa responsabilité. Par ailleurs, le

container fournira une panoplie de services techniques critiques (transaction, sécurité, etc.).

Hibernate et Spring Mapping Objet/Relationnel Hibernate

Hibernate est une librairie de Mapping Objet/Relationnel. Hibernate propose une approche faiblement intrusive, fortement basée sur les mécanismes d'introspection et d'interception/proxy de classe. Le gestionnaire de persistance est capable de stocker une instance d'un objet simple à l'aide d'un fichier de mapping et en inspectant les valeurs et les relations de l'instance.

Hibernate est pratique à utiliser, car il ne nécessite aucune contrainte au niveau de la structure de l'objet à persister et supporte toutes les notions objets du langage Java (héritage, classe abstraite, composition/agrégation, etc.).

Pour donner une idée de la différence entre les EJB CMP et le mécanisme O/R Hibernate, nous présentons à travers le tableau ci dessous, un exemple minimaliste d'implémentation avec le mapping pour les deux technologies.

Là où les EJB requièrent 3 classes spécifiques, avec au niveau de l'implémentation la déclaration de méthodes propre au cycle de vie de l'objet, Hibernate se contente d'une seule classe simple.

Même si à l'exécution le résultat est relativement proche, le nombre de classes à écrire et de déclarations pour la gestion des exceptions, pour la prise en compte des cycles de vie, etc. est plus important. La plupart des environnements actuels de développement facilitent la génération de code pour les EJB, cependant la solution proposée par Hibernate reste plus simple.

Spring

Du point de vue du code, la responsabilité du développeur dans la gestion de persistance se limite (dans les grandes lignes) à la déclaration programmatique des sessions, des transactions, la sauvegarde/mise à jour des instances et le chargement des données à l'aide de requêtes de l'API d'Hibernate.

Spring propose, à l'aide de 'template', de

Classes EJB CMP

```
public interface MyClass extends EJBObject
{
public Integer getld() throws
RemoteException;
public String getNom()throws
RemoteException;
}

public interface MyClassHome extends EJBHome {
public MyClassEJB create ( Integer id, String nom )
public MyClassEJB findByPrimaryKey (Integer id) throws
FinderException, RemoteException;
}

public abstract class MyClassEJB implements EntityBean
{

public abstract Integer getld ();
public abstract void setld ( Integer id );

public abstract String getNom();
public abstract void setNom( String nom );

public Integer ejbCreate ( Integer id, String nom )
throws CreateException
{
setld( id );
setNom( nom );
return null;
}

public void setEntityContext (EntityContext context){}
public void unsetEntityContext (){}
public void ejbActivate (){}
public void ejbPassivate (){}
public void ejbRemove ()
throws javax.ejb.RemoveException{}
public void ejbLoad (){}
public void ejbStore (){}
public void ejbPostCreate ( Integer id, String nom ){}
}
```

Classe O/R Hibernate

```
public class MyClass
{

private Integer m_id;
private String m_nom;

public Integer getld(){
return m_id;
}

public void setld( Integer id){
m_id = id ;
}

public String getNom(){
return m_nom;
}

public void setNom( String nom ){
m_nom = nom;
}
}
```

Mapping EJB CMP

```
<entity>
<ejb-name>MyClass</ejb-name>
<home>MyClassHome</home>
<remote>MyClass</remote>
<ejb-class>MyClassEJB</ejb-class>
<persistence-type>Container</persistence-type>
<prim-key-class>java.lang.Integer</prim-key-class>
<reentrant>False</reentrant>
<cmp-version>2.x</cmp-version>
<abstract-schema-name>MyClass</abstract-schema-
name>
<cmp-field><field-name>nom</field-name></cmp-field>
<prim-key-field>id</primkey-field>
</entity>
```

Mapping Hibernate

```
<class name="MyClass">
<id
name="id"
type="java.lang.Integer"
>
<generator class="increment" />
</id>

<property
name="nom"
type="java.lang.String"
/>
</class>
```

prendre en charge proprement le code technique répétitif : ouverture de session, début de transaction, commit de transaction et fermeture de session avec gestion des exceptions.

Les templates fournissent un contexte propre et encapsulent l'exécution de codes faisant appel aux API d'Hibernate.

Transaction avec Spring

Le mécanisme d'AOP intégré à Spring va permettre de définir des interceptions d'appels de méthodes en charge de gérer les transactions, indépendamment du mécanisme de persistance (Hibernate, JDO, etc.).

À travers la configuration d'un composant, il est possible de déclarer les méthodes cibles faisant état d'une interception. Tout appel à ces méthodes est redirigé vers une autre méthode d'un composant dédié à la mise en place et la gestion d'un contexte transactionnel. De même, le retour de la méthode cible sera également pris en charge pour, par exemple, supprimer le contexte transactionnel. Dans le cas de l'utilisation avec Hibernate, l'intercepteur concerné est également capable de prendre en charge création/fermeture d'une session fournie par le gestionnaire de persistance Hibernate.

Exemple de configuration Spring :

```
<bean id="mainService" class="MainService">
<property name="sessionFactory"><ref
local="sessionFactory"/></property>
</bean>

<bean id="mainServiceTx"
class="org.springframework.transaction.
interceptor.TransactionProxyFactoryBean">
<property name="transactionManager"><ref
bean="transactionManager"/></property>
<property name="target"><ref bean="main
Service"/></property>
<property name="transactionAttributes">
<props>
<prop key="save*">PROPAGATION_
REQUIRED</prop>
</props>
</property>
</bean>

<bean id="sessionFactory" class="org.springframe
work.orm.hibernate.LocalSessionFactoryBean">
<!--configuration Hibernate -->
```

```
...
</bean>
```

Ici, nous définissons pour la classe 'MainService' un proxy qui se chargera d'intercepter les appels aux méthodes qui commencent par 'save' ('save*'). Cet intercepteur doit créer un contexte transactionnel et le propager jusqu'au retour de la méthode (PROPAGATION_REQUIRED).

```
public class MainService extends Hibernate
DaoSupport{

    public void service( String agence
Name ) throws Exception {
        Agence agence = new Agence();
        agence.setName( agenceName );

        saveAgence( agence );
    }

    private void saveAgence( Agence agence )
throws Exception{
        getHibernateTemplate().saveOrUpdate
( agence );
    }
}
```

Au niveau du code, la méthode publique 'service' se charge d'effectuer quelques opérations et appelle la méthode saveAgence. L'appel est donc intercepté pour créer la session et la transaction nécessaire à la mise à jour. Une fois dans la méthode de saveAgence, l'instance 'agence' est sauvegardée. Au retour, la transaction est validée dans l'intercepteur et la session clôturée.

Ici, toutes les opérations techniques fastidieuses sont prises en charge par l'intercepteur.

Gestion de la sécurité avec Spring

À l'instar du mécanisme de transaction, un intercepteur peut être mis en place pour vérifier l'autorisation d'accès aux méthodes. Dans une approche couplée avec le mécanisme d'authentification de JAAS (Java Authentication Autorisation Service), il est possible d'associer l'utilisateur authentifié (Subject/Principal) au contexte d'exécution courant et de s'assurer, lors d'une interception, que l'accès à la méthode cible est autorisé. Par ailleurs, toujours sur le même principe, il

est également aisé, lors de l'interception, de filtrer les données de retour, en fonction de l'utilisateur authentifié (par exemple, supprimer les éléments d'une liste qui ne devrait pas être transmis à un utilisateur sans les privilèges adéquats).

Gestion de l'accès à distance

Toujours sur le principe de la programmation par aspect (AoP), Spring va permettre de définir un proxy pour un objet qui sera en charge, du côté client, de simuler l'exécution locale d'un objet physiquement présent sur un serveur. Et du côté serveur, un intercepteur pourra déléguer l'exécution du service à l'implémentation requise.

Spring intègre des intercepteurs s'appuyant sur le protocole RMI en toute transparence : aucune différence pour le développeur, l'objet reste tel quel. Le mécanisme d'interception s'applique à n'importe quel objet et pourra déporter l'exécution sur une jvm distante.

Bilan

Spring arrive à couvrir le même domaine fonctionnel qu'un container d'EJB gérant des entités CMP et ce, en offrant :

- La simplicité : la barrière d'entrée pour l'utilisation de Spring est faible, la persistance avec Hibernate coule de source, la documentation sur Spring et Hibernate est accessible et très complète (avec une communauté très active)
- La modularité : les composants sont interchangeableables (le mécanisme de persistance peut être changé à moindre coût)
- Aspect : les fonctionnalités (transaction, remote, sécurité) sont proposées en toute transparence, sous forme d'aspect, ce qui n'impacte quasiment pas sur le code existant

Cependant...

- Spring n'est pas un standard, contrairement aux EJB ; et à ce titre est plus sujet à des variations rapides de spécification
- La richesse de Spring permet différentes approches pour un même problème et une prolifération d'astuces et de composants en tous genres au sein d'un même projet
- Les CMP utilisent un schéma abstrait de base de données qui offre une grande souplesse pour intégrer une application sur une infrastructure existante
- Spring n'offre pas de mécanisme simple de clustering et de tolérances aux pannes

(même si pour la plupart des implémentations EJB cette fonctionnalité est relativement réduite).

Conclusion

Spring ouvre une nouvelle voie dans le développement d'applications J2EE. Couplé avec Hibernate il réussit le pari de réduire singulièrement la complexité de mise en œuvre d'une application intégrant un mécanisme de persistance transactionnelle. Le potentiel de Spring reposant sur une utilisation simple des techniques de programmation aspect, offre des perspectives quasi illimitées.

En comparaison, les EJB CMP accusent le coup, avec leur héritage historique d'une norme maintenant vieille de plus de 6 ans (ce qui est tout à leur honneur). Heureusement, la norme 3.0 s'oriente fortement vers un mécanisme de mapping basé sur JDO (Java Data Object), aux concepts similaires à Hibernate, et contribuera à simplifier le développement, tout en respectant des standards. De son côté, Hibernate a rejoint le JBoss Group (JBoss étant une implémentation Open Source des EJB) et devrait, dans sa version 3, proposer une couche d'abstraction pour permettre une utilisation à travers une API JDO.

■ Alexis Agahi

Expert XML/J2EE, Objet Direct, filiale de Homsys Group

À propos de Homsys Group :

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



**Abonnez - vous
en ligne**

www.programmez.com

WINDOWS/LINUX/MAC OS X :

Construisez vos interfaces graphiques avec FLUID/FLTK et DEV-C++

Si vous ne programmez que ponctuellement en C++, et que vous désirez développer une application portable, le trinôme DEV-C++/FLUID/FLTK répondra parfaitement à vos besoins. Et c'est gratuit...

Si vous désirez programmer en C++ une application graphique (commerciale ou non) qui soit portable, une kyrielle de possibilités s'offre à vous. A la condition de bien vouloir consentir à réaliser un investissement vous pouvez évidemment choisir le leader en la matière, c'est à dire QT (dont une version est téléchargeable pour Windows, à condition que l'application ne soit pas d'usage commercial). Sinon, dans le désordre, vous pouvez vous rabattre sur WxWindows (riche en widgets), XToolkit (du projet Mozilla), V, TK, GTK+ (le toolkit de Gimp sans doute le plus populaire après QT), TK (fort utilisé par des interpréteurs comme PERL, TCL ou Python), GLOW, FOX, CORAL, etc.

Présentation de FLTK

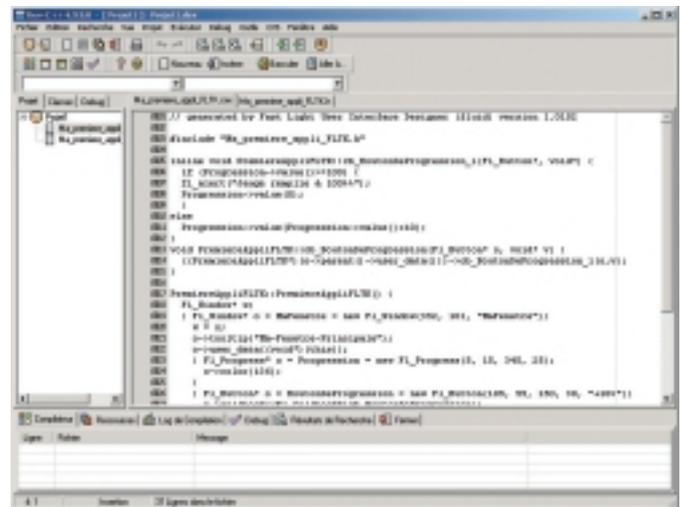
Nous allons vous présenter un choix alternatif supplémentaire qui tient la route : FLTK (Fast Light Toolkit). «Fulltick», c'est ainsi qu'il y a lieu de prononcer, est disponible sous Windows, mais aussi évidemment sous Linux et MacOS X. Ce toolkit est distribué selon les termes de la GNU Library General Public License (LGPL) version 2, ce qui signifie que vous pouvez développer une application commerciale multi-plate-forme avec FLTK, sans enfreindre les termes de sa licence.

FLTK a été conçu pour être lié de manière statique. Le code non commun aux diverses plates-formes est de l'ordre de 10 %, assurant un comportement similaire d'une grande compatibilité quel que soit l'OS cible. FLTK fait appel directement à Xlib (Unix), ou aux API WIN32 (Windows), ou encore à Carbon (Mac OS X). En résumé, vous n'avez pas besoin de vous soucier des dépendances lors du déploiement de votre application. Selon les Widgets utilisés par votre code, la taille de l'exécutable grossira de +- 150 Ko (fenêtre minimaliste), à +- 550 Ko (avec l'ensemble des Widgets).

Sous Linux, cette manière de procéder paraîtra non satisfaisante si FLTK fait partie intégrante de votre distribution favorite. Par contre, sous Windows, il y a peu de chance qu'une autre application FLTK soit déjà déployée sur vos postes clients. Quoi qu'il en soit, il est évidemment toujours possible de lier dynamiquement le toolkit à l'exécutable, mais vous serez alors contraints de vous soucier de cette dépendance, lors du déploiement de votre application.

Une installation facile réalisée avec DEV-C++

DEV-C++ est un IDE (Integrated Development Environment) de développement en licence GPL qui utilise mingw (port du compilateur gcc issu du projet GNU Compiler Collection) pour créer des exécutables win32. Nous avons téléchargé et installé la dernière version 5 (bêta). Une fois l'IDE installé, et la langue française convenablement configurée (Menu Outils/Options d'Environnement), nous pouvons installer FLTK en chargeant le package correspondant, via Internet (Menu Outils / Nouvelles versions/packages). Il s'agit ici de la version 1.1.2. Vous pouvez également l'installer directement en vous rendant sur le site de DEV-C++ et



en y téléchargeant l'exécutable du nom de fltk-1.1.2.Devpak.exe.

Avec ce package est inclus un outil graphique de création d'interface, baptisé FLUID (sous l'arborescence C:\Dev-Cpp\bin). Nous y reviendrons plus loin.

(**)* MinGW, "Minimalistic Gnu for Windows", est un compilateur C/C++ qui produit des exécutables Win32, constituant une alternative au Visual C++ de Microsoft. Son package logiciel contient un pré-processeur, un compilateur, un linker, de même qu'une très grande partie des fichiers en-têtes pour programmer sous Win32. MinGW s'exécute par la ligne de commande et c'est donc une excellente idée de l'employer avec un IDE comme Visual-mingw ou DEV-C++.

Une première application minimaliste

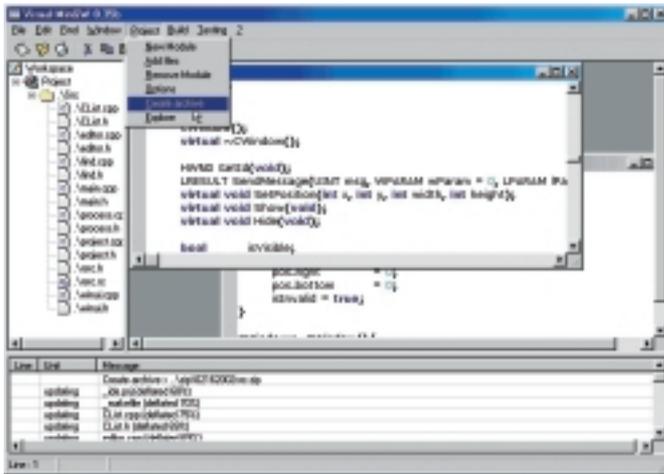
Maintenant lorsque vous créez un nouveau projet sous DEV-C++, vous pouvez choisir GUI/FLTK.

En fait, DEV-C++ ajoute le drapeau -DWIN32 comme option du compilateur et les paramètres -lfltk -lole32 -luuid -lcomctl32 -lwsck32 -lm pour l'édition des liens.

Le code suivant a été automatiquement créé. Vous pouvez commencer par le compiler et l'exécuter (touche F9) pour déterminer que tout fonctionne convenablement.

```
#include <FL/Fl.H>
#include <FL/Fl_Window.H>
#include <FL/Fl_Box.H>

int
main (int argc, char *argv[])
{
    Fl_Window *window;
    Fl_Box *box;
```



```

window = new FL_Window (300, 180);
box = new FL_Box (20, 40, 260, 100, "Hello World!");

box->box (FL_UP_BOX);
box->labelsize (36);
box->labelfont (FL_BOLD+FL_ITALIC);
box->labeltype (FL_SHADOW_LABEL);
window->end();
window->show (argc, argv);

return Fl::run();
}
    
```

N'importe quel programme utilisant FLTK doit inclure au minimum FL/Fl.H. Si vous faites appel à d'autres classes, comme c'est le cas ici, vous devez inclure les fichiers correspondants. Les classes fl_Window et fl_Box étant utilisées, le code débute par les inclusions :

```

#include <FL/Fl.H>
#include <FL/Fl_Window.H>
#include <FL/Fl_Box.H>
    
```

Pour créer un Widget, il suffit de faire appel à l'opérateur new, comme pour instancier n'importe quelle autre classe. Pour créer une fenêtre, les paramètres sont : la largeur, la hauteur et éventuellement le titre.

```

window = new FL_Window (300, 180);
    
```

Ensuite il y a création d'un cadre qui contiendra le texte statique "Hello World!" :

```

box = new FL_Box (20, 40, 260, 100, "Hello World!");
    
```

Vous pourrez toujours afficher un contenu différent avec la méthode label :

```

box->label("Bonjour !");
    
```

Les instructions suivantes modifient le type du cadre, la taille de la police, et le type de police (gras et italique), et le type d'étiquette :

```

box->box (FL_UP_BOX);
box->labelsize (36);
    
```

```

box->labelfont (FL_BOLD+FL_ITALIC);
box->labeltype (FL_SHADOW_LABEL);
    
```

Les trois dernières lignes sont importantes. Par défaut, le conteneur window est actif («en cours») jusqu'à ce que la méthode end() lui soit appliquée, ou bien qu'un autre conteneur soit créé. On termine donc l'élaboration de la fenêtre par end(), puis on l'affiche via show().

```

window->end();
window->show(argc, argv);
    
```

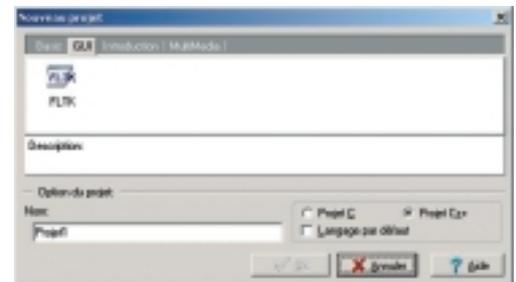
L'application appelle, en fait implicitement et de manière répétitive Fl::wait() jusqu'à ce que toutes les fenêtres soient fermées. Puis l'application peut quitter avec un return Fl::run().

Créons une interface avec FLUID

Ceci est bien beau, mais il serait beaucoup plus convivial de créer l'interface graphique à l'aide d'un outil créateur d'interfaces. FLUID (Fast Light User-Interface Designer) a été conçu dans ce but. Vous le trouverez dans le répertoire bin de DEV-C++.

Une fois lancés, nous allons commencer par créer une nouvelle classe (Menu New / code / classe). Nous la baptiserons PremiereAppliFLTK. A cette classe, il y a lieu d'associer un constructeur du même nom PremiereAppliFLTK() (Menu New / code / fonction/method).

Sous ce constructeur, nous initions une fenêtre du nom de maFenetre (Menu New / group / Window). Comme Extra Code de l'onglet C++, nous indiquerons #include <FL/fl_ask.H>. Le label sera MaFenetre (onglet GUI), et le nom de cette classe fenêtre sera également MaFenetre (onglet C++). L'inclusion du fichier fl_ask est nécessaire, car nous allons faire appel à fl_alert().



Sous le niveau de la fenêtre, nous ajoutons une barre de progression (Menu New / Other / Progress) du nom de classe de Progression et un bouton (au même niveau) du nom de BoutonDeProgression (Menu New / Buttons / Button). Comme étiquette de ce dernier, on notera «+10 % ».

A ce stade, il ne reste plus qu'à créer la méthode main et un callback qui permettra de faire évoluer la barre de progression de 0 à 100 %.

L'écriture du code de main et du callback

Au même niveau que le constructeur de la fenêtre, nous allons coder main. Il suffit de créer une méthode vide (Menu New / code / fonction/method). Automatiquement FLUID créera main. Son code sera très simple (Menu New / code / code), le constructeur se chargeant de la création des Widgets, de l'affectation du callback, puis la fenêtre est affichée :

```

PremiereAppliFLTK* Instance_Appli = new PremiereAppliFLTK();
Instance_Appli->MaFenetre->show(argc,argv);
    
```

Un callback réagira à l'évènement «enfoncer le bouton». Attention, vous pouvez demander (sous FLUID évidemment ou en le codant), de réagir soit au changement d'état (FL_WHEN_CHANGED), ou encore au click (FL_WHEN_RELEASE par défaut). Les lignes nécessaires au callback

seront encodées dans le cadre «callback» de l'onglet C++ du widget. En fait, le prototype de cette méthode est `void FL_Widget::callback(FL_Callback*, void* = 0)`. Le premier paramètre à passer est une fonction callback, et en deuxième un pointeur vers les données. FLUID s'occupe de tout. Il vous suffit d'ajouter le code voulu réagissant à l'évènement :

```
if (Progression->value()>=100) {
    fl_alert("Jauge remplie à 100%%");
    Progression->value(0);
}
else
    Progression->value(Progression->value()+10);
}
```

Comme on le voit, rien de plus simple. Maintenant, pour générer le code, il faut se diriger vers le menu File / Write Code. Le fichier `xx.FL` (au format FLUID) reprenant les éléments de l'interface sera transformé en un fichier d'inclusion (d'extension `.h`) et un fichier source (d'extension `.cpp`).

Sous DEV-C++ il ne vous reste plus qu'à créer un nouveau projet GUI/FLTK et d'y ajouter ces deux fichiers. Ils contiennent tout le code nécessaire à sa bonne compilation et exécution.

Voici le code d'inclusion final généré :

```
#ifndef Ma_premiere_appli_FLTK_h
#define Ma_premiere_appli_FLTK_h
#include <FL/Fl.H>
#include <FL/Fl_Window.H>
#include <FL/fl_ask.H>
#include <FL/Fl_Progress.H>
#include <FL/Fl_Button.H>

class PremiereAppliFLTK {
public:
    PremiereAppliFLTK();
    FL_Window *MaFenetre;
    FL_Progress *Progression;
    FL_Button *BoutonDeProgression;
private:
    inline void cb_BoutonDeProgression_i(FL_Button*, void*);
    static void cb_BoutonDeProgression(FL_Button*, void*);
};
#endif
```

Et le code source final généré :

```
// generated by Fast Light User Interface Designer (fluid) version 1.0102

#include "Ma_premiere_appli_FLTK.h"

inline void PremiereAppliFLTK::cb_BoutonDeProgression_i(FL_Button*, void*) {
    if (Progression->value()>=100) {
        fl_alert("Jauge remplie à 100%%");
        Progression->value(0);
    }
}
```

```
else
    Progression->value(Progression->value()+10);
}

void PremiereAppliFLTK::cb_BoutonDeProgression(FL_Button* o, void* v) {
    ((PremiereAppliFLTK*)(o->parent()->user_data()))->cb_BoutonDeProgression_i(o,v);
}

PremiereAppliFLTK::PremiereAppliFLTK() {
    FL_Window* w;
    { FL_Window* o = MaFenetre = new FL_Window(352, 101, "MaFenetre");
        w = o;
        o->user_data((void*)(this));
        { FL_Progress* o = Progression = new FL_Progress(5, 15, 345, 25);
            o->color(136);
        }
        { FL_Button* o = BoutonDeProgression = new FL_Button(105, 55, 150, 30, "+10%");
            o->callback((FL_Callback*)cb_BoutonDeProgression);
        }
        o->end();
    }
}

int main(int argc, char **argv) {
    PremiereAppliFLTK* Instance_Appli = new PremiereAppliFLTK();
    Instance_Appli->MaFenetre->show(argc,argv);
    return Fl::run();
}
```



Étendre FLTK tout en apprenant

FLTK est moins riche en widgets que Qt ou WxWindows, pour ne citer que ceux-là. Cependant, des développeurs se sont penchés sur la question et proposent des extensions. Il s'agit d'autres widgets, programmés en FLTK, et souvent placés dans le domaine public, avec leurs codes sources, ou bien des extensions proposant d'autres fonctionnalités, comme le multi threading, la connexion à une base de données (Simply Powerful Toolkit), ou la programmation réseau !

Le mot de la fin

Un portage FLTK pour DOS existe, ainsi que des interfaces pour les langages Eiffel, Python, PERL et Ruby. Avec FLTK le développeur n'a pas à se préoccuper de coder un destructeur pour démanteler les Widgets. L'application est maintenant exécutable directement sous Windows 98 ou bien Windows XP. Après recompilation sous Linux (utilisez le script `ftk-config`) ou Mac OS X, elle tournera sans problème sous ces deux derniers OS.

Bonne programmation !

■ Xavier Leclercq - Xavier.Leclercq@programmez.com

Passer à PHP 5 : réalisation d'une FAQ.

PHP 5 ne s'utilise pas comme PHP4. A partir d'une application concrète, nous verrons les nouveautés du langage et en particulier sur l'aspect "métier" de notre application.

Dialogue avec la base de données

Comme la principale nouveauté de PHP5 est son modèle objet, il est évident que nous allons en profiter.

```
<?php
class DBTools {
    private $_connection = null;
    private $_dbPath = null;
    private static $_singleton = null;

    private function __construct (){
        //on regarde s'il existe un fichier de configuration
        if (! is_readable ('project.conf.xml')){
            throw new Exception('Pas de fichier de configuration');
        }else{
            $projectConfig = simplexml_load_file('project.conf.xml');
            //on regarde si les informations de connexion sont présentes
            if (!isset ($projectConfig->database->filePath)){
                throw new Exception('Pas d\'information de connexion dans le
fichier de configuration');
            }else{
                $this->_dbPath = $projectConfig->database->filePath;
            }
            //on regarde si la base existe. Si non, on la crée avec la partie
            //initialisation du fichier de configuration
            if (! is_readable ($this->_dbPath)){
                $this->_connectIfNeeded ();
                if (isset ($projectConfig->database->script)){
                    try {
                        $this->doQuery ('BEGIN');
                        foreach ($projectConfig->database->script->query as $query){
                            $this->doQuery ($query);
                        }
                        $this->doQuery ('COMMIT');
                    }catch (Exception $e){
                        $this->doQuery ('ROLLBACK');
                        $this->disconnect ();
                        unlink ($this->_dbPath);
                        throw new Exception ('Impossible de créer la base de données
['.$e->getMessage ().']');
                    }
                }
            }
        }
    }

    function __destruct () {
        $this->disconnect ();
    }
}
```

```
public static function getInstance (){
    if (DBTools::$_singleton === null){
        DBTools::$_singleton = new DBTools ();
    }
    return DBTools::$_singleton;
}

public function doQuery ($query){
    $this->_connectIfNeeded ();
    if (!$result = sqlite_query ($query, $this->_connection)){
        throw new Exception ('Impossible d\'exécuter la requête ' . $query);
    }
    return $result;
}

public function connect ([...]
public function disconnect ([...]
private function _connectIfNeeded ([...]
public function fetch ($rs)[...]
}
?>
```

Voici une classe que nous n'aurions pas pu faire en PHP4 standard, ou en tout cas que nous aurions eu plus de mal à réaliser.

Variables statiques, encapsulation et Singleton

Nous décidons ici que notre objet de manipulation de base de données respecte un singleton. Concrètement, cela signifie que n'importe où dans l'application, nous allons manipuler une seule instance de l'objet. Cela présente l'avantage de gérer la connexion à un seul endroit, et de ne pas risquer de les multiplier. Avec cette technique, on ne fermera pas non plus un canal qui serait utilisé par un autre objet, empêchant un clone de fonctionner correctement.

Nous aurions pu arriver à un résultat presque équivalent en PHP4, en utilisant une variable statique dans la méthode « getInstance ». Toutefois, rien n'empêche l'utilisateur d'instancier directement la classe DBTools. De plus, il aurait suffi d'oublier à un seul endroit l'affectation ou le passage par référence de l'objet pour en obtenir un clone dont nous ne maîtrisons pas l'état. Ici, comme le constructeur est privé, la seule façon d'appeler l'objet est l'utilisation de la méthode statique getInstance, et comme les objets sont automatiquement passés par références, nous travaillons toujours sur le bon objet.

SQLite, embarquer la base de données.

Un autre élément intéressant réside dans la base de données embarquée. En PHP4, nous n'avions pas la possibilité de transporter ainsi la base de données. Nous devions nécessairement passer par un serveur, et très souvent nous utilisions MySQL. Il n'était pas possible, sans en connaître les paramètres (hôte, login, mot de passe) de s'y connecter, et encore moins de créer une base de toutes pièces.

Comme nous avons maintenant cette possibilité, nous en profitons pour

faire en sorte que notre application puisse créer sa propre base de données si cette dernière n'existe pas. Nous n'avons pas à demander à l'utilisateur de connaître les paramètres de connexion à la base, notre application est prête à l'emploi.

XML pour le paramétrage

Même s'il était possible en PHP4 d'analyser et de lire les données provenant de fichiers XML, il faut avouer que ce n'était pas tâche facile, sans passer par un objet tiers. Cette restriction n'existe plus, et nous avons maintenant l'opportunité de créer des fichiers de configuration au format XML très rapidement.

Dans notre application, c'est ce que nous avons fait :

```
<project>
<database>
<filePath>project.db</filePath>
<script>
<query>CREATE TABLE category ( id_cate INT PRIMARY KEY , caption_
cate VARCHAR ( 255 ) );</query>
<query>CREATE TABLE question ( id_ques INT PRIMARY KEY , caption_
ques VARCHAR ( 255 ) , answer_ques VARCHAR( 255 ) , id_cate INT );</query>
</script>
</database>
</project>
```

Nous indiquons la base à utiliser (database->filePath) et les scripts de création nécessaires dans l'optique où la base n'existe pas. Comme vous pouvez le constater, la lecture du fichier se fait en quelques lignes, très naturellement. Il sera toutefois nécessaire, pour des raisons de sécurité évidentes, de ne pas placer ce fichier XML dans un emplacement lisible par l'internaute. Il en va de même avec le fichier de base de données.

Les exceptions, une méthode exceptionnelle

Les exceptions marquent le début d'un monde où les développeurs PHP ne seront plus sans cesse obligés de tester les moindres codes retour des fonctions pour en traquer les erreurs. Utilisez sans restriction throw, try et catch pour vous simplifier la vie, et gérez les erreurs de façon globale. Le code d'installation de la base de données, sans l'utilisation des exceptions, aurait été moins naturel et aurait pu ressembler à ceci :

```
$this->doQuery ('BEGIN');
foreach ($projectConfig->database->script->query as $query){
    if (! $this->doQuery ($query)){
        $this->doQuery ('ROLLBACK');
        $this->disconnect ();
        unlink ($this->_dbPath);
        $this->error = 'Impossible de créer la base de données. La commande
'.'.$query.'.ne fonctionne pas.';
        return ;
    }
}
$this->doQuery ('COMMIT');//si nous sommes ici, alors tout c'est bien passé.
```

Plusieurs problèmes en découlent :

- Tout code applicatif désirant utiliser l'objet doit en tester le code erreur.
- Il est nécessaire de tester le code retour de doQuery, à chaque appel.
- C'est à l'appelant de dire quelle requête n'a pas fonctionné, alors que

la méthode doQuery semble la plus apte à faire ceci.

Le code avec exceptions est ordonné d'une façon plus compréhensible :

```
try {
    $this->doQuery ('BEGIN');
    foreach ($projectConfig->database->script->query as $query){
        $this->doQuery ($query);
    }
    $this->doQuery ('COMMIT');
} catch (Exception $e){
    $this->doQuery ('ROLLBACK');
    $this->disconnect ();
    unlink ($this->_dbPath);
    throw new Exception ('Impossible de créer la base de données
['.$e->getMessage ().']');
}
- On démarre la transaction
- On lance les requêtes
- On valide la transaction
- En cas de problème, on annule la transaction en lançant un nouveau
message d'erreur.
```

Destructeur : terminer proprement, au bon moment

Un dernier point des plus prometteurs, les destructeurs en PHP nous donnent enfin l'opportunité de rétablir un état applicatif propre au moment où notre objet a terminé son travail.

Nous nous en servons ici pour clore la connexion. Notre application n'a ainsi pas besoin de se soucier de l'état du canal de communication.

Manipuler les enregistrements

Maintenant que nous savons dialoguer avec la base de données, nous allons créer un objet standard capable de manipuler les enregistrements (lecture, écriture). Nous n'aurons qu'à surcharger cet objet pour prendre en compte les cas particuliers que sont les catégories et les questions.

```
abstract class DBRecord {
    protected $_modified = false;
    protected $_record = null;

    function __construct ($id){
        if (($object = $this->_load ($id)) === false){
            throw new Exception ('L'enregistrement demandé n'existe pas.('.$id.')');
        }else{
            $this->_record = $object;
        }
    }
    function __set ($prop_name, $prop_value){
        $this->_record->$prop_name = $prop_value;
        $this->_modified = true;
    }
    function __get ($prop_name){
        return $this->_record->$prop_name;
    }
    public function modified () {
        return $this->_modified;
    }
    public function save (){
```

```

    if ($this->_modified){
        $this->_save ();
    }
}
abstract protected function _load ($id);
abstract protected function _save ();
}

```

Notre classe est déclarée abstraite, car elle n'implémente pas l'ensemble de ses comportements (load et save). Toutefois, elle est capable de gérer un enregistrement, et elle en contrôle l'état (modifié ou non) via les méthodes magiques « __get » et « __set ».

A noter que les méthodes __get et __set ne sont appelées que si la propriété demandée n'existe pas dans l'objet lui-même. Ce qui signifie qu'on ne passera jamais par cette fonction lors de la lecture des propriétés _modified et _record.

Spécialisation pour les catégories

```

final class DBCategoryRecord extends DBRecord {
    protected function _load ($id){
        if ($id !== null){
            $ct = DBTools::getInstance ();
            $rs = $ct->doQuery ('select id_cate, caption_cate from category
where id_cate=' . intval ($id));
            return $ct->fetch ($rs);
        }else{
            $object->id_cate = null;
            $object->caption_cate = null;
            return $object;
        }
    }
    protected function _save (){
        $ct = DBTools::getInstance ();
        if ($this->id_cate === null){
            $query = "insert into category values (null, '" . addslashes ($this->caption_cate) . "')";
        }else{
            $query = "update category set caption_cate = '" . addslashes ($this->caption_cate) . "' where id_cate=" . intval ($this->id_cate);
        }
        $ct->doQuery ($query);
    }
}

```

On spécifie dans la classe spécialisée que la classe est « finale », c'est-à-dire qu'il ne sera plus possible de la surcharger. On implémente par la même occasion les deux méthodes abstraites de la classe mère, la lecture et la sauvegarde.

On pourra ainsi se servir de la classe de cette manière :

```

$category = new DBCategoryRecord (null);
$category->caption_cate = 'Premier libellé';
$category->save ();

```

```

$category = new DBCategoryRecord (1);
$category->caption_cate = 'Nouveau libellé';
$category->save ();

```

Spécialisation pour les questions, un nouveau rôle pour __set

```

final class DBQuestionRecord extends DBRecord {
    protected function _load ($id) [...]
    protected function _save ()[...]
    function __set ($prop_name, $prop_value){
        if ($prop_name !== 'id_cate'){
            parent::__set ($prop_name, $prop_value);
        }else{
            try {
                new DBCategoryRecord ($prop_value);
                $this->id_cate = $prop_value;
            }catch (Exception $e){ //aucune catégorie de l'identifiant donné. }
        }
    }
}

```

Nous profitons ici de la méthode __set pour empêcher la saisie de valeurs erronées de clef étrangère (propriété id_cate). En effet, la méthode testera à chaque affectation l'existence de la catégorie avant de mettre à jour le champ en question.

Toutes les autres propriétés restent gérées par le parent DBRecord.

Voici un exemple d'utilisation de la classe :

```

//création d'un enregistrement
$question = new DBQuestionRecord (null);
$question->caption_ques = 'Question ?';
$question->answer_ques = 'Réponse !';
$question->id_cate = 1;
$question->save ();

```

//ne prendra pas en compte la catégorie erronée

```

$question = new DBQuestionRecord (1);
$question->id_cate = 1000;//La catégorie 1000 n'existe pas
$question->save ();

```

On pourra ainsi étendre ce genre de contrôle sur différents champs.

Cerise sur le gâteau

Pour des raisons d'organisation, nous avons développé plusieurs objets dans plusieurs fichiers (un objet par fichier et réciproquement).

L'idéal serait de pouvoir nous servir de tous ces objets sans avoir à en inclure la définition, ce que rend possible « l'auto loader » de PHP 5. Rajoutez simplement le code de la méthode __autoload dans votre fichier d'entrée ou dans votre include principal, et le reste sera chargé à la demande. Dans notre exemple :

```

function __autoload($class){
    if (is_readable ($filename = strtolower ($class) . '.class.php')){
        require_once(strtolower ($class) . '.class.php');
    }
}

```

Vous pouvez télécharger les sources de l'exemple sur <http://copix.aston.fr/programmez2.zip>

■ **Gérald Croës** - Consultant Open Source ASTON



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

DotFuscator : obfuscation .NET

Si vous ne disposez pas de DotFuscator CE, vous pouvez toujours la télécharger sur le site web de l'éditeur. L'obfuscation est une technique qui vise à transformer un code, de manière à empêcher le reverse-engineering. En employant DotFuscator CE, les noms des variables et des méthodes seront transformés afin de produire un code extrêmement difficile à lire (humainement parlant). En cas de décompilation, le code re-généré reste obscur à un tiers programmeur qui ne pourra que très difficilement en comprendre les rouages. L'obfuscation aide à sécuriser votre application aux yeux des crackers, mais non à en boucher les trous de sécurité. DotFuscator Community Edition (CE) est la version légère et intégrée à Visual Studio 2003 de DotFuscator Pro.

Le problème sous .NET

Que vous adoptiez le modèle .net ou java, vous devez accepter un fait : les applications sont compilées en un byte code intermédiaire, qui est à son tour interprété par une super entité : une machine virtuelle d'exécution. En C sous Windows, le code est compilé en code nativement compréhensible par le compilateur cible. Le programmeur a même la possibilité de positionner certaines optimisations pour le processeur ad hoc (le code natif produit n'étant exécutable que par le processeur de destination).

Un code C# est un code .NET qui sera compilé en byte code intermédiaire appelé précisément MSIL. Ce code MSIL n'est pas interprété nativement par le processeur, mais bien par le CLR, le moteur d'exécution de .net, en quelque sorte l'équivalent de la JVM. Le CLR va recompiler à son tour le code MSIL à la volée (Just In Time) pour produire au final un code qui sera reconnu directement par le processeur. Bref, l'exécutable proprement dit, c'est-à-dire l'assemblage .net produit sous Visual Studio, est facilement 'démontable'. N'importe qui peut demander de le rétro-générer en un programme codé dans un langage .NET.

Sans obscurcir votre code MSIL, vos éventuels mots de passe codés en dur et vos algorithmes confidentiels tomberont sous la coupe des crackers. Si vous êtes sceptiques, vous pouvez utiliser le célèbre décompilateur Salamander pour rétro-générer votre code MSIL en C#. L'éditeur vous laisse d'ailleurs la possibilité sur sa page Web de décompiler 10 méthodes de chaque classe d'un assemblage .net.

L'éditeur de Salamander propose également RemoteSoft Salamander

Il s'agit d'un obfuscateur proposant deux types de protection, la première de type classique, et la deuxième de type protecteur. Le grand avantage de ce logiciel est que le cracker ne peut plus utiliser Salamander pour pratiquer du reverse-engineering...

Voici un code MSIL désassemblé :

```
private void button1_Click(object sender, EventArgs e)
{
    BabelFishService babelFishService = new BabelFishService();
    try
    {
        textBox3.Text = babelFishService.BabelFish(textBox1.Text, textBox2.Text);
        textBox4.Text = "Traduction réussie";
    }
    catch (WebException)
    {
        textBox4.Text = "Echec de la traduction (service Web indisponible ?)";
    }
}
```

Et voici le même code rétro-généré après obfuscation à l'aide de DotFuscator Community Edition :

```
private void b(object A_0, EventArgs A_1)
{
    a A = new a();
    try
    {
        d.Text = A.a(b.Text, c.Text);
        e.Text = "Traduction réussie";
    }
    catch (WebException)
    {
        e.Text = "Echec de la traduction (service Web indisponible ?)";
    }
}
```

Malgré ces modifications, parfois fort nombreuses du code MSIL, le comportement de l'application n'en sera pas modifié. Le degré d'obfuscation n'est pas extraordinaire et il y aurait moyen de l'améliorer en passant à la version professionnelle en cryptant les symboles, les méthodes, et autres types. DotFuscator CE se contente de renommer en préservant ou non la hiérarchie ou les espaces de noms. La version professionnelle propose aussi un service d'optimisation qui aura le mérite d'accélérer l'application en supprimant tout code inutile. Par exemple, si vous appelez une seule méthode d'une classe alors qu'elle en comporte 50, les 49 inutilisées ne seront pas incorporées (DotFuscator a baptisé cela : le procédé de réduction (*)).

(*) A propos de la réduction

Dotfuscator Professional Edition analyse de manière statique une application pour y repérer le code qui n'est pas employé. Par code inutile, il faut comprendre des types, des méthodes et des champs non utilisés. De cette manière, le code MSIL généré sera toujours le plus compact possible. Et qui dit compact, dit gain de vitesse au chargement, par exemple.

L'analyse consiste à parsemer le code de déclencheurs, c'est-à-dire à baliser les points d'entrées d'une application (Main est par exemple définie en tant que déclencheur). Vous pouvez vous-même définir vos propres déclencheurs. En parcourant le code de chaque déclencheur, Dotfuscator retiendra les méthodes et types utilisés, puis toutes les méthodes appelées. L'introspection ne s'arrêtera qu'à partir du moment où toutes les méthodes appelées auront été analysées. Dotfuscator peut alors déterminer un ensemble minimum de types et de membres nécessaires à l'exécution, qui eux seuls seront inclus dans l'assemblage. PreEmptive Solutions, l'éditeur de Dotfuscator, affirme que certains de ses clients ont constaté jusqu'à 70 % de réduction de leur exécutable.

En outre, Dotfuscator CE modifiera certaines informations, afin d'empêcher l'utilisation par un cracker de l'outil ILDASM de Microsoft. L'API de réflexion sera ainsi incapable d'analyser votre code source, rendant le désassemblage par cet utilitaire impossible (mais ce qui n'arrêtera pas Salamander).

Un peu de pratique

Nous utiliserons l'obfuscateur Dotfuscator CE intégré à VS.NET 2003. Vous pouvez lancer Dotfuscator Community Edition via le menu 'outils' de VS.NET 2003. Dotfuscator s'exécute et une fenêtre présentant plusieurs onglets se présente à vous. Commencez par créer un nouveau projet et spécifiez dans BUILD un répertoire temporaire et un répertoire de sortie. (Fig 1)

Le fichier de configuration est un document XML qui s'initialise lors de cette phase de création d'un nouveau projet.

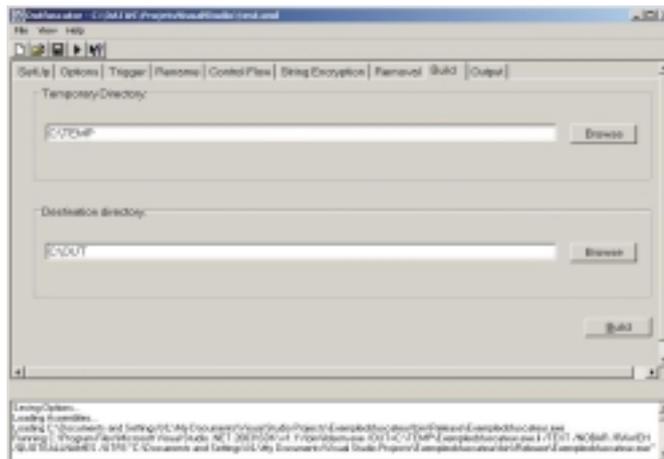
Cliquez sur l'onglet Trigger en y indiquant le chemin du code MSIL que vous devez obscurcir. Il suffit d'invoquer le bouton Browse pour retrouver votre fichier assemblage d'extension «exe». (Fig 2)

En sélectionnant l'onglet Build, cliquez sur le bouton Build, et la progression de l'obfuscation s'affichera dans la fenêtre inférieure (surtout si vous avez coché verbose). Dotfuscator CE affichera "Build finished" et votre code MSIL est désormais protégé. Pour vous en convaincre, rétro-générez son code source avec Salamander.

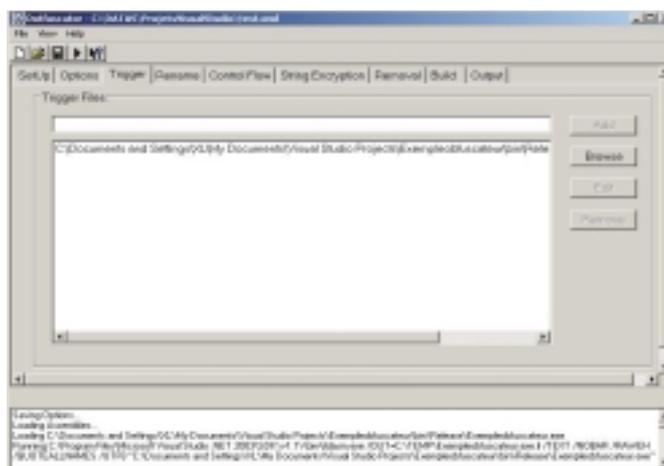
Comme nous l'avons signalé plus haut, nous pouvons choisir de conserver les noms des espaces de noms, soit la hiérarchie, soit ni l'un ni l'autre. (Fig 3)

Voici le code source d'origine de test :

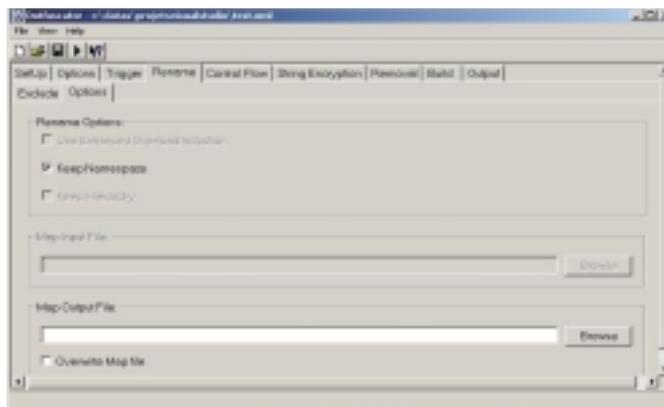
```
using System;
namespace Exempleobfuscateur
{
    public class Exemple
    {
        public void AfficheBonjour()
    }
}
```



(Fig 1)



(Fig 2)



(Fig 3)

Dotfuscator Community Edition (CE)

Points forts :

- Intégré à Visual Studio 2003 ou gratuitement téléchargeable ;
- Système Overload induction ;
- Rapidité et simplicité de la version GUI sont au rendez-vous.

Points faibles :

- L'utilisation en ligne de commandes n'est pas évidente en raison de la mise en œuvre du fichier de configuration XML ;
- L'optimisation du code (l'opération de réduction) n'est pas incluse. Vous devez passer obligatoirement à la version PRO ;
- Idem pour le cryptage des chaînes, fonctionnalité uniquement accessible avec la version PRO.

```

        Console.WriteLine("Bonjour chers lecteurs !");
        Console.ReadLine();
    }

    public static void Main()
    {
        Exemple bj = new Exemple();
        bj.AfficheBonjour();
    }
}

```

Voici le code source rétro-généré en ayant demandé à DotFuscator CE de conserver les espaces de noms :

```

using System;
namespace Exempleobfuscateur
{
    public class a
    {
        public void a()
        {
            Console.WriteLine("Bonjour chers lecteurs !");
            Console.ReadLine();
        }

        public static void Main()
        {
            new a().a();
        }
    }
}

```

Voici le code source rétro-généré en ayant demandé à DotFuscator CE de conserver la hiérarchie :

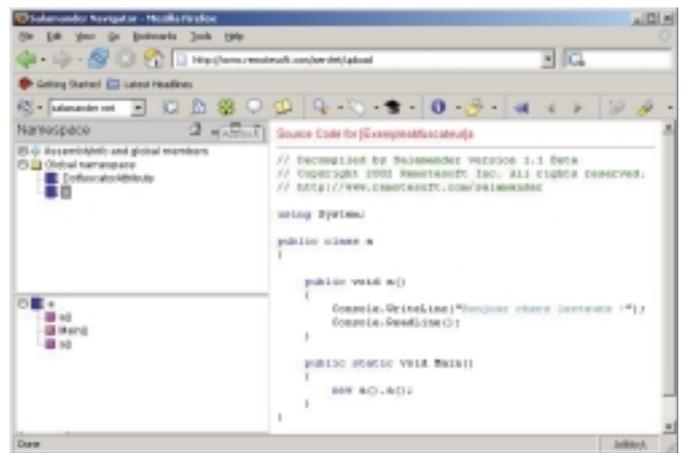
```

using System;
namespace a
{
    public class a
    {
        public void a()
        {
            Console.WriteLine("Bonjour chers lecteurs !");
            Console.ReadLine();
        }

        public static void Main()
        {
            new a.a().a();
        }
    }
}

```

Le système de renommage de Dotfuscator renomme plusieurs méthodes différentes avec un même nom (une technologie brevetée de renommage



(Fig 4)

appelée «overload induction»). Pour nous en convaincre, prenons un exemple encore plus frappant, issu de la documentation, et dont voici le code source d'origine :

```

private void CalcPayroll(SpecialList employeeGroup) {
    while (employeeGroup.HasMore()) {
        employee = employeeGroup.GetNext(true);
        employee.UpdateSalary();
        DistributeCheck(employee);
    }
}

Et voici le code rétro-généré :
private void a(a b) {
    while (b.a()) {
        a = b.a(true);
        a.a();
        a(a);
    }
}

```

D'autres logiciels d'obfuscation se contentent d'un simple renommage, allant même jusqu'à remplacer des noms par des caractères non imprimables, ce qui est totalement inutile, car Salamander est capable de renommer de telles chaînes. Par contre, le fait de nommer plusieurs méthodes du même nom (ici 'a') rend la compréhension du résultat décompilé très ardue.

Le mot de la fin

Nous attirons votre attention sur les nombreux avantages apportés par la version pro, comme le cryptage des chaînes, rendant le reverse-engineering encore plus ardu, mais surtout la possibilité de réduction du code source qui allégera vos exécutables. Ce dernier point n'est pas à négliger, surtout si vous programmez une application embarquée, la taille des exécutables étant alors cruciale.

Le site Web de Preemptive, l'éditeur de DotFuscator : <http://www.preemptive.com/>

Le décompilateur Salamander : <http://www.remotesoft.com/salamander/index.html>

■ Xavier Leclercq - Xavier.Leclercq@programmez.com

Maîtriser la barre des tâches de Windows

La barre des tâches est un élément de l'interface utilisateur du bureau de Windows. Si elle se fait facilement oublier, le programmeur peut aussi souhaiter communiquer avec elle. Découvrons ensemble comment procéder.

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

L'interface utilisateur de Windows est appelée le Shell. Le terme de Shell implique que celui-ci permet à l'utilisateur de demander au noyau le chargement et l'exécution d'applications. Sous Unix/Linux, un Shell est un simple terminal dédié à la saisie de lignes de commandes. Sous Windows, les choses sont très différentes, car ce système a été orienté "interface graphique" dès le départ. Le Shell est alors ici, selon la définition de Microsoft, un ensemble d'applications et de bibliothèques. L'application graphique principale est le bureau, et dans ce bureau l'application graphique principale est la barre des tâches. Idée directrice du Shell oblige, tout ce petit monde collabore au chargement et à l'exécution d'applications, mais il est important d'avoir à l'esprit que certains éléments du Shell sont relativement indépendants, voire autonomes. Ainsi en est-il de notre barre des tâches. Celle-ci est une simple application (mais pas nécessairement une application simple ;). Cela entraîne qu'il est possible de la remplacer par une autre, ce qui pourrait faire l'objet d'un autre article. Cela entraîne également que sa programmation est plutôt hybride. Par sa situation elle doit collaborer avec le Shell ou les applications, et ceci se fait parfois par messages, parfois avec COM qui est à la base du Shell, parfois avec des APIs qui encapsulent COM, parfois avec un mélange des deux. Ce désordre a

sans doute aussi des raisons historiques si l'on pense à l'évolution parallèle des systèmes NT et 9x qui tendent à se rejoindre dans Windows 2000/XP.

On voit qu'il n'est pas forcément évident de savoir par quel bout prendre les choses. Nous allons essayer de clarifier tout cela par la pratique. Nous nous proposons d'améliorer le programme de hook le clavier que nous avons

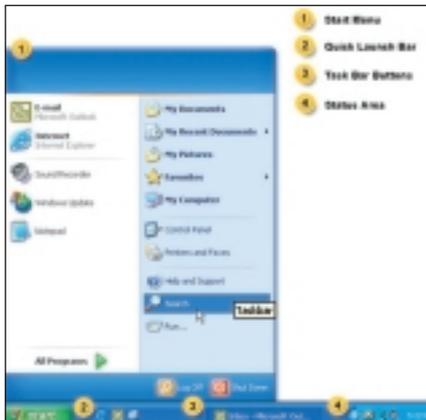


Fig.1 : L'organisation de la barre des tâches. Images extraites de la MSDN de Microsoft.

écrit le mois dernier. Nous ne sommes pas satisfaits, car le programme traîne sur le bureau pendant son utilisation. Nous préférons l'icôner dans la zone de statut de la barre des tâches. (Fig.1) Pour un petit utilitaire de ce genre, nous pouvons aussi souhaiter qu'il n'affiche jamais de boutons dans la barre de boutons, et enfin nous pouvons souhaiter créer un raccourci dans le menu Programmes/Démarrage de la barre des tâches, afin que notre petit outil soit actif dès le démarrage. Ou encore, nous pouvons choisir de placer le raccourci dans la barre de lancement rapide (Quick Launch Bar). Nous fournissons un projet pour Visual C++

6.0 traitant tous les exemples et un projet C++Builder 6.0 montrant comment, avec ce dernier, icôner une application et recevoir les messages qu'elle peut émettre. Nous conseillons vivement au lecteur de reprendre les projets et leurs configurations telles quelles. On se souvient, par exemple, que dll de hook et programme d'installation, doivent cohabiter dans un même répertoire, ce qu'assure chaque projet. Enfin pour la création du raccourci, on considère que l'application pointée est située dans un sous répertoire de c:\taskbar, ceci pour alléger le code. Si vous modifiez cette localisation, vous devrez adapter le code de Raccourci.cpp. Comme de coutume vous trouverez les sources de tous les exemples sur le site officiel (www.programmez.com). Enfin, en cas de besoin, les exemples "bas niveau" écrits sous Visual C++ pourront être compilés sous C++Builder sans difficulté.

Iconifier une application

C'est le travail le plus simple. On renseigne une structure NOTIFYICONDATA, puis on invoque l'API Shell_NotifyIcon, comme on le voit (Encadré 1). Cet extrait de code provient d'une application basée dialogue en API Windows pure et dure, écrite avec Visual C++. Malgré sa simplicité, ce code mérite quelques commentaires. Comme nous l'avons dit, Windows a subi de nombreuses évolutions. La structure NOTIFYICONDATA a suivi le mouvement, mouvement encore conjoint à celui d'Internet Explorer, comme en témoigne l'épouvantable déclaration, à géométrie variable, que voici :

```
typedef struct _NOTIFYICONDATA {
    DWORD cbSize;
    HWND hWnd;
    UINT uID;
    UINT uFlags;
    UINT uCallbackMessage;
    HICON hIcon;
    #if (_WIN32_IE < 0x0500)
        TCHAR szTip[64];
    #else
        TCHAR szTip[128];
    #endif
    #if (_WIN32_IE >= 0x0500)
        DWORD dwState;
        DWORD dwStateMask;
        TCHAR szInfo[256];
        union {
            UINT uTimeout;
            UINT uVersion;
        } DUMMYUNIONNAME;
        TCHAR szInfoTitle[64];
    }
```

```

DWORD dwInfoFlags;
#endif
#if (_WIN32_IE >= 0x600)
    GUID guidItem;
#endif
} NOTIFYICONDATA, *PNOTIFYICONDATA;

```

Le système s'y retrouve en testant la taille de la structure. Il convient donc de renseigner `cbSize` correctement.

Ensuite, nous voyons qu'un membre contient un handle de fenêtre. Il est courant de passer ici le handle de la fenêtre principale, mais rien n'y oblige. On peut très bien, si on le souhaite, passer le handle d'une fenêtre qui serait toujours cachée. C'est la procédure de rappel de cette fenêtre qui devra traiter les messages de la souris en provenance de la barre des tâches. Le membre `uId` est un identificateur d'icône. En effet, si une application ne peut s'icôner qu'à partir d'une seule fenêtre à la fois, elle peut placer plusieurs icônes dans la barre de statut. Pour que la raison de l'émission d'un message puisse être clairement définie, l'identificateur d'icône est transmis en paramètre des messages émis par la barre des tâches. Cet identificateur est à l'usage du programmeur et non du système. Il peut donc être quelconque, même si pour l'exemple, nous avons utilisé l'identificateur de la ressource. Le membre `uFlags` doit être renseigné rigoureusement, car le système lui fait aveuglément confiance, en ce qui concerne la validité des autres membres.

Un message personnalisé

Vient maintenant le membre `uCallbackMessage`. Il s'agit d'un entier non signé, correspondant à un message utilisateur. C'est cette valeur qui sera la valeur de message que la barre des tâches enverra à votre application via l'API `SendMessage`. La question est de savoir comment choisir cette valeur? Il y a deux méthodes. La première méthode consiste à suivre la recommandation de la MSDN et à demander au système de fournir cette valeur, via l'API `RegisterWindowMessage`, qui reçoit une chaîne quelconque en paramètre.

```

nid.uCallbackMessage =
::RegisterWindowMessage("HookClavierProgrammez");

```

La MSDN nous raconte que cette méthode est la bonne, quand deux applications différentes travaillent avec un même message, ce qui est le cas ici : la barre des tâches d'un côté, notre application de l'autre. Seulement, voilà, si on veut porter l'application d'exemple vers MFC et son mécanisme de routage de messages, ce procédé ne nous facilite pas... la tâche :-).

Le deuxième procédé consiste à simplement prendre une valeur parmi celles autorisées à l'utilisateur, d'où dans notre code :

```
#define WM_TASKBAR WM_USER+1
```

L'expérience montre que tout fonctionne bien, et si l'on travaille avec MFC ou C++Builder (encadré 2), tout est plus simple.

Avez-vous mal à l'en-tête ?

Les derniers membres de la structure sont moins cruciaux, mais pas moins intéressants, car ils permettent de gérer finement les bulles d'aides qui surgiront quand le pointeur de souris survolera l'icône. (Fig.2) Pour ces détails, le lecteur voudra bien se reporter à la MSDN

(<http://msdn.microsoft.com>) Le cas échéant, il ne sera pas trop surpris d'un refus de compiler de Visual C++. En effet, il y a gros à parier que Visual C++ 6.0 ne dispose pas des tout derniers entêtes Windows. Dans ce cas, il est toujours possible de se procurer ceux-ci en téléchargeant la SDK de Microsoft.

Enfin, pour une application comme la nôtre, il est de bon ton de masquer la fenêtre principale dès le lancement, d'où le

```
::ShowWindow(hDialog, SW_HIDE);
```

dans le programme d'exemple.

Menus travaux

Comme la fenêtre principale de l'application a maintenant disparu, nous ne pouvons plus accéder aux fonctionnalités de notre application que par son icône sur la barre des tâches. Cette dernière émet des messages relatifs aux événements souris uniquement. Le comportement classique est d'ouvrir la fenêtre principale en cas de clic sur le bouton gauche et de proposer un menu surgissant en cas de clic sur le bouton droit. (Fig. 3). C'est ce que fait notre fonction de rappel, dont voici un extrait (encadré 3). En ce qui concerne l'affichage du menu, on note que l'on s'enquiert au préalable de la position du curseur et surtout, que l'on amène notre fenêtre à l'avant-plan, bien qu'elle ne soit pas visible. Si l'on ne procède pas ainsi et que la fenêtre n'est pas en avant-plan au moment de l'appel à `TrackPopupMenu`, le menu ne disparaît pas lorsque l'utilisateur clique à l'extérieur, ce qui fait désordre.

Un remède contre les boutons

Par défaut, lorsqu'on ouvre notre fenêtre, la barre des tâches crée un bouton correspondant dans la zone des boutons. (Fig. 4) Messenger, par exemple, conserve ce comportement par défaut. Il est néanmoins possible de bricoler cela, car, à partir d'Internet Explorer 4.0 (!) la barre des tâches implémente une interface COM, `ITaskbarList`, qui permet de retirer (ou même d'ajouter) un bouton, pourvu que nous disposions d'un handle de fenêtre. (Fig. 5) Puisque cette fois nous avons à faire à COM, nous utiliserons le pointeur intelligent que nous avons donné dans l'article sur la programmation du planificateur de tâches (Programmez! 67). Moyennant quoi, le code est tout simple :

```
void WindowOnTaskbar(BOOL on)
{
```



Fig.2 : Notre application icônifiée affiche une bulle d'aide.



Fig. 3 : Notre application affiche un menu surgissant.



Fig. 4 : Par défaut, une fenêtre visible à l'écran est matérialisée par un bouton sur la barre des tâches.

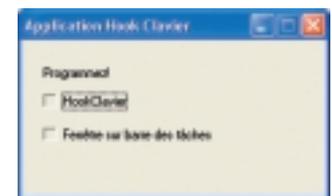


Fig. 5 : Notre programme de démonstration permet ou non de conserver le bouton sur la barre des tâches.

```

HRESULT hr;
TSmartComPtr<ITaskbarList> tbl;
try
{
    hr = ::CoCreateInstance(CLSID_TaskbarList,
                          NULL,
                          CLSCTX_INPROC_SERVER,
                          IID_ITaskbarList,
                          (void **)&tbl);

    TestOk(hr);
    // obligatoire
    hr = tbl->HrInit();
    TestOk(hr);

    if(on)
        hr = tbl->AddTab(hDialog);
    else
        hr = tbl->DeleteTab(hDialog);
    TestOk(hr);
}
catch (TComException ce)
{
    ce.Show();
}
}
    
```

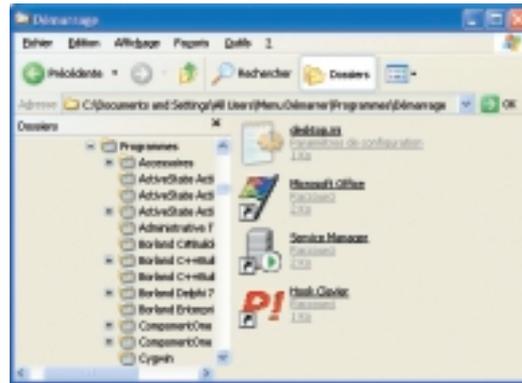


Fig. 6 : Les applications lancées au démarrage du système sont pointées par des liens localisés dans un répertoire particulier.

Bien entendu, COM est initialisé via l'API CoInitialize, avant tout appel à notre routine.

COM vous rebute ? Il existe un autre moyen de procéder, ou plutôt de bidouiller. On crée une fenêtre qui reste toujours invisible et la fenêtre "principale" de l'application, celle qui peut apparaître à l'écran, sera alors un enfant de la première. On peut agir sur le bouton correspondant à cette seconde fenêtre moyennant une petite bidouille :

```

if (Visible)
    ModifyStyleEx(0, WS_EX_APPWINDOW);
else
    ModifyStyleEx(WS_EX_APPWINDOW, 0);
    
```

avec MFC. En API pure et dure, on passera par SetWindowLong pour modifier les bits de style de la fenêtre.

Les raccourcis en long et en large

La barre des tâches a encore pour rôle d'afficher un menu. Ce menu est construit à la volée, d'après le contenu de répertoires surveillés en permanence. Ces répertoires contiennent des raccourcis ou liens qui pointent sur les applications réelles. Succinctement, un raccourci est un fichier contenant le nom de l'application ou du document pointé. Donc, si nous voulons ajouter une entrée au menu de la barre des tâches, par exemple, dans Programmes/Démarrage, il faut créer un raccourci dans le bon répertoire (Fig. 6). Notre problème se décompose en deux parties. D'abord connaître le répertoire qui nous concerne, ensuite, savoir créer un lien. On résout le problème à coup d'API, de COM (voir exemple raccourci.cpp) et de gymnastique compliquée. Pour connaître le répertoire sans risquer de nous faire piéger par un problème de langue, par exemple, nous invoquons l'API SHGetSpecialFolderLocation qui, moyen-

nant un identificateur en paramètre, retourne un pointeur sur un objet COM dit PIDL, ce qui se traduit en gros par "pointeur sur une liste d'identificateurs d'Item". Il faut comprendre que cet objet contient le répertoire découpé comme un saucisson. (Fig. 7). Nous récupérons alors le nom entier de notre répertoire dans un tampon, via l'API SHGetPathFromIDList. Attention maintenant. Nous avons invoqué des

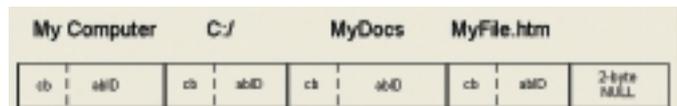


Fig. 7 : Exemple de saucissonnage au sein d'un PIDL (diagramme issu de MSDN).

APIs, mais c'est COM qui a travaillé en réalité. Ceci implique que l'objet pointé par PIDL a été alloué par COM et son interface de gestion-mémoire IMalloc et qu'il doit être libéré de la même manière. Il existe une API pour obtenir un pointeur sur IMalloc. Après quoi, nous pouvons libérer la mémoire et bien sûr, nous devons libérer IMalloc elle-même ensuite. Ceci conduit à la formule magique :

```

LPMALLOC pMalloc;

::SHGetMalloc(&pMalloc);
pMalloc->Free(pidl);
pMalloc->Release();
    
```

Le lecteur consultera la MSDN pour connaître tous les identificateurs que peut recevoir SHGetSpecialFolder. Si l'on veut placer l'application dans le Quick Launch, l'identificateur est CLSID_APPDATA, mais cela n'est toutefois pas suffisant (beau manque d'homogénéité...). Il faut concaténer la chaîne :

```
"\\Microsoft\\Internet Explorer\\Quick Launch"
```

au répertoire obtenu pour aboutir au résultat.

Une fois que nous avons obtenu notre répertoire, nous lui concaténons le nom du raccourci, sans omettre l'extension.lnk. Ceci fait, le raccourci se crée au moyen des méthodes de deux interfaces COM : IShellLink et IPersistFile. Cette dernière étape ne présente pas de difficulté, si les principes de programmation COM expliqués dans Programmez! 67 ont été assimilés. Dans ce cas, le code d'exemple est explicite. La barre des tâches n'a désormais plus (ou plus beaucoup) de secrets pour vous. A bientôt pour de nouvelles aventures dans les profondeurs de Windows.

Encadré 1**Code pour icôner une application**

```
#define WM_TASKBAR WM_USER+1

void PutOnTaskbar(HWND hwnd)
{
    NOTIFYICONDATA nid = {0};

    HICON icon = ::LoadIcon(instance, MAKEINTRESOURCE(IDI_PROGRAMMEZ));

    nid.cbSize = sizeof(NOTIFYICONDATA);
    nid.hIcon = icon;
    nid.hWnd = hwnd;
    strcpy(nid.szTip, "Hook clavier");
    nid.uCallbackMessage = WM_TASKBAR;
    nid.uFlags = NIF_ICON | NIF_MESSAGE | NIF_TIP;
    nid.uID = IDI_PROGRAMMEZ;
    // Si on dispose des dernier entêtes
    //nid.szInfoTitle = "Programmez!";
    // etc...
    ::Shell_NotifyIcon(NIM_ADD, &nid);

    if (icon)
        ::DestroyIcon(icon);
}
```

Encadré 2**Intercepter un message personnalisé avec C++Builder**

Nettement plus évoluée que MFC, la librairie graphique VCL cache son mécanisme de routage de messages au programmeur. S'il s'agit d'intercepter un message personnalisé, il est possible de le faire explicitement, à l'aide d'un sandwich aux macros :) Voici les points importants : On doit déclarer l'interception et la méthode de rappel dans une section protected de la classe. La macro END_MESSAGE_MAP reçoit le type de la classe de base en paramètre. Si le message n'est pas traité par la fonction de rappel (ce n'est pas le cas pour notre programme d'exemple), il est transmis à la classe de base. Enfin, on garde à l'esprit que la VCL encapsule les messages dans un objet TMessage, même dans le cas d'un message personnalisé, d'où l'apparition de ce type dans les paramètres de OnTaskbar, la méthode de rappel, et dans la macro VCL_MESSAGE_HANDLER:

```
class TForm1 : public TForm
{
    //etc...
protected:
    void __fastcall OnTaskbar(TMessage &Message);

    // interception d'un message personnalisé
    BEGIN_MESSAGE_MAP
        VCL_MESSAGE_HANDLER(WM_TASKBAR, TMessage, OnTaskbar)
    END_MESSAGE_MAP(TForm)

private:    // User declarations
```

```
HINSTANCE HookClavierDll;
```

```
// etc...
```

```
};
```

Encadré 3**Code partiel de la fonction de rappel**

```
void OnTaskbar(HWND hwnd, WPARAM wParam, LPARAM lParam)
{
    UINT id;
    UINT msg;

    id = (UINT)wParam;
    msg = (UINT)lParam;

    if(id != IDI_PROGRAMMEZ)
        return;

    switch(msg)
    {
    case WM_LBUTTONDOWN:
        ::ShowWindow(hwnd, SW_SHOW);
        // etc.
        break;

    case WM_RBUTTONDOWN:
        POINT pt;
        UINT item;
        ::GetCursorPos(&pt);
        ::SetForegroundWindow(hwnd);
        item = ::TrackPopupMenu(Menu,
            TPM_RIGHTALIGN | TPM_BOTTOMALIGN |
            TPM_RETURNCMD | TPM_NONOTIFY,
            pt.x, pt.y, 0, hwnd, NULL);

        switch(item)
        {
        case MENU_OUVRIR:
            ::ShowWindow(hwnd, SW_SHOW);
            // etc.
            break;
        case MENU_QUITTER:
            // etc.
            break;
        default:
            break;
        }
        break;

    default:
        break;
    }
}
```

■ Frédéric Mazué - fmazue@programmez.com



Manipuler les fichiers avec Java

Java offre des fonctionnalités d'entrées/sortie très riches et permet de travailler finement dans un environnement complexe. Découvrons ensemble les bases du travail avec les fichiers.

Tout langage de programmation se doit de résoudre le problème présenté par la gestion des entrées/sorties. Au sens large, ce terme recouvre tout ce qui entre ou sort de l'ordinateur, via des périphériques physiques comme un modem, une connexion réseau, un périphérique de stockage, ou des périphériques virtuels, comme la console Windows, ou une URL. Au sens restreint, le terme recouvre les fichiers et la console. Des langages de bas niveau comme C ne traitent que l'aspect restreint, et se reposent sur le système d'exploitation pour l'implémentation du reste. Certains systèmes d'exploitation, comme UNIX, homogénéisent d'ailleurs les choses en considérant qu'un périphérique est avant tout un fichier. Les langages de haut niveau ont souvent une démarche semblable à celle d'UNIX, mais en raison de leur approche haut niveau, ils considèrent que tout est flux (ou flot) de données, la source de données du flux pouvant être n'importe quoi. Cette approche est celle qui a été adoptée par les concepteurs de Java. Sous Java tout est donc flux de données (fichier, URL, simple chaîne, etc.) et la quantité de classes encapsulant des flux plus ou moins compliqués est tout à fait impressionnante. Le propos de cet article est d'aider le lecteur habitué à des choses plus rudimentaires (langage C) et qui débute dans ce domaine, à s'y retrouver dans ce vaste monde. Nous traitons seulement le cas où un fichier est la source de données du flux.

Le bas niveau avec Java

Java, comme tout langage, doit traiter directement avec le système d'exploitation pour manipuler les fichiers. Java se voulant portable et les systèmes d'exploitation étant tous différents, la classe `java.io.File` propose un ensemble de fonctionnalités qui se veulent être le dénominateur commun des divers systèmes. Cette classe gère le concept de fichier au plus bas niveau. Ainsi, elle permet de créer un fichier, de tester son existence, connaître la date de sa dernière modification, tester si le fichier est en lecture seule, etc. Mais elle ne permet pas de positionner finement les droits d'accès à la manière d'UNIX, puisque l'équivalent n'existe pas sous Windows par exemple. Le lecteur issu du langage C sera surpris de ne trouver dans la classe `File` aucune méthode pour lire ou écrire un octet. C'est en fait logique, car ces opérations sont du ressort des flux, conformément à ce que nous avons dit plus haut. De plus, un objet `File`, construit à partir d'un nom de fichier, ne crée pas celui-ci sur le disque pour autant. Pour que la création soit effective, il faut invoquer la méthode `createNewFile`. En corollaire, un objet `File` ne va jamais détruire un fichier, à moins que sa méthode `delete` ne soit invoquée explicitement. Les types `File` de Java et de C n'ont donc rien en commun. En Java, un objet `File` représente une entité sur un disque, et rien de plus.

En revanche, on sera étonné à juste titre de constater que la classe `File` encapsule aussi bien le fichier que les répertoires, ce qui est plutôt désordonné. Cela tient sans doute à ce que les concepteurs de Java ont

une culture UNIX où un répertoire est un fichier, puisque tout est fichier sous UNIX. La classe propose les méthodes `isDirectory` et `isFile`, pour aider à s'y retrouver. `File` propose la méthode `mkdir` pour créer un répertoire.

Les systèmes de fichiers et Java

Java ou pas, écrire du code parfaitement portable n'est pas immédiat, car les systèmes de fichiers ont des représentations différentes. Ainsi, sous Windows, la racine pourra être `C:\` alors que sous UNIX/Linux elle sera simplement `/`. Le programmeur doit faire avec. Il y a deux façons. D'abord la paresseuse qui consiste à travailler dans le répertoire de l'utilisateur uniquement. C'est évidemment cette méthode que les exemples joints à cet article (sur le site www.programmez.com) utilisent :) Voici le code qui crée un répertoire de nom `demofichier` dans, par exemple, `C:\Documents and Settings\Fred` ou `/home/fred`. A la suite de quoi, le répertoire créé est défini comme le répertoire courant.

```
public static void setWorkingDirectory() {
    String prefix;
    String rep_name = "demofichier";
    String fullpath;
    File rep;

    prefix = System.getProperty("user.home");
    fullpath = prefix + File.separator + rep_name;
    rep = new File(fullpath);
    if (!rep.exists()) {
        rep.mkdir();
    }
    System.setProperty("user.dir", fullpath);
}
```

La seconde méthode consiste à savoir sur quel système d'exploitation on se trouve, et à gérer par programmation les préfixes des chemins de fichiers. Pour connaître le système :

```
System.getProperty("os.name");
```

Et on instancie un objet fichier bidon, pour connaître toutes les racines ou unités de disque si besoin est :

```
// fichier_bidon n'a pas besoin
// d'exister sur le disque.
File f = new File("fichier_bidon");
File[] tab = f.listRoots();
for(int i=0; i<tab.length; i++)
```

```
System.out.println(tab[i]);
```

Quelle que soit la méthode choisie, on retiendra que le constructeur de File doit recevoir un nom de fichier dit "abstrait", ce que l'on traduit concrètement (si l'on peut dire...) par complet à savoir :

```
Préfix + chemin_complet + nom_de_fichier.
```

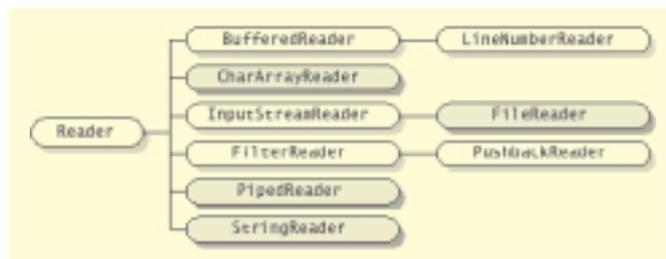
Flux et reflux

Il est maintenant temps de lire ou d'écrire quelque chose dans un fichier :-). A la base de tous les flux Java se trouvent 4 classes abstraites aux noms pas tellement explicites InputStream, OutputStream, Reader et Writer. (Figures ci-contre). L'utilisateur ne peut pas instancier ces classes, car elles sont abstraites, mais il peut arriver que d'autres classes en renvoient une instance, par exemple la classe URL, dont la méthode openStream retourne un InputStream.

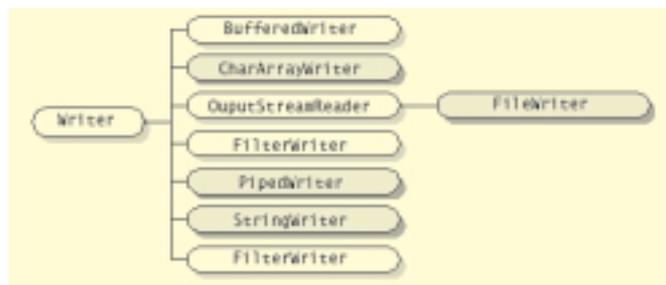
Pourquoi y a-t-il 4 classes à la base de la hiérarchie des flux ? Il y a deux groupes de classes, selon que l'on veut traiter des données binaires (InputStream et OutputStream), ou des caractères (Reader et Writer). Et dans chaque groupe, une classe gère la lecture et l'autre l'écriture. Ainsi, il est normalement impossible d'écrire dans un flux ouvert en lecture, ce qui est une bonne chose. Avec une exception toutefois, en ce qui concerne la classe RandomAccessFile, que nous mentionnerons plus loin. Remarquons que les méthodes read ou write de ces classes sont tout ce qu'il y a de rudimentaires, car elles ne permettent de manipuler qu'un ou une série d'octets (y compris en ce qui concerne Reader et Writer), et ne gèrent même pas les types natifs de Java. On utilisera donc normalement des classes de plus haut niveau en essayant de ne pas se perdre dans cette jungle. Voici un tableau 'minimum vital' clarifiant qui fait quoi, et devant vous aider à faire le bon choix.

Pour travailler efficacement avec ce tableau, certaines remarques s'imposent. Des noms que nous n'avons pas mentionnés jusqu'ici, apparaissent dans la colonne 'Base'. Ne pas s'en inquiéter, la hiérarchie de classes est très complexe, avec chaque fois une petite spécialisation d'une classe à l'autre. Les classes telles que FilterInputStream ne sont (en général) pas utilisées directement par le programmeur, mais viennent s'insérer dans la hiérarchie..

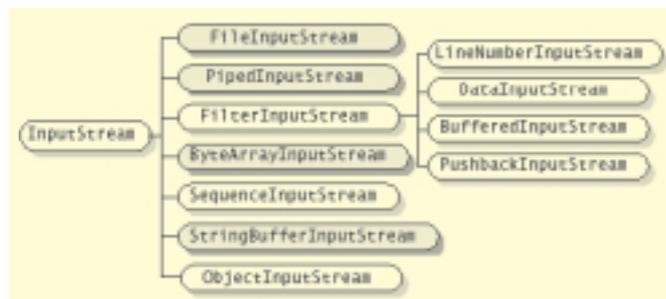
Les constructeurs des classes DataInputStream et DataOutputStream ne reçoivent que des objets InputStream ou OutputStream en argument. On ne peut donc leur donner directement un fichier comme source de don-



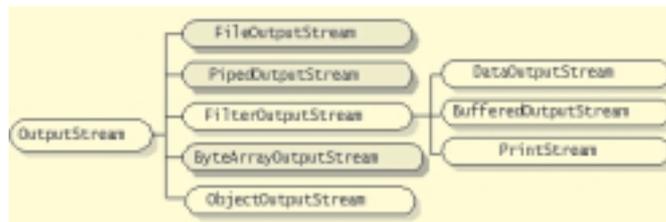
Quelques-unes des classes descendant de Reader. Ce diagramme ainsi que les autres diagrammes de hiérarchies de classes proviennent du site de Sun.



Quelques-unes des classes descendant de Writer.



Quelques-unes des classes descendant de InputStream.



Quelques-unes des classes descendant de OutputStream.

Classes	Base	Type de sortie	Description
DataInputStream DataOutputStream	FilterInputStream FilterOutputStream	Binaire	Permet de lire/écrire des types natifs Java. Permet aussi de lire/écrire des caractères, ceux-ci étant codés sur deux octets, Unicode oblige. Ces classes sont toutefois peu appropriées au travail avec des données texte.
FileInputStream FileOutputStream	InputStream OutputStream	Binaire	Permet de définir un fichier comme source de données.
BufferedInputStream BufferedOutputStream	FilterInputStream FilterOutputStream	Binaire	Permet de réaliser des entrées/sorties tamponnées (bufferisées) en mode binaire.
FileReader FileWriter	InputStreamReader OutputStreamWriter	Caractères	Permet de définir des fichiers en mode texte comme source de données.
PrintWriter	OutputStreamWriter	Caractères	Cette classe est utilisée pour les sorties textes. Un type natif Java est converti en une chaîne humainement lisible.
BufferedReader BufferedWriter	Reader Writer	Caractères	Permet de réaliser des entrées/sorties tamponnées en mode texte.
RandomAccessFile	DataInput (Interface) DataOutput (Interface)	Binaire/Caractères	Permet les accès directs dans les fichiers. Le RandomAccessFile est le type qui ressemble le plus au type FILE * de C. Il peut d'ailleurs être ouvert à la fois en lecture et en écriture.

nées. On utilise pour cela, par exemple, pour la lecture, un objet `FileInputStream` qui dérive lui aussi de `InputStream`. Cette classe sert en quelque sorte de pont entre le flux de données et le fichier. Lorsqu'il s'agit d'écrire, on se méfiera du constructeur de `FileOutputStream` qui, s'il reçoit uniquement un nom de fichier, écrasera systématiquement ce dernier s'il existe. Pour éviter cela et passer en mode append, passer `true` comme second argument au constructeur. Toutes les entrées/sorties binaires sont effectuées au format Big Endian, quelle que soit l'architecture de la plate-forme hôte.

Par défaut, les entrées/sorties ne sont pas tamponnées. Ceci implique que chaque opération voit une activité physique du disque lui correspondre. Pour gagner en performance, on utilisera des classes bufférisées. Une telle classe ne peut toutefois être utilisée seule, car son constructeur ne reçoit qu'un `InputStream` (ou un `OutputStream`) et ses méthodes sont peu utiles. (voir plus loin)

Enfin, même si rien ne le suggère dans le tableau, les flux consomment des ressources et il est pertinent de les fermer après usage par la méthode `close`.

Un flux peut en cacher un autre

Nous voyons que chaque classe sait faire un petit quelque chose qui nous intéresse, mais pas la totalité. C'est la philosophie de Java. Pour avoir un résultat complexe, on construit une chaîne d'objets, chaque élément apportant sa touche au résultat final. Voulons nous écrire un entier dans un fichier en mode tamponné ? Voici la démarche.

- créer le fichier (objet `File`)
- créer un objet `FileOutputStream` à partir de `File`, afin de pouvoir affecter le fichier comme source de données au flux à venir.
- créer un objet `BufferedOutputStream` à partir de `FileOutputStream`, afin de passer en mode tamponné.
- Créer un objet `DataOutputStream` à partir de `BufferedOutputStream`, afin de disposer des méthodes pour écrire la donnée binaire.
- écrire la donnée. (Notre entier).
- vider le tampon (`flush`) afin de forcer l'écriture.
- Fermer le flux `DataOutputStream`.

On procède dans cet ordre, car vous voulez disposer de la méthode `writeln` de `DataOutputStream`, mais cette méthode doit s'appuyer sur la méthode tamponnée `write` de `BufferedOutputStream` qui est intercalée dans la chaîne.

Tout cela conduit à un code remarquablement lourdingue et difficile à relire. Mais d'un autre côté, cela apporte l'avantage de pouvoir piloter les entrées/sorties très finement.

Voici le code :

```
import java.io.*;

public class EcritInt {
```

```
public static void main(String[] args) {
    File f;
    FileOutputStream fos;
    BufferedOutputStream bos;
    DataOutputStream dos;
    int entier = 128;

    FileUtilities.setWorkingDirectory();

    try {
        String nom = System.getProperty("user.dir")
+ File.separator + "monfichier_entier.bin";
        System.out.println(nom);
        f = new File(nom);
        if (!f.exists())
            f.createNewFile();
        fos = new FileOutputStream(f, true);
        bos = new BufferedOutputStream(fos);
        dos = new DataOutputStream(bos);
        dos.writeInt(entier);
        dos.flush();
        dos.close();
    }
    catch(IOException ioe) {
        ioe.printStackTrace();
    }
}
```

Un dernier conseil

Les opérations d'entrées/sorties sont bloquantes en Java. Cela signifie qu'un programme risque d'attendre indéfiniment dans un flux, un octet qui ne viendra jamais. Il est possible et même recommandé d'éviter de risquer un tel blocage, en effectuant la lecture dans un thread. On peut encore préalablement s'assurer que des octets sont prêts à être lus :

```
int octets_disponibles = flux.available();
if(octets_disponibles > 0)
{
    byte[] data = new byte[octets_disponibles];
    flux.read(data);
}
```

Nous en avons terminé avec cette initiation, mais l'univers des I/O Java est immense et mérite d'être exploré plus avant. Une autre fois nous nous retrouverons pour approfondir ces notions.

■ Frédéric Mazué - fmazue@programmez.com

✓ **L'actualité** en ligne

www.programmez.com

✓ **Abonnez-vous** en ligne

www.programmez.com/abonnement

Envoyez une carte de vœux virtuelle Flash, animée et ludique !

Le Nouvel An est un événement propice pour reprendre contact avec d'anciennes relations, d'anciens clients, d'anciens partenaires de travail ou tout simplement avec ses proches et ses amis. La carte de vœux est ainsi devenue l'outil numéro un pour transmettre ses vœux de Nouvel An.

Pour 2005, Programmez vous offre une carte de vœux virtuelle au format Flash, accompagnée d'un jeu online, tous les deux entièrement personnalisables et paramétrables en fonction de vos besoins. Voici pour vous une manière ludique et originale d'envoyer vos vœux par email.

L'email, l'arme fatale...

Nous limiterons notre champ d'action à la conception d'un email HTML simple, sans aucun attachement afin de garantir le maximum de chance d'atteindre vos destinataires. Malheureusement, les outils anti-spams et autres programmes de sécurité bloquent majoritairement toute intrusion indésirable dans la plupart des messageries courantes. A titre d'exemple, le logiciel Outlook (qui reste l'outil de messagerie le plus utilisé aujourd'hui), désactive par défaut la réception d'email intégrant un fichier au format Flash.

C'est pourquoi, nous vous proposons de suivre la méthode utilisée par la plupart des concepteurs d'e-cards sur Internet : une image fixe proposant un lien direct vers l'animation Flash disponible sur un serveur FTP sécurisé. Vous trouverez le fichier HTML ecard_fixe.htm, sur www.programmez.com.

Intégration du fichier ecard_fixe.htm : vous noterez que ce fichier est personnalisable à votre convenance. Vous pouvez par exemple remplacer le logo central par le logo de votre choix et surtout, modifier le lien présent sur le bouton " CLIQUEZ ICI ".

Ce dernier bouton permet d'accéder à un autre fichier HTML : le fichier ecard_anim.htm, qui lui, propose la version animée en Flash de votre carte virtuelle. Vous trouverez le lien pour le fichier en question sur site programmez (et sur le prochain CD Rom)

Revenons à présent à la conception de notre carte fixe par email. Si vous avez la curiosité d'observer le source HTML du fichier ecard_fixe.htm, vous trouverez le lien direct vers la carte virtuelle animée (en rouge dans le code suivant) :

```
<html>
<head>
<title>Bonne année 2005 !</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#000000">
<table width="100%" border="0" cellpadding="0" cellspacing="0" align="center">
<tr>
<td align="center"><div align="center"><a href="http://www.programmez.com"><imgsrc="http://timy.clic.free.fr/outerspace/ecard_fixe/ecard_fixe_01.jpg" width="567" height="86" border="0"></a></div></td>
<tr>
<td><div align="center"><a href="http://timy.clic.free.fr/outerspace/ecard_anim/ecard_anim.htm"></a></div></td>
<tr>
<td><div align="center"><a href="http://www.cyrille-renouvin.net"></a></div></td>
```



```
</tr>
</table>
</body>
</html>
```

Code source du fichier ecard_fixe.htm

Vous pouvez donc directement modifier ce lien vers la carte de vœux animée, en remplaçant le lien existant par le lien HTTP de votre choix, comme par exemple l'URL suivante :

http://www.mondomaine.com/carte/ecard_anim.htm. N'oubliez pas non plus de remplacer les liens relatifs aux trois images JPEG de votre carte virtuelle par ceux que vous mettrez sur votre FTP (voir les liens en bleu foncé dans le code source du fichier ecard_fixe.htm).

Ensuite, pour concevoir votre email, vous pouvez récupérer la totalité du code source HTML modifié et l'intégrer simplement dans un email au format HTML bien entendu. Votre carte est prête à l'envoi. N'oubliez pas de faire

quelques tests pour contrôler la validité des différents liens. Afin de vous aider à personnaliser votre carte de vœux fixe, nous mettons également à votre disposition un fichier graphique au format Photoshop avec ses différents calques.

La carte de vœux animée Flash

Comme toute bonne e-card qui se respecte, cette dernière est une carte de vœux conçue en Flash, le format d'animation numéro 1 de Macromedia, disponible sur plus de 98% des ordinateurs connectés à Internet.

Version d'évaluation gratuite de Flash MX 2004 : <http://www.macromedia.com/fr/downloads/>

Vous trouverez également cette carte de vœux entièrement personnalisable. Le fichier ecard_anim.swf est le programme Flash compilé. Son source est le fichier ecard_anim fla. Le troisième fichier, qui se nomme ecard_anim.htm, intègre un lien direct vers le fichier Flash compilé.

La carte virtuelle animée, comprend un logo que vous pouvez remplacer par le vôtre. Pour ceci, ouvrez le fichier ecard_anim fla avec Flash MX 2004 et sélectionnez le script de frame présent sur le premier calque du scénario. Dans la fenêtre Actions située vers le bas, vous trouverez le code source ActionScript du fichier Flash, localisez ensuite la fonction suivante :

```
_root.onEnterFrame = function() {
    // when a new image is selected,
    fade out, load new image, and fade in
    if (square._alpha>10 && fadeOut) {
        square._alpha -= 10;
    }
    if (square._alpha<10) {
        loadMovie("http://timy.clic.free.fr/outerspace/logo_prog.jpg", "square");
        fadeOut = false;
        fadeIn = true;
    }
    if (square._alpha<100 && fadeIn && !fadeOut) {
        square._alpha += 10;
    } else {
        fadeIn = false;
    }
}
```

```
// limit input field
if (input>5) {
    input = 5;
}
};
```

Fonction ActionScript de chargement du logo

Localisez en particulier la ligne de code suivante :

```
loadMovie("http://timy.clic.free.fr/outerspace/logo_prog.jpg", "square");
```

Cette instruction permet de charger en mémoire un fichier JPEG, en l'occurrence le fichier de logo logo_prog.jpg qui se trouve physique-



ment sur le serveur à l'adresse du site programmez. Vous pouvez remplacer ce lien par un lien vers un autre fichier JPEG de votre choix. Vous pouvez également modifier le titre et le texte de votre carte de vœux, directement dans le fichier Flash.

Le bouton "PLAY" permet de lancer le jeu en ligne que nous vous proposons gratuitement avec vos cartes de vœux virtuelles. Si vous sélectionnez ce bouton sur la scène, vous pourrez visualiser le code source ActionScript qui lui est associé :

```
on (press) {
    getURL("http://timy.clic.free.fr/outerspace/OuterSpace.htm");
}
```

Cette instruction permet au destinataire de votre e-card de jouer au jeu en ligne. Là aussi,

il vous faudra remplacer le lien URL du jeu par celui que vous utiliserez personnellement sur votre FTP pour héberger le jeu. N'oubliez donc pas de le modifier.

Concept du jeu

Vous avez finalement personnalisé vos cartes de vœux et vous les avez positionnées sur le FTP de votre choix. Il vous reste maintenant à personnaliser le jeu. Commençons donc par détailler le concept de ce jeu afin de mieux définir les différents paramètres de personnalisation accessibles.

Sur le site programmez (et le CD Rom du mois prochain), vous trouverez le code compilé du jeu, au format Shockwave. Shockwave est le second plug-in de l'éditeur Macromedia dédié au média Internet. Il permet d'aller plus loin que Flash dans la conception de contenu animé et interactif pour Internet. Il dispose même d'un moteur 3D online, qui s'est imposé aujourd'hui comme la référence des moteurs 3D dédiés à Internet. Notre programme prend en charge la détection et l'installation automatique du dit plug-in.

Spécifications du jeu

Le jeu en ligne que nous vous proposons est un jeu de palet original et amusant dans lequel vous irez affronter une série d'extra terrestres les plus loufoques. Le jeu s'intègre

directement dans le navigateur Internet par défaut de votre ordinateur (il est également compatible FireFox). Une version de démonstration jouable est intégrée sur le site.

Après une brève introduction, une synthèse vocale (en anglais uniquement) vous souhaitera la bienvenue et vous invitera à jouer en sélectionnant le mode de votre choix.

Remarque : la synthèse vocale nécessite la présence de l'interface SAPI 4.0 minimum pour les ordinateurs Windows. Sur Windows XP, cette interface vocale est présente par défaut. Sur Windows 98, il est nécessaire parfois de la télécharger par exemple ici : <http://activex.microsoft.com/activex/controls/sapi/spchapi.exe>. Pour les ordinateurs Macintosh, l'interface vocale serait disponible par défaut.

Après avoir sélectionné le mode de jeu (parmi 3 modes : Championnat, Coupe et Duel), vous

êtes invité à sélectionner un adversaire

Le premier à atteindre le score de 6 points remporte la victoire. Vos adversaires n'ont bien entendu pas tous le même niveau, ils dépendent tous de deux paramètres déterminants : la vitesse de réaction et la célérité d'action.

Aléatoirement, des bonus ou des malus apparaissent sur la scène de jeu. A la fin du mode championnat ou du mode Tournois, un classement virtuel s'affiche, suivi du classement des 10 meilleurs joueurs sur Internet. Nous décrivons un peu plus loin la procédure d'affichage et d'enregistrement des 10 meilleurs scores sur un serveur web

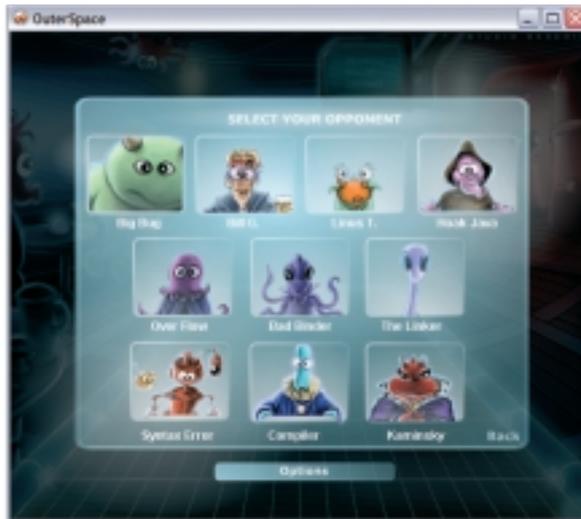
Pour finir avec la description des spécifications du jeu, vous remarquerez également l'intégration d'un écran d'option. Cet écran permet, entre autres, d'afficher les meilleurs scores en ligne et de désactiver les effets sonores, la musique ou la synthèse vocale.

Personnalisation de premier niveau

Vous trouverez donc plusieurs fichiers entièrement consacrés à la personnalisation de votre jeu. Trois fichiers en particulier retiendront notre attention : les fichiers OuterSpace.htm, data.txt et OuterSpace.dcr.

Le fichier HTML, décrit un peu plus haut dans cet article lors de l'étape de personnalisation de la carte animée, intègre dans son code HTML un fichier d'extension DCR, le fichier OuterSpace.dcr, qui est le programme du jeu compilé au format Shockwave. Afin de modifier le code source de ce programme, vous aurez besoin du programme Director MX 2004 de l'éditeur Macromedia (à récupérer sur le site : <http://www.macromedia.com/fr/downloads/>).

Mais pour vous faciliter la tâche, nous avons fait en sorte de pouvoir personnaliser ce jeu sans devoir obligatoirement éditer le code source avec le logiciel Director. En effet, le fichier data.txt est un fichier de personnalisation externe prévu à cet effet à l'aide d'un simple éditeur de texte. Ouvrez donc ce fichier texte au format ASCII :



```

http://timy.clic.free.fr/outerspace/logo_prog.jpg
NAMES*****
Big Bug
Bill G.
Linus T.
Hoak Java
Over Flow
Bad Binder
The Linker
Syntax Error
Compiler
Kaminsky
CONGRATULATIONS*****
Well done !
Fantastic !
Waow !
Not Bad !
You're fast !
Unbelievable !
ERRORS*****
Oh no !
keep it easy !
Come on !
Too hard for you !
You loose again !
No comment !
LOGO URL LINK*****
http://www.programmez.com/
MISC SPEECHES*****
your score is recorded.
Here is the top ten scores. Your score is
Welcome to OuterSpace Duel
Here comes a new challenger !
Sounds effects off
Sounds effects on
Music off
Music on
    
```

LOGO URL

```

Speeches off
Speeches on
Well done. Here is the championship classment.
Welcome in dual mode. Please, select your opponent.
ENABLE OR DISABLE SPEECH SYNTHESIS BY DEFAULT
1
    
```

Contenu du fichier de personnalisation data.txt

Comme vous pourrez le constater, le fichier data.txt est un simple fichier de données contenant diverses informations relatives au jeu. Pour commencer, à la ligne n°2, vous trouverez le lien URL de l'image JPEG de votre logo. Vous pouvez donc modifier le lien de ce logo et positionner un lien vers votre propre logo sur votre propre FTP. Pour vous aider dans le dimensionnement de votre logo, vous trouverez dans les sources les fichiers JPEG logo_tsm.jpg et logo_prog.jpg qui pourront vous servir comme exemple à suivre.

Attention : pour activer la personnalisation de votre jeu, vous devez obligatoirement disposer d'un espace FTP où transférer vos fichiers. Cette personnalisation, via le fichier de donnée externe ne fonctionnera pas en local sur votre disque dur.

Personnalisez le nom des personnages

Après la ligne 3 du fichier data.txt, vous trouverez une liste de dix noms. Ces noms, vous le constaterez rapidement, sont les noms des adversaires du jeu. Vous pouvez donc ici les remplacer par les noms de votre choix, comme ceux de vos clients, de vos collègues de travail ou pourquoi pas par les noms des membres de votre propre famille ?

Les lignes suivantes sont les différents messages de la synthèse vocale qui apparaissent durant le jeu. Si vous souhaitez désactiver cette synthèse vocale en anglais, vous pouvez ajuster la dernière ligne du fichier de configuration à 0. Cette valeur sera lue par le programme et désactivera la synthèse vocale dans le jeu.

Pour finir avec la description de ce fichier de personnalisation, vous remarquerez la présence ligne 29 d'un lien URL. Ce lien correspond au lien ajusté sur votre logo. Ici, il renvoie vers le site du magazine Programmez, mais bien entendu, vous pouvez le remplacer par le lien vers le site Internet de votre société ou de votre page web personnelle.

Pour aller plus loin dans la personnalisation de votre jeu, il vous sera indispensable d'éditer le

code source du jeu. Le code source est intégralement disponible sur le site et le prochain CD-Rom, avec le fichier OuterSpace.dir. Nous vous rappelons que ce code source est éditable et modifiable avec le logiciel Director MX 2004 de Macromedia.

Mise en ligne du jeu et des cartes

La procédure de mise en ligne de votre jeu est très simple. Un espace FTP quelconque sera suffisant. Par exemple, nous hébergeons la version de démonstration de ce jeu et des cartes de vœux sur un espace gratuit : Dans la fenêtre de notre client FTP, vous apercevrez les différents fichiers nécessaires au bon fonctionnement du jeu, à savoir les 3 fichiers décrits précédemment (OuterSpace.htm, OuterSpace.dcr et le fichier de personnalisation data.txt). Vous apercevrez également deux fichiers au format JPEG, ce sont les logos que nous utilisons dans la démonstration du jeu. Les dossiers ecard_anim et ecard_fixe intègrent les différents fichiers nécessaires au fonctionnement de la carte de vœux fixe et animée.

Module de sauvegarde des meilleurs scores

Nous terminerons ce dossier par la description du module de sauvegarde des meilleurs scores sur Internet. En effet, cette possibilité est offerte avec le jeu. Elle permet non seulement de fidéliser vos joueurs en créant une émulation collective autour du jeu, via l'esprit de compétition, mais aussi de contrôler la popularité et la jouabilité de votre jeu.

Les contraintes techniques liées au module de sauvegarde sont simplifiées : seuls seront sauvegardés en ligne, le score numérique et le nom saisi par le joueur. A ce stade, il est inutile de mettre en œuvre une base de données en ligne, un simple fichier texte ASCII de sauvegarde des données suffira, avec deux scripts CGI en Perl pour interfacer le jeu et le fichier de sauvegarde.

Vous trouverez également ces scripts CGI en langage Perl, un fichier pour la lecture et un second pour l'écriture du fichier texte ASCII :

```
#!/usr/bin/perl
```



Les 10 meilleurs scores des joueurs sont enregistrés sur Internet

```
print "Content-type: text/html\n\n";

# Ouvrir le fichier "mars_scores.txt" qui intègre
# les 10 meilleurs noms+chronos de la planète Mars :
open(SCORES_TXT,"scores_os.txt") || exit;

# Enregistrer les données de ce fichier dans la
# variable @liste_scores :
@liste_scores = <SCORES_TXT>;

# Fermer le fichier :
close(SCORES_TXT);

# Trier les données :
@liste_scores = sort { $b <=> $a } @liste_scores;

# Définir le nombre de scores à lire :
$scores_max = 20;
$index = 0;

# Pour chaque ligne, retourner les scores et les
# noms associés du plus petit au plus grand
# chrono :
foreach $ligne_score (@liste_scores)
{
    print $ligne_score;
    $index = $index + 1;
    if ($index >= $scores_max)
    {
        last;
    }
}

# Finir lecture des scores
exit;
```

Script CGI en langage PERL pour la lecture des 10 meilleurs scores en ligne (fichier read_os.cgi)

```
#!/usr/bin/perl
```

```
# Récupérer la taille des données postées :
$taille = $ENV{'CONTENT_LENGTH'};

# Récupérer les données postées :
read(STDIN, $donnees, $taille);

# Conversion de caractères :
$donnees =~ s/%([\dA-Fa-f][\dA-Fa-f])/pack("C",hex($1))/eg;
$donnees =~ tr/\+\/ /;

# Créer un tableau de données avec les données postées :
@split_data = split("&",$donnees);
foreach $index (@split_data)
{
    ($index1, $data1) = split("=", $index);
    $info{$index1} = $data1;
}

# Ouvrir le fichier texte :
open(TXT_SCORES,">>scores_os.txt") || exit;

# Copier les données "score" et "nom" dans le fichier :
print TXT_SCORES "$info{'score'}&$info{'nom'}&\n";

# Fermer le fichier :
close(TXT_SCORES);

# Finir l'écriture du score et du nom :
exit;
```

Script CGI en langage PERL pour l'écriture des 10 meilleurs scores en ligne (fichier write_os.cgi)

Pour interfacer vos scripts CGI avec le jeu, il ne vous sera pas obligatoire d'éditer le code source du jeu qui utilise déjà les fichiers disponibles sur un serveur FTP.

Nous espérons que le jeu ne nuira pas trop à votre productivité pour l'année 2005...

Par ailleurs, si vous souhaitez obtenir plus d'information sur comment aller plus loin dans la conception de jeux en ligne pour Internet, ou sur CD-Rom, nous restons à votre disposition sur l'adresse email suivante : tech@tsm-internet.com. A vous de jouer maintenant et Bonne Année 2005 !

■ Laurent Jayr

Auteurs du jeu : Laurent Jayr - tech@tsm-internet.com - Graphisme : Cyrille Renouvin cyrille@cyrille-renouvin.net