

# Programmez!

Mensuel • Novembre 2003 • N°58 • 5,95 €

## CRYPTAGE p38

La science du secret

**MULTI  
PLATE-FORME :**  
Terrain miné

**NOUVEAU :**  
Microsoft  
OFFICE System

**TESTS :**

Borland C#Builder  
s'impose sur .net

Macromedia  
Dreamweaver MX 2004

### PHP p64

- Développer un Framework avec PHP
- PHP Documentor

### TECHNO p54

Synchroniser des Codes Sources sous Windows et Linux

Architectures distribuées : La solution Remoting

### FORMATION p34

EPITECH : L'apprentissage par Projet

### PRATIQUE p66

Mettre en œuvre RMI

Printed in France - Imprimé en France  
BELGIQUE 6,45 € - SUISSE 12 FS  
LUXEMBOURG 6,45 € - Canada 8,95 \$ CAN  
[www.programmez.com](http://www.programmez.com)

M 04319 - 58 - F: 5,95 E



# L'Open Source se fédère

## Vers une collaboration Apache - ObjectWeb ?

**Président du comité exécutif d'ObjectWeb, Christophe Ney nous explique pourquoi et comment le consortium qu'il représente et Apache vont coopérer autour J2EE, notamment sur le projet Geronimo.**



**Programmez ! : Vous avez récemment pris contact avec Greg Stein à l'ASF. Quels étaient vos objectifs ?**

**C.N.** La Fondation Apache utilise une licence de type BSD alors qu'ObjectWeb recommande l'utilisation de la licence LGPL. ObjectWeb peut donc utiliser et utilise déjà les composants d'Apache, mais la réciproque n'est pas vraie. Il était donc naturel pour ObjectWeb d'étudier une solution pour permettre le renforcement de notre coopération avec ASF. Après s'être réunis, le comité directeur et le collègue d'architectes ont recommandé aux membres une évolution de la licence du code, dont ils détiennent le copyright, vers BSD, pour les composants qui seraient de l'intérêt de la communauté Apache, dans le cadre du projet Geronimo.

**Programmez ! : Concrètement, êtes vous avancé dans votre collaboration ?**

**C.N.** Oui. La communauté Apache a reçu très favorablement notre proposition. Dans une lettre ouverte, son comité directeur a vivement encouragé les développeurs intéressés par Geronimo à regarder de très près les composants disponibles chez ObjectWeb. Pour commencer, ils envisagent d'utiliser les composants ObjectWeb que sont JOTM (moniteur transactionnel) ou ASM (manipulation de bytecode Java).

**Programmez ! : Allez-vous vous limiter à l'échange de briques logicielles ou travaillerez-vous ensemble sur de nouveaux projets ?**

**C.N.** ASF est une fondation nord-américaine, tandis qu'ObjectWeb, d'origine européenne, se révèle plus multiculturelle. Les deux communautés peuvent donc rapidement devenir très complémentaires : elles ont toutes deux un rayonnement mondial et partagent des ambitions et un fonctionnement similaires. Certains développeurs contribuent déjà à Apache et à ObjectWeb. Permettre la réutilisation dans l'esprit de compétition qui anime les deux communautés me semble tout à fait sain.

**Programmez ! : Quid d'une fusion Geronimo / JOnAS ?**

**C.N.** Il n'y a pas grand chose à fusionner pour l'instant. Geronimo n'est qu'un projet, JOnAS est un serveur complet J2EE 1.3, bientôt J2EE 1.4, intégrant des composants ObjectWeb et Apache. A terme, nous pourrions effectivement développer un serveur d'application libre en commun. Mais il est encore bien trop tôt pour s'engager sur ce point et des incompatibilités de licence demeurent.

■ **Propos recueillis par David Thévenon**

## Geronimo : J2EE made in Apache

La fondation Apache va créer un serveur d'application J2EE complet et open source. En plus de JOnAS du consortium ObjectWeb et de JBoss, édité par le JBoss Group, les développeurs disposeront d'ici quelques longs mois d'une troisième alternative J2EE open source : Geronimo. Lancé par la fondation Apache, cette plate-forme J2EE complète reposera sur un socle JMX et intègrera de nombreuses briques open source existantes telles qu'OpenJMS, Castor ou bien encore Tomcat. Geronimo devrait être distribué sous licence ASL. Cette dernière permet à un éditeur de logiciels commerciaux de revendre Geronimo sous code fermé. Développeurs et entreprises disposeront donc d'un choix supplémentaire. Comme le faisait remarquer Greg Stein, qui gère ce projet à la fondation Apache, à notre confrère Inforworld : " les entreprises disposeront ainsi d'un choix supplémentaire. Rien ne les empêchera de se rabattre vers des produits commerciaux si leur expérience de l'open source les déçoit ".

Le site officiel : [incubator.apache.org/projects/geronimo.html](http://incubator.apache.org/projects/geronimo.html)

## JOnAS sur la voie de la certification

Le serveur d'application J2EE open source d'ObjectWeb sera bientôt distribué par Red Hat. Grâce au programme "scholarship" de Sun, JOnAS pourrait également être le 1er serveur d'application open source certifié J2EE. Lancé en 1999, le projet JOnAS (Java Open Application Server) en est aujourd'hui à sa version 3.3. Ce serveur d'application J2EE open source implémente l'ensemble de la norme J2EE 1.3, au travers de sous-projets tels que JORAM (JMS) ou JOTM (JTA, JTS). JOnAS n'est en effet



qu'un des 35 projets abrités par le consortium ObjectWeb. Fondé en 2002 par des entreprises telles que Bull, France Télécom et l'INRIA, il a été récemment rejoint par Red Hat au mois d'août. L'éditeur au chapeau rouge inclura d'ici la fin de l'année JOnAS, Tomcat (JSP / servlet) et le serveur web Apache à son offre Red Hat Advanced Server. Il se devait donc de prendre part à l'activité du consortium. ObjectWeb est également la première communauté open source à bénéficier du programme "scholarship" de Sun. Ce programme permet à une

organisation à but non lucratif de bénéficier du soutien de l'éditeur tout au long du processus de certification J2EE. Concrètement, ObjectWeb bénéficiera de l'appui de techniciens de chez Sun pour certifier JOnAS et n'aura pas à payer la redevance annuelle imposée aux éditeurs pour faire partie du " club J2EE ". Pour Stacey Quandt, analyste au cabinet Forrester Research, "JOnAS dispose désormais d'un support commercial qui est indispensable pour convaincre les entreprises américaines. Et en obtenant la première certification décernée à une solution open source, JOnAS pourrait rogner des parts de marché à IBM et BEA". ■ D.T.



**Nouveau ! : [www.emploi-programmez.com](http://www.emploi-programmez.com)**

# La révolution industrielle du logiciel



**D**ès les années 70, la programmation orientée objet est apparue afin de gérer la complexité des développements informatiques. Cette complexité n'a jamais cessé de croître et ce, à cause de l'exigence endémique mais légitime des utilisateurs. Or l'esprit humain n'a pas pu évoluer en quelques décennies. Pour résoudre ce tour de force, des techniques et des outils ont fait leur apparition.

L'erreur serait de restreindre les connaissances informatiques à l'Outil, car ce dernier n'est qu'un moyen et en aucun cas une finalité. Ce raisonnement d'une banalité affligeante pour tout artisan n'est pas forcément acquis pour l'informaticien. Un beau rabot n'a jamais fait un bon menuisier !

Dès le 19ème siècle, la production industrielle s'est théorisée. Ceci a permis l'émergence de méthodes et de techniques de production encore d'actualité (Fordisme, taylorisme, ...). L'informatique connaît les mêmes mutations, l'avènement de nouvelles techniques de développement (Merise, UML, MDA, RUP, ...) permet de structurer les étapes de production d'un logiciel. Dans cette chaîne de production logicielle, chacun a sa place et, à l'instar des chaînes de manufacture, les rotations de postes et les promotions sociales sont également possibles. Cette vision industrielle du développement logiciel peut déranger, mais force est de constater que seule cette rigueur permet la création de logiciels robustes, fiables et aisément maintenables et à moindre coût. Ce nivellement par le haut des processus de développement est une marque de professionnalisme. Sur le long terme, cette démarche s'avère fructueuse tant pour les clients qui possèdent des produits aux comportements prédictibles et donc aisément maintenables, que pour les développeurs qui perçoivent la valorisation de leur travail. Le chef de projet lui aussi aborde l'avenir avec plus de sérénité, confiant que ses développements sont à même d'absorber sans difficulté de multiples évolutions.

Dans l'industrie, les tâches pénibles et fastidieuses ont été allégées par une automatisation du processus de production, les productions ont été rationalisées par la mise en place de normes, de standards. On constate une tendance similaire en génie logiciel, normes et standards s'appellent UML et design patterns et conduisent à l'élaboration de logiciels plus construits, plus compréhensibles et par là même, plus évolutifs. L'automatisation logicielle est également émergente, et se traduit par la mise en place de processus permettant la génération de codes sources (Ex : Rational XDE, DTM...). De gros progrès ont été constatés ces dernières années dans ce domaine et ces outils, dits de méta-génération, offrent de réels avantages concernant la réduction des processus de développement. Faire mieux en travaillant moins, n'est ce pas là l'objectif de tout informaticien, voire de tout homme ? Le progrès n'est qu'un corollaire de la fainéantise humaine, dans les sciences comme en informatique !

■ **Nicolas CHABRIER**

Nicolas CHABRIER, 26 ans, ingénieur en informatique diplômé de l'ESSAIM,  
est PDG et co-fondateur avec Sébastien ROS  
de Evaluant, éditeur de Data Tier Modeler - [www.evaluant.com](http://www.evaluant.com)

# Office 2003 : un mille-feuille applicatif

**Lancé simultanément dans toutes les langues le 21 octobre, la nouvelle édition de la suite bureautique a fait sa mue technologique, avec 4 grandes innovations : XML à toutes les saucées pour l'entreprise (ce qui n'est pas une nouveauté sauf pour Office), IRM pour gérer et sécuriser les échanges de données sensibles, SharePoint pour faciliter le travail collaboratif et enfin l'encre numérique (évolution logique vers le TabletPC).**

**R**ebaptisée "Microsoft Office System", elle hisse XML au cœur de son fonctionnement. Plus que jamais, l'éditeur vise l'entreprise, et propose un environnement pur Windows de bout en bout de la chaîne, en introduisant le formulaire dynamique, via InfoPath. Au-delà d'InfoPath qui concentre l'essentiel des commentaires, Office 2003 possède d'autres nouvelles fonctions. En soi, les outils Office évoluent avec leur temps sans apporter de bouleversantes fonctions. Sur certains aspects, ils rattrapent le retard pris sur la concurrence. Le nouveau cheval de bataille est bel et bien l'organisation, essentiellement la manière de mettre en place du collaboratif.

## Office System : 4 couches

Pour obtenir une chaîne homogène de bout en bout (pour verrouiller diront certains), Microsoft a créé Office System. Cet ensemble vise plusieurs objectifs : travail collaboratif, meilleure sécurisation des échanges, environnement homogène, formulaire XML, gestion documentaire simplifiée. La masse toujours croissante des informations, documents et mails, pose de nouveaux problèmes aux entreprises et aux utilisateurs. Office System y répond en offrant un "mille-feuille" fonctionnel, réparti en 4 couches. En façade, on a les déclinaisons d'Office et les divers outils de productivité (FrontPage, Visio, Project, formu-

laire, etc.). En dessous, les serveurs, puis la partie service et enfin les solutions. Tout cela est orienté entreprise / bureautique / processus métiers et tourne sur une plate-forme dual : systèmes et serveurs (Biztalk, SQL Server, Exchange Server, Windows XP, Windows Server 2003). Si on voit immédiatement l'intérêt d'une telle construction modulaire, cela obligera à maîtriser les nouveaux outils et à les déployer.

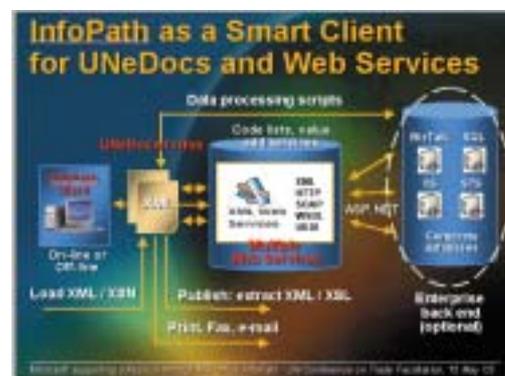
## Protection des données : vers un verrouillage ?

L'effort est louable de vouloir sécuriser au maximum les échanges de données. Les fonctions DRM et eDRM, fournissent tout un ensemble de mécanismes pour identifier les utilisateurs, les machines, les applications et même le périmètre d'accès d'une utilisation dans un document. Il est aussi possible de mettre en place toute une stratégie de protection des données (quel que soit le format). Par exemple, on peut interdire que le texte d'un email soit imprimé, copié (puis collé) ou de faire un copie-écran, ou exclure une machine non "certifiée" pour le serveur. On peut affiner les auto-

risations d'utilisation d'une application, d'un document, selon les autorisations de chaque utilisateur. Si on souhaite mettre en place un eDRM, il faut utiliser le WRMS (Windows Rights Management Services) et Windows 2003 Enterprise Edition. Mais, il faut aussi que le poste client soit équipé du Windows Rights Management. De plus, les applications doivent être compatibles RMS (Office System l'est). Pour les autres, il faudra attendre des mises à jour. On peut aussi implémenter les API adéquates. Si, on peut être raisonnablement favorable à un tel système de sécurité et de confidentialité, force est de constater que cela oblige à définir précisément l'architecture et à implémenter de nouveaux serveurs et clients.

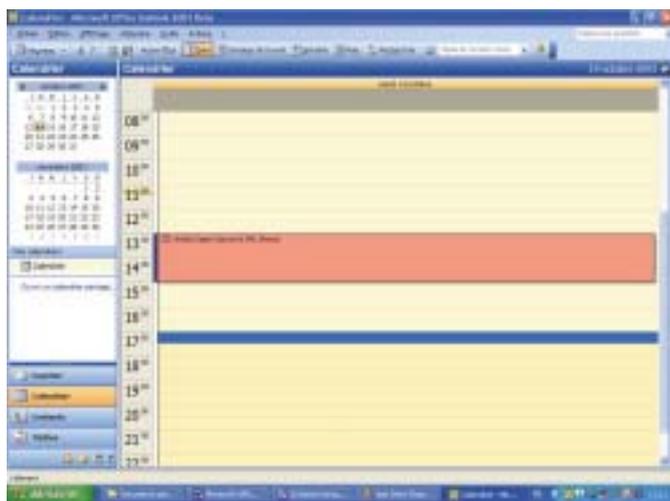
## InfoPath : des qualités et des défauts

D'Office 2003, on retient bien souvent un unique élément : InfoPath. Celui-ci s'appuie sur XML et les schémas. InfoPath permet de créer des formulaires dynamiques que l'on utilise ensuite. On construit les formulaires directement dans l'outil



Office InfoPath 2003. On dispose d'un véritable IDE. Il est très facile de créer des interfaces de saisie complexe, avec des zones extensives. Mapper une base de données sur des champs InfoPath est presque un jeu. La compatibilité avec SQL Server est assurée, tout comme les autres SGBD, comme Oracle. Si on comprend l'intérêt d'un tel dispositif, il existe aujourd'hui des limitations à InfoPath. Tout d'abord pour exploiter toutes les possibilités d'un formulaire, il faut disposer d'Office 2003. Actuellement, il fonctionne uniquement en environnement Windows. Un formulaire InfoPath ne fonctionne ni sur environnement non Windows, ni sur PocketPC ou Smartphone, et encore moins sur le Web. Le seul terminal supporté, est le TabletPC. Le déploiement en réseau se fait via le SharePoint. Si on souhaite une publication large, il est possible d'utiliser le fichier XML puis de le passer dans une page ASP.NET, voir Xform. Pour un formulaire complexe, on peut envisager un format PDF. InfoPath rentre en concurrence directe avec Xform. Xform est le standard soutenu par le W3C et surtout utilisable, via un simple navigateur. Cependant, Xform n'a pas la richesse fonctionnelle d'un InfoPath, et demeure lié à la présentation des données et ne mêle pas les aspects bureautiques. On doit dis-





poser de SharePoint Services pour les fonctions avancées et de Windows Server 2003. L'avantage d'InfoPath est de pouvoir offrir une solution puissante mais simple de conception. Enfin, n'oubliez pas qu'InfoPath est un outil strictement indépendant. Il n'est pas utile de posséder Word ou Excel. Et les outils de bureautique Office fonctionnent aussi sans InfoPath installé. Par contre, on peut utiliser un Word 2003 pour servir d'application d'accueil d'un formulaire InfoPath. InfoPath intéressera pas

ou très peu les utilisateurs lambda. Il vise surtout l'entreprise.

### Côté développeur

Et VBA dans tout cela ? Non il n'est pas encore mort. La raison en est toute simple. Il faut bien pouvoir utiliser les multitudes d'applications bureautiques Office conçues avec ! Mais, il ne faut pas se faire trop d'illusions. Le modèle de développement .NET primera. Ainsi, il est possible de développer un programme Office avec Visual Studio .NET.

## InfoPath s'installe à la Commission économique des Nations Unies pour l'Europe (UNECE)

Cette vénérable institution a été créée en 1947. Elle a su évoluer avec son temps et les technologies modernes. Pour preuve, depuis plusieurs mois, InfoPath est utilisé avec Office Professional édition 2003. Le but était de construire des formulaires XML rapidement et à moindre coût, tout en respectant les standards. InfoPath intervient dans les UNeDocs. Il s'agit d'un projet devant être l'équivalent papier (UN/EDIFACT). Il s'agissait de reprendre le contenu papier et d'utiliser la souplesse des documents XML. En fait, l'UNECE a défini un standard des documents commerciaux internationaux. L'institution a démarré la version électronique de ce standard en 2002, basé sur les schémas XML et en utilisant les Web Services. C'est alors que les technologies InfoPath et l'ensemble Office attirent le regard des responsables. Un prototype fut alors mis au point et dévoilé il y a quelques mois. UNeDocs est le standard électronique d'UN/EDIFACT utilisé par plus de 300 000 entreprises à travers le monde ! Il définit les éléments basiques des documents commerciaux. Il reprend la syntaxe du standard papier. On pourra donc utiliser UNeDocs dans des applications d'entreprise .NET. Plusieurs éditeurs proposent déjà une implémentation. UNeDocs est d'une grande souplesse. Il est possible de valider un document soit par un script, soit par un Web Service, idem pour la conversion en HTML ou en PDF.

L'arrivée d'Office 2003 modifie quelque peu le développement d'applications Office. Si le nouvel office continue à supporter VBA, Office supporte désormais VB.NET, et surtout C#. Bref, Microsoft harmonise son modèle de développement d'applications Office vers le modèle .NET, ce qui était impossible à réaliser directement avec VBA. Le code VBA issu d'Office XP est compatible avec Office 2003 (dixit Microsoft). L'outil de conception change aussi. On peut tirer profit de Visual Studio .NET 2003.

Le code .NET que l'on écrira sera utilisable sous Word et Excel. L'évolution de VBA vers le modèle .NET se vérifie dans les outils. Avec Office 2003, on dispose du Visual Studio Tools for Office. Il s'agit de bénéficier de VS.NET et du framework .NET en créant des applications Word ou Excel. C'est clairement le futur pour les développeurs. Il est possible de convertir du code VBA en code VB.NET, malheureusement, la conversion n'est pas automatique. Pour utiliser VS Tools for Office, il est nécessaire de posséder VS .NET Professional 2003 (ou une version supérieure). Avec l'implémentation du XML, on bénéficie aussi du support des Web Services. Microsoft introduit aussi l'idée de Smart Document. Il utilise le concept des "smart tags" déjà présent dans Office XP. Il s'agit de simplifier, l'utilisation et le travail sur des documents. On peut ainsi créer des "documents riches" basés sur XML pour Word et Excel.

Une API est disponible pour les développeurs. On peut utiliser n'importe quel langage supporté par la CLR. On n'oubliera pas de mentionner la nouvelle version de FrontPage. Il prend un sérieux coup de jeune et tente de jouer dans la cour des grands en proposant un outil bien plus pro. On peut utiliser les fonctions de SharePoint Server, ASP.NET pour les sites dynamiques, XML et les XSLT.

d'InfoPath ou encore des nouvelles fonctions de messagerie instantanée "vidéo", un peu à la iChat d'Apple, il faudra installer les derniers outils serveurs de Microsoft. D'ailleurs l'orientation entreprise se confirme par la disponibilité de fonctions dans l'édition professionnelle d'Office et non dans la version standard. Ainsi, l'utilisation de XML n'est pas possible en Office standard (sauf en sauvegarde). Cette limitation peut étonner quand on sait qu'Open Office et StarOffice le proposent. De plus, mettre en avant la productivité n'est pas suffisant. Il faut convaincre de l'intérêt d'Office 2003. D'autre part, une guerre des prix existe en entreprise. Microsoft n'hésite plus à consentir de gros rabais. Sun fait la même chose avec StarOffice. Les dernières enquêtes sur la migration Office 2003 montrent un délai important. On pourra mesurer l'impact d'Office 2003 au plus tôt, à la fin du printemps 2004.

■ **François Tonic**

### Un défi économique pour Microsoft, stratégique pour l'entreprise

Pour Microsoft l'enjeu est énorme. Office est depuis des années, le véritable trésor de guerre de l'éditeur. La migration vers Office 2003 et au-delà vers Office System est un facteur de réussite ou d'échec. Office 2003 n'est qu'un aspect de l'architecture souhaitée par Microsoft. Office System montre parfaitement cette version. Si on veut véritablement exploiter les outils de travail collaboratif,

## Fiche technique

### Points forts :

InfoPath  
Office System  
Interface  
Intégration des outils  
Gestion des droits

### Points faibles :

Manque d'ouverture d'InfoPath  
XML non généralisé  
Trop orienté Windows  
Infrastructure logicielle lourde

# Macromedia Dreamweaver MX 2004 :

## Plus dynamique que jamais

Avec quelques semaines d'avance sur la nouvelle année, Macromedia nous gratifie de toute une série de mises à jour de sa suite Internet. Pour ce présent test, nous nous limiterons à Dreamweaver MX 2004.

Dès la page de démarrage, le ton est donné : HTML, ColdFusion, PHP, ASP, ASP.NET (via C# ou VB), JSP, CSS ou encore WML. Il ne manque que iMode ou Smartphone pour compléter la panoplie des choix. Cette bonne impression se conjugue avec la toute nouvelle interface utilisateur. Désormais, la même interface est reprise dans l'ensemble des outils de Studio MX.

La présentation et l'ergonomie générale sont plutôt réussies avec un accès rapide aux principales fonctions. Mais attention à l'overdose de fenêtres sur des écrans trop petits ! La fenêtre de conception est bien agencée. Trois modes d'affichages sont possibles : code, design, ou mixte (le fameux split). Le mode split affiche le code et le design. Il a notre préférence pour l'accès aux deux éléments. Par contre, il est surprenant de constater un fonctionnement différent en cas de modification.

Si celle-ci est faite directement dans l'interface, le code est bien modifié à la volée, par contre, quand on modifie à partir du code un objet, il faut cliquer sur un autre objet pour mettre à jour l'objet modifié dans la partie interface. On aurait apprécié un comportement identique.

La fenêtre propriétés n'est pas une réussite au niveau présentation et ergonomie. Une structure "plus fenêtre" aurait été souhaitable. On apprécie le choix dans l'affichage de la barre d'outils contenant les objets d'interface (par onglet ou via un menu pop-up).

### Intégration des autres outils

Pour faciliter la vie du développeur et éviter de recoder des fonctions très souvent utilisées, il peut mettre à profit le module Snippets. Par défaut, Macromedia met à notre disposition plusieurs dizaines de codes prêts à l'emploi. L'accent a été mis sur la meilleure gestion et création d'une feuille CSS. On a droit à un meilleur rendu à l'écran, à des nouvelles pro-

priétés CSS, on a même droit à des conseils de codage CSS. L'intégration des autres outils de la gamme Studio est un point important. Un onglet flash est disponible, permettant d'incorporer rapidement des éléments conçus en Flash, petit plus sympathique.

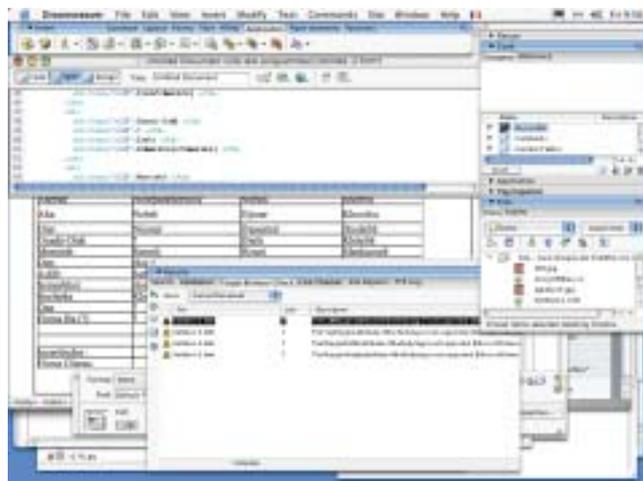
On bénéficie aussi de la technologie Fireworks pour les images. On peut ainsi directement réaliser de petits traitements (recadrage, redimensionnement, balance). La pertinence de ces nouvelles fonctions reste à démontrer, même si la fonction de recadrage peut rendre quelques services. Sur la version testée, nous avons eu quelques soucis de réactivité de ce module.

Les petites astuces pour le développeur que Macromedia propose sont intéressantes, mais guère plus : menu contextuel pour manipuler un bloc de code, une fonction rechercher – remplacer modifiée, un inspecteur de balises, une nouvelle présentation colorée de la syntaxe. Rien de bien transcendant, juste de quoi améliorer le travail au quotidien, et c'est déjà bien... On appréciera le support d'Unicode et surtout, la nouvelle fonction permettant de récupérer par simple copier-coller une sélection d'un document Word ou Excel. Pour les tableaux, c'est un vrai bonheur. Surtout, on peut modifier la taille et le style des cellules.

Une des forces de Dreamweaver est le support des pages dynamiques. Il devient plus facile d'intégrer les sources ColdFusion, l'outil maison : nouvelle version d'HomeSite+, accès au débogueur ColdFusion MX, support balises de la version MX 6.1 ainsi que des composants. Mais surtout, l'accent a été mis sur l'intégration d'ASP.NET 1.1. Côté Java, on peut aisément

intégrer un JavaBeans dans une page JSP. Pour la vérification du code, l'ensemble joue son rôle. On regrettera que Dreamweaver ne vérifie pas le code Safari, Mozilla ou encore iMode. On aurait pu aussi s'attendre à une meilleure vérification de code et des balises à la volée. En tout cas, on ne peut pas dire que Macromedia ne fait pas d'efforts. La nouvelle interface utilisateur gagne en lisibilité et productivité, même si quelques défauts d'affichage peuvent être gênants. Si certaines nouvelles fonctions paraissent assez anecdotiques, d'autres sont les bienvenues, comme sur le CSS. Un excellent produit qui demeure la référence du marché.

■ François Tonic



## Fiche technique

Nom du produit : Dreamweaver MX 2004  
 Editeur : Macromedia  
 Configuration minimale : Pentium III 600 (Win98SE, Win2000, XP, Server 2003) ou PowerMac G3 (MacOS X 10.2.6 ou supérieur)  
 Prix : 479 €

**Points forts :**  
 Interface réussie  
 Bibliothèque de codes  
 Gestion CSS  
 Unicode

**Points faibles :**  
 Lenteurs sous MacOS X  
 Prise en compte des modifications en mode split  
 Fenêtre propriétés  
 Supports terminaux mobiles

# Avec C#Builder, Borland s'impose sur .Net

Borland concurrence Microsoft sur son propre terrain et propose la première version d'un produit qui fait déjà figure de ténor du développement C#.

Proposer un outil de développement pour C# et .Net est une démarche courageuse car cela impose de faire mieux que ce qui existe déjà. En testant C#Builder, voyons si Borland a réussi son pari. Le produit se décline en 4 versions : personnelle, professionnelle, entreprise et architecte. La version personnelle est librement téléchargeable pour le développement d'applications non commerciales. Saluons au passage ce bel effort de la part de Borland. Nous testons la version architecte.

## Prise de contact

Le produit s'installe bien. Les produits qui l'accompagnent également. Signalons la présence de CaliberRM, qui est l'outil qui automatise la gestion des besoins, afin de faciliter la priorisation des tâches et des mises à jour des projets. Viennent également Interbase 7 avec une

licence de développement, l'outil d'analyse des performances Optimizeit, La plate-forme ECO dont nous reparlerons, le système de gestion de version StarTeam et Janeva qui jette un pont entre .Net et CORBA et EJB. (Liste non exhaustive).

Au premier contact avec le produit, nous remarquons que Borland a abandonné le look à la Delphi pour adopter celui de Visual Studio .Net (figure 1). La

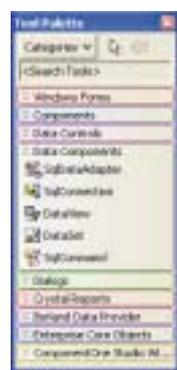


Figure 2 : La palette de composants a été totalement remaniée.

ressemblance va jusqu'à proposer, au démarrage de l'outil, une page HTML contenant des liens vers les derniers projets développés, des didacticiels et des liens vers des pages d'informations relatives au produits Borland. Un air de déjà vu. Nous retrouvons malgré tout des éléments connus tels que l'inspecteur d'objets, le gestionnaire de projets, etc. Nous remarquons que la palette de composants a été entièrement remaniée (figure 2). Celle-ci est maintenant arrimable à la manière d'autres éléments de l'EDI et utilisable en tout sens tel un béret basque. Nous avons particulièrement apprécié cette nouvelle palette qui s'intègre

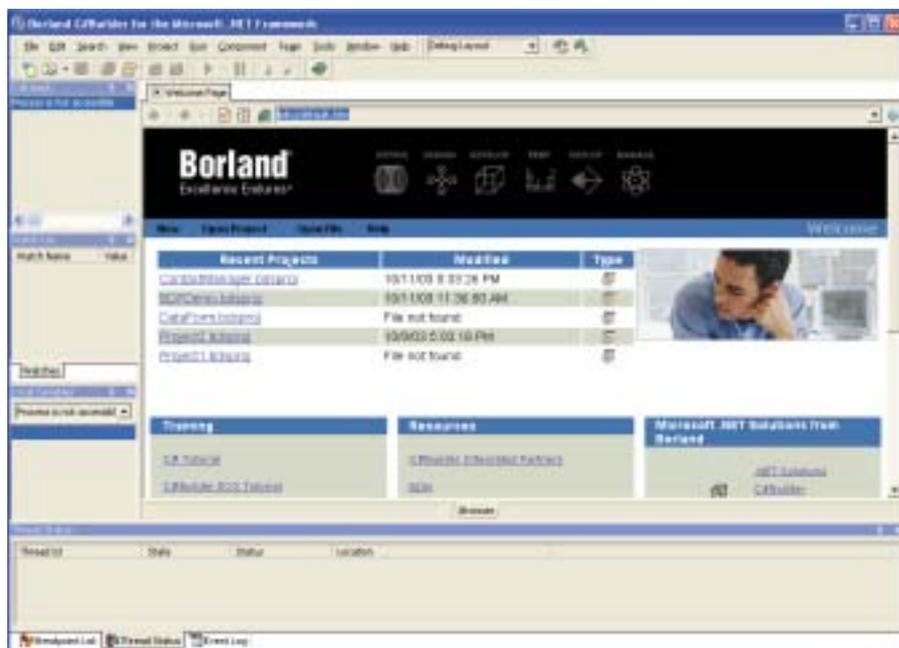


Figure 1 : Comme un air de déjà vu... mais la ressemblance s'arrête là.

remarquablement bien à l'ensemble, s'ajuste selon le contexte et en mode édition propose des échantillons de code, à insérer dans le source d'un simple clic. Nous apprécions encore particulièrement que chaque élément du projet dispose d'un élément de fermeture dans son propre onglet. Très pratique. Concluons en disant que nous avons un outil de développement rapide bien pensé, qui permet la navigation aisée entre codes, conception des fiches et UML (figure 3) et doté de trouvailles astucieuses, même si nous pensons que l'auto-indentation du code pourrait être améliorée. Notons encore C#Builder sait importer/exporter des projet de/vers Visual Studio .Net.

## Les bases de données.

C#Builder, comme les autres outils Borland, permet de développer en live, avec de nombreuses bases de données éventuellement distantes. Ici SQL Server, DB2, Oracle, Interbase. C#Builder a des composants ADO améliorés par rapport à ce que l'on trouve dans Visual Studio .Net. et le processus de développement n'en est que plus agréable. Nous regrettons toutefois vivement que C#Builder ne permette pas de développer avec MySQL, ou même

pourquoi pas, PostgreSQL. Puisqu'il possible de faire coopérer .Net et MySQL (Voir l'article de Xavier Leclercq dans Programmez! 43), c'est possible aussi sous C#Builder, mais manuellement. Nous souhaitons que cette lacune soit comblée dans une prochaine version. Nous avons rencontré quelques fonctionnalités géniales: La génération d'une chaîne de connexion par simple glisser-déposer d'une table, ou la génération de requêtes SQL uniquement à partir de clic de souris.

## La modélisation

Comme les autres produit Borland, C#Builder met UML à la place d'honneur. Fort bien. Nous avons en outre ECO (Entreprise Core Objets) qui est une nouveauté de taille. ECO est l'implémentation du standard MDA (Model Driven Architecture) de l'OMG. Il est difficile d'expliquer en peu de mots ce qu'est ECO. Simplifions en disant que c'est un outil au dessus de UML. Alors qu'UML permet de modéliser des classes et les relations entre elles, ECO permet de modéliser l'application elle même. Travailler avec ECO c'est en quelque sorte, faire de la conception incrémentale. Les modifications de conception se traduisent vers le bas par la

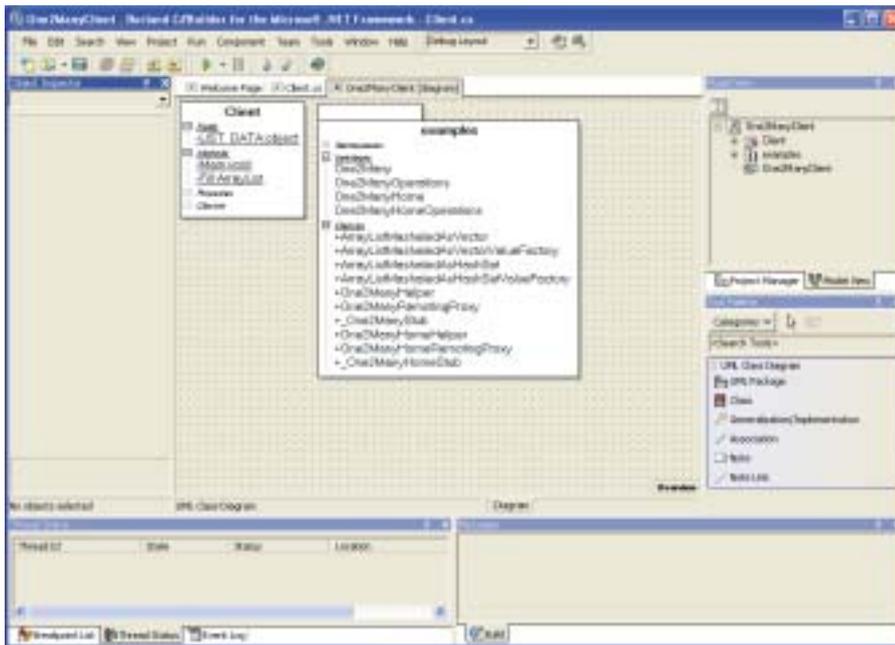


Figure 3 : A tout instant vous avez accès à la modélisation UML de votre projet.

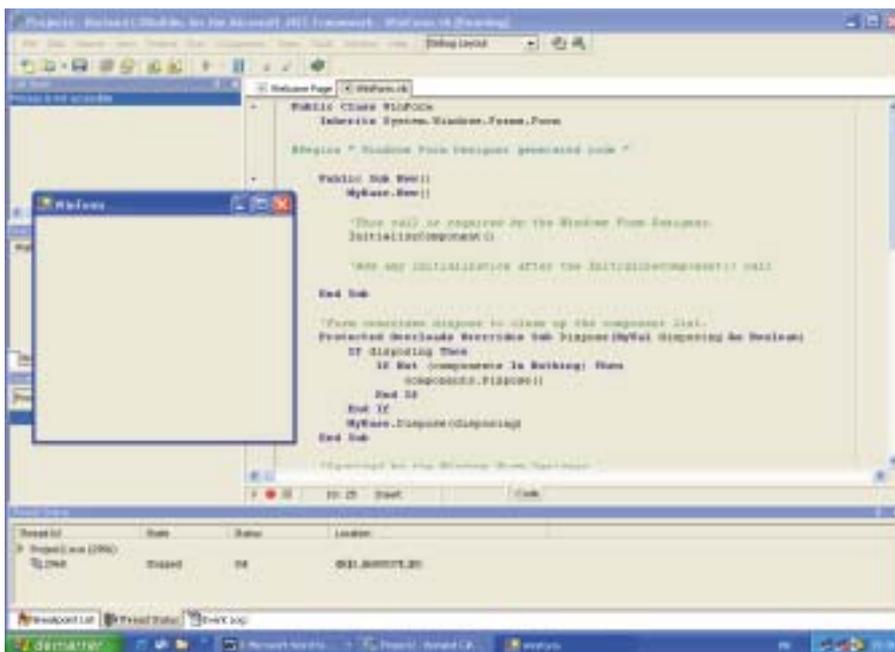


Figure 4 : C# Builder permet de programmer...en VB.

génération automatique de code. Les modifications de codes se traduisent vers le haut par une modification du modèle. Nous avons donc un outil qui chapeaute toute la chaîne de développement et les concepteurs n'ont pas besoin de connaître le langage utilisé. Indirectement ECO évite que le développeur interprète le modèle et le cantonne à un rôle d'implémentation. A l'exécution le modèle architectural réside dans un espace au sein duquel vivent les instances de ces classes. Si un objet est une instance d'un composant d'une application, par

comparaison, on peut dire que l'espace ECO est une instance de l'architecture elle-même. L'espace ECO est une sorte de container d'objets, de cache et de contexte transactionnel. A l'intérieur de cet espace, des composants de persistance peuvent jeter un pont entre l'application et des fichiers XML ou des SGBDR. Pour bien se rendre compte de ce que vaut un outil comme ECO, il faut l'utiliser très longuement, ce que nous n'avons évidemment pas pu faire. Toutefois sa présence au sein de C#Builder nous semble heureuse.

### C#Builder vs Visual Studio .Net

Grande est la tentation de comparer ces deux là. Nous allons donc nous y essayer. Visual Studio .Net est certes excellent. Son éditeur de code est (peut être) un chouïa meilleur. Nous pensons que son aide contextuelle est bien meilleure de celle de C#Builder. Visual Studio .Net permet de travailler avec d'autres langages que C#. La plate-forme .Net ayant des qualités indéniables, y travailler avec son langage préféré est une perspective alléchante. C#Builder comme son nom ne l'indique pas, permet de programmer en VB.Net (figure 4) et bien sûr pour ASP. L'éditeur propose d'ores et déjà une coloration syntaxique pour HTML/XML, Php, JavaScript et SQL. Autant dire tout pour le développement Web. En revanche, lors de nos essais nous n'avons pas su voir comment C#Builder se comporterait avec un plugin Python pour .Net par exemple. Sauf erreur de notre part, la documentation de C#Builder est muette sur ce point. Le gros problème de Visual Studio .Net est que Microsoft ne connaît guère les technologies qui ne viennent pas de Microsoft. C#Builder a pour lui de connaître l'existence de nombreux SGBDR et surtout il est ouvert à d'autres technologies. Nous pensons bien sûr à EJB et CORBA. Nous apprécions encore que C#Builder propose des outils de modélisations modernes. Pour ces raisons, si nous devons investir dans un outil de développement pour .Net nous choisirions C#Builder.

■ Frédéric Mazué - [fmazue@programmez.com](mailto:fmazue@programmez.com)

## Fiche technique

Nom du produit : C#Builder Architecte

Editeur : Borland

Configuration minimum : Windows 2003 server ou Windows XP Pro ou Windows 2000 Pro ou Server 128 Mo de ram (Nous recommandons beaucoup plus)

Prix : 2499 € HT pour une version architecte (Edition Entreprise : 1799 € HT, Professionnel : 999 € HT, Personnel : 69 € HT)

#### Points forts :

Certaines fonctionnalités de l'EDI notamment palette de composant et fonctionnalités base de données. L'ouverture du produit sur le monde extérieur. L'efficacité globale.

#### Points faibles :

L'absence de composants ADO pour MySQL. Trop peu d'exemples d'applications. Appréciation générale C#Builder est un outil formidable et surtout prometteur pour l'avenir.

# InstallShield DevStudio 9 : l'atelier pour installateur

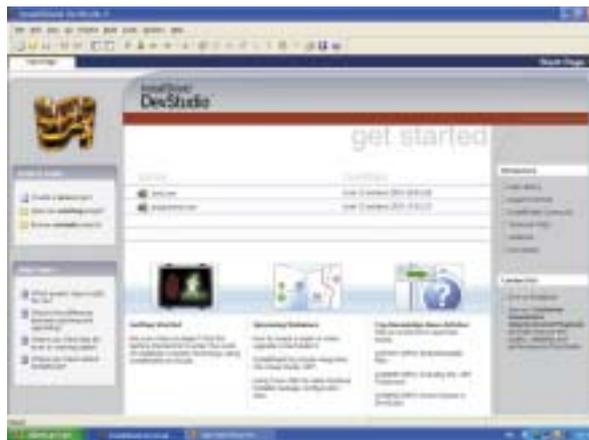
InstallShield sort une importante mise à jour de son atelier à fabriquer des installateurs Windows : InstallShield DevStudio 9 ! Outre la prise en compte des dernières technologies Windows, l'éditeur a modifié son produit dans le bon sens.

Dans le monde Windows, InstallShield demeure une des grandes références pour créer des installateurs simples ou complexes. DevStudio 9 contient de nombreuses améliorations et nouvelles fonctions. Le support de MSI a été amélioré, garantissant le respect des spécifications Microsoft et des services Windows Installer. Le langage script interne a, lui aussi, connu quelques modifications. Autres améliorations : le débogueur, le reporting de création, une meilleure prise en compte de Windows Server 2003, des terminaux mobiles sous Windows (PocketPC 2003 & Smartphone 2002), ou encore de l'intégration avec Visual Studio .NET. Parmi les nouveautés, les plus marquantes concernent le nouveau format des fichiers projets, qui est maintenant au format XML. Com+ est reconnu par DevStudio 9. Sans doute plus anecdotique pour la majorité des utilisateurs, le support du 64 bits. Ainsi, il devient possible de déployer

sur processeur et Windows 64 bits. On voit aussi l'apparition du " One-Click Install ". Cette technique doit pouvoir simplifier au maximum la phase d'installation vue par les utilisateurs. Il y a aussi l'apparition d'un nouvel assistant, pour créer et gérer un projet.

## Un IDE qu'il faut maîtriser

Le premier contact avec DevStudio 9 est plutôt agréable. La fenêtre de création d'un nouveau projet offre de nombreuses possibilités. Une fois créé le nouveau projet, on arrive dans l'environnement lui-même. Si la page de démarrage est d'une grande simplicité d'accès, on ne peut pas en dire autant de l'onglet Installation Designer. Heureusement pour rendre bien accessible DevStudio 9 on bénéficie désormais du Project Assistant. Ce dernier centralise et simplifie la gestion du projet. On navigue dans les différentes étapes, via des icônes. On accède aux options de base du projet. Si on souhaite aller encore au-delà dans la personnalisation et la finesse des réglages, il faudra passer dans l'onglet Installer Designer. Le comportement de l'assistant est bon et évite de se perdre dans la multitude des options. L'ajout des fichiers sources est rapide. D'ailleurs, on peut tout de suite noter la bonne performance et réactivité de l'environnement d'InstallShield. Le seul véritable ralentissement constaté est durant l'ajout des fichiers sources, d'environ 60 Mo. On peut simultanément créer 3 types d'installation : stand alone, Internet et CD-Rom. On ne peut qu'apprécier la qualité de la fenêtre "information" durant la compilation d'un projet. Le log permet de visualiser immédiatement le moindre problème. Bien entendu, on peut vérifier s'il y a des dépendances dans les fichiers sources. On peut aussi définir les bibliothèques redistribuables. À l'instar d'un vrai IDE de codage, il est possible de tester en direct son installateur, via la fonction



Run. De plus, on dispose d'un puissant mode debug. Très pratique quand on crée un installateur complexe. On a un accès à l'ensemble de l'interface utilisateur du futur installateur. Cependant, les modifications via le "RAD" sont limitées. Les thèmes proposés en standard manquent d'originalité. DevStudio 9 est un outil impressionnant pour ceux qui font uniquement des installateurs Windows. Dans le cas contraire, un outil plus universel sera nécessaire. Dommage que l'on soit parfois obligé d'utiliser des outils externes à l'éditeur. La compression des données est de bonne qualité. DevStudio 9 se destine avant tout aux gros projets, dans bien des cas, on pourra utiliser une solution plus légère de l'éditeur : Jexpress.

■ F.T.

## Fiche technique

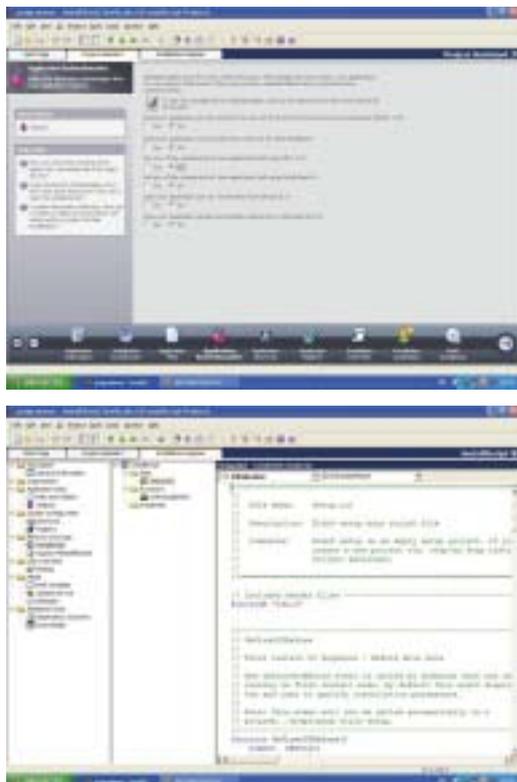
Nom du produit : InstallShield DevStudio 9  
 Editeur : InstallShield  
 Configuration minimale : PIII, 256 mo ram, 1 Go de disque dur, WinNT SP6, Win98, ME, XP ou Windows Server 2003  
 Prix indicatif : 1 199,00 € H.T.

### Points forts :

Project Assistant, Terminaux mobiles, Debug, Meilleur support MSI, XML

### Points faibles :

Complexité du designer, Personnalisation interface, Uniquement Windows



# Multi plate-forme : terrain (toujours) miné

Il y a des notions d'informatique qui font parfois peur aux responsables. Le multi plate-forme est longtemps resté une bête noire des développeurs. Mais avec l'arrivée de nouveaux outils et technologies (Java, Internet, XML, Web Services, EAI, JDBC, etc.), le multi plate-forme tend à se simplifier, ou tout du moins à devenir moins contraignant dans certains projets. Mais, ne vous réjouissez pas tout de suite, le multi plate-forme recouvre toujours une multitude de réalités.



Cela va d'un langage portable, à la prise en compte des différents terminaux, en passant par les bibliothèques d'interface, les IDE, les compilateurs, l'intégration, les tests ou encore les Web Services. Le multi plate-forme fait partie du quotidien de l'entreprise et du développeur. Dans certains cas, il est possible de s'affranchir largement de cette problématique, notamment dans le cadre du Web qui offre une couche d'abstraction suffisante. Même chose pour le Web Services, surtout depuis la disponibilité du WS-I Basic Profile 1.0.

## Multi plate-forme : c'est (presque) pas sorcier

Comme toute chose en informatique, le cross platform ne s'invente pas. Selon votre projet, vous utiliserez telle ou telle solution. Aujourd'hui, qu'on aime ou non, le langage de référence demeure Java. Cependant, si la portabilité du code est une réalité (si, si, je vous l'assure), mieux vaut avoir une chaîne homogène. Cela signifie qu'il faut que chaque plate-forme visée ait (de préférence) une JDK iden-

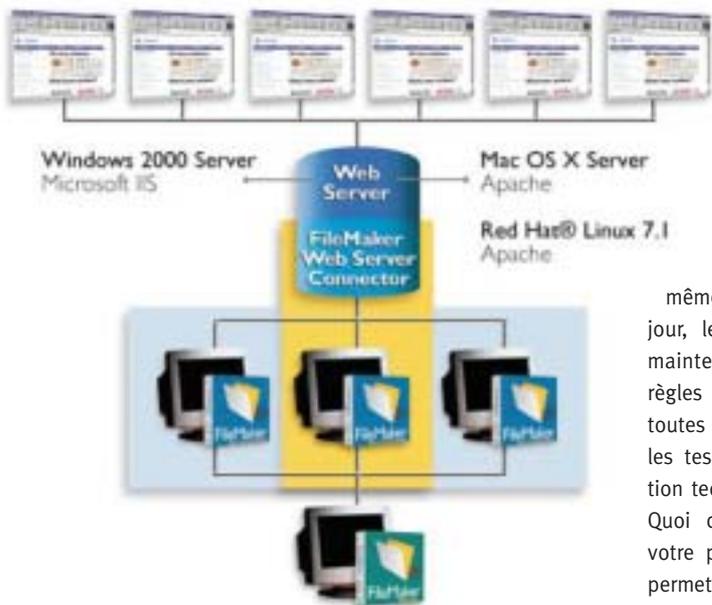
tique (dans la version et surtout l'éditeur). Car, malheureusement, selon l'implémentation, la JDK peut réagir différemment, idem selon le système. Si vous êtes obligé d'utiliser des JDK

de provenances différentes, pensez impérativement aux tests. Si vous êtes en intra ou extranet, essayez ici aussi d'uniformiser le tout. Un autre avantage est la simplification de la main-

## Les "basic like" bougent encore !

Tous les développeurs MacOS connaissent sans aucun doute, au moins de nom, deux IDE capables de créer à partir d'un RAD, une application MacOS X, Windows, voire Linux, RealBasic et Revolution. Sont-ils viables ?

RealBasic est un véritable IDE au sens habituel du terme : un éditeur, un RAD, un debug. Il permet de construire des applications tout à fait standard. Revolution pour sa part à une logique toute autre. Les conceptions ont repris le principe de la pile et des cartes proposées il y a presque dix ans par Apple, avec le mythique HyperCard. Si cette conceptualisation de l'application n'a rien de choquant, il faut tout de même s'y habituer, et ce n'est plus du tout la logique actuelle. RealBasic est donc un avantage sur ce point. Pour le RAD, RealBasic est bien plus complet dans le choix des objets d'interfaces et les options. La complétion est plutôt efficace dans RealBasic. Dans Revolution, la complétion n'existe pas, mais on bénéficie d'une aide textuelle statique. Revolution convient aux petits besoins alors que RealBasic peut supporter des applications de moyenne importance. Si les besoins ne sont pas considérables, c'est un IDE intéressant et plutôt riche au niveau fonctionnel. La compilation est assez rapide.



tenance... Il n'y a pas de petites économies. Si vous optez pour Java et que vous souhaitez "porter" le code sur chaque plate-forme, utilisez le même IDE partout. Eclipse constitue le premier choix, car disponible sur de nombreux systèmes. Évitez absolument d'utiliser plusieurs IDE Java (idem pour les autres langages hormis PHP ou Python). La compatibilité n'est pas toujours au rendez-vous, loin s'en faut. Souvent, on part d'une application A tournant sur un système Z, pour réaliser un portage sur un système Y. Si c'est du Java, en théorie, pas trop de souci. Si vous utilisez Delphi, vous pourrez utiliser Kylix et inversement. Dans le cas de C++, l'affaire se corse, tout comme en C, Pascal, VB, Basic et autres langages subtils... Si votre application de départ est du VB et qu'elle doit pouvoir fonctionner sur Linux ou OS X, une bonne réécriture s'impose (même si dans certains cas on peut utiliser des IDE reprenant une partie du code VB). Cela pourrait être une bonne occasion pour uniformiser le langage et remplacer ainsi VB sous Windows. Tant faire ce peut, il faut que le code soit le moins spécifique possible à chaque

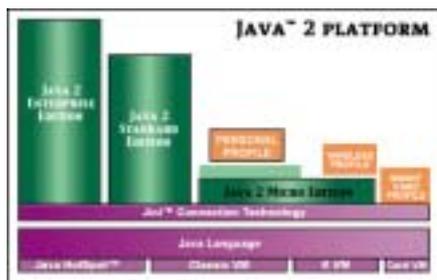


plate-forme. Pourquoi ? Plus le code de chaque système cible est différent et spécifique, plus sa maintenance sera lourde et d'une portabilité limitée. Avoir un maximum de code générique (ou en tout cas, construit de la

même manière) facilite les mises à jour, les modifications et donc sa maintenance. Il faut définir des règles de maintenance unique pour toutes les plates-formes. Idem pour les tests, idem pour la documentation technique, etc.

Quoi qu'il en soit. Conceptualisez votre projet multi plate-forme. Cela permettra d'établir les listes des outils nécessaires, les contraintes,

les objectifs. Chaque contrainte peut restreindre le choix des outils. La couche de données sera la moins problématique, même si pour les terminaux mobiles, une déclinaison spéciale peut être requise. La couche d'accès peut, par contre, se modifier selon les choix des IDE et des langages. Dans un projet multi plate-forme, on peut mettre en place un outil de type CVS pour centraliser les sources et les versions.

### Opter pour un IDE tout-en-un ?

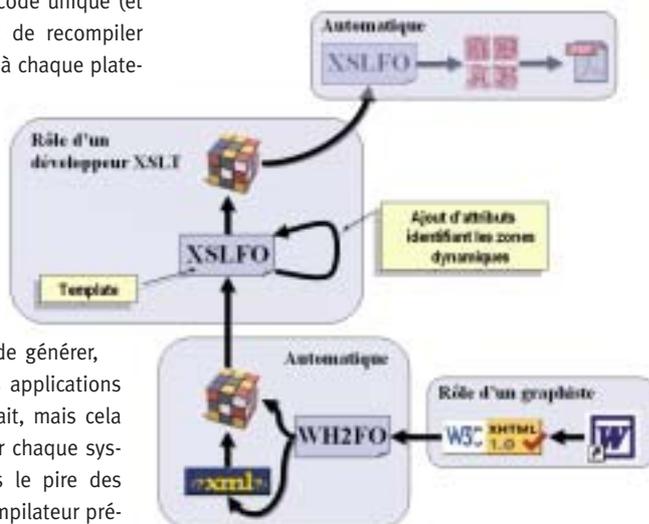
L'idéal serait de disposer d'un code unique (et strictement unique) qu'il suffit de recompilier avec un compilateur spécifique à chaque plate-forme. Des IDE permettent cela. Mais en général, on hésite à les utiliser pour de gros projets. Ce type d'outils rend des services, quand on souhaite rapidement mettre en place des applications natives. Il existe un assez grand nombre d'IDE capables de générer, à partir d'un code unique, des applications multi cibles. Delphi / Kylix le fait, mais cela nécessite une recompilation sur chaque système (Windows / Linux). Dans le pire des cas, vous pouvez utiliser un compilateur présent sur différentes plates-formes. Le plus répandu est GCC. Par contre, cela nécessitera une adaptation du code. CodeWarrior de Metrowerks propose du cross platform en C / C++, mais là encore, impossible de le faire à partir d'un code unique.

### Le multi plate-forme c'est aussi du multi terminal

Le multi plate-forme cache, bien entendu, une multitude de réalités, la prise en compte dans une application (quelle qu'elle soit) de plusieurs types de terminaux, pose aussi quelques problèmes, dans le langage, les compilateurs, mais aussi (et surtout) au niveau de l'interface. Les contraintes sont radicalement différentes quand on parle d'interface PocketPC, TabletPC, Smartphone ou desktop. Et d'un point de vue encore plus global, dans le monde de l'embarqué.

Deux cas de figures peuvent alors survenir. En environnement homogène, les problèmes sont moindres. Ainsi, si vous êtes en tout Microsoft Windows, il est possible d'utiliser des technologies et des outils proches ou identiques. En .NET, la déclinaison mobile peut être utilisée, même si des contraintes existent. En Java aussi, on peut avoir une certaine uniformité entre J2SE et J2ME, mais là également, les contraintes de la version mobile sont à considérer. Dans tous les cas, il faudrait maintenir plusieurs codes sources si le projet fonctionne sur Desktop et des terminaux mobiles. Dans le cas de .NET utilisé sur Desktop et TabletPC, on peut envisager un code unique avec des tags d'activation pour les fonctions spécifiques.

Plus que jamais dans un contexte multi terminal, séparez strictement la couche fonctionnelle (bref toute la partie code pur) et la couche



de présentation (donc l'interface). Cette stricte séparation permet dans le cas de portage d'être plus souple. Car, dans certains cas, il faut refondre toute l'interface. Il est plus facile de le faire quand celle-ci est indépendante du

code fonctionnel. Il sera alors aisé de construire plusieurs interfaces : une pour chaque terminal et ensuite d'adapter si nécessaire le code. C'est d'autant plus vital dans les projets Web. Dans le monde de l'embarqué pur, on optera souvent pour du C/C++. Là, la librairie d'interface est donc essentielle. Il faut pouvoir porter une interface complexe et riche sur diverses plates-formes. Des librairies telles que ZooLib, WxWindows, GTK+ ou Qt permettent de tels développements. Dans des développements Windows - Linux ou OS X - Windows, les IDE prennent généralement en charge l'interface.

## Les bases de données

Pour faire un multi plate-forme rapidement, avec une gestion un peu complexe des données, pourquoi ne pas utiliser un SGBD couplé à un RAD pour créer une application ? Des solutions éprouvées existent depuis longtemps. 4D ou FileMaker sont les dignes représentants de cette catégorie. Vous créez l'application et la base de données sur une plate-forme, le portage se limitera bien souvent à une compilation croisée, voire une recompilation sur la plate-forme cible. Si vous avez deux parties distinctes (1 application, 1 base de données), il faudra prévoir une couche d'accès aux données, à partir de l'application. Le SGBD devra être capable de travailler dans une telle configuration. Définissez avec rigueur les besoins, les protocoles et le type de couche à mettre en place : du mapping, des requêtes, un simple fichier d'échange ? Le SGBD et l'application devront bien entendu savoir travailler les données entrantes et sortantes. Il faudra aussi penser aux problèmes de droits d'accès, des priorités ou encore de la synchronisation. Dans le cas d'utilisateur nomade et d'applications mobiles, la problématique est différente : accès en mode connecté à la base ? Comment synchroniser les données locales et celles du serveur ?

## Les Web Services sont-ils multi plates-formes ?

En théorie, oui. En pratique, il faudra attendre quelques mois. Pour ceux utilisant les solutions IBM ou Microsoft, cela ne cause guère de

souci. Côté Java, ce n'était pas forcément la joie. Des librairies tierces pouvaient être utilisées, mais rien de natif en standard. Il faudra attendre la sortie de la 1.4 de J2EE. La garantie du multi plate-forme d'un service web se fera dans l'avenir avec l'estampille : WS-I Basic Profile. C'est lui qui doit assurer l'interopérabilité entre tous les services conçus avec des outils l'implémentant. Il y avait aussi des soucis avec les premières versions de SOAP, la mouture 1.2 résout en grande partie cela. Quand on n'a pas besoin d'un composant lourd de type EJB, ou d'applications Web via un serveur d'applications, le Web Services peut rendre d'énormes services pour fédérer des applications et données hétérogènes. Il peut y avoir encore plus simple si on veut juste échanger des données : un fichier XML qu'il suffit de traiter. Le Web Services peut éviter de porter physiquement une application sur plusieurs plate-formes, à condition de modifier l'architecture applicative pour prendre en compte le service web.

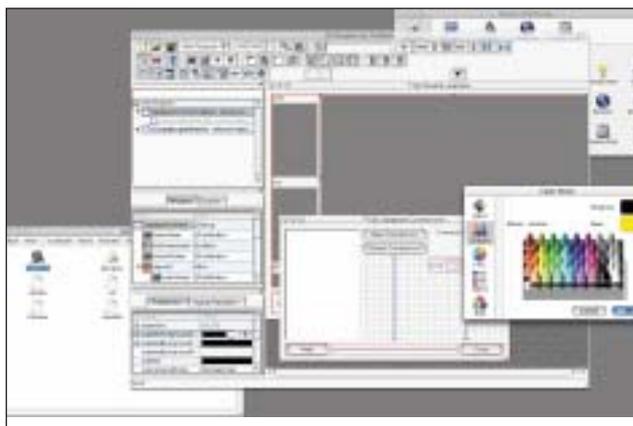
## Documents, formulaires : les nouveaux défis du multi plate-forme

Le multi plate-forme, on ne le répétera jamais assez, est multi visage. Un aspect nouveau, pour beaucoup, apparaît : la notion de formulaires et de documents. Comment une entreprise, peut elle communiquer un document ayant une forme et donc une présentation unique à toutes les plates-formes ? Même remarque pour le formulaire. Comment puis-je proposer un formulaire de saisie / consultation unique valable partout ? Avec la maturation technologique, on dispose d'outils et de techniques fiables (en principe) permettant cela, sans à avoir besoin de recourir à une plomberie énorme et ingérable.

Si on met de côté la technologie Macromedia, encore un peu jeune, Xform est la seule capable d'être véritablement universelle, car ne nécessitant aucun ajout logiciel et possédant une spécification W3C. De plus, Xform se base sur XML. Si InfoPath se base aussi sur XML et les schémas, il se limite pour le moment à Windows. Pour le mettre sur d'autres plates-formes ou sur le Web, il faut recourir à pas mal de manipulations en utilisant le fichier

XML et/ou ASP.NET. Mais si on souhaite un formulaire dynamique riche unique partout, la technologie PDF est ce qu'il y a de mieux actuellement. Surtout, il est possible de générer des documents PDF avec à partir de plusieurs sources et langages (ex. : .NET ou Java). Cela est très pratique quand on a besoin de générer à la volée des documents, des formulaires. Bien entendu, l'inconvénient du PDF est son poids relatif, mais sa qualité de présentation est indéniable. Pour générer du PDF à la volée, on peut utiliser XSLFO en environnement .NET, FOP en Java. On peut aussi le faire, à partir de PHP via la classe FPDF (gratuite) ou encore avec ReportLab pour Python. Choix intéressant, quand on a un site PHP pour créer un dossier ou un rapport. Cependant, attention, les librairies et API pour générer à la volée du PDF nécessitent de mettre en place un minimum de code rien que pour créer le document, les champs, récupérer les données, etc. Une relative maîtrise du PDF (au niveau format) est un plus, quand on souhaite créer des documents lourds et complexes.

■ François Tonic



# Effets spéciaux et multi plate-forme

**D**uboi, internationalement reconnu dans le monde des effets spéciaux, à plusieurs cordes à son arc : post-synchronisation, son - entrée et sortie, étalonnage des couleurs en temps réel. Son outil SFX, Dutruc est un environnement composite qui est en quelque sorte le Photoshop des images cinématographiques. Plus de 100 films, dont Alien



la Résurrection, Asterix & Obelix I et II et la Neuvième Porte, ont bénéficié de la magie numérique de Dutruc. En l'occurrence, l'un des derniers films sur lequel Duboi a travaillé, Le Fabuleux Destin d'Amélie Poulain, est un succès mondial. La problématique de départ était simple. En avril 2000, le chef de projet Dutruc chez Duboi se posait la question suivante : comment porter l'environnement d'effets spéciaux maison, du système Irix au système Windows NT. L'application pesait tout de même presque 800 000 lignes de code, sans compter les technologies liées. " Je me suis senti comme la souris dans Tex Avery — mes pieds sont collés au sol et je regarde en l'air et vois un piano tombant du ciel qui arrive vers moi " explique Olivier Tubach, le chef de projet Dutruc.

## Deux raisons : le temps et encore le temps

Car à l'époque, tous les outils de Duboi fonctionnaient uniquement sur IRIX, une situation qui devenait intenable. Mais comment réaliser un portage aussi complexe vers de nouvelles plates-formes ? L'équipe de Duboi avait donc besoin d'une boîte à outils multi plate-forme, capable de fonctionner avec OpenGL et Windows NT, et qui soit de plus, Open Source. Deux solutions étaient alors envisageables :

WxWindows et Qt de Trolltech. Les deux librairies furent testées en interne durant une quinzaine de jours. Le choix se porta sur Qt. Deux raisons poussèrent Duboi à prendre le produit Trolltech : le temps et le temps ... L'impératif du temps était cruciale. Duboi devait absolument respecter des délais très courts, et cela obligeait les développeurs à être immédiatement opérationnels pour réaliser le portage. Qt apparut la solution la plus productive. Outre l'aspect gain de temps, trois facteurs furent en faveur de Qt : une excellente documentation, un accès au code source et un support auprès de l'éditeur. Ces trois éléments fournissaient aux développeurs de Duboi des avantages certains. Le support était important. Car en cas de problème, il fallait une réponse rapide, dans les heures qui suivaient la question. L'historicité de Qt, la qualité des produits Trolltech ont fait le reste. Olivier Tubach avoue avoir apprécié le fait que Qt soit à la base de KDE, ainsi que la robustesse de la librairie. " Nous avons besoin que Dutruc soit performant pendant de longues heures d'affilée. Les artistes l'ouvrent tôt le matin et ne le ferment qu'à leur départ. Il doit donc être très stable et ne doit jamais tomber en panne ! (Je croise les doigts !)." Qt répondait facilement à ces besoins, malgré les exigences posées par Dutruc. " Une session Dutruc implique la création et la destruction de milliers de dialogues " souligne Olivier Tubach, " Qt doit donc être très robuste. "

## Accros !

Aucun des quatre développeurs ne connaissait Qt. Cette lacune ne fut pas un réel obstacle. Ils ont rapidement appris à le maîtriser, lui et ses subtilités : "Qt est si simple à utiliser que nous avons passé notre temps à travailler et non à batailler contre l'interface graphique" poursuit Olivier Tubach. "L'interface API est également cohérente. Nous n'étions pas obligés de nous noyer dans la documentation. Les gadgets logiciels ont fait leur travail."

Comme Motif, initialement utilisé, comportait peu de gadgets logiciels complexes, l'ancien Dutruc ne contenait pas de commandes basiques. En raison des nombreux raccourcis de Qt, il a été facile d'ajouter des commandes sophistiquées, comme des infobulles dyna-

miques et un affichage en arborescence au cours du portage. "Avec Motif, il est difficile de programmer toutes ces fonctions, tandis qu'avec Qt, ce type d'amélioration a été très simple à réaliser" précise Tubach. L'équipe devait bien sûr ré-écrire l'interface graphique, mais une fois ce travail effectué, le portage pouvait être effectué pour n'importe quelle plate-forme nécessaire par simple recompilation. "Avec Qt, nous sommes libres. Nous pouvons facilement nous rendre là où nous devons aller."

## Flexibilité multi plate-forme

Une fois le portage de Qt terminé, le client était ravi car l'application pouvait fonctionner sur des plates-formes qu'il possédait déjà. L'arbre source unique Qt est maintenant utilisé chaque jour sur les plates-formes NT du client, en interne sur Linux, IRIX, et une récente compilation pour Mac OS X a également été effectuée.

"Qt résout un grand nombre de nos problèmes complexes en étant multi plate-forme " se réjouit Olivier Tubach. " Nous sommes désormais en mesure de dire à nos clients 'Dutruc fonctionnera de la même façon sur Mac, NT, Irix, Linux, etc.'" Les programmeurs Dutruc déclarent qu'ils ont reçu une assistance de la part de Trolltech, qui dépasse de loin l'assistance fournie habituellement. "Ils ont même ajouté des fonctions à l'interface API Qt, uniquement parce que nous en avons besoin" explique O. Tubach. Son équipe et lui n'ont eu qu'une petite plainte à émettre sur Qt, et lorsqu'ils s'en sont ouverts à Trolltech, il leur a été répondu que le fixage était en projet. "le seul point négatif que j'ai pu trouver sur Qt était déjà en cours de règlement. :) " Après son expérience avec ce portage, Duboi a décidé de baser tous ses futurs développements sur Qt. En tout et pour tout, le portage de Dutruc et l'implémentation de Qt ont pris à peine cinq mois. "Nous avons commencé à apprendre à utiliser Qt en avril et nous avons fini le portage en septembre" a-t-il précisé. "Et au cours du portage, nous avons pris deux semaines de vacances ! Vous imaginez ça ? Nous sommes accros à Qt maintenant ! Nous ne voulons plus entendre parler de Motif, de MFC, ou de toute autre solution."

JEAN CHRISTOPHE CIMETIÈRE (MICROSOFT FRANCE) :

# “Le multi plate-forme chez Microsoft est une vision un peu particulière”



**Comment Microsoft intègre-t-il le concept de multi plate-forme dans son offre ? Est-il identique à la définition que l'on en donne habituellement ?** Programmez ! s'est entretenu avec Jean-Christophe Cimetière (jccim@microsoft.com), chef de projet plate-forme d'entreprise chez Microsoft France.

**Programmez ! : N'est-il pas surprenant de constater une sorte de raréfaction des outils multi plates-formes, capables de générer une application multi cible à partir d'un code unique ? Quelles sont les causes d'une telle tendance de la part des éditeurs ?**

**Jean-Christophe Cimetière :** Il y a 5 ou 6 ans, il y avait effectivement de nombreux outils dédiés au Client / Serveur et au multi plate-forme, tel qu'Uniface ou Forté avant son rachat par Sun. Des outils plus ou moins performants. La problématique du développement multi plate-forme sur les postes clients était commune à beaucoup d'entreprises. Mais l'Internet est arrivé, amenant une réelle indépendance vis à vis du système d'exploitation grâce au HTML. Le Web a donc supplanté le multi plate-forme. Aujourd'hui, les entreprises hésitent à reprendre des développements d'applications riches, pour le poste client.

**P ! : Quelle est la vision de Microsoft ?**

**JCC :** Microsoft fait du multi plate-forme autour de Windows, du PDA au data center. À ce jour, Microsoft décline deux noyaux de Windows, Windows CE (décliné en PocketPC et SmartPhone) et Windows XP & Windows Server 2003. Au-dessus de l'OS, nous avons maintenant le Framework.NET qui propose un modèle de programmation unifiée, facilitant le travail du développeur. Le multi plate-forme chez Microsoft est une vision différente de nos concurrents.

**P ! : Justement, en parlant de .NET, est-il possible de faire un développement multi plateforme avec le langage C#, même si Microsoft ne propose pas d'outils spécifiques ?**

**JCC :** Oui c'est possible mais avec des restrictions. Tout d'abord, il faut savoir que Microsoft à sou-

mis à la standardisation certaines parties du Framework .NET: la CLI et le langage C#. Cela permet d'envisager des portages. Microsoft l'a fait, via le projet Rotor tournant sous MacOS X et BSD. Cela démontre la faisabilité d'un tel travail. Un des objectifs est éducatif. Il permet d'offrir aux étudiants et écoles la possibilité de découvrir les technologies Microsoft. Dans Rotor, on dispose des codes sources. En parallèle, la communauté Open Source a créé le projet Mono. Pour porter le Framework .NET, il "suffit" de ré-implémenter les API, le moteur d'exécution, et un compilateur respectant le standard. Mono est un projet assez avancé mais à l'heure actuelle, cette implémentation n'est pas totale.

**P ! : Au-delà de Mono, allez-vous sortir des outils purement multi plates-formes ? Allez-vous supporter d'autres projets, de type Mono, pour sortir C# et .NET de leur environnement Windows ?**

**JCC :** Tout d'abord, il faut savoir que nous avons de bonnes relations avec les auteurs du projet Mono, sans pour autant supporter, ni encourager cette initiative. On a un modèle multi plate-forme dans le sens Windows. Il ne faut pas oublier que le Framework .NET est disponible en téléchargement, gratuitement. C'est un des composants qui contribue à donner sa valeur à Windows. Mais un projet tel que Mono montre, qu'il est possible de réaliser des portages. On ne peut pas recommander à un développeur de programmer en C# sous Windows pour ensuite déployer sur un Linux. Mono est encore jeune et a un avenir incertain.

**P ! : Est-ce à dire que développer sur C# avec Linux ou Mac OS X n'est pas faisable ?**

**JCC :** Techniquement c'est possible, on peut choi-

sir de coder en C# sous Linux, en tout cas ce n'est pas supporté par Microsoft.

**P ! : J'ai une application VB. Je souhaite que celle-ci puisse fonctionner sur d'autres systèmes, comment puis-je faire ?**

**JCC :** Si on veut faire cela, on va plutôt s'orienter vers du C / C++. Microsoft ne fournit pas ce type d'outils pour du multi plate-forme "classique". Une alternative est d'utiliser une application Web avec ASP.NET.

**P ! : XML et les Web Services ne seraient ils pas l'un des futurs du multi plate-forme ?**

**JCC :** Les Services Web ont atteint un premier niveau de maturité. Il existe un consensus autour de ce standard. Les Services Web font abstraction de l'interface utilisateur. On peut échanger des messages XML en totale transparence sur différentes plates-formes. Sur le client final, on peut alors choisir l'interface la mieux adaptée. Pour une interface riche, on pourra utiliser des WinForms, pour quelque chose de plus léger, du simple HTML.

**P ! : Un des problèmes, qui est de porter une application quelle qu'elle soit, ne vient-il pas de la non séparation stricte de l'interface et du code fonctionnel ?**

**JCC :** Oui. Il faut tendre vers ce modèle d'architecture orientée service. On n'a pas assez mis l'accent sur la séparation de l'interface, des objets métiers et de l'accès aux données. On allait à la facilité, d'autant que les anciens outils de développement, y compris les outils de Microsoft n'encourageaient pas à maintenir cette séparation. Finalement, l'architecture multi couche de l'application est une notion prise en compte relativement récemment. Cette prise de conscience se traduit notamment par l'importance croissante du métier d'architecte. Mais réfléchir sur l'architecture en couche de son application est une chose, disposer des outils pour l'implémenter naturellement en est une autre. Pour certains développeurs, c'est un choc culturel et cela a un coût au niveau des phases de conception.

■ Propos recueillis par François Tonic

# EPITECH : l'apprentissage par projet



Créée il y a quatre ans, l'EPITECH a vu fleurir sa première promotion en 2002. Petite sœur de l'EPITA, elle propose un enseignement qui repose essentiellement sur la connaissance des systèmes et des réseaux et de leur mise en application lors de missions au sein d'entreprises. Elle propose un cursus à pédagogie inductive, fortement axé sur l'apprentissage par projet.



L'enseignement dispensé à l'Epitech sort clairement des sentiers battus, ce qui explique la spécificité des profils retenus à l'entrée de l'école. "Nous ne retenons pas les candidats trop scolaires mais plutôt les autodidactes, autonomes et passionnés d'informatique", précise Nicolas Sadirac, directeur de l'Epitech. Les étudiants ne sont pas pour autant tous des petits génies de la débrouille informatique. "En première année, la moitié seulement des étudiants savent déjà programmer et moins de 10% programment bien, précise-t-il. Quasiment aucun des étudiants qui arrivent ne savent programmer en groupe." Hormis l'anglais, il n'y a pas d'évaluation sur les connaissances, mais uniquement sur leur mise en pratique. Une des spécificités de l'école est qu'il n'y a pas de cycle prépa ou d'examens "sur table". Réparti sur 5 années d'études, le cursus d'Epitech se découpe en deux cycles de 3 et 2 ans. Fondé sur la pratique intensive de l'informatique, il est ponctué de nombreux projets personnels. La sélection est vive et la notation objective sans appel : la mission de l'élève est avant tout de rendre des projets qui fonctionnent.

## Un système d'enseignement basé sur la capacité à innover

Chaque année, l'étudiant devra valider une vingtaine de projets. Durant les premières années, les projets sont plutôt orientés programmation Unix. "Un des premiers projets consiste en la réalisation d'un Shell Unix, se souvient Eyal Dotan, ancien et enseignant à

l'Epitech. Puis des applications réseau telles qu'un client / serveur FTP et IRC et un jeu client / serveur, nommé "Zappy", dont un serveur TCP/IP gérant plusieurs machines clientes de jeu, le tout en langage C, évidemment." Si comme dans la plupart des écoles, l'étudiant apprend à utiliser des outils de haut niveau tels que .Net, Java, Delphi, à l'Epitech, il doit comprendre le fonctionnement de base de tous ces outils. "On ne vous laisse pas utiliser la commande "printf" avant de savoir la coder. De même, on vous demande de faire une machine virtuelle Java avant de débiter dans ce langage."

"En France, l'apprentissage reste très normatif et ne met pas souvent l'accent sur la capacité d'innovation", déplore Nicolas Sadirac. A l'Epitech, les professeurs interviennent plus comme "réparateurs" qu'"apporteurs" de connaissance. L'idée est également de mixer les matières. Sur un projet, l'étudiant sera ainsi confronté aussi bien à des notions de mécanique, de mathématique que de stratégie." La sélection au fil des ans est importante : 20% des étudiants sont éliminés la première année, 13 à 14% la deuxième et à nouveau 20% la troisième année. A la sortie, il y aura eu plus de 50% d'éliminés. Au final, les étudiants auront le titre d'experts informatique et deviendront chefs de projet ou administrateurs système et sécurité. L'investissement de l'étudiant est conséquent avec parfois 60 à 70

heures de travail sur les projets. L'école est ouverte 24h/24 et 7j/7. Certains étudiants vont même jusqu'à habiter sur place et les deux douches de l'école sont très régulièrement fréquentées !

■ Anne-Françoise Moal

## Carte d'identité de l'Epitech

Date de création : 1999  
 1ère promotion : 2002  
 Nombre d'étudiants en formation : 1.200  
 Cursus : En 5 ans  
 Admission : Accessible après le bac et en 2ème année et 3ème année sur admission parallèle après DEUG, DUT et BTS.  
 Cycle : Le cursus prévoit des certifications professionnelles (Cisco, Sun, Microsoft, Checkpoint...) en 3ème et 4ème années.  
 Finalité : Obtenir une compétence d'ingénieur informaticien très complète, particulièrement sur les nouvelles technologies et Internet, avec une composante opérationnelle forte et un spectre de connaissances très large.  
 Nombre d'heures de cours : Entre 15h00 et 25H00, regroupées sur 3 jours  
 Prix de la scolarité : Entre 5 400 euros et 6 000 euros selon le niveau d'années  
 Pour plus d'information : [www.epitech.net](http://www.epitech.net)



## Nicolas Sadirac : "apprendre à apprendre"

*Nicolas Sadirac, directeur de l'EPITECH : "Même s'il y a des bases incontournables, l'informaticien ne doit pas être un puits de connaissance mais avant tout quelqu'un capable d'apprendre, de savoir-faire"*

Après une formation en physique aux Etats-Unis, Nicolas Sadirac poursuit sa scolarité à l'EPITA en 1986. En 1987, il est embauché en fin de tronc commun par l'école, comme assistant de l'administrateur système. En 1989, il devient lui-même administrateur système pour l'EPITA. Passionné de sécurité, il aura l'occasion de créer différentes sociétés dans ce secteur. Aujourd'hui, à 35 ans, il est directeur technique du groupe et directeur de l'EPITECH dont il a participé à la création en 1999. " L'objectif de notre enseignement est de former des gens qui ne dépendent ni des modes, ni des technologies, explique-t-il. L'important est de raisonner en termes d'objectifs à atteindre, grâce à la mise en place de points de validation qui sont eux-mêmes à inventer. Notre enseignement se base plus sur une capacité à analyser les problématiques que sur la maîtrise de connaissances théoriques. " Les meilleurs éléments sont intégrés dans les laboratoires comme 3IE, LES, LERIA qui s'autofinancent et ont des missions d'audit ou de développement. Ils peuvent également devenir assistants des professeurs, être intégrés dans les activités de recherche. " L'une des grandes difficultés sans doute à surmonter est la jeunesse des étudiants : ils sont souvent forts techniquement, mais ont besoin d'être structurés, estime-t-il. Un des gros dangers est d'être performant sur de petites choses, mais ne pas être capable d'assurer un service fini et continu. " Passionné d'informatique et d'enseignement, Nicolas Sadirac ne perd pas de vue pour autant les contraintes d'un marché toujours plus exigeant.

### Les projets

L'une des grandes forces d'Epitech est le nombre très important de projets informatiques intégrés au cursus. Ils se répartissent en 3 catégories :

- Les mini-projets, à réaliser individuellement

- ou par binôme, dans un délai moyen de une à deux semaines. Ils ont pour but de mettre directement en application les nouvelles connaissances acquises en cours et aux séances de travaux pratiques.

- Les projets, dont les périodes de développement, sont comprises entre 2 et 8 mois, avec parfois des séances de suivi au cours de cette période. Ce sont des projets très complets et relativement complexes, à réaliser par groupe de 4 à 10, imposant ainsi une répartition des tâches. Ils permettent à l'étudiant d'acquérir

une très grande expérience de développement au sein d'une équipe.

- Les projets personnels dont le temps de réalisation varie de 1 à 3 ans. Ils peuvent être réalisés en collaboration avec une entreprise.

L'outil informatique de l'EPITECH répond à des objectifs techniques et pédagogiques ambitieux. Il ne trouve d'équivalent qu'au sein de très grosses entreprises. C'est donc en grandeur réelle que chacun développe et teste ses projets. Mieux, l'ensemble du système fait partie intégrante de la vie quotidienne de l'école.

### Portrait d'un ancien de l'Epitech

## La passion n'attend pas le nombre des années

Eyal Dotan a commencé à pianoter sur un ordinateur à l'âge de ... 4-5 ans. A 8 ans, il faisait ses premiers pas dans la programmation et à 16 ans, ce virtuose de l'informatique développe un logiciel de protection contre les virus : Viguard, particulièrement innovant, puisqu'il ne se base pas sur une connaissance préalable des virus. En 1998, le logiciel de ce lycéen est commercialisé par la société Tegam qui le diffuse à ce jour en France et dans tous les pays francophones. Après deux ans d'études aux Etats-Unis, Eyal Dotan rejoint en 1999 l'Epitech pour un cursus de trois ans. Il explique son choix par l'aspect pratique de la formation proposée. " On y apprend la programmation par la pratique, et non par la théorie, précise-t-il. Ayant débuté dans la programmation de façon autodidacte, cette formation me correspondait parfaitement. Il n'y a pas de contrôle ni d'examens " crayon et papier " car l'évaluation se base sur ce que l'étudiant est capable d'accomplir. Le travail s'effectue en groupe dans l'objectif de réaliser un projet, à l'identique d'une situation en entreprise. De plus, l'apprentissage se concentre sur l'informatique dès le départ, il n'y a pas de cycle prépa ; cela permet d'entrer directement dans le vif du sujet : la programmation. Personnellement, je considère que l'Epitech est certainement l'une des meilleures écoles européennes dans le domaine de la programmation système et de la sécurité informatique. "

Depuis 2002, il poursuit ses recherches dans la sécurité et enseigne au sein de l'Epitech, la sécurité et la programmation sous Windows. "C'est un domaine sur lequel peu d'écoles offrent des formations, et qu'il est pourtant de plus en plus indispensable d'appréhender, estime-t-il. Ma manière d'enseigner est typique de l'Epitech : quelques cours suivis de projets et de soutenances qui valident le savoir acquis durant l'année."



*Eyal Dotan, enseignant à l'EPITECH : " En informatique, ce ne sont pas tant les connaissances qui comptent, que la façon d'apprendre et de s'adapter aux nouvelles technologies. "*

# 3IE, une pépinière de talents

Créé en 1999, l'Institut d'Innovation Informatique pour l'Entreprise (3I.E.) est le département Nouvelles Technologies de l'Information et de la Communication de l'EPITA (Ecole Pour l'Informatique et les Techniques Avancées).



Véritable fer de lance des NTIC de l'Epita, 3IE est un partenaire technologique proche de la SSII qui sort des sentiers battus. Association loi 1901, 3IE dispose de tous les pôles de compétences d'une société de services informatiques.

"En créant 3IE, l'idée était de proposer une structure capable de mettre en place une veille technologique pour détecter les nouvelles technologies, les tester, les confronter au marché et si besoin d'intégrer leur apprentissage au cursus proposé par l'EPITA, explique Yannick Lejeune, directeur général de 3IE. L'école fait en effet appel à 3IE pour l'intégration de nouveaux modules de cours, ou encore demande à ses membres de dispenser certains cours." Initialement prévue pour analyser le marché des NTIC et en intégrer les résultats au cursus de l'EPITA pour permettre à l'école de rester concurrentielle, la petite cellule de veille a vite développé son champ d'ac-

tion. La veille stratégique interne s'est transformée en service vendu à des sociétés extérieures qui ont souhaité conseil et développement, afin de mettre en place des technologies que le marché ne connaissait pas encore. C'est donc tout naturellement que 3IE s'est pourvu d'un pôle Recherche & Développement, ainsi que d'un pôle Consulting et Formation.

3IE compte 50 membres, pour moitié des permanents salariés en CDI, et pour moitié des étudiants de l'EPITA. Ces derniers sont sélectionnés sur les résultats obtenus au cours de leurs études, au minimum bac+3, mais aussi sur leur connaissance des technologies existantes, leur motivation et leur ouverture d'esprit. Lorsqu'ils commencent à travailler au sein de 3IE, ils suivent un cursus hors norme. Ils continuent à aller en cours et à passer leurs examens comme les autres et effectuent, en plus de leurs projets scolaires, un véritable travail de consultants, développeurs, programmeurs, ou chefs de projet pour les clients de 3IE. Ils sont rémunérés au forfait. "Parmi tous les étudiants travaillant au sein de 3IE, nous en sélectionnons environ 5 chaque année pour lesquels nous prendrons en charge les frais de scolarité, précise Yannick Lejeune. En contrepartie, ils s'engagent à former la promotion suivante." Ils sont coachés par un encadrement spécialisé dans les diffé-

rents domaines de compétence de 3IE : Informatique, Veille, Knowledge Management, Business Development, Graphisme, etc.

## Quand passion rime avec performance

Nicolas Sorel, ingénieur développeur, travaille depuis mi-juillet pour 3IE. Cet autodidacte, qui a créé différentes structures spécialisées dans les prestations dédiées aux SSII, apprécie spécialement l'ambiance start-up qui règne au sein de 3IE. "Pour les étudiants de l'EPITA, travailler au sein de 3IE est une formidable opportunité dont le moteur est indubitablement une motivation profonde et une très grande disponibilité. Il n'y a en effet pas d'aménagement spécifique du temps de travail pour ces étudiants. Si ce travail à mi-temps peut être un bon moyen pour financer ses études, ce n'est pas la première des motivations qui anime les étudiants qui ont fait ce choix et ont été sélectionnés."

Les étudiants sont très conscients de cette valeur ajoutée. Ils apprennent très tôt à rédiger un cahier des charges technique et fonctionnel, dans l'objectif d'un usage précis et d'un retour sur investissement. Pour les étudiants c'est un formidable moyen de comprendre le business model de leur client.

■ Anne-Françoise Moal

## Carte d'identité de 3IE

Date de création : 1er octobre 1999

Statut : département de l'Epita

Compétence :

- Recherche et Développement NTIC
- Consulting, développement et expertise
- Veille Stratégique
- Service aux entreprises
- Formation

Effectif : 50 personnes, dont une vingtaine d'étudiants travaillant à temps partiel

Moyens : Ouvert 24h/24 et 7jrs/7. 30 serveurs répartis sur plusieurs salles blanches, liaison spécialisée 20Mo, stations de travail renouvelées tous les 2 ans environ.

Les membres :

Yannick Lejeune, directeur général  
Alexandre Pinhel, directeur technique  
Vincent Guigui, directeur assistant

Le reste de l'équipe est composé d'un responsable administratif, d'une équipe commerciale, de chefs de projets, de chargés de veille, d'une vingtaine d'étudiants et d'autant de salariés travaillant à temps plein.

CA : 1 million d'euros pour 3IE, le chiffre d'affaire de l'EPITA étant de 15 millions d'euros en 2002

## Des références

### Recherche

- 5ème programme cadre européen : Développement d'un chatterbot (agent conversationnel) avec les sociétés Cybion, Innovia et la Cantoche
- X2PMS : conception d'un outil de gestion de projet basé sur eXtreme Programming (XP)

### Développements applicatifs

- BNP Paribas : application CRM
- Cybion : Conception d'un outil de gestion d'annuaires, collaboratif
- Schumann Bourse : Outil d'automatisation de la mise à jour des indices boursiers
- Netpartnering.com : Création de la première version de l'intranet international

### Veille technologique et conseil

- Piren Conseils : Etude des performances de serveurs d'applications J2EE

- 24PM : Pré-conseil technologique pour la création des systèmes d'affiliation
- Educplanet.com : Etudes de marché sur le e-learning et ses acteurs

### Formation

- Auditeurs et Conseils associés : Formation NTIC (téléphonie, PDA, ASP) pour une délégation d'entrepreneurs affiliés au MEDEF du CHER.
- CFI : Formations XML, Perl, C++, Delphi, Javascript, Java, JSP, Lotus Domino/Notes, Linux

### Systèmes et Réseaux

- Palm : tests de plate-forme
- Drianlexa : Audits de sécurité
- Ecole eartsup : administration système et infogérance

### Création Web

- Leguide.com : Audit d'optimisation du site
- Com et Dia : Pré-études de faisabilité pour différents sites



## INTERVIEW

**A 25 ans, Yannick Lejeune est directeur général de 3IE. Il nous présente ici, une structure atypique qui s'adosse à l'EPITA et qui a tout d'une SSII.**

**P : Quels sont les liens de 3IE avec l'EPITA ?**

**YL :** 3IE a été créée le 1er octobre 1999 à l'initiative de l'EPITA. Je suis moi-même un ancien de l'EPITA et à la fin de mon cursus en 1999, j'ai été sollicité pour créer la structure de 3IE. J'ai alors recruté 3 étudiants en cours d'études pour constituer l'équipe. Nous avons très vite démarré avec des prestations de conseil sur les technologies innovantes. Du repérage au déploiement, il n'y avait qu'un pas. Notre service de veille technologique permet également à l'EPITA de rester à la pointe des dernières technologies dans ses enseignements. Il n'est pas rare en effet, que 3IE collabore à la mise en place de nouveaux modules de cours dispensés aux étudiants de l'école. Cela leur permet d'acquérir des connaissances et des compétences qui seront très recherchées par les entreprises, dès leur sortie du cursus. Certains de ces étudiants ont même la possibilité de travailler à temps partiel au sein de notre structure.

**P : Comment intégrez-vous ces étudiants ?**

**YL :** Ils sont avant tout sélectionnés sur leur motivation. Elle doit être forte, car intégrer 3IE nécessite une réelle disponibilité. Ils travaillent à mi-temps pour nous et doivent poursuivre comme n'importe quel autre étudiant de l'EPITA leur cursus. Nous commençons par leur expliquer le fonctionnement de notre entreprise dans son entier, de la maintenance à la rédaction d'un contrat de services. Nous bénéficions d'un véritable potentiel de futurs entrepreneurs. Nous leur proposons bien plus que de l'alternance, car travailler au sein de 3IE leur permet d'acquérir des connaissances et des compétences en se confrontant directement au monde de l'entreprise. Ils sont suivis

personnellement par un coach qui les accompagne dans chacune de leur mission.

**P : 3ie initie des projets, de quoi s'agit-il ?**

**YL :** De manière à acquérir une expertise réelle dans le domaine des nouvelles technologies, 3IE dirige de nombreux projets destinés à mettre en oeuvre son savoir-faire et à parfaire les connaissances de ses membres. Ces projets sélectionnés par rapport à leur degré d'innovation ou leur adéquation au marché sont parfois effectués sous le parrainage de sociétés qui bénéficient alors du transfert de connaissances correspondant à l'ouvrage. Parmi ces projets, on trouve des systèmes décisionnels, du multimédia enrichi, de la voix sur IP, des agents intelligents, des innovations en informatique mobile...

**P : comment les entreprises ont-elles réagi ?**

**YL :** Microsoft a été l'une des premières à nous suivre. 3ie fait désormais partie de l'Advisory Board de leur équipe éducation supérieure. Notre activité ne se cantonne pas à prêcher pour une seule paroisse : Open source, Sun, Solaris, Microsoft, etc, tout dépendra du client final. Nous veillons particulièrement, lors de la sélection des étudiants, à ne pas laisser se développer un certain intégrisme.

**P : Qui sont vos clients ?**

**YL :** Au départ, ce sont des start-up créées par des anciens de l'EPITA. Nous bénéficions également de deux autres types de prescripteurs : les clients de ces derniers et des organismes tels que l'Anvar ou le Critt. Aujourd'hui, de nombreux projets ont abouti et de grands noms de l'industrie nous font confiance. A titre d'exemple, nous avons travaillé pour Palm sur

des tests de plate-forme, Schneider Electric pour la mise en place d'outils, ou encore BNP Paribas pour l'installation d'une plate-forme. Nous avons également des clients atypiques, pour lesquels nous avons pu apporter des solutions originales comme des applications de gestion de Musées. Des organismes, comme la chambre de commerce et d'industrie, nous ont demandé de mettre en place des conférences sur la vulgarisation de la veille technologique.

**P : Comment vous placez-vous sur le marché ?**

**YL :** Nos journées hommes ne sont pas les moins chères du marché mais nos équipes sont nettement plus "véloces". Nos membres sont non seulement d'une extrême compétence mais surtout, animés d'une fabuleuse motivation. Ce sont des équipes jeunes, flexibles et la bureaucratie n'a pas de place au sein de 3IE. Les clients voient en nous un partenaire et non un prestataire. De par le statut de 3IE, nous recherchons avant tout l'équilibre financier. Nous comptons actuellement 1 million d'euros pour les coûts de fonctionnement, ce qui correspond à notre chiffre d'affaires.

**P : Quels sont vos rapports avec les SSII ?**

**YL :** Nos prestations s'étendent du développement à l'expertise, en passant par la formation. Les sociétés de services ne sont pas nos concurrents car nous ne nous positionnons pas sur le marché en lui-même, mais sur le marché de demain, celui qui existera dans 6 mois ou 1 an. Certaines SSII sont même nos clientes, nous apportant une partie d'un lot qu'elles doivent honorer pour leurs propres clients.

■ AFM

# Synchroniser des Go de codes sources sous Windows et Linux en quelques minutes

Dans cet article vous saurez tout ce qu'il faut savoir sur la manière de sauvegarder automatiquement vos données sur un serveur FTP distant et sur la mise à jour d'une copie miroir des données.

**P**our le travail d'archivage et de copie, vous pouvez envisager d'installer une machine Linux pour effectuer ce genre de travail de copie sur disque dur et d'archivage. Un PC à première vue obsolète, comme un AMD K6 cadencé à 300 Mhz avec 92 Mo de mémoire vive, peut alors suffire. Vous économiserez le prix d'acquisition d'un PC neuf et du système d'exploitation Windows (qui de toute manière ne peut pas s'installer sur ce type de matériel).



Unité SDLT de sauvegarde sur bandes.

Les dossiers partagés des machines Windows seront "montés" sous Linux comme partages SMBFS. Il suffira d'éditer votre fichier /etc/fstab pour les y ajouter :

```
...
//venus/backup /mnt/backup smbfs username
=guest,password=guest
```

Notez bien que si Linux vous rebute, il sera toujours possible de reproduire ce que nous allons vous expliquer sous Windows.

## La synchronisation des données et non la copie totale

Dans la plupart des cas il n'est pas réalisable de recopier chaque soir l'intégralité des données. Par exemple, si votre volume quotidien moyen de nouvelles données est de 10 Mo, vos données de bases sont probablement 1000 fois plus lourdes. Vous ne pouvez pas

recopier 5 ou 10 Gigas de données chaque nuit, sous prétexte de conserver une copie de sauvegarde actualisée. Cette copie prendrait trop de temps et de ressources (bande passante, CPU) et ne vous permettrait pas de revenir quelques jours en arrière.

Cependant, si par exemple vous désirez sauvegarder sur une machine distante votre site Web local ou vos dernières modifications à vos codes sources, une solution très simple est d'utiliser TAR en combinaison avec NCFPT. Ces deux commandes, faut-il le préciser, sont disponibles sous Linux mais également en version binaire pour Windows (mais vous devez aussi ajouter gzip si vous compressez l'archive (option -z)).

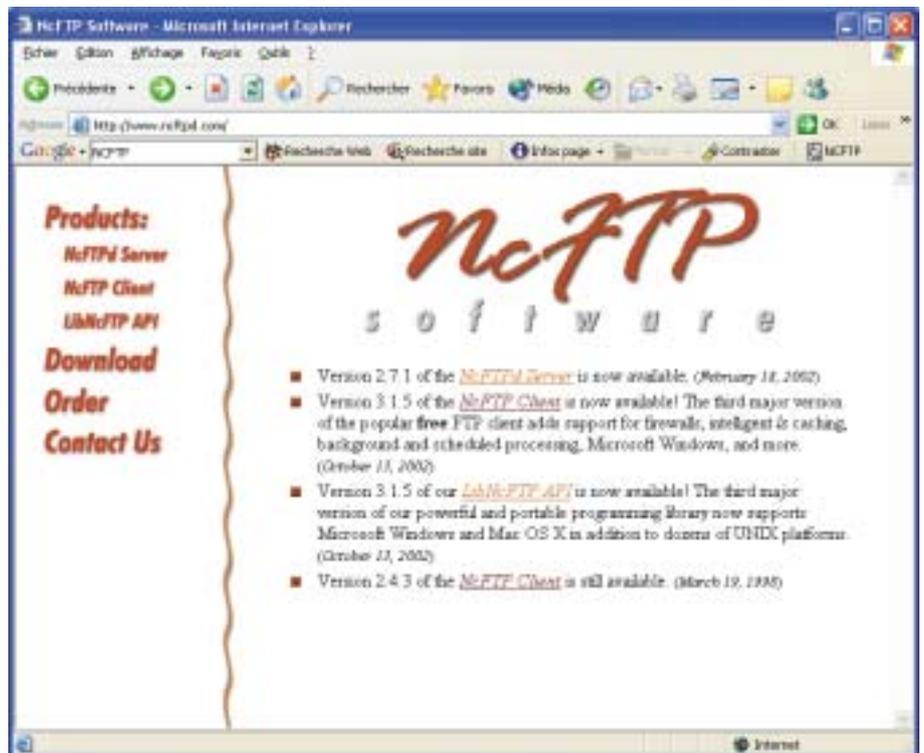


Une cartouche de haute capacité "SuperDLTtape" (SDLT).

Par exemple :

```
tar -czf /home/backup/sauvegarde.tar.gz
/var/www/html/
ncftpput -u login -p pass sitedistant.fr /backup/
/home/backup/sauvegarde.tar.gz
```

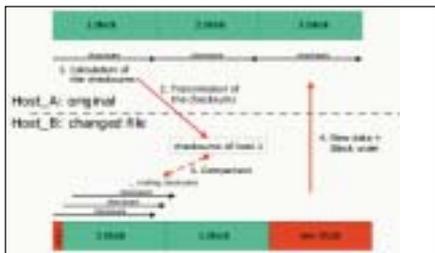
ncftpput "uploadera" le fichier. L'option "-u" permet d'indiquer le login, l'option "-p" le mot de passe, spécifiez ensuite l'adresse du serveur FTP distant, le répertoire distant (ici /backup/) et le fichier local à transférer.



Home page de NCFPTD : vous pouvez y télécharger une version cliente Windows ou/et Linux.



Logo de RSYNC.



L'algorithme de RSYNC découpe chaque fichier en un ensemble de blocs puis en calculant des sommes de contrôles.



Connexion cliente NCFTP sous Linux.

Pour éviter de transférer l'intégralité des données chaque nuit, il y a lieu de réaliser une copie différentielle. Ainsi à partir d'un premier "miroir" des données, vous n'aurez plus qu'à synchroniser chaque nuit les "différences" constatées.

## Une synchronisation universelle avec "tar"

Une combinaison des commandes tar (archivage) et find (établissement de la liste des derniers fichiers modifiés) permet de réaliser une archive des modifications. Voici le script public shell de Sébastien Viot qui combine ces deux possibilités :

```
#!/sbin/sh
jour=`/usr/bin/date +%d`
num_jour=`/usr/bin/date +%a`
mois=`/usr/bin/date +%m`
annee=`/usr/bin/date +%Y`
date_sauve=$jour'.'$mois'.'$annee
nom_fichier_log=/tmp/SI_`date_sauve`
position=0
```



Extrait de la page de manuel de rdiff-backup : reportez-vous y pour plus de détails sur son utilisation.

```
case $num_jour in
'Mon'|'lun') critere=-3
/usr/bin/rm /tmp/SI*
position=0;
'Tue'|'mar') critere=-1
position=1;
'Wed'|'mer') critere=-1
position=2;
'Thu'|'jeu') critere=-1
position=3;
'Fri'|'ven') critere=-1
position=4;
esac
cd /
echo Recherche des fichiers a sauvegardes
find home/users/* ! -type d -mtime $critere |
grep -v .dt | grep -v .fm > $nom_fichier_log
echo Fin de la recherche : $nom_fichier_log
echo Rembobinage de la cassette
mt -f /dev/rmt/1n rewind
if ((test "$num_jour" = "lun") || ( test "$num_jour" = "Mon")) ;
then
echo Effacement de la cassette
mt -f /dev/rmt/1n erase ;
fi
echo Positionnement sur la cassette
mt -f /dev/rmt/1n fsf $position
echo Debut du tar
echo $nom_fichier_log
tar cvf /dev/rmt/1n -I $nom_fichier_log

echo Fin du tar
echo Ejection de la cassette
```

```
#mt -f /dev/rmt/1n rewoffl
#mt -f /dev/rmt/1n offline
echo Sauvegarde incrementale OK
```

Et voici le script de restauration :

```
#!/sbin/sh
echo "Restauration de fichiers a partir de la sauvegarde incrementale"

# Teste le nombre d'arguments, lors de l'exécution du script
if test $# -lt 3
then
echo "Usage : $0 <nom du jour> <utilisateur> <criteres de recherche>"
exit 1
fi

# Pre-positionnement sur la cassette en fonction du jour
position=0
case $1 in
'lundi') position=0;;
'mardi') position=1;;
'mercredi') position=2;;
'jeudi') position=3;;
'vendredi') position=3;;
esac
if ( (test "$1" != "lundi") && (test "$1" != "mardi") && (test "$1" != "mercredi") && (test "$1" != "jeudi") && (test "$1" != "vendredi") ) ;
then
echo "Erreur : jour = lundi, mardi, mercredi, jeudi et vendredi"
```

```

exit 1;
fi

# Création du répertoire d'extraction
cd /tmp

# Teste l'existence du répertoire d'extraction
retour=`ls | grep extraction`
if (test "$retour" != "extraction") ;
then
mkdir -m 777 extraction ;
fi
cd extraction

# Lancement de l'extraction
mt -f /dev/rmt/1n rewind
mt -f /dev/rmt/1n fsf $position
echo "Lancement de la recherche"
retour=`tar lf /dev/rmt/1n | grep $2 | grep $3`
if (test "$retour" = "") ;
then
echo "Fichier(s) non trouve(s)"
exit 1;
fi

```

Le gros défaut de cette technique est d'archiver un fichier dans sa totalité ! On peut faire de nos jours beaucoup mieux, en ne transférant que le volume qui a réellement changé au sein d'un fichier et non le fichier complet (dont la date de stockage sur disque a été modifiée).

## Une commande presque parfaite : rsync

La commande rsync réalise ce travail, que ce soit sous Linux ou sous Windows ([http://optics.ph.unimelb.edu.au/help/rsync/rsync\\_pci.html](http://optics.ph.unimelb.edu.au/help/rsync/rsync_pci.html)). Rsync fonctionne en mode client/serveur (la commande est exécutée sur le serveur d'archivage Linux/Windows et sur les postes clients) en n'envoyant que les "différences" entre les fichiers (et des sommes de contrôles). En effet, son algorithme découpe chaque fichier en un ensemble de blocs, puis calcule au besoin deux sommes : une somme faible sur 32 bits et une somme forte sur 128 bits. On part du principe que si les blocs testés sont différents il n'y aura pas de correspondance des sommes. La somme forte est calculée uniquement lorsque la somme faible d'un bloc ne correspond pas. Par rapport à d'autres moyens (dump, diff, tar, etc), c'est rsync qui transmettra le moins de données. En revanche la consommation CPU est parfois plus élevée (en raison du calcul des sommes

```

root@endymion: ~ - 192.168.0.29 VT
File Edit Setup Control Window Help
[root@endymion /root]# rdiff-backup -l INETPUB_BU
Found 3 increments:
  increments.2003-06-06T15:13:04+02:00.dir  Fri Jun  6 15:13:04 2003
  increments.2003-06-06T16:35:23+02:00.dir  Fri Jun  6 16:35:23 2003
  increments.2003-06-06T17:12:04+02:00.dir  Fri Jun  6 17:12:04 2003
Current mirror: Tue Jun 10 13:25:25 2003
[root@endymion /root]#

```

Commande "rdiff-backup -l INETPUB\_BU" qui liste les points de synchronisation actuels.

```

root@endymion: ~ - 192.168.0.29 VT
File Edit Setup Control Window Help
[root@endymion /root]# cat copie.log
-----[ Session statistics ]-----
StartTime 1055243672.36 (Tue Jun 10 13:14:32 2003)
EndTime 1055244324.27 (Tue Jun 10 13:25:24 2003)
ElapsedTime 651.90 (10 minutes 51.90 seconds)
SourceFiles 26205
SourceFileSize 1690433874 (1.57 GB)
MirrorFiles 26205
MirrorFileSize 1690129681 (1.57 GB)
NewFiles 2
NewFileSize 3056 (2.98 KB)
DeletedFiles 2
DeletedFileSize 3049 (2.98 KB)
ChangedFiles 67
ChangedSourceSize 350160084 (334 MB)
ChangedMirrorSize 349855898 (334 MB)
IncrementFiles 71
IncrementFileSize 388515 (379 KB)
TotalDestinationSizeChange 692708 (676 KB)
Errors 0
-----
[root@endymion /root]# █

```

La synchronisation via rdiff-backup a duré 10 minutes. Plus de 26000 fichiers ont été passés en revue et 67 fichiers ont été modifiés depuis la dernière synchronisation...

fortes). Rsync est parfaitement indiqué pour réaliser un miroir de données FTP, sauvegarder des répertoires ou des partitions, synchroniser des documents de bureau à partir de son ordinateur à la maison, etc.

## L'outil ultime : rdiff-backup

Cependant cette manière de pratiquer ne permet toujours pas de revenir sur des changements antérieurs. La synchronisation des données sur le serveur d'archivage est rapide mais c'est finalement le seul avantage. C'est pourquoi nous préconisons l'emploi de rdiff-backup qui reprend le principe de rsync tout en y ajoutant la sauvegarde différentielle (<http://rdiff-backup.stanford.edu/>). Entre parenthèses, rdiff-backup est un script Python.

La commande rdiff-backup est exécutable sous Windows, par l'intermédiaire du projet cygwin (<http://sources.redhat.com/cygwin/>) en raison

de l'implémentation de la bibliothèque de synchronisation. Elle offre une sauvegarde avec historique différentiel pour n'importe quel type de fichiers y compris des liens symboliques. Les "différences" par rapport au miroir sont stockées sous la forme d'archives compressées "tar.gz". Le format est donc standard.

```
rdiff-backup /repertoire/localsource utilisateur@machinedistante::/repertoire-utilisateur/cible
```

Et en ce qui concerne la restauration :

```
rdiff-backup -r 3D /home/destination /usr/restore
```

Le paramètre "-r 3D" signifie "restaure" en revenant trois jours "en arrière dans le temps". Par ailleurs de très nombreuses options existent, comme la possibilité d'exclure des répertoires en indiquant des expressions génériques. La commande man rdiff-backup vous renseignera plus en détails.

■ Xavier Leclercq

# Architectures distribuées : le remoting comme solution ?

2<sup>ème</sup> partie

Les architectures distribuées ne sont pas apparues par hasard. En effet, le prix du matériel, ainsi que ses performances, ont souvent été des facteurs limitants, quant à la montée en puissance d'applications. C'est pour cela qu'est apparue progressivement la notion d'architecture distribuée qui utilise la puissance de plusieurs machines pour effectuer une tâche qui auparavant était effectuée sur une seule.

## AUTHENTIFICATION

Par authentification on comprendra le fait de pouvoir identifier de façon certaine les acteurs (le client et/ou le serveur). L'acte d'authentification se fait généralement en fournissant un identifiant : le couple login/password en est un exemple.

L'authentification peut être envisageable des deux côtés (client et serveur). Le serveur peut souhaiter limiter l'accès à une ressource particulière en fonction du client. Mais le client peut également souhaiter authentifier le serveur, afin de ne pas envoyer des informations confidentielles à n'importe qui.

La mise en place d'une authentification n'est pas toujours nécessaire. Par exemple, dans le cadre d'un Web Services qui donne la température dans une ville donnée, il ne sera pas utile d'authentifier les intervenants : cela ne pose pas énormément de problèmes de sécurité. Mais dans le cadre de manipulations de données inter entreprises, des précautions devront être prises et cela, afin d'empêcher quiconque d'utiliser un service qui ne lui est pas destiné. HTTP possède les moyens d'authentification suivants :

- Basic.
- Digest.
- Windows Integrated \*
- SSL.

\* Windows uniquement.

### Basic et Digest

- Basic permet à un client de fournir un login/password pour permettre un usage d'une ressource.
- Digest est la même chose que Basic, sauf qu'au lieu de faire circuler les données login/password en clair, celles-ci sont encodées de façon réversible...

Dans les deux cas, seul le client est authentifié et les communications qui suivront ne seront pas du tout encryptées. Il est donc particulièrement déconseillé d'utiliser ces protocoles dans le cadre d'applications d'entreprises. De plus, ce type d'authentification n'assure aucune protection contre les attaques par replay (replay attack).

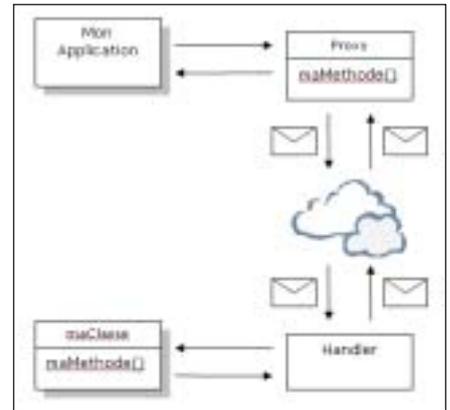
### Windows Integrated

Windows possède un mode d'authentification particulier : Windows Integrated. Ce mode d'authentification utilise la couche SSPI de Windows afin d'authentifier le client. Dans le cadre d'une utilisation avec Kerberos, le serveur est également authentifié.

### SSL

SSL est bien plus qu'un moyen d'authentification, c'est également un système d'encryptions des données. Ici sera considéré uniquement la partie authentification de ce protocole.

Lors de la négociation, le client peut connaître



l'identité du serveur en demandant la validation du certificat de serveur à l'autorité qui l'a générée. Le serveur, quant à lui, peut exiger que le client fournisse un certificat afin de valider son identité par rapport.

## AUTORISATION & CONFIDENTIALITE

Du point de vue du serveur, une fois que le client est authentifié, il est possible de lui attribuer des autorisations. L'objectif des autorisations est de définir des droits d'usage d'une ressource. Cela peut aller, par exemple, des droits de lecture pour les simples employés d'une entreprise, à des droits de modification pour l'administrateur réseau.

.NET possède un modèle de sécurité pour l'exécution du code. Il est possible d'appliquer une identité à un thread en fonction de l'authentification, ce qui permettra ensuite d'appliquer des règles d'accès pour l'exécution.

Utiliser IIS et ASP.NET permet de mettre en œuvre facilement la gestion des authentifications et par la même occasion des autorisations, en utilisant un attribut Principal Permission pour accorder ou non le droit d'exécution d'une portion d'implémentation.

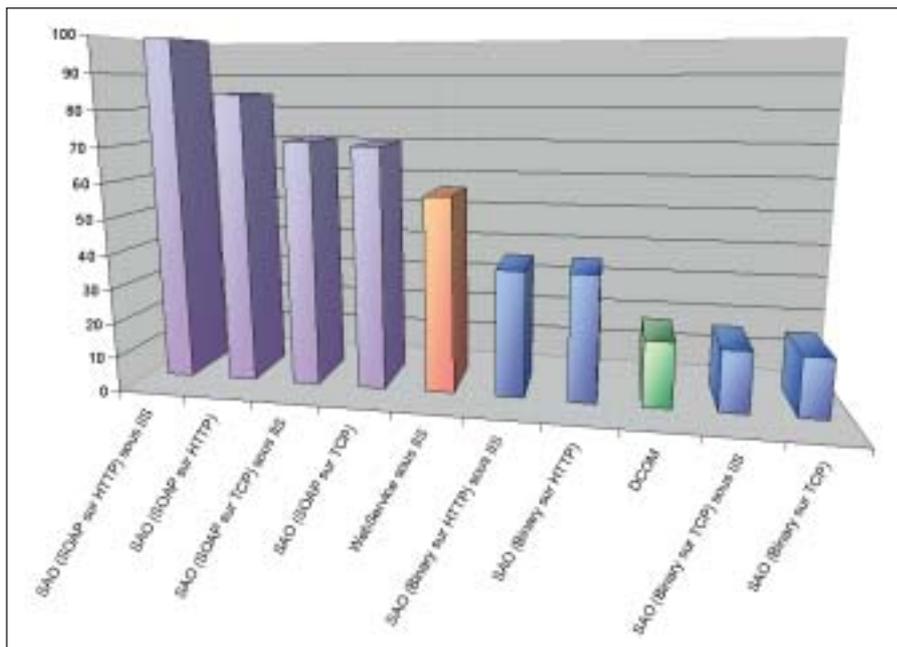
La confidentialité passe souvent par l'usage d'une encryption des données. Actuellement, sur HTTP il existe un protocole standard : le SSL. Il s'utilise au moyen d'une URL possédant le protocole HTTPS.

## FIABILITE DU TRANSPORT

HTTP n'est pas un protocole fiable dans le sens où une requête peut être envoyée plus d'une fois... HTTP utilise en général TCP, donc ce n'est

Dans le cadre de la comparaison performance/services offerts, il est bon de comparer deux configurations qui seront couramment utilisées :

	SAO (Binary sur TCP)	SAO (Binary sur http) sur http sous IIS
Performance brute	✓	
Mise en cluster	Windows, hardware	Windows, hardware
Recyclage du serveur		IIS
WebGarden		IIS 6
Mise en œuvre de la sécurité		IIS (SSL, Authentification...)
Monitoring	EventLog	Compteurs de performances pour IIS, EventLog



pas un problème des données, mais bel et bien un problème du protocole HTTP qui ne spécifie pas correctement le comportement d'un client HTTP face aux erreurs. Le vague de la spécification de HTTP (pléthore de "MAY" et "SHOULD") est donc clairement en cause ici.

Cela implique que les services accédés au moyen d'HTTP devront être idempotents : une requête devra donner le même résultat quel que soit le nombre d'exécutions effectuées. Jusqu'à temps que le HTTP soit remplacé par un protocole fiable, toute méfiance sera donc la bienvenue. Pour corriger ce défaut, des spécifications de protocoles commencent à apparaître, comme le HTTPR, afin d'en faire profiter les applications. Dans le remoting, il est possible de changer de protocole. L'usage de protocole fiable tel que TCP sera préférable mais cela aura énormément d'impact sur la sécurité :

- Développement de ses propres couches de sécurité.
- Développement de ses propres couches de qualité de service.

## SECURITE AVEC LE REMOTING

Les objets à utiliser au moyen du remoting peuvent être exposés au moyen d'IIS. Dans ce cas, il suffira de configurer correctement IIS. Attention toutefois, il existe une limitation avec HTTPS : le TransportSink HTTP actuel ne permet pas à un client de fournir son certificat. Le serveur ne pourra être assuré de l'identité du client dans de tels scénarios.

En dehors de l'usage d'IIS, la sécurité est dure

à mettre en œuvre sans mettre la main dans le cambouis. En effet, il n'existe rien pour sécuriser les accès via remoting !

## DEPLOIEMENT

L'usage de composant distribués doit faciliter la distribution de l'application. En effet, si à chaque fois qu'une nouvelle version apparaît côté serveur il fallait redéployer des composants côté client, le coût de gestion pourrait atteindre des proportions inquiétantes. Heureusement, les WS et le remoting fonctionnent avec la notion d'interface. Une interface est une façade qui est exposée au monde. Cette façade doit être immuable :

- Au niveau des méthodes exposées.
- Au niveau des exceptions levées.

Cette problématique doit être prise en compte dans le cycle de vie du logiciel : le versionning. Le respect de l'immuabilité de l'interface n'est en général pas un problème (encore moins en .NET, puisque l'on peut ajouter des méthodes sans casser la compatibilité). Le problème se situe surtout au niveau de l'activation. En effet, .NET est un framework qui fonctionne en utilisant des assemblies avec version. Cela risque donc de poser des problèmes d'activation à ce niveau.

Le remoting est un peu plus compliqué à déployer que le Web Service puisqu'il nécessite la présence côté client d'une description de l'objet à activer. Il faut ainsi une assembly décrivant exactement le type de l'instance à activer. A première vue, une solution de facilité

est de recopier le composant sur le poste client, mais cela risque de poser des problèmes de déploiement et de sécurité :

- Déploiement : chaque fois qu'il y aura une modification côté serveur.
- Sécurité : les composants peuvent parfois contenir des données sensibles qu'il n'est pas souhaitable de partager.

Une autre solution est d'extraire les informations nécessaires pour le remoting. L'utilitaire **soapsuds** (livré avec le .NET SDK) est présent pour résoudre ce problème, en extrayant tout ce qui sera nécessaire pour une activation à distance. Malheureusement, soapsuds apporte plus de problèmes qu'il n'en résout. La génération à la main des proxies sera donc la méthode préférée pour contrôler finement les versions côté client.

Le remoting se découpe en deux grandes catégories : les objets SAO et CAO. Les objets SAO sont toujours activés par le serveur. Lors d'une activation par le client, ce dernier ne peut préciser la version de l'objet à activer, cela sera toujours la version déclarée lors de la configuration du serveur. Le seul moyen pour qu'un client puisse activer la version de son choix est donc de proposer des endpoints différents.

Quant aux CAO, ils sont toujours activés par le client (c'est d'ailleurs pour cela qu'il est impossible de préciser la version lors de la configuration du serveur). Lors d'une activation, le remoting envoie toutes les informations nécessaires concernant le type à activer vers le serveur :

- Assembly.
- Signature de l'assembly.
- Version de l'assembly.
- Type à activer.

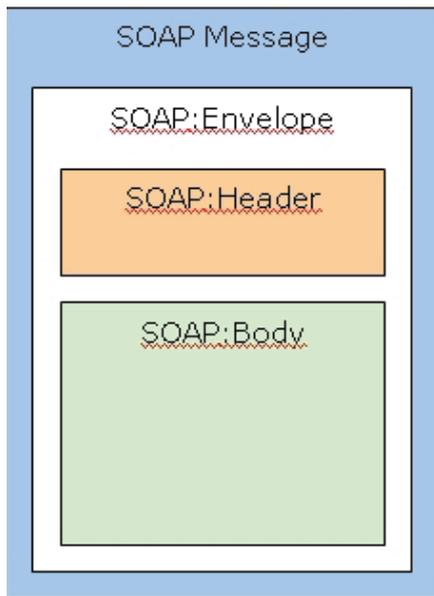
Cela peut donc poser d'énormes problèmes d'activation lorsque le versionning est utilisé.

## OBJET MARSHALÉ PAR VALEUR AVEC LE REMOTING

Les objets marshalés par valeur dans le remoting sont un peu un cas à part :

- Le serveur instancie et initialise une classe dans la version de son choix.
- Le client consomme une copie de cette instance.

Il risque donc d'y avoir des problèmes de compatibilité de versions : il peut être envisageable que le serveur possède une version plus récente des composants, provoquant ainsi des problèmes de compatibilité avec le client. Le problème est plus large que cela en fait : la sérialisation d'une instance se fait non seulement



en stockant ses données internes mais également en stockant la version du composant (dans le strong name). Cela amène donc des problèmes d'activation et non d'interfaçage. Pour éviter cela, il est possible de supprimer la sérialisation de la version avec le strong name. Cela permettra au client de continuer à utiliser un composant, même dans une version plus récente.

```
<formatter
  includeVersions="false"
/>
```

Attention toutefois à ne pas abuser de cette possibilité. L'inhibition de la sérialisation de la version de composant se fait de manière globale au formateur du channel, ce qui veut dire que tous les composants marshalés par valeur subiront le même traitement.

Lorsque le serveur envoie la description d'un type marshalé par valeur, la version du type est également envoyée (puisque faisant partie du strong name). Le problème est que pour une nouvelle version du composant, celle-ci n'aura pas été forcément déployée sur le poste client. Il est donc possible de spécifier côté client qu'il faut ignorer la version du type. Pour cela, il suffit de configurer le formateur du channel de la façon suivante :

```
<formatter
  strictBinding="false"
/>
```

Comme pour l'attribut " includeVersions ", tout le channel est impacté.

■ **Pierre Chalamet** - (Neoxia)

## QUELLE EST LA MEILLEURE METHODE POUR EXPOSER DES COMPOSANTS ?

Il faut tout d'abord savoir quels sont les systèmes qui devront communiquer. En effet, choisir le remoting pour faire dialoguer une application Java sous Solaris et une application .NET sous Windows n'est sûrement pas la meilleure chose à faire (même s'il existe des ponts entre ces deux mondes). Les WS sont présents pour répondre aux problèmes d'interopérabilité en milieu hétérogène. L'objectif principal n'est donc pas la performance mais bel et bien la mise en place d'un moyen de communication standardisé. Les WS utilisent HTTP car c'est un protocole largement répandu dans les entreprises. SOAP est utilisé car il permet l'interprétation de données sur différents systèmes (le point fort d'XML est l'échange de données). En revanche, le remoting vise d'autres cieux. Son principal objectif est de conserver la sémantique de l'appel de méthode, quel que soit le protocole employé. Un autre objectif du remoting est la performance. L'usage de SOAP est pratique dans le cas de systèmes hétérogènes, mais lorsque toutes les applications fonctionnent sous .NET, son usage devient très discutable. SOAP est du XML et donc les messages SOAP ont tendance à être très volumineux. Le besoin de faire des transactions distribuées est également un point crucial. En effet, l'usage des WS et du remoting dans l'état actuel ne permet pas de faire des transactions distribuées. Il faudra donc faire très attention à ce point là, lorsqu'il s'agit d'applications d'entreprises utilisant plusieurs bases de données, par exemple. Il est toutefois possible de faire des transactions distribuées avec les WS ou le remoting, mais au prix d'un effort de développement assez conséquent.

En dehors du problème de la gestion de l'état, le remoting permet également à l'objet serveur de faire des appels sur le client. Cela est utilisé dans le cadre des événements, par exemple (events). Un objet côté serveur pourra envoyer des notifications vers un client, sans que ce dernier ait à faire de " polling ". Cela peut contribuer à réduire l'usage de la bande passante, surtout si le client doit être notifié le plus rapidement possible.

Le remoting CAO est à déconseiller dans la plupart des cas, mais son usage offre de nombreuses possibilités, comme par exemple la création d'EnvoySink ou de ChannelSink, apportant ainsi un contrôle très fin dans le développement d'architectures distribuées. Mais pour la majorité des applications, l'usage des WS pour l'interopérabilité ou du remoting SAO pour les performances suffira largement !

	Web Services	Remoting
<b>Protocoles</b>		
HTTP	✓	✓
HTTPS	✓	Partiellement
Autres		TCP, MSMQ, Jabber, SMTP, NamedPipes...
<b>Formateurs</b>		
SOAP 1.1	✓	✓
Binaire		✓
Autres		XML-RPC...
<b>Activation</b>		
Stateless	✓	✓
Statefull		✓
Partage de références statefull		CAO uniquement ou fabrique de composant (SAO)
<b>Transaction</b>		
Locale	✓	✓
Distribuée	Envisageable	Envisageable
<b>Sécurité</b>		
SSL	✓	✓
Authentification	IIS	IIS
Autorisation	ACL, .NET	.NET (si hébergé dans IIS)
<b>Interopérabilité</b>		
Java	✓	Envisageable
Unix	✓	Envisageable
Win32	SOAPTK	Envisageable
Win32 managé	✓	✓

# Développer un framework PHP

A la suite des articles précédents sur les Frameworks, concentrons-nous cette fois sur la création d'un framework PHP, l'organisation du code en limitant les redondances.

## Première méthode, l'inclusion

Le premier mot qui nous vient à l'esprit lorsqu'il s'agit de découper et d'organiser du code en PHP est le fameux "include" (ou "require"), instruction connue de tous. Nous allons créer un en-tête, un pied de page, et inclurons ces éléments dans chacun des scripts que nous allons développer. L'en-tête gèrera les actions prioritaires comme le démarrage de la session, l'éventuelle vérification des paramètres de l'url, et pourquoi pas, de l'affichage de la bannière. Le pied de page se chargera de toutes les actions normales de fin de page, comme les calculs de temps de génération de la page, l'écriture dans un cache, en plus du simple fait d'afficher le dit pied de page. Nous pourrions envisager notre exemple comme suit :

### header.inc.php

```
<html>
<head>
<title>Page pour un simple framework</title>
</head>
<body>
<div id="head"><h1>Site de démonstration
</h1></div>
<?php
// nous utilisons toujours les tags php complets
pour être compatible quelle
// que soit la valeur de l'option "short tag"
session_start ();
list($micro,$time) = explode(' ', microtime());
$_startTimer = $micro + $time;
//génération d'un identifiant de visite.
//nous choisissons de ne pas utiliser l'identifiant
de la session.
isset($_SESSION['visitId']) || $_SESSION
['visitId'] = UniqId ();
$_SESSION[$_SESSION['visitId']]['count']
+;//incrémente le nombre de visites de l'utilisateur
?>
```

### footer.inc.php

```
<?php
//nous allons sauvegarder le nombre des visites.
```

```
mysql_connect ();
//...
//Affichage du temps de calcul.
list($micro,$time) = explode(' ', microtime ());
$_stopTimer = $micro + $time;
echo "Page générée en ", $_stopTimer - $_
startTimer;
?>
<div id="footer">
<p>Copyrights foo, php rules.</p>
<p>Réalisé avec Side http://phpside.org</p>
</div>
</body>
</html>
```

### Un exemple d'utilisation

```
<?php
//nous donnons l'extension .php à toutes nos
pages, quelles qu'elles soient,
//pour des raisons de sécurité. (nous ne souhaitons
pas qu'un visiteur puisse obtenir
//par mégarde l'ensemble des paramètres de
connexion à notre site)
include ('header.inc.php');
?>
<p>Bienvenue sur notre site de démonstration
</p>
<?php
include ('footer.inc.php');
?>
```

Les avantages de cette méthode résident dans sa simplicité de mise en oeuvre. Toutefois, nous risquons d'être confrontés à des problèmes de souplesse. En l'état, il est exclu, au développement de notre page, de contrôler ce qui se passe à l'intérieur des fichiers header et footer. En effet, si nous souhaitons pour un script donné (par exemple la page d'accueil) changer le texte de l'en-tête, ou simplement sa couleur, c'est impossible.

De même, si nous disposons de pages qui ne génèrent aucun affichage (une redirection après un traitement par exemple), il nous est impossible d'utiliser les "header" et "footer",

qui génèrent du contenu et nous empêchent ainsi de manipuler l'en-tête (nb: sauf utilisation de l'option output\_buffering, qui a l'inconvénient de rendre notre application dépendante de la configuration).

Bien sûr, nous pouvons développer une entité spécifique de nos en-têtes et pied de page pour les cas particuliers, mais nous perdons alors tout l'intérêt de notre framework, censé centraliser les actions communes.

Un autre inconvénient de taille, nous allons devoir développer pour chaque site un exemple de nos "header" et "footer". Pas question ici de capitaliser sur nos développements, si ce n'est en devenant agiles aux copier / coller en triant les éléments spécifiques à chacun de nos sites !

## Deuxième méthode, bibliothèque de fonctions

En développant une bibliothèque de fonctions pour réaliser les tâches communes, il est indispensable de penser à ce que nous voulons faire, et comment nous voulons le faire. En d'autres termes, ne fonçons pas tête baissée et évitons de nous disperser dans les méandres de la programmation.

L'exemple choisi ici est toutefois suffisamment simple pour ne pas nous perdre dans ces dérives.

Nous allons créer les fonctions doHeader () et doFooter (). Toutes deux attendront en paramètre un tableau d'options régissant leur fonctionnement, pour éviter de nous retrouver dans un contexte où les dites fonctions ne seraient que des alias des précédents includes.

```
<?php
//once, pour éviter les risques de redéclaration.
require_once ('myLib.php');
doHeader (array ('color'=> '#cccccc', 'session'
=>true, 'timeCounter'=>true, 'html'=> 'head.htm'))
//le code de notre page.
doFooter (array ('timeCounter'=>true, 'html'=>
'foot.htm'));
?>
```

Encore une fois, cette méthode présente l'avantage d'être simple à réaliser et à mettre en oeuvre. Nous obtenons une certaine maîtrise des éléments systématiques par l'intermé-

diaire de tableaux d'options. Nous entrevoions une capitalisation de nos développements plus aisée car il suffira de transporter notre bibliothèque "en l'état" pour pouvoir l'utiliser dans un autre site, les éléments spécifiques à l'application étant dissociés.

Il subsiste toutefois quelques difficultés. Nous voyons dans notre exemple que nous devons spécifier par deux fois que nous souhaitons afficher les temps de calcul de la page, alors que si nous en avons demandé le calcul dans l'en-tête, il est logique que nous souhaitions l'afficher dans notre pied de page. Pour éviter ce problème, les plus pressés utiliseraient des variables globales pour régir nos calculs. C'est cependant une méthode très risquée : ces variables peuvent être écrasées en cours de route par d'autres intervenants. Notre framework devant être réutilisable, nous ne pouvons demander à ses utilisateurs d'en connaître l'implémentation. Autre inconvénient qui ne transparait pas ici, le risque de conflits des noms de fonction. Si nous souhaitions créer des méthodes d'authentification, de gestion de cache, d'abstraction des bases de données, nous arriverions à une multiplication des éléments à même de rendre la tâche "d'invention des noms de fonction" pénible, sans quoi nous provoquerions des conflits avec les bibliothèques externes ou existantes. La liste n'étant pas exhaustive, nous sommes aussi confrontés à une problématique de maintenance. Nous souhaitons ajouter à toutes nos pages des comportements communs, via des options ? Nous devons modifier tous les scripts impliqués, ou alors modifier les comportements par défaut de nos fonctions.

### Troisième méthode, bibliothèque de classes

PHP apporte un support objet assez limité, mais nous en avons suffisamment pour ce qui nous intéresse, et disposons souvent de "hacks" pour pallier les manques évoqués.

Nous allons créer une classe *Framework*, qui implémentera toutes les actions dont nous avons besoin.

Cette classe s'appuiera sur un fichier de configuration pour régir les comportements par défaut, comportements que nous pourrions former lors du développement de nos pages.

```
<?php
include ('framework.class.php');
//l'opérateur new retourne bizarrement (en
```

```
PHP4) une copie de l'objet créé.
//pour pallier ce problème, nous forçons la
récupération de la référence à
//l'objet créé.
$framework = & new Framework ();
$framework->setConfigVar ('Titre', 'Exemple
simple de framework');
$framework->start ();
echo "Autre framework simple";
$framework->end ();
?>
```

Le constructeur de notre classe *Framework* et notre fichier de configuration peuvent être de la forme :

#### framework.class.php

```
<?php
class Framework {
/**
 * variables de configuration.
 */
var $_config;
function Framework ($configFile = 'framework.
config.php'){
//notre fichier de configuration est ici écrit
en php, car rien ne s'analyse
//plus vite en php que du php.
include ('framework.config.php');
$this->_config = get_defined_vars ();
}
}
?>
```

#### framework.config.php

```
<?php
$timer = true;
$header = 'header.htm';
$footer = 'footer.htm';
//autres paramètres.
?>
```

Nous pouvons donc avoir plusieurs fichiers de configuration, regroupés par thèmes. Un fichier de configuration pour la page d'accueil, un fichier de configuration pour les pages non visibles, et plus généralement, un fichier de configuration par groupe de fonctionnalités. L'avantage d'une telle méthode est de profiter de tout ce que l'objet peut nous offrir, modularité et évolutivité.

L'inconvénient de l'utilisation d'une telle vision est le coût d'apprentissage. Plus notre framework sera évolué, plus il comportera d'API diverses, de couches d'abstractions,

d'éléments et de normes associés, plus il sera difficile à aborder par les néophytes.

### Un élément à considérer, les "fichiers déclaratifs".

Toutes les méthodes que nous avons passées en revue, proposent l'utilisation directe des éléments de notre framework. Nous nous positionnons naturellement dans le contexte "à une url correspond un script". C'est en effet la plus naturelle des solutions, mais pas l'unique solution.

Nous pouvons parfaitement envisager de développer un script qui soit capable d'aller chercher les informations, grâce à des paramètres précis de l'url. En procédant ainsi, nous serions capables, via un fichier de configuration, de maîtriser l'ensemble des scénarios de notre site, et de maîtriser l'accès à notre site depuis un seul fichier "exécutable".

Pour cet exemple, nous allons repartir de notre classe *Framework*, en lui apportant quelques corrections. Nous allons créer le fichier *index.php*, point d'entrée unique de notre site, pour qu'il soit minimaliste.

#### index.php

```
<?php
include ('framework.class.php');//framework
.class.php est situé dans le include_path PHP.
$framework = & new Framework ();
$framework->execute ();
?>
```

Difficile de faire plus court. Comme vous pouvez le voir, nous incluons notre bibliothèque, instancions un objet de type *Framework* et en demandons son exécution immédiate. Mais comment peut-on réaliser un site complet grâce à un seul fichier ? La réponse se cache dans ces fameux fichiers déclaratifs, qui vont contenir toutes les actions possibles de notre site, et qui vont définir les règles associées à ces actions. Regardons en premier lieu un exemple de ce que peut être un fichier déclaratif.

#### all.declare.php

```
<?php
// encore une fois, notre fichier déclaratif est
écrit en PHP, car rien
// n'est analysé plus vite en PHP que du PHP.
// Il est toutefois parfaitement envisageable
d'opter pour des choix différents,
// comme des fichiers .INI ou un format XML.
$default = & new FrameworkAction ('default.
php', array ('head'=>'header.htm'));
```

```
$redirect = & new FrameworkLink ('http://
copix.aston.fr');
?>
```

Nous avons déclaré deux actions. La première, *default*, exécutera le script *default.php*. Nous donnons à cette action un paramètre supplémentaire, un tableau associatif. Ce tableau associatif nous permet de donner à chacune de nos actions un nombre de paramètres illimité. La deuxième action, *redirect*, se propose de réaliser une redirection vers un site extérieur. Nous ne donnons pas de paramètre supplémentaire à cette dernière. Il ne nous reste plus qu'à voir l'implémentation de la méthode *execute* de la classe *Framework*.

```
<?php
class Framework {
    function execute (){
        include ('all.declare.php');//inclusion de
notre fichier de déclaration.
        // nous vérifions s'il existe un paramètre
action dans l'url,
        // si c'est le cas, c'est l'action que nous
allons tenter d'exécuter,
        // si ce n'est pas le cas, nous chercherons
l'action nommée default.
        $action = isset ($_GET['action']) ? $_GET
['action'] : 'default';
        //nous vérifions si l'action existe.
        if(!isset ($$action)){
            //l'action n'existe pas, nous affichons
un message d'erreur 404.
            $this->_headerError (404);
            exit;
        }
        //l'action existe, nous demandons son
exécution.
        $this->_processAction ($$action);
    }
}
?>
```

Voilà en quelques lignes de code comment peut fonctionner un routage simple de nos scripts PHP, avec un point d'entrée unique. Nous avons défini des actions, et ce sont elles qui vont déterminer par quoi les fonctionnalités attendues sont implémentées. Aujourd'hui, votre page d'accueil est une simple liste de news. Demain, alors que des lois menacent la pérennité des logiciels libres, vous voulez que votre page d'accueil se transforme en un message expliquant à tous pourquoi les brevets logiciels sont une mauvaise idée. Tout ce que vous avez à faire, c'est de définir une implé-

mentation différente pour votre action par défaut "default", sans pour autant modifier votre "vraie" page d'accueil.

Par exemple :

```
<?php
$default = & new FrameworkAction ('danger
OfPatents.php');
$action = & new FrameworkAction ('default
.php', array ('head'=>'header.htm'));
$redirect = & new FrameworkLink ('http://
copix.aston.fr');
?>
```

De cette manière, vous avez modifié la structure de votre site sans pour autant modifier le code de ce dernier. Autre exemple, vous avez développé un site sur l'actualité des logiciels libres. Pour gérer l'agenda des sorties des logiciels, vous avez développé en vitesse une fonctionnalité de "news du jour". Alors que le temps vous le permet, au bout d'un an d'existence de votre site, vous mettez à jour cette fonctionnalité par un vrai agenda. Toutefois, vous souhaitez recycler l'ancienne implémentation en une vraie gestion de nouvelles. Encore une fois, il vous suffit de modifier vos fichiers de déclaration. Par exemple, voici le fichier de description avant l'apparition de l'agenda :

```
<?php
$action = new FrameworkAction ('news.list
.php');
?>
```

Le voici après apparition de l'agenda.

```
<?php
$action = new FrameworkAction ('agenda.
php');
$news = new FrameworkAction ('news.
list.php');
?>
```

Résultat, les urls n'ont pas bougé et vous avez conservé tout le code existant.

Voyons maintenant comment implémenter notre fonction *\_processAction (\$action)*.

```
<?php
class Framework {
    /**
     * par convention, nous commençons les
noms de méthodes privées par un "_"
     * pour indiquer aux utilisateurs de notre
classe qu'il ne faut pas les utiliser directement.
     */
    function _processAction ($action){
        //nous demandons dans un premier temps
l'exécution des paramètres (tableau d'option)
```

```
$this->_processStartingOptions ($action->
params);
if (is_a ($action, 'FrameworkLink')){
    header ('Location:' + $action->url);
}
else if (is_a ($action, 'FrameworkAction')){
    require_once ($action->fileName);
}
//nous terminons par l'exécution des
actions de fin
$this->_processEndingOptions ($action->
params);
}
}
?>
```

En deux mots, cette méthode demande l'exécution des options de démarrage, demande l'inclusion du script qui implémente notre action puis demande l'exécution des options de fin de script. C'est dans nos méthodes *\_processStartingOptions* et *\_processEndingOptions* que nous allons tester les paramètres du tableau pour traiter les éléments.

Pour aller plus loin, il nous est possible de réaliser un système d'enregistrement des options. Au démarrage du framework, nous indiquerions que les options 'html' sont traitées par la méthode de notre choix, ou alors que l'option 'checkUser' est implémentée par une autre. De cette manière, nous serions arrivés à un système générique, ou toutes nos options seraient capitalisées au fur et à mesure de nos développements. Une fois une option développée, son implémentation est transportable de site en site, car parfaitement identifiable et indépendante. Je vous invite à télécharger les sources complètes des exemples à l'adresse <http://copix.aston.fr/articles/> pour les tester vous mêmes et les approfondir.

La suite au prochain numéro.

■ **Gérald Croes**

**Créée en 1990, Aston propose des prestations de conseil, assistance à maîtrise d'ouvrage, réalisation d'applications au forfait, en assistance technique, tierce maintenance applicative et formation.**

**Ses domaines de prédilection sont : le développement, l'intégration, le workflow, le décisionnel, les portails et Internet. Les résultats de sa cellule de veille technologique font l'objet de publications régulières dans la presse. Par ailleurs, Aston organise périodiquement des séminaires sur des sujets techniques d'actualité.**

Renseignements et inscriptions : [www.aston.fr](http://www.aston.fr)



# PhpDocumentor : L'usine à documentation

Nous avons vu le mois dernier quels étaient les bons réflexes à prendre pour commenter ses sources en vue de générer des documentations de qualité pour des applications PHP. Dans cette seconde partie, on va s'intéresser aux fonctions plus avancées de phpDocumentor, sur la base d'exemples concrets.

## Mise à jour

Premièrement, rappelons que phpDocumentor est un logiciel libre, dont le développement (très actif) est permanent. Aussi, il est préférable de mettre régulièrement à jour son installation. Pour ce faire, consultez de temps à autres l'adresse suivante : <http://phpdoc.org>, section "downloads". Nous considérerons dans cet article que phpDocumentor est installé à l'adresse <http://localhost/phpDocumentor>. Notre version de phpDocumentor est la 1.2.2.

## Sources de l'exemple

Afin de bien saisir le fonctionnement des différentes options, voici une partie des sources des deux fichiers sur lesquels nous allons travailler :

```
userManager.php :
<?
/**
 * Pseudo gestionnaire d'utilisateurs
 *
 * Programme factice, servant à illustrer les possibilités
 * de génération de documentations par phpDocumentor
 *
 * @package userManager
 * @author Gauthier Delamarre
 */
/**
 * Classe principale
 *
 * Cette classe va servir à manipuler les objets de type "user".
 */
class userManager {
    /**
     * Configuration générale
     * @var array
     */
    var $_CFG;
    /**
     * Constructeur
     *
     * @param string chemin d'accès au fichier de configuration
     */
    function userManager($cfgPath) { ... }
    /**
     * Ajout d'un utilisateur
     *
     * @param object Objet de type "user"
     * @return bool
     */
    function addUser($userObj) { ... }
    /**
     * Gestion d'erreur
     */
}
```

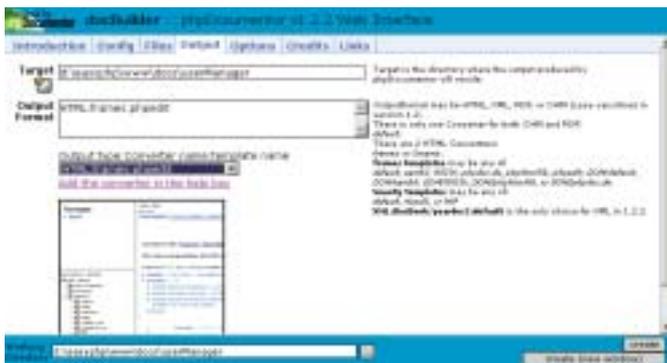
```
* @param string Message d'erreur à afficher
* @param bool Indique si l'exécution doit être stoppée ou non
*/
function raiseError($msg,$skill = TRUE) { ... }
}
?>

user.php :
<?
/**
 * Définition de la classe "user"
 *
 * @package userManager
 */
/**
 * Classe définissant les utilisateurs
 */
class user {
    /**
     * Créer un nouvel utilisateur
     *
     * @param string Nom de l'utilisateur
     * @param string Prénom
     * @param string Adresse email
     *
     * @return bool
     */
    function create($nom,$prenom,$email) { ... }
    /**
     * Charge les données utilisateurs depuis la base
     *
     * @param int Identifiant de l'utilisateur à charger
     * @return bool
     */
    function load($id) { ... }
}
?>
```

Dans notre environnement de test, ces fichiers se situent dans `d:\easy-php\www\userManager\`. Nous aurons besoin de l'indiquer à phpDocumentor pour qu'il puisse les scanner. Note : Vous pourrez télécharger les fichiers complets sur [programmez.com](http://programmez.com) pour effectuer les tests sur votre propre machine.

## Premier essai

Maintenant que nous avons installé le générateur de documentations, et que nous disposons de sources d'exemple, nous allons pouvoir créer notre première documentation. Pour cela, pointez un navigateur sur <http://localhost/phpDocumentor/>. L'interface web apparaît. Dans un premier temps, nous n'allons nous intéresser qu'aux onglets "files" et "output". Dans le premier onglet, un seul paramètre est indispensable : le chemin d'accès aux fichiers. Dans le second, seul le dossier où stocker la documentation générée est obligatoire. Vous pouvez cependant en profiter pour choisir un format et/ou un thème pour votre documentation. Le choix est assez riche, et il est possible d'en sélectionner plusieurs simultanément.



indiquer le chemin `d:\easyphp\www\dows\userManager` pour y stocker notre documentation. Il ne nous reste plus qu'à lancer le traitement des sources pour que les documentations soient créées aux formats demandés, en cliquant sur le bouton " create ". Pour consulter le résultat, pointez un nouveau navigateur sur <http://localhost/docs/userManager> :

Résultat de notre premier essai avec le convertir HTML:frames:DOM/earthli

*Note : si vous souhaitez essayer le convertir CHM (format d'aide Windows), l'utilitaire Microsoft Help Workshop (gratuit, à télécharger sur le site de Microsoft).*

## Tuning

Nous sommes maintenant parés pour nous pencher sur les options et fonctionnalités supplémentaires de phpDocumentor. Voyons d'abord l'onglet " options " de docBuilder.

Les options ici proposées sont assez simples à comprendre, car chacune est accompagnée d'un descriptif précis. Voici un résumé de ces options :

Custom tags	Liste de tags personnalisés que l'on veut que le parser prenne en compte
Parse @access private	Détermine la prise en compte des méthodes et propriétés privées.
Generate Highlighted Source Code	Ajoute à la documentation une copie colorisée des sources. De plus, chaque fonction ou classe sera liée à sa documentation, et chaque fonction PHP détectée renverra vers la documentation officielle en ligne !
JavaDoc-compliant Description Parsing	Active la compatibilité avec la syntaxe JavaDoc de descriptif des classes (la première phrase, i.e. jusqu'au premier point, correspond toujours à la description courte).
PEAR Package Repository parsing	Active la prise en compte des règles de codage spécifiques à PEAR (Chaque module est dans un répertoire propre, portant le même nom que le module, les méthodes et propriétés dont le nom débute par un _ sont considérées comme privées, et enfin, la méthode _NomDeLaClasse() est associée au destructeur de la classe.

## Types personnalisés

Maintenant, voyons comment phpDocumentor gère les types de données personnalisés, à savoir, en PHP, les objets. Dans notre exemple, une fonction attend en paramètre un objet de type user. Nous avons précisé avec le tag `@param` que l'argument doit être un objet, mais nous n'avons précisé son type que dans la description du paramètre. Sachez que phpDocumentor accepte en fait n'importe quelle chaîne pour qualifier un type d'argument, et qu'il est possible de commenter ainsi :

```
@param user Objet user à ajouter au gestionnaire d'utilisateurs
```

Sous forme textuelle, cela ne change pas grand chose, certes. Mais là où cela devient intéressant, c'est que dans un cas comme celui-ci, phpDocumentor fera directement un lien depuis " user " vers la définition de la classe correspondante !

## Renvois

Un autre tag essentiel est le tag `@link`. Celui-ci permet de générer dynamiquement des liens vers des pages spécifiques de la documentation. Vous pouvez l'utiliser dans les blocs de page comme dans les blocs de fonction, de classes ou même de variables. Exemple :

```
@param user Objet de type " user " (voir {@link user::create()} ou {@link user::load()})
```

Une autre syntaxe permet de réaliser le même type de liens, mais par le biais d'un tag à part entière, et non pas embarqué dans une description :

```
@see user::load(),user::create(),raiseError
```

Notez que la méthode `raiseError()` est indiquée telle quelle, sans spécifier sa classe, ni préciser qu'il s'agit d'une fonction (en ajoutant des parenthèses vides à la fin), car elle fait partie de la classe courante.

## Liste de tâches

Si vous souhaitez mentionner dans vos sources les tâches qu'il reste à faire, vous pouvez ajouter des tags `@todo`. Ceux-ci seront ensuite regroupés en une seule page par package, pour en consulter la liste d'un coup d'oeil. Exemple :

```
@todo Ajouter le support de PostgreSQL pour l'accès à la base.
```

## Fichiers README, INSTALL et CHANGELOG

Si phpDocumentor trouve un ou plusieurs de ces fichiers (sans extension), il en intégrera directement le contenu dans la documentation.

## Exemples dans les commentaires

Vous avez également la possibilité de placer des exemples dans vos commentaires. Vous avez pour ce faire deux méthodes. Soit vous spécifiez le chemin d'un fichier contenant l'exemple :

```
@example exemples/exemple1.php Exemple d'utilisation
```

Soit vous placez l'exemple dans le descriptif :

```
/**
 * Charge les données utilisateurs depuis la base
 *
 * Exemple d'utilisation :
 * <code>
 * $userObj = new user;
 * $userObj->load(12);
 * </code>
 * ...
```

Dans les deux cas, le code sera mis en couleur, et tous les termes renvoyant vers des fonctions, méthodes, variables ou constantes seront liés vers la page de documentation correspondante.

## Conclusion

Nous voici arrivés au terme de notre plongée dans le monde merveilleux de phpDocumentor, qui est véritablement un outil incontournable pour qui veut documenter rapidement et efficacement ses sources. Il vous reste à essayer les différents modèles, puis à consulter la documentation officielle de phpDocumentor ([www.phpdoc.org/manual](http://www.phpdoc.org/manual)) pour en tirer le meilleur parti.

■ Gauthier Delamarre

# Programmation d'un jeu 3D

4<sup>ème</sup> partie

## *Dynamique physique et pseudo intelligence artificielle*

Développer un jeu 3D pour Internet est toujours un pari difficile. Il faut pouvoir l'intégrer dans un navigateur Internet standard, ce qui nécessite une optimisation des processus classiques, notamment dans le domaine de la simulation physique et de l'intelligence artificielle.

Dans le domaine physique, nous avons vu dernièrement quelle technologie utiliser, en l'occurrence le moteur de simulation Havok. Nous avons précédemment opté pour le moteur Shockwave 3D, moteur développé par Intel et qui apparaît optimal pour la création de jeux 3D sur Internet.

Ce mois ci, nous abordons deux notions critiques pour le jeu : la dynamique physique du véhicule et un système de pseudo intelligence artificielle pour le *game engine*. Nous utiliserons le terme pseudo intelligence artificielle, car il ne nous sera pas possible, vu les contraintes de notre cahier des charges, d'intégrer un véritable moteur d'intelligence artificielle dans ce jeu. Nous allons donc proposer une solution intermédiaire basée sur un concept original et déterministe pour créer un semblant d'IA dans notre jeu.

Mais avant toute chose, terminons l'intégration de la dynamique physique de notre véhicule, à savoir le paramétrage et la programmation du moteur Havok pour les véhicules dynamiques du jeu.

### Dynamique des véhicules

La dernière fois, nous vous avons présenté le moteur Havok, un moteur de simulation physique très réaliste et performant et qui a la possibilité d'être utilisé dans notre *jeu online*. Ce dernier est assez complet et nous allons l'utiliser ici pour la programmation de la dynamique de nos véhicules.

Nous ne reviendrons pas sur l'installation, l'intégration et l'initialisation du moteur Havok dans notre jeu. En revanche, nous allons vous présenter comment nous programmons le moteur physique, de manière à ce que notre véhicule réponde dynamiquement dans l'univers virtuel ainsi créé.

Le module de programmation correspondant se nomme "module pilotage" et vous le trouverez dans le fichier *AI\_demo.dir* disponible en



Figure 1 : programme *AI\_demo.exe* en action.

téléchargement ici : [http://dev.net.online.fr/cs\\_4.zip](http://dev.net.online.fr/cs_4.zip). Ce fichier n'est pas bien entendu le code source final du jeu, il synthétise simplement les fonctions de simulation physique du véhicule dans un environnement 3D. Ce programme est donc un programme intermédiaire qui a la particularité de résumer les fonctions les plus importantes de notre *game engine*. Ouvrez donc le fichier *AI\_demo.dir* avec Director (9.0 de préférence, 8.5 minimum) et avec le package Havok pré-installé.

### Le module pilotage

Le module pilotage est un script Lingo fondamental, car il intègre tous les processus liés au calcul de notre exemple. Il permet en réalité de contrôler le comportement du modèle 3D utilisé pour le véhicule. Ce dernier doit être

capable d'accélérer, de décélérer et de tourner à gauche ou à droite. Le contrôle de la direction du modèle doit aussi être possible via les touches du clavier, en utilisant les flèches haut, bas, droite et gauche pour le pilotage du véhicule.

En parallèle à ces fonctionnalités, le modèle 3D doit aussi avoir une vitesse maximum et un cercle de rotation maximal. Il faudrait aussi pouvoir glisser et déraiper avec le modèle, tout en permettant le contrôle du véhicule et en optimisant le contre braquage pour corriger la trajectoire.

Bref, toutes ces caractéristiques font de ce comportement un élément essentiel et complexe à mettre en œuvre, probablement le plus complexe de toute l'étude de cas, car il doit reproduire au mieux le comportement d'un

véhicule soumis à des effets dynamiques. Heureusement, l'utilisation du moteur de simulation physique Havok devrait nous permettre de faciliter la mise en œuvre de toutes ces fonctionnalités.

Toutes les propriétés du comportement du véhicule doivent être paramétrées. Dans l'exemple *AI\_demo.dir*, nous avons ainsi défini un grand nombre de paramètres, comme la vitesse maximum, la vitesse de rotation maximale, l'accélération, la décélération, la puissance de freinage, la puissance de pivotement, le grip et la résistance aérodynamique.

Le module Pilotage intègre un gestionnaire d'événement utilisé pour activer les fonctions *getKey()*, *updateHover()* et *updateDrive()*. La première fonction permet de vérifier, avant chaque début de frame, si une ou plusieurs touches du clavier sont appuyées par l'utilisateur et en particulier les touches de direction du clavier. La seconde fonction, *updateHover()*, permet de vérifier le comportement et l'état des suspensions du véhicule. Ces caractéristiques vont bien entendu directement interagir sur le comportement global du véhicule, tout comme le feraient de vraies suspensions sur une voiture. La troisième fonction, *updateDrive()* permet d'appliquer la simulation physique sur le modèle 3D, en fonction des valeurs des propriétés définies préalablement.

Définissons à présent quelques propriétés et commandes du moteur de simulation physique. (figure 1)

## La restitution des corps physiques.

Il existe une propriété du moteur Havok correspondant à la "restitution des corps rigides". Cette dernière définit en réalité la capacité de rebondissement des corps après une collision. Nous avons choisi pour nos modèles 3D une valeur de 0.5, qui correspond en fait à une valeur de 50%. Ainsi, la perte d'énergie après chaque collision sera réduite de moitié.

Cette propriété permet d'initialiser ou de récupérer la restitution ou le rebondissement d'un corps rigide. La restitution représente la perte ou le gain d'énergie après la collision d'un objet. Si la valeur de restitution d'un objet est de zéro, alors toute l'énergie est perdue pen-

dant la collision et l'objet ne rebondira pas. En revanche, une valeur de 1 pour la restitution d'un objet génère un rebond parfait. Une valeur de restitution supérieure à 1 signifie qu'un objet rebondissant gagne de l'énergie après chaque collision. Ainsi, une balle rebondissante, atteindra une hauteur plus élevée après chaque impact avec le sol.

Ensuite, le moteur Havok nous permet de paramétrer le coefficient de friction des modèles 3D, soit leurs adhérences respectives. Le script poursuit avec le calcul des vitesses linéaires et angulaires courantes et récupère la masse des modèles 3D.



Figure 2 : trajectoire "optimale" possible (en rouge)

```
pChassisRB.restitution = 0.5
rbSurface.restitution = 0.5
rbChk.restitution = 1
```

## La commande "Impulsion"

Cette commande applique une impulsion sur le centre de gravité d'un corps rigide. Un vecteur Lingo spécifie les valeurs de l'impulsion. Un exemple pouvant définir cette impulsion serait celui d'une voiture tapant un mur et s'arrêtant immédiatement. Une impulsion, contrairement à une force, a un effet immédiat sur la vitesse du corps rigide et permet ainsi d'avoir un degré de contrôle sur l'objet.

Notre programme permet aussi d'accentuer le

ralentissement du bolide, un peu comme si un frein automatique s'enclenchait si le modèle n'avancait plus.

```
diff = pMaxSpeed - currentSpeed
- Apply an impulse proportional to the difference
- between our desired speed and our actual speed
imp = (diff * pAccGain * pChassisMass) *
worldForward * pPowerCoeff
pChassisRB.applyImpulse( imp )
```

## La commande "Pivotement"

Nous souhaitons également gérer le contrôle du modèle via les touches "gauche" et "droite", autrement dit, permettre que le modèle

puisse tourner à gauche ou à droite. Mais avant de poursuivre, nous devons avant tout vérifier que le modèle soit bien capable de tourner en fonction de propriétés liées à la vitesse courante et la vitesse de rotation maximale. Ensuite, nous pouvons gérer la rotation éventuelle du modèle à gauche ou à droite, en vérifiant que la variable locale *canTurn*, vérifiée dans le script précédent, a bien pour valeur booléenne TRUE et que bien sûr, les propriétés *pGoingRight* ou *pGoingLeft* sont elles aussi actives et sur TRUE.

Dans les deux cas (gauche ou droite), nous appliquons une petite impulsion linéaire (un cinquième de la valeur "normal") au modèle, avant de lui appliquer une impulsion angulaire qui lui permettra de pivoter sur son axe à gauche ou à droite. Cette impulsion linéaire

de faible amplitude permet de donner la force nécessaire au modèle pour tourner dans le cas où ce dernier serait à l'arrêt. Il est en effet impossible de faire pivoter un véhicule à l'arrêt, à cause de la force de gravité et d'autres paramètres comme l'adhérence. Mais nous souhaitons cependant activer cette particularité dans notre étude de cas.

```
- Check if we can turn.
- We check if the magnitude of our angular
velocity is less than our max -- turn speed
- Max Turning speed depends on velocity
mTS = min(abs(currentSpeed), (1.0 - propSpeed)
* pMaxTurnSpeed)
canTurn = (abs(angularVel.dot(worldAxis)) < mTS)
```

```

- If we can turn and we're turning left
if (canTurn and pGoingLeft) then
- Apply an angular impulse to turn us left
imp = worldAxis * (pTurnGain * pChassis
Mass) * pPowerCoeff / 2
if (pGoingBackward) then imp = -imp
pChassisRB.applyAngularImpulse( imp )
end if
    
```

La commande Havok *applyAngularImpulse* nous permet d'appliquer une impulsion angulaire au centre de gravité du modèle, qui doit pivoter à gauche ou à droite. Cette commande permet d'appliquer une impulsion angulaire sur un corps rigide au centre de sa masse. Un vecteur Lingo spécifie la valeur de cette impulsion. Une impulsion angulaire a un effet immédiat sur la vitesse angulaire de l'objet de la même façon qu'une impulsion linéaire aurait un effet immédiat sur la vitesse linéaire de cet objet.

## Les propriétés de glissement et de dérapage du véhicule

Ensuite, nous allons calculer dans quelle mesure le modèle peut glisser sur la droite ou la gauche, en fonction de paramètres physiques comme la vitesse, le grip et la masse du véhicule. Nous appliquons dans notre programme une forte compensation à cet effet, pour améliorer la maniabilité et réduire le dérapage du modèle, pouvant rendre le contrôle du véhicule trop difficile.

Le dernier script de ce comportement applique une impulsion angulaire pour compenser un excès éventuel de pivotement du modèle. Il applique ensuite une impulsion linéaire pour simuler la résistance aérodynamique liée à tout corps en mouvement dans un environnement quelconque.

```

- Work out how fast we're sliding left/right
- and compensate according to the grip property
- We just allow the car to slide if its braking
slideSpeed = worldRight.dot(currentVel)
if not (pBraking) then
imp = worldRight * (slideSpeed * pChassis
Mass * pGrip) * pPowerCoeff
- Apply an impulse to compensate for sliding
pChassisRB.applyImpulse( imp )
end if
- Apply an angular impulse to compensate for
spinning
if (canTurn=false) or ((pGoingLeft=false) and
    
```

```

(pGoingRight=false)) then
imp = worldAxis * angularVel.dot(world
Axis) * (pTurnGain * pChassisMass) * pPowerCoeff
pChassisRB.applyAngularImpulse( imp )
end if
    
```

## La gestion de la caméra

Le comportement de la caméra est le dernier comportement du moteur physique Havok que nous avons intégré dans le programme. Il permet de placer la caméra juste derrière le modèle 3D symbolisant le véhicule et le suit durant toute son évolution. Un gestionnaire d'événement va nous permettre de calculer la nouvelle position de la caméra, son orientation et son angle de rotation. Un autre gestionnaire d'événement permettra de récupérer la position de la cible (le point virtuel vers lequel va pointer la caméra) et la position de la caméra.

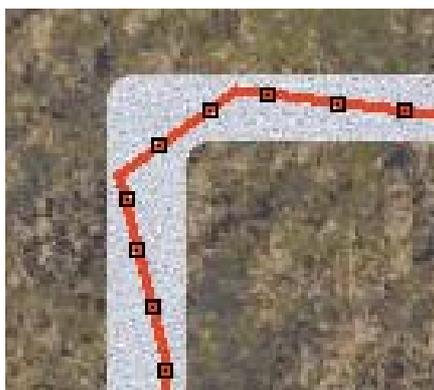


Figure 3 : la trajectoire est échantillonnée

L'étude du moteur de simulation physique est à présent terminée. Ce moteur est relativement complexe à mettre en œuvre, mais son apport dans l'étude de cas est fort appréciable, car il ajoute une dimension réaliste très importante. Il serait fort dommageable de s'en passer pour une étude de cas tridimensionnelle de cet ordre.

Les fonctions et comportements développés dans le programme exemple sont à peu de chose près identiques à celles que nous allons intégrer dans l'étude de cas finale. Cependant, quelques ajustements seront nécessaires pendant la phase d'intégration finale, mais l'architecture du moteur physique restera la même. Passons maintenant à l'étude de l'intelligence artificielle du jeu.

## Pseudo IA

Le second volet de cet article aborde le développement d'un module dédié à la génération

d'un programme d'intelligence artificielle. Nous n'avons pas les moyens de créer un véritable programme d'intelligence artificielle, comme ceux utilisant de la logique floue ou des réseaux neuronaux, aussi nous allons "ruser" en générant une banque de données de trajectoires pré-calculées, donnant l'impression que notre jeu intègre un véritable programme IA.

## Présentation

Mais avant de poursuivre la description du programme, nous allons vous présenter le concept sous-jacent à ce programme. Tout d'abord, il faut savoir qu'un programme IA, dans un jeu, détermine le comportement des "adversaires". Autrement dit, dans notre cas précis, le programme IA va déterminer le comportement des adversaires du joueur, à savoir certaines caractéristiques comme la rapidité, l'agressivité, la précision, la capacité à produire des erreurs, etc.

La solution que nous proposons pour ce jeu est simple. Nous devons préalablement déterminer une trajectoire optimale pour le circuit (figure 2).

Bien entendu, la trajectoire que nous proposons dans cet exemple n'est pas parfaite aux yeux d'un puriste de la course automobile. C'est pourquoi les conseils d'un consultant expert sont souvent utilisés pour pallier ce défaut dans les productions de jeux de course. Une fois la trajectoire optimale définie, nous allons échantillonner cette dernière, en fonction d'un frame rate précis, afin d'obtenir une liste de coordonnées spatiales, comme dans l'extrait décrit avec la figure 3.

La figure X montre une portion du circuit avec la trajectoire prévue en rouge. Cette trajectoire sera échantillonnée par notre utilitaire *AI\_demo.dir*. Autrement dit, notre programme va enregistrer les coordonnées (x,y,z) du véhicule ainsi que la rotation selon l'axe X, Y, Z du même véhicule. La fréquence d'échantillonnage sera déterminée par le *frame rate* du programme (soit la fréquence d'image par seconde). Ainsi, plus le FPS (*Frame per Second*) sera élevé, plus le nombre d'enregistrement sera important.

Les informations enregistrées sont partiellement visibles dans la fenêtre du programme (en bas à droite de la figure 4).

Bien entendu, le véhicule devra se déplacer sur la piste et suivre la trajectoire pendant l'enregistrement des données. Et c'est là que

réside l'astuce de cette méthode : la trajectoire n'est pas parfaite, mais elle est personnalisable. En réalité, pendant une session d'enregistrement, vous déplacez vous même le véhicule et vous êtes libre de suivre ou pas la trajectoire. De plus, vous pouvez enregistrer autant de sessions que vous souhaitez, ce qui vous permettra de créer un nombre de scénarios illimité. Prenez le cas d'un adversaire très dur à battre : vous enregistrez pour lui une session avec une trajectoire optimale et des paramètres optimaux (vitesse, agressivité, etc.). Dans le cas contraire, un adversaire très lent et peu adroit, vous pouvez enregistrer pour lui une session dans laquelle il commettra de nombreuses erreurs avec, par exemple, de nombreuses sorties de pistes. Bref, vous pouvez créer une multitude de contextes et de scénarios possibles, sans limite du nombre de courses et d'adversaires.



Figure 4 : information sur la session d'enregistrement du programme

A la fin de chaque session, (une session = une course = trois tours), vous avez la possibilité de cliquer sur le bouton " SAVE " pour sauvegarder toutes vos données dans un fichier texte ASCII externe (le fichier AI.dat). Ces informations seront ensuite stockées dans des fichiers de données intégrés au jeu final.

### Conclusion de la quatrième partie

La quatrième partie de ce dossier touche à sa fin. Vous pouvez retrouver tous les codes

sources de cet article en téléchargeant le fichier [http://dev.net.online.fr/cs\\_4.zip](http://dev.net.online.fr/cs_4.zip). La prochaine fois, nous vous montrerons comment sauvegarder les scores et les chronos des joueurs. Nous traiterons en premier lieu la sauvegarde des chronos sur disque dur (avec tri automatique en fonction du score) et la sauvegarde en ligne, pour ceux qui disposent d'une connexion Internet et qui désirent comparer leurs chronos avec d'autres joueurs.

■ Laurent Jayr - [dev.net@skeddio.com](mailto:dev.net@skeddio.com)