

DÉVELOPPEUR

RÉFÉRENCE

LA LETTRE
BIMENSUELLE
DU DÉVELOPPEMENT

www.devreference.net

v2.03 | 9 novembre 2001

EDITO



Les services Web portent la promesse pour les développeurs de réaliser un vieux rêve : faire dialoguer les applis entre elles, même entre des mondes aussi éloignés que .NET et J2EE. Mais déjà la dissidence commence à gronder dans les rangs du clan Java, censé se serrer les coudes face au géant de Redmond. Gosling dénigre SOAP, un des fondements des services Web. Pour le père de Java, il n'existe qu'une seule plateforme pour l'interopérabilité : J2EE. IBM conteste la pertinence de JMS dans J2EE 1.3 et s'affiche de plus en plus avec Microsoft. Les membres de la coalition sourient autour de la table du JCP mais se tirent dans le dos en coulisses. Car si les services Web sont une technologie ouverte, où gagner de l'argent ? Déjà Sun avec iPlanet, IBM avec Eclipse, HP avec HP-AS sont prêts à donner leurs outils et leurs serveurs d'applications pour attirer les clients vers leurs technologies. L'histoire des tentatives d'un Unix unifié semble se répéter. Décidément, les intérêts du marketing et des développeurs sont rarement compatibles. Et pendant ce temps-là, malgré ses procès, Microsoft continue son petit bonhomme de chemin...

Pierre Tran
Rédacteur en chef

Développement objet

Best Practices

SI LES NOUVELLES TECHNOLOGIES FONT BRILLER LES YEUX DES DÉVELOPPEURS,
LE CHEF DE PROJET SE TROUVE QUANT À LUI EN PROIE À DE NOMBREUSES INTERROGATIONS :
MON PROCESSUS DE DÉVELOPPEMENT EST-IL DÉPASSÉ ? COMMENT GÉRER LA COMPLEXITÉ
TECHNOLOGIQUE ? COMMENT ORGANISER MES ÉQUIPES ?

En nous appuyant sur nos retours d'expérience, nous montrerons comment adapter son processus de développement de façon à concilier qualité des réalisations et complexité technologique. Cet article s'adresse à des chefs de projet toutes expériences confondus.

Mon processus de développement est-il dépassé ?

La construction du macro-planing est une étape clé du projet : le chef de projet définit le processus de développement (figure 1), composé d'une succession de phases, et

propose pour chacune d'entre elles une durée, des moyens à mettre en œuvre, des documents à produire et des profils d'intervenants. Le succès du projet dépend, dès lors, de l'adéquation du projet au processus de développement.

[suite page 8]

Sommaire

Actualité

Événements, tendances, nouveaux produits 2

Du côté de chez Borland .. 3
Compte-rendu de la conférence française et retour d'expérience sur Kylix 1.0.

Microsoft PDC 2001 5
La Professional Developer's Conference Microsoft s'est tenue du 22 au 26 octobre à Los Angeles.

Méthode

Développement objet Best Practices 8
Intégrer les nouvelles technologies tout en conservant la qualité des réalisations est une des problématiques des chefs de projet aujourd'hui. Voici quelques bonnes pratiques, issues de retours d'expérience, à partager sans modération avec tous les membres de l'équipe de développement.

Programmation

Programmation par contrat 13
Les contrats répondent aux exigences de qualité logicielle dans le contexte d'architectures complexes.

Outil

Lotus Domino Rnext bêta 2 17
La prochaine version de Lotus Notes/Domino passée en revue.



Du côté de chez Borland

MOINS RICHE EN CONTENU TECHNIQUE QUE CELLE DE SA MAISON MÈRE, LA CONFÉRENCE ANNUELLE DE BORLAND FRANCE (BORCON), ORGANISÉE LE 8 ET 9 OCTOBRE DERNIERS, A CEPENDANT PERMIS DE PRENDRE LA TEMPÉRATURE DE L'ÉDITEUR SUR LE MARCHÉ FRANÇAIS.

Par Emmanuel Fayet*

Par rapport à la conférence Borland US qui s'est déroulée cet été à Long Beach en Californie, le discours est ici à Paris sensiblement plus marketing que technique, avec un effort de décryptage des nouveaux concepts mis en œuvre dans les produits Borland.

Citons par exemple le concept de services Web, une idée qui fait clairement son chemin dans les esprits. Différents scénarios d'utilisation de ces services applicatifs apparaissent : projets d'e-commerce,

mais aussi intranet de grandes entreprises. Philippe L'Heureux-Bouron, informaticien à la banque BNP-Paribas détaille son scénario : « Nous disposons en interne du portail d'entreprise EIP d'Hummingbird. Nous voulons maintenant l'enrichir avec des outils en ligne de gestion d'annuaire ou de réservation de salles. Le service Web m'intéresse car il me semble un standard simple d'intégration. » L'idée est séduisante. Il reste cependant à attendre que les outils de développement suivent et facilitent l'utilisation ou la création des services Web. C'est aujourd'hui

le cas de Delphi 6 et de Kylix 2.0, et des prochaines versions de JBuilder.

Du côté de Kylix, la situation est contrastée. Nous sommes maintenant six mois après la sortie en France de ce Delphi pour Linux. Mais les développeurs semblent toujours dans la phase de prise de contact avec Linux et de l'évaluation des conditions de portage du code Delphi vers Kylix. Un premier développeur Delphi regrette l'absence de générateur d'états sous Kylix, un autre celle d'un système de gestion d'aide. Les premières applications Kylix ne viendront pas forcément du

portage d'applications existantes mais sans doute de nouveaux développements ou de solutions

mixant l'utilisation de Delphi et de Kylix (voir à ce sujet notre entretien avec Alexandre Guillien, de la société Scripto). Présent à la conférence, Jean-Pierre Le Blanc, directeur du développement de la division RAD de Borland, annonçait le sortie de Kylix 2.0 pour le 23 octobre. Cette nouvelle version apporte le support des services Web, ainsi qu'une mise à niveau des classes de la bibliothèque de Kylix sur celle de la dernière version de Delphi. Nous ne manquerons pas de suivre dans les prochains mois cette incursion de Borland sur ce marché du développement rapide d'applications pour Linux.

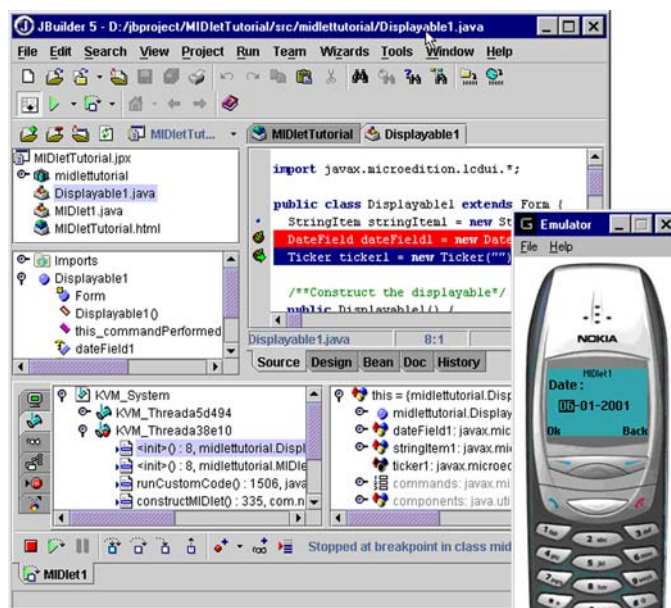
Notons enfin plusieurs sessions animées par des entreprises ayant mis en œuvre des solutions basées autour des outils Borland. La société Lapeyre, par exemple, présentait son site Web d'e-commerce basé sur le serveur d'applications Java Application Server 4.5 et un dialogue avec son réseau de magasins au travers d'un service d'envoi de messages JMS (Java Message Service). Autre exemple, la société de services Dassault Data Services dévoilait une architecture originale basée sur une approche par composants, mise en œuvre avec Delphi dans le cadre du développement d'un logiciel de gestion de bancs de tests et de mesures réalisés pour Alcatel Optronic.



JBuilder 5, le passage au j-business

Présenté lors de la conférence US en juillet, JBuilder 5 est disponible pour Windows, Linux et Solaris. Il inclut le moteur Tomcat 3.2 pour la génération de pages Web dynamiques selon les technologies JSP 1.1 et Servlet 2.2. Mais surtout, il facilite le développement d'objets Java respectant la norme EJB et assure le déploiement automatique vers un serveur d'applications compatible J2EE, dont Application Server 4.5 de Borland mais aussi WebLogic 6.0 de BEA et WebSphere 3.5 d'IBM.

Fort de sa position de leader sur le marché des outils de développement Java, Borland a engagé une série de partenariats : Entreprise Studio for Java regroupe JBuilder et Application Server avec l'outil de modélisation de classes Rational Professional pour JBuilder de Rational Rose et l'éditeur de sites Web Dreamweaver UltraDev de Macromédia.



Ici, JBuilder Mobile Set Edition Nokia en action : le débogage d'une application Java, destinée à fonctionner sur un téléphone Nokia, se fait dans JBuilder. L'émulateur fourni par Nokia sert aux tests de l'application dans la phase de développement.

* Emmanuel Fayet est fondateur de la société Ecoswitch, spécialisée dans la formation et l'appui technique aux développeurs. Il collabore régulièrement à DÉVELOPPEUR RÉFÉRENCE et peut être joint à l'adresse efayet@ecoswitch.fr

Kylix 1.0 : premier retour d'expérience chez Scripto

Propos recueillis par Emmanuel Fayet

Alexandre Guillien est chef de Projet à l'agence lyonnaise de Scripto, une société spécialisée dans l'ingénierie et le traitement de l'information. Il détaille pour nous l'architecture du logiciel Scripto Meta-Portal de gestion de portail Internet et nous livre son retour d'expérience suite à l'utilisation de Kylix dans le cadre de ce développement.

DÉVELOPPEUR RÉFÉRENCE : Pouvez-vous nous présenter en quelques mots Scripto ?

Alexandre Guillien : Le groupe Scripto est spécialisé dans la reprise d'existant et le traitement informatique des données des entreprises. Cela comprend autant la numérisation que la frappe, la conversion, la mise en base de données et la mise à disposition par des applications client-serveur ou multitières en intranet-extranet ou Internet. Nous intégrons toute la chaîne allant de la production de documents à leur présentation sur Internet.

Scripto dispose d'une offre de portail Internet. Quelles sont les composantes techniques de votre offre ?

Nous avons bâti une solution mixte Linux-Windows et adopté un principe de séparation du contenu et de la présentation des informations du portail.

Le cœur du portail tourne sur une machine Linux qui héberge un serveur Web Apache, ainsi qu'une base de données MySQL. Cette base contient tous les éléments de contenu et de présentation selon une structure bien définie. La génération des pages Web se fait dynamiquement par l'intermédiaire d'un script PHP qui fusionne les éléments de contenu et de présentation.

La partie édition et mise à jour des données du portail s'effectue à l'aide d'une application cliente Windows. Cette application, écrite en Delphi 5, accède aux données par le biais d'un serveur d'applications écrit en Kylix. Ce serveur d'applications sert ainsi d'intermédiaire transparent entre la base de données et le logiciel client. D'un point de vue technique, le dialogue entre le client Delphi et le serveur Kylix baptisé SAS (Simple Application Server) se fait en utilisant les composants TCP/IP Indy et en échangeant des paquets de données au format XML.

Le premier atout de cette approche 3 tiers est de conserver une

ergonomie de logiciel Windows pour la partie édition du portail. En effet, l'idée d'une interface Web d'édition a été rapidement abandonnée du fait des limitations importantes imposées par cette plate-forme. L'autre atout : ce sont bien évidemment les technologies légères (Linux, MySQL, Apache, PHP) utilisées pour le côté serveur du portail.

Donc côté Linux, votre serveur écrit en Kylix réceptionne les éléments de contenu et insère ces éléments dans la base MySQL. Avez-vous rencontré des difficultés pour alimenter cette base à partir de Kylix ?

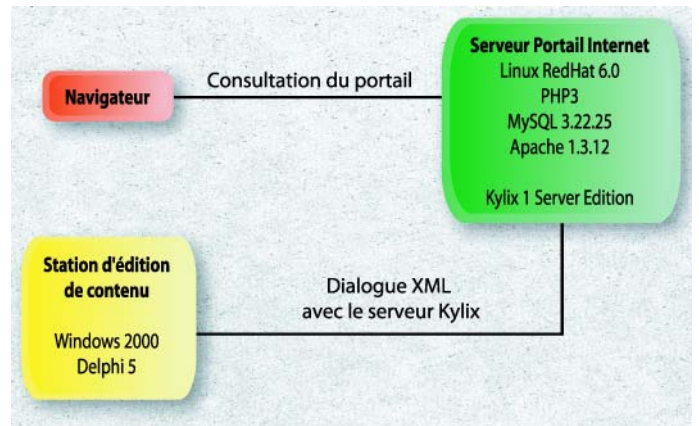
Oui et non. Nous pensions utiliser le nouveau framework de Borland pour l'accès aux données : DBExpress. Mais le driver DBExpress pour MySQL fourni avec Kylix posait de gros problèmes dans la gestion des champs de type BLOB avec des plantages aléatoires de l'application. Ces champs BLOB sont pour nous essentiels car c'est là que nous stockons les données HTML. Il s'agit sans doute d'une erreur de jeunesse mais nous avons été forcés d'abandonner cette voie. Nous avons alors trouvé sur Internet une suite de composants Delphi donnant un accès natif à MySQL au travers des fichiers DLL du client standard de cette base.

Nous avons porté ces composants vers Kylix. Au final, nous avons perdu plus de trois jours à essayer de faire fonctionner le couple DBExpress/MySQL alors qu'en moins d'une journée la librairie MySQL native était convertie et opérationnelle !

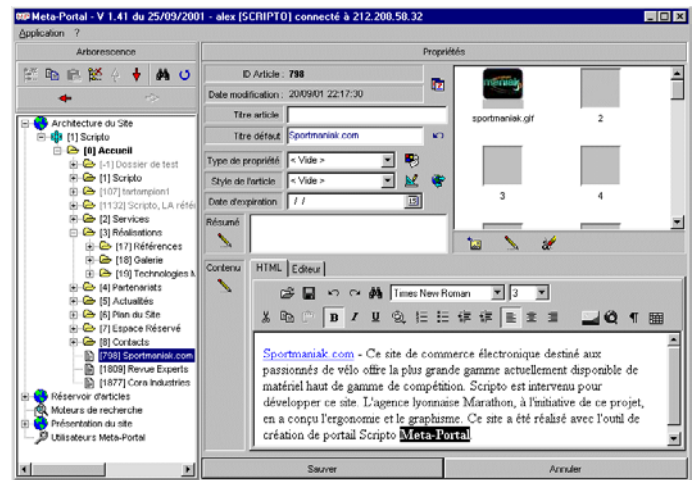
Quel jugement portez-vous sur Kylix à la lumière de cette première expérience ?

Comme je vous l'ai dit, les principaux problèmes que nous avons rencontrés venaient de DBExpress/MySQL. D'autres petits bugs inattendus nous ont déçus, mais ces derniers étaient contournés rapidement. Une grande frustration est aussi venue de nombreux raccourcis clavier qui ne fonctionnent pas sous KDE. En revanche, et malgré les mises en garde de Borland, il est intéressant de noter que nos développements ont fonctionné sous une RedHat 6.0 sans le moindre souci.

En fait, les difficultés portent plus sur la méconnaissance des spécificités de Linux que sur le code en lui-même. Pour tout dire, la conversion



Architecture de la solution Meta-Portal de Scripto avec une partie édition de contenu sous Windows et une autre partie de publication Web du contenu sous Linux.



Un rédacteur est en cours d'édition d'un article à paraître sur le site de Scripto.

de projets d'un système à l'autre se fait de manière quasi transparente : 80 % de nos composants sont passés sous Linux en un temps record. Dans le cadre de notre projet, nous n'avons pas poussé à bout la conception d'interfaces graphiques de Kylix. Nous n'avons eu besoin que de quelques boutons et de mémos pour nos tests...

Avez-vous des recommandations à faire à vos collègues développeurs souhaitant démarrer avec Kylix ?

Je dirais que Kylix et Delphi sont dans l'utilisation courante deux produits à l'ergonomie et aux fonctionnalités identiques : c'est vraiment bluffant ! Quiconque sait utiliser

Delphi est aujourd'hui en mesure de développer sous Linux avec un minimum de formation.

ALLER PLUS LOIN

Pour visiter des sites réalisés avec Scripto Meta-Portal :

- <http://www.scripto.fr>
- <http://www.sportmaniak.com/>
- <http://www.revue-experts.com/>

La librairie VG MySQL pour Delphi, portée sous Kylix par Scripto, est disponible sur le site de son auteur Vladimir Gaitanoff :

<http://www.vglib.com/link-5.html>

Événement

Microsoft PDC 2001

CETTE ANNÉE, PAS DE SURPRISE, LA PDC (PROFESSIONAL DEVELOPERS CONFERENCE) 2001 A MIS L'ACCENT SUR .NET ET LES SERVICES WEB XML.

PARMI LES GRANDES ANNONCES, ON RETIENDRA LA DISPONIBILITÉ DE VISUAL STUDIO .NET RELEASE CANDIDATE.

Par Pierre Tran

La PDC est sans aucun doute l'événement le plus important avec la JavaOne dans le monde des développeurs. Pour Microsoft, c'est le lieu privilégié pour présenter et détailler ses toutes dernières technologies. C'est ici que se dessinent les grandes tendances logicielles voire matérielles des prochaines années. La PDC, c'est aussi l'occasion d'annoncer des produits majeurs pour les développeurs.

Comme de coutume, la PDC a officiellement démarré par le discours inaugural de Bill Gates devant un parterre de 6 800 participants. Bill Gates a également débuté son show par une vidéo humoristique, la journée d'un architecte en chef à Redmond. Puis il a présenté les grands lignes stratégiques de la firme, les annonces produits et sa vision du futur, le tout entrecoupé de démonstrations des nouvelles technologies.



La nouvelle architecture .NET intègre la notion de services.

Les outils arrivent

Il y a un peu plus d'un an, la PDC dévoilait pour la première fois la plate-forme .NET, son architecture, son modèle de programmation. Pour ceux qui y étaient, ce fut un grand choc. Cette année, pas de révélations fracassantes mais une concrétisation des annonces faites tout au long de l'année. Le modèle de programmation prend consistance, et les outils de développement sont maintenant là. La plate-forme s'enrichit de clients, de serveurs et de services. Le concept de développement client-serveur évolue vers celui du client-serveur-services.

Nouveau cette année, le concept de "services Web" prend l'appellation "services Web XML" (XML Web services). Une précision qui s'impose face à l'arrivée des services au sens applicatifs que sont .NET My Services, et qui sont eux construits à partir de services Web XML au sens composants. Une appellation qui par ailleurs renforce l'image d'adoption généralisée du standard XML par Microsoft, et où on peut y voir une subtile tentative marketing d'appropriation du concept de service Web basé sur les standards XML.

La première grande annonce concerne la disponibilité de Visual Studio .NET et du .NET Framework, tous deux en version Release Candidate. Le .NET Framework est la plate-forme multilangage qui offre tout ce qu'il faut pour construire, déployer et exécuter les services Web et les applications .NET : la CLR (Common Language Runtime), les bibliothèques de classes du framework .NET... Par rapport à la version bêta 2, la RC de Visual Studio ne contient aucune nouvelle fonctionnalité, elle n'apporte que des corrections de bogues. Cette version expire le 28 février 2002. La version finale est prévue pour la fin de l'année 2001. Destinée aux architectes et chefs de projet, la version Enterprise Architect comprenant la technologie de modèles d'entreprise n'était pas hélas disponible à la PDC.

.NET pour tous les clients

La mobilité sera une des tendances fortes de cette PDC. Côté clients .NET, on voit émerger les smart devices (téléphones, PDA...). Ainsi deux produits dans cette direction

sont annoncés. Tout d'abord le .NET Compact Framework, un sous-ensemble du .NET Framework destiné aux appareils sous Pocket PC et Pocket PC 2002 (et par la suite au futur Windows CE "Talisker"). Ce Compact Framework permet aux appareils légers avec peu de mémoire de faire tourner du code managé et des services Web. Concrètement, cela veut dire que l'on pourra faire du peer-to-peer entre smart devices grâce à la technologie de services Web .NET. Autre produit annoncé, les Smart Device Extensions (SDE) for Visual Studio .NET s'intègrent à Visual Studio et permettent aux programmeurs habitués à cet environnement de développer, de déboguer et de déployer des applications pour les smart devices.

La future génération de Windows CE (nom de code "Talisker") s'appellera Windows CE .NET et supportera en standard le .NET Compact Framework et les services Web XML. A noter également la disponibilité d'un Mobile Internet Toolkit facilitant le développement d'applications Web à destination des téléphones Wap/cHTML, des pager et des PDA, et capables ainsi de consommer des services Web.

La future version de Windows CE supportera les services Web en natif.



Les annonces

Versions finales

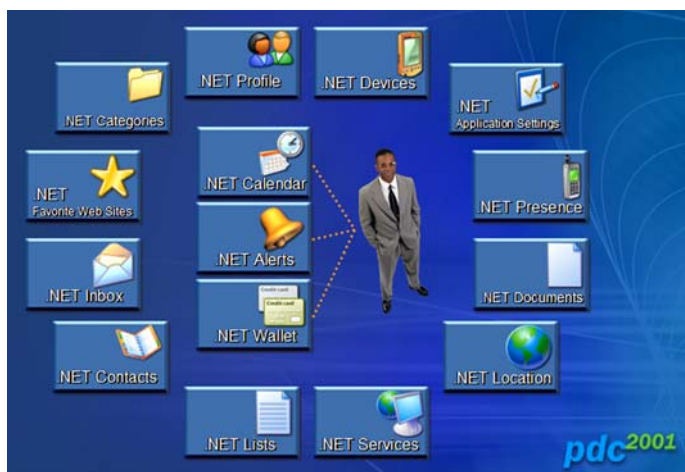
- Windows XP
- XML Core Services v4.0
- SQLXML 2.0 for SQL Server 2000

Release Candidates

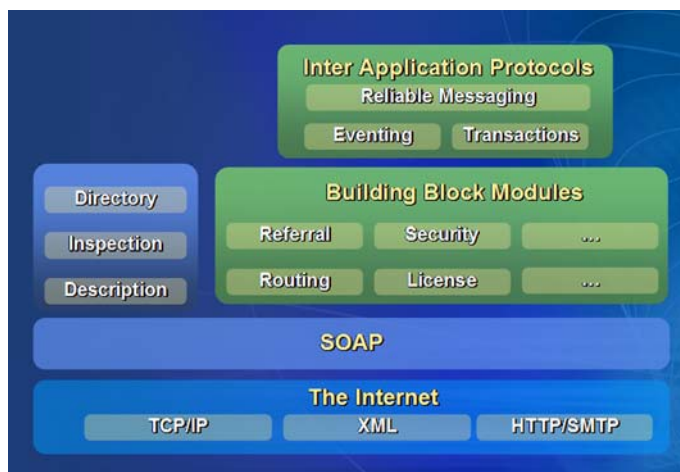
- .NET Framework
- Visual Studio .NET

Technology previews

- .NET Compact Framework
- Smart Device Extensions for Visual Studio .NET
- Tablet PC SDK
- .NET My Services SDK
- .NET Alerts SDK
- Office XP Web Services Toolkit
- Commerce Server 2002
- .NET Speech SDK



.NET My Services est une collection de services Web centrés sur l'utilisateur et fédérés par Microsoft



Global XML Web Services Architecture (GXA), une proposition IBM/Microsoft, se propose d'étendre les services Web.

A ranger dans les clients également : Windows XP, dont le lancement mondial tombait, comme par hasard, en plein milieu de la PDC. Nous avons eu droit à la retransmission en temps réel du lancement à New York, cérémonie un peu pompeuse avec chant gospel et show à l'américaine dont nous aurions pu nous passer. Malgré la disponibilité d'un Visual Studio .NET Toolkit pour Windows XP permettant d'exploiter les nouvelles fonctionnalités de ce système, la présence de cette annonce au beau milieu de la PDC nous a semblé superflue. Mais les participants sont repartis contents avec une boîte de Windows XP sous le bras et convaincus que MS-DOS était bel et bien mort. Il faut cependant garder à l'esprit que Windows XP constitue une brique essentielle de la stratégie .NET, puisqu'il se destine à être le premier client .NET déployé à grande échelle, dans ses différentes incarnations (domestique, professionnelle, serveur et embarquée).

Orchestrer des services Web

Côté serveurs .NET, Microsoft annonçait un prototype de BizTalk Orchestration for Web Services. Ce composant de BizTalk Server 2000 se présente comme une suite pour orchestrer, exécuter et gérer les services Web .NET, le tout dans un environnement de conception visuel. Les diagrammes de règles métier ainsi conçus peuvent être compilés en un fichier exécutable XLANG, le langage XML de définition de règles métier spécifique à BizTalk. Combinés aux services Web, les planifications XLANG permettent d'agréger ces services Web pour former une règle métier et créer un service Web frontal pour ces règles métier. La prochaine génération de XLANG,

appelée XLANG/s, permettra de créer des scripts de règles métier compilés en assemblage .NET qui pourra s'exécuter dans la CLR.

Vers une société de services

L'autre grande annonce, détaillée lors de la conférence de Bob Muglia (vice-président .NET Services Group), concernait le lancement de la plate-forme de développement pour .NET My Services (anciennement HailStorm), une collection de services Web XML fédérés par Microsoft. .NET My Services est un ensemble de services applicatifs en ligne, centré sur l'utilisateur, permettant de stocker des informations telles que son profil personnel, ses contacts, ses listes, son agenda, sa géolocalisation, ses documents, ses paramètres d'applications, ses favoris... La plupart de ces services sont encore en bêta, excepté Passport qui est opérationnel depuis 1999. Ce service d'authentification unique, supporté par 75 partenaires, revendique 200 millions d'utilisateurs, avec 3,5 milliards d'authentifications par mois. Les autres services sont prévus à terme pour l'horizon fin 2002, avec des versions bêtas intermédiaires courant 2002.

Pour construire ces services, Microsoft annonçait pour la PDC trois produits : .NET My Services SDK, .NET Alerts SDK et Passport SDK. Le service .NET Alerts permet d'utiliser le canal MSN Messenger (Windows Messenger sous XP) pour afficher des alertes préprogrammées sur le poste ou le téléphone de l'utilisateur. On pourra par exemple recevoir une alerte de eBay à chaque fois qu'une surenchère est faite sur un objet que l'on suit. C'est un peu le grand retour du push, mais totalement programmable.

Les trois services de base Passport, .NET Alerts et .NET Presence resteront gratuits pour les utilisateurs. En revanche, le modèle économique derrière les autres services prévoit de faire payer à la fois les fournisseurs de services et les utilisateurs. Pour les développeurs, la plupart des outils pour développer des .NET My Services seront gratuits. Les participants à la PDC ont par ailleurs reçu un ouvrage de 500 pages sur les spécifications .NET My Services.

.NET My Services révèle un nouvel aspect de l'activité de Microsoft, qui s'étend désormais vers un modèle de société de services applicatifs en mode locatif.

Une moisson de SDK

Pour compléter le tour d'horizon des annonces faites lors de la PDC, citons enfin :

- SQLXML 2.0 for SQL Server 2000, un ensemble de composants pour étendre le support de XML par SQL 2000 Server ;
- XML Core Services v4.0, un ensemble de services (parseur XML, moteur XSLT, support de XML Schema...) permettant aux développeurs d'ajouter le support de XML dans leurs programmes sous Visual Studio .NET ;
- Commerce Server 2002 Technology Preview, outil pour développer des sites de commerce, s'intègre dans Visual Studio .NET et supporte les composants du .NET Framework ;
- Office XP Web Services Toolkit, pour intégrer des services Web dans les solutions Offices XP développées en VBA (disponible à la fin de l'année) ;
- .NET Speech SDK Technology Preview, un kit pour développer des

extensions de contrôle vocal dans les applications .NET via des PC, des téléphones ou des PDA.

Les technologies de services Web XML prennent de l'ampleur et les problématiques d'interopérabilité et de sécurité commencent à se faire pressantes. Pour faire avancer le schmilblic, Microsoft propose à la standardisation une architecture pour les services Web baptisée Global XML Web Services Architecture (GXA), composée de spécifications destinées à étendre les fonctionnalités de SOAP et des services Web. Ce sont quatre spécifications qui sont annoncées à la PDC, axées sur la sécurité et le routage. La sécurité tout d'abord, avec WS-Security, qui décrit comment utiliser certaines spécifications W3C telles que XML Signature et XML Encryption, et WS-License pour associer les créances aux messages SOAP. Le routage d'autre part, avec WS-Routing (anciennement SOAP-RP), qui décrit comment placer les informations d'adresses de routage dans les messages SOAP, et WS-Referral qui permet le routage dynamique de messages entre des nœuds SOAP. Ces spécifications, élaborées en collaboration avec IBM, ont leur origine dans l'initiative Web Services Framework déjà présentée en juillet 2000 à la XML DevCon et soumise en avril 2001 au W3C.





Le PC du futur

Une des attractions de cette PDC fut la présentation de la technologie Tablet PC, ou PC ardoise. Le matériel, encore sous forme de prototype, se présente comme un écran LCD sur lequel on écrit avec un



stylet, tel un portable auquel on aurait enlevé la partie clavier/UC. A l'intérieur, il s'agit d'un véritable PC tournant sous un Windows XP, un surensemble de XP intégrant des fonctions de reconnaissances d'écriture et vocale. Un vieux projet que Microsoft tente de sortir de ses cartons périodiquement mais qui aujourd'hui, grâce à la puissance des machines, commence à se concrétiser. Les démos sont d'ailleurs assez bluffantes et permettent d'entrevoir des applications novatrices pour les professions itinérantes. Microsoft annonçait à cet effet un Tablet PC Beta SDK pour engager les développeurs à innover dans cette voie.

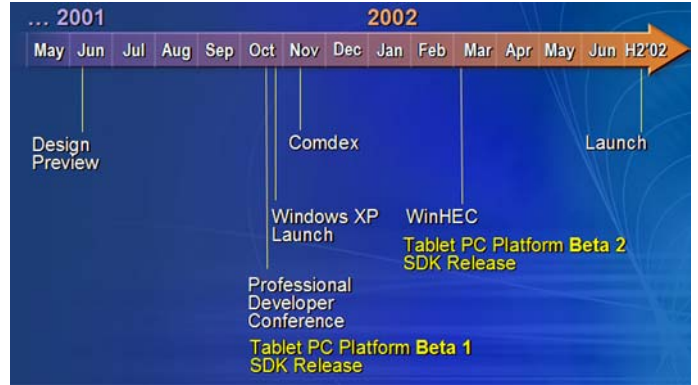
Vers un monde plus .NET

Outre les conférences plénières, la PDC c'était aussi un ensemble de 130 sessions techniques, des ateliers, un hall avec 90 exposants, une organisation et une logistique exemplaires : 600 machines étaient à la disposition des participants et le site disposait d'un réseau sans fil 802.11b permettant aux portables et aux Pocket PC de se connecter au Net de n'importe où. Les participants sont repartis avec six CD contenant les produits annoncés et le plein d'informations techniques.

Ce que l'on retiendra de cette cuvée 2001, c'est l'accent mis sur la mobilité et les services applicatifs, ainsi que la disponibilité des outils.

	2001	2002	2003
Clients	Windows XP Pocket PC Smart phone	Tablet PC	Windows "Longhorn"
Serveurs	Windows 2000 .NET Enterprise Servers	Windows .NET Server .NET Enterprise Servers 2002	Windows "Longhorn" Server .NET Enterprise Servers 2003
Services	.NET Passport .NET Alerts	.NET My Services	.NET My Services+
Outils	Visual Studio 6.0 SOAP Toolkit	Visual Studio .NET	Visual Studio .NET+

Roadmap de la plate-forme .NET.



Roadmap Tablet PC.

Le géant de Redmond poursuit sa stratégie à long terme visant à imposer .NET partout ("anytime, anywhere, on any device"). Pour ce faire, l'éditeur avance avec une tactique ouverte, la promotion des services Web XML avec comme message : la meilleure plate-forme pour développer des services Web, c'est .NET. Et pour y parvenir, Microsoft dispose d'une arme redoutable auprès des développeurs : Visual Studio .NET.

ALLER PLUS LOIN

Le site de la PDC 2001

msdn.microsoft.com/events/pdc/

Les annonces de la PDC 2001

www.microsoft.com/presspass/events/pdc01/material.asp

.NET My Services Community beta site

www.betaplace.com

Web Services Framework

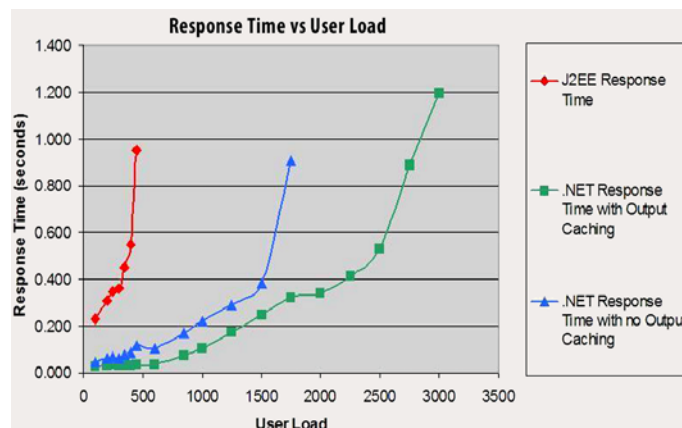
www.w3.org/2001/03/WSWS-popa/paper51

Mon .NET est plus rapide que ton J2EE

Annoncée durant la PDC, une étude comparative entre .NET et J2EE a été publiée sur le site GotDotNet. Cette étude, conduite par la société Vertigo, comporte cinq volets. Le premier est une comparaison entre l'application Pet Store sous J2EE et son implémentation .NET. Rappelons que Pet Store est l'application de référence qui montre comment implémenter les technologies J2EE. Résultat : l'implémentation .NET utilise trois fois moins de lignes de code. Le deuxième volet effectue des benchmarks plus précis sur les deux implémentations, notamment en termes de temps de réponse, d'utilisation CPU et de montée en charge. Ici, le résultat est spectaculaire : .NET est 28 fois plus rapide ! Le troisième volet montre que pour développer des services Web, .NET est plus productif que IBM WebSphere (deux fois moins de code, 24% moins de temps, un seul outil au lieu de six...). Quatrième comparatif : le benchmark de l'application de

e-commerce Nile @Bench révèle que la version ASP.NET est trois ou quatre fois plus rapide que la version EJB/J2EE (sans ou avec cache). La dernière étude détaille l'implémentation du benchmark Nile avec .NET et WebSphere 4.0 mais sans résultat, afin que les utilisateurs puissent faire leurs propres mesures.

Les réactions au comparatif Pet Store ne se sont pas fait attendre dans la communauté Java, réactions parfois épidémiques. Par le passé, Microsoft a souvent tenté de manipuler l'opinion avec des FUD (Fear, Uncertainty and Doubt, peur, incertitude et doute). Mais ici, les chiffres sont là, et tout un chacun peut les vérifier.



Pour autant, on peut s'interroger sur la pertinence de certains comparatifs. L'application Pet Store n'a jamais été prévue pour servir de benchmark, mais plutôt de tutorial pour implémenter J2EE. A un niveau atomique, cela reviendrait à faire des benches sur le programme "Hello World" ! Par ailleurs, l'implémentation .NET utilise à son avantage des astuces telles que les procédures stockées qui faussent les résultats.

Mais la véritable question est : ces comparatifs ont-ils vraiment un sens ? Car le choix de Java n'a jamais été dicté par les performances mais l'interopérabilité et l'indépendance vis-à-vis des fournisseurs. Peut-on comparer un train et une voiture ? Toujours est-il que Microsoft dispose là d'une arme marketing redoutable pour aller convaincre les entreprises encore indécises de la supériorité de .NET.

L'étude est téléchargeable sur <http://www.gotdotnet.com/team/compare/>

Développement objet

Best Practices



Stève Sfartz est directeur technique du pôle Stratégie et Développement de la société IMPROVE. De formation télécoms, ingénieur ENST de Bretagne, Stève réalise depuis 1996 des missions de développement et de conseil en architecture Objet, Java, Web, XML... Par ailleurs, Stève anime régulièrement des conférences et des formations sur ces thèmes.

[Suite de la page 1]

Le chef de projet est libre de définir son propre processus projet, ou bien d'en choisir un parmi ceux proposés par les éditeurs. La figure 2 (page suivante) présente une synthèse des processus les plus en vo-

gue dans la communauté objet et nouvelles technologies.

Force est de constater que toutes ces méthodologies proposent de travailler de façon itérative, que ce soit au niveau des plannings, des spécifications ou encore des développements...

Si l'itératif s'est imposé, c'est parce qu'il réduit la complexité de

réalisation des phases, en travaillant par approches successives et incrémentales. Il est alors possible de présenter rapidement aux utilisateurs des éléments de validation.

De plus, l'itératif permet une gestion efficace des risques, en abordant dès les premières itérations les points difficiles. Par exemple, les premières itérations de la phase

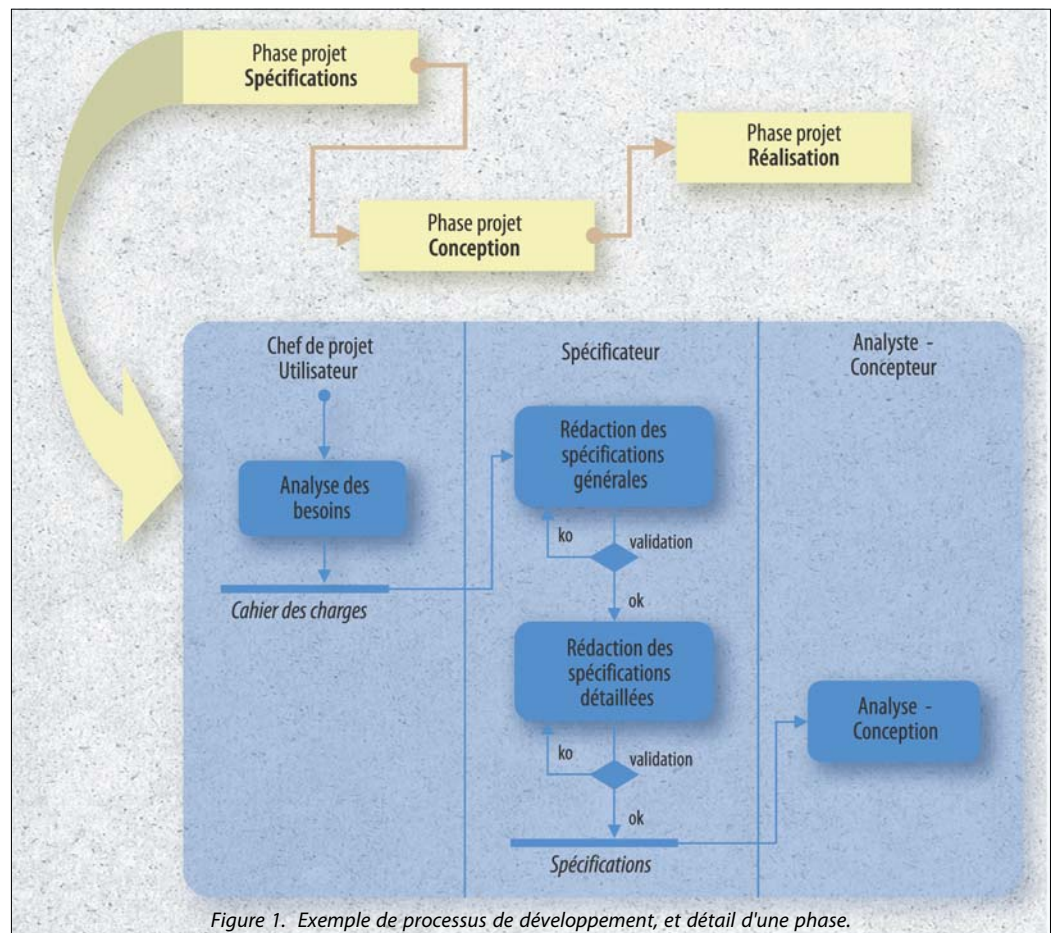


Figure 1. Exemple de processus de développement, et détail d'une phase.

	Description	Points forts	Points faibles
Cascade	<ul style="list-style-type: none"> Propose de dérouler les phases projet de manière séquentielle Cité pour des raisons historiques 	<ul style="list-style-type: none"> Distingue clairement les phases projet 	<ul style="list-style-type: none"> Non itératif Ne propose pas de modèles de documents
RUP Rational Unified Process	<ul style="list-style-type: none"> Promu par Rational Le RUP est à la fois une méthodologie et un outil prêt à l'emploi (documents types partagés dans un référentiel Web) Cible des projets de plus de 10 personnes 	<ul style="list-style-type: none"> Itératif Spécifie le dialogue entre les différents intervenants du projet : les livrables, les plannings, les prototypes... Propose des modèles de documents, et des canevas pour des projets types 	<ul style="list-style-type: none"> Coûteux à personnaliser : batterie de consultants Très axé processus, au détriment du développement : peu de place pour le code et la technologie
XP eXtreme Programming	<ul style="list-style-type: none"> Ensemble de « Best Practices » de développement (travail en équipes, transfert de compétences...) Cible des projets de moins de 10 personnes 	<ul style="list-style-type: none"> Itératif Simple à mettre en œuvre Fait une large place aux aspects techniques : prototypes, règles de développement, tests... Innovant: programmation en duo, kick-off matinal debout... 	<ul style="list-style-type: none"> Ne couvre pas les phases en amont et en aval du développement : capture des besoins, support, maintenance, tests d'intégration... Elude la phase d'analyse, si bien qu'on peut dépenser son énergie à faire et défaire Assez flou dans sa mise en œuvre: quels intervenants, quels livrables ?
2TUP Two Track Unified Process	<ul style="list-style-type: none"> S'articule autour de l'architecture Propose un cycle de développement en Y Détaillé dans "UML en action" (voir Aller plus loin) Cible des projets de toutes tailles 	<ul style="list-style-type: none"> Itératif Fait une large place à la technologie et à la gestion du risque Définit les profils des intervenants, les livrables, les plannings, les prototypes 	<ul style="list-style-type: none"> Plutôt superficiel sur les phases situées en amont et en aval du développement : capture des besoins, support, maintenance, gestion du changement... Ne propose pas de documents types

Figure 2. Synthèse des méthodologies utilisées dans le cadre de développement objet et nouvelles technologies.

technique aborderont les aspects sécurité et transaction.

.....
Best Practice

Définir pour chaque phase, le nombre d'itérations et leur contenu précis.

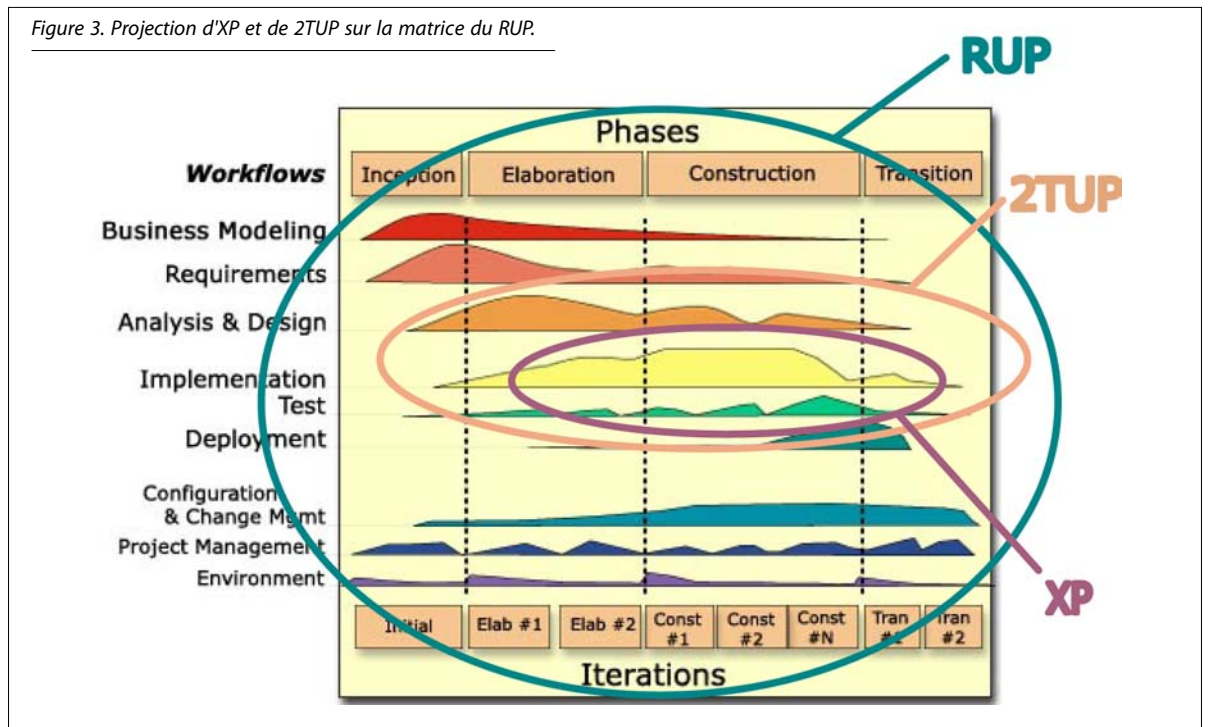
Exemple. Dans le cadre d'un projet Web qui comporterait un module de personnalisation, la réalisation de ce module pourrait être composé des itérations suivantes :

- Itération n° 1 (longue/technique) : création du moteur de personnalisation des interfaces Web (spécifications des besoins de personnalisation, élaboration de l'ergonomie, modélisation du moteur en adéquation avec l'architecture applicative, développement des composants logiciels, développement d'une maquette à des fins de validation) ;
- Itération n° 2 (courte/technique) : revue de l'ergonomie suite à la démonstration et intégration de nouveaux éléments à

la maquette (mise à jour de l'ergonomie, mise à jour de la modélisation du moteur, mise à jour des composants logiciels, réalisation de la maquette à des fins de démonstration) ;

- Itération n° 3 (courte/fonctionnelle) : intégration d'éléments fonctionnels (définition d'une charte graphique et adaptation de la maquette à des fins de validation).
-

Figure 3. Projection d'XP et de 2TUP sur la matrice du RUP.



avec des risques importants de remise en cause des réalisations (décalage de planning, démotivation des équipes...).

Exemple. Faire reposer la phase de réalisation sur un processus en Y, en s'attachant à :

- intégrer les risques dès les premières itérations ;
- isoler les solutions techniques dans des frameworks ;
- valider l'avancement par des prototypes fonctionnels.

Nous avons insisté sur la place à accorder à la technologie. Cependant, cela se traduit de façon différente selon le type de réalisation. Par exemple, dans le cas d'une application intranet de gestion de forum, on privilégiera les aspects montée en charge sur les aspects sécurité. De plus, on pourra remettre en question la nécessité de développer des frameworks techniques ou de formaliser les règles de développement.

La figure 5 définit les objectifs de la branche technique du cycle de développement en Y en distinguant trois types d'applications : One-Shot, départementale et produit.

Selon le type d'application réalisée, on observe des exigences différentes en termes d'équipe, d'architecture et de règles de développement. C'est pourquoi ce n'est qu'après la réalisation de ces exigences que le chef de projet a une vision précise de la suite de son projet : c'est alors le moment de réactualiser le macro-planning à partir des éléments révélés par la phase technique, notamment la productivité des équipes.

Best Practice

En règle générale, on peut être confiant quant à la réussite du projet lorsque le prototype fonctionnel est validé et testé. En effet, à cette étape, les équipes de développement sont productives (formées

sur les frameworks techniques, bibliothèques de composants et règles de développement...).

Comment organiser mes équipes ?

Dans la plupart des projets nouvelles technologies, la gestion des ressources se révèle rapidement un véritable casse-tête : les experts ne sont jamais disponibles, les équipes de développement sont sur-dimensionnées sur certaines phases et sous-dimensionnées sur d'autres, les transferts de compétences n'ont pas lieu si bien que les ressources critiques restent les mêmes tout au long du projet...

Nous présentons ci-dessous quelques règles pour éviter de tomber dans ces pièges.

Best Practice

Se concentrer sur la phase de développement en cours.

Ne pas essayer de nourrir ses équipes à tout prix : il est normal que certaines ressources ne soient pas affectées sur certaines phases du projet. Mieux vaut parfois les laisser vacantes, que de leur demander de lancer leur développement en dépit du manque d'informations, ou pire encore, de les affecter à des tâches qui ne sont pas de leur ressort (car c'est autant d'énergie en gestion de ressources, transfert de compétences, qui ne vont pas directement au bénéfice du projet).

Exemple. Les spécifications ont pris du retard. Les experts techniques sont en train de travailler sur les problèmes de sécurité des sessions utilisateurs sur HTTP et l'interfaçage des transactions CICS. Plusieurs développeurs n'ont pas de tâches affectées.

- Ne pas leur demander de commencer à

lire les spécifications et démarrer le développement !

- Pourquoi ne pas laisser ces développeurs vacants (formation, revue de code existant) ?
- Etudier s'il est possible de les affecter aux jeux de tests ou à la récupération de l'existant ?

Best Practice

Rechercher la simplicité.

Ne pas regretter de ne pas avoir que des experts techniques. En effet, les développeurs seniors sont une menace pour le projet. D'une part, ils risquent de produire du code non maintenable, car trop élaboré, trop objet ! D'autre part, ils deviennent inégalement des ressources critiques.

- Identifier les besoins en expertise technique.
- Faire réaliser à des développeurs seniors des bibliothèques de composants ou des modèles de développement qui puissent être réutilisés par les équipes moins expérimentées.

Best Practice

Réactualiser son planning de façon itérative.

S'il faut prendre le temps de bien définir son processus de développement en début de projet, et proposer un macro-planning réaliste, il faut rester conscient que les charges estimées évoluent régulièrement jusqu'à la validation du prototype fonctionnel, moment à partir duquel on a une bonne idée de la productivité des équipes.

Aussi, plutôt que de mettre constamment à jour son planning, le revoir plutôt à la fin de certaines itérations.

Figure 5. Objectifs et organisation de la phase technique selon le type d'application.

	Exigences	Composantes de la branche technique		
		Equipe	Architecture	Règles de développement
Application	<ul style="list-style-type: none"> • Maintenabilité • Robustesse • Pas de réutilisation 	<ul style="list-style-type: none"> • Développeurs seniors sur les technologies retenues 	<ul style="list-style-type: none"> • Définie par les développeurs 	<ul style="list-style-type: none"> • Règles de codage et de nommage
Application d'entreprise	<ul style="list-style-type: none"> • Maintenabilité • Robustesse • Réutilisation 	<ul style="list-style-type: none"> • Distinction des rôles : - chef de projet technique, - architecte, - développeur 	<ul style="list-style-type: none"> • Définie par l'architecte • Bibliothèques de composants fonctionnels 	<ul style="list-style-type: none"> • Réalisation de prototypes • Règles de codage, de nommage • Stratégie de gestion des exceptions
Produit	<ul style="list-style-type: none"> • Maintenabilité • Robustesse • Configurabilité 	<ul style="list-style-type: none"> • Distinction des rôles : chef de projet technique, architecte, développeur technique, développeur fonctionnel 	<ul style="list-style-type: none"> • Définie par l'architecte • Frameworks techniques • Frameworks fonctionnels 	<ul style="list-style-type: none"> • Réalisation de prototypes • Règles de codage, de nommage • Stratégie de gestion des exceptions et des logs

Best Practice

Spécialiser ses équipes de développement.

Le temps d'apprentissage des nouvelles technologies peut se révéler interminable (Java + XML + HTML + JavaScript + JDBC + SQL + EJB + objet distribué + suivi de session) sans parler de l'apprentissage du fonctionnel !

Aussi, nous conseillons de :

- *ne pas chercher le mouton à 5 pattes !*
- *spécialiser ses équipes (interfaces graphiques ou code métier, ce qui correspond assez bien aux aspirations de chacun), et libre ensuite à chacun de changer de spécialité ;*
- *profiter des phases de spécifications et d'analyse pour former ses équipes de développement ;*
- *placer des développeurs seniors et juniors en paire pendant quelques heures ou jours d'apprentissage se révèle souvent plus efficace qu'effectuer des relectures de code !*

Best Practice

Démarrer tôt la phase d'architecture.

Exemple. Aussi longtemps que le travail d'architecture n'est pas terminé, les développements sont bloqués, en attente des éléments techniques (choix de produits tiers, frameworks, règles de développement).

- *Pourquoi ne pas lancer en parallèle les phases de spécifications et d'architecture ?*

Enfin, la meilleure garantie de succès, c'est de s'assurer que chaque intervenant remplit au mieux le ou les rôles qui lui sont attribués, ainsi que l'illustre la figure 6.

Nous espérons vous avoir éclairé sur les processus de développement objet et nouvelles technologies, et surtout sur la façon de maîtriser le risque, tout en travaillant de façon itérative. A vous maintenant de doser ces exigences en fonction de votre type d'application (One-Shot, entreprise ou produit), et du cadre de sa réalisation (régie, forfait, ou régie forfaitée). Bonne aventure...

ALLER PLUS LOIN

From Waterfall to Iterative Lifecycle - A tough transition for project manager
<http://www.rational.com/media/whitepapers/TP173A.pdf>

Completing the Unified Process With Process Patterns, by Scott W. Ambler
<http://www.ambysoft.com/unifiedProcess.html>

A Comparaison of RUP and XP
<http://www.rational.com/media/whitepapers/TP167.pdf>

Présentation de l'eXtreme Programming
<http://www.extremeprogramming.org>

Présentation du processus 2TUP
 UML en action, par Pascal Roques et Franck Vallée, éditions Eyrolles

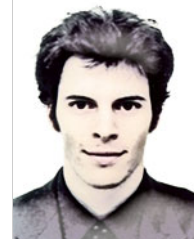
Le mythe du mois-homme, Essais sur le génie logiciel par Frederick P. Brooks Jr, Thomson Publishing

Figure 6. Rôles et Best Practices des intervenants du projet.

	Fonction	Best Practices
Chef de projet	<ul style="list-style-type: none"> • Elabore le processus de développement et le macro-planning • Rend compte de l'avancement du projet 	<ul style="list-style-type: none"> • Etre ferme face aux demandes des utilisateurs • Trancher dans le sens de la simplification
Chef de projet utilisateur	<ul style="list-style-type: none"> • Recueille le besoin des utilisateurs • Participe à l'élaboration des spécifications fonctionnelles 	<ul style="list-style-type: none"> • Etre sensible aux contraintes de réalisation
Spécificateur	<ul style="list-style-type: none"> • Rédige les spécifications fonctionnelles générales et détaillées • Identifie les flux de données 	<ul style="list-style-type: none"> • Se plonger dans le modèle physique pour cerner le niveau d'adéquation avec l'existant • Creuser les points fonctionnels plutôt qu'attendre les retours de l'analyse, et ce afin de minimiser le nombre d'itérations
Architecte	<ul style="list-style-type: none"> • Recense les besoins techniques dans le dossier technique • Elabore les architectures logicielle et applicative • Argumente ses choix technologiques • Identifie les besoins en produits tiers et frameworks techniques 	<ul style="list-style-type: none"> • Expert objet et nouvelles technologies • Assisté par les consultants des technologies tierces sélectionnées • Faire valider l'architecture par un consultant externe • Prendre connaissance du fonctionnel et assister le concepteur sur les premières modélisations
Concepteur	<ul style="list-style-type: none"> • Réalise les modèles de conception selon l'architecture applicative • Identifie les besoins d'évolutions de l'existant (modèle de données, interfaces, batchs...) 	<ul style="list-style-type: none"> • Bonne connaissance de l'objet • Bonne connaissance du fonctionnel et des données (SQL, site central...) • A fait l'expérience d'un projet objet et nouvelles technologies
Chef de projet technique	<ul style="list-style-type: none"> • Assure le suivi des plannings de développement • Responsable de la qualité des développements 	<ul style="list-style-type: none"> • Expert objet et nouvelles technologies • Valide les modèles de conception • Assistance technique aux équipes de développement • Relecture de code
Développeur technique	<ul style="list-style-type: none"> • Réalise les bibliothèques de composants et les frameworks techniques en collaboration avec l'architecte 	<ul style="list-style-type: none"> • Expert objet et nouvelles technologies • Forme les équipes fonctionnelles sur les réalisations techniques
Développeur fonctionnel	<ul style="list-style-type: none"> • Réalise les applications décrites dans les spécifications fonctionnelles, en tenant compte de l'architecture applicative, des frameworks techniques et des règles de développement 	<ul style="list-style-type: none"> • Bonne connaissance du fonctionnel • Spécialisé sur un domaine : interface graphique ou composants métiers

Programmation par contrat et qualité logicielle

AVEC LA COMPLEXITÉ GRANDISSANTE DES SYSTÈMES INFORMATIQUES SE MULTIPLIENT LES OCCASIONS DE FAIRE DES ERREURS. OR DES MÉTHODES ET DES OUTILS ADAPTÉS EXISTENT, QUI ONT POUR OBJECTIF DE GARANTIR LA QUALITÉ ET LE SUCCÈS DE CES ARCHITECTURES COMPLEXES. DEPUIS UNE DIZAINE D'ANNÉES, LES CONTRATS RÉPONDENT À CES EXIGENCES.



Nicolas Chauvat a fondé la société Logilab, spécialisée dans l'application industrielle de techniques issues de la recherche en informatique et pionnière des assistants personnels intelligents.

Récemment, l'un des piliers de l'industrie informatique annonçait la commercialisation dans quelques années d'ensembles de machines à même de s'administrer et de réparer certains de leurs problèmes matériels de manière autonome. Cela illustre bien la croissance continue de la complexité des systèmes informatiques, soutenue par l'évolution parallèle de la puissance des machines, de l'abstraction des langages et des méthodes de développement. Avec une complexité croissante, les occasions de faire des erreurs se multiplient. Pour garantir qualité et succès de ces systèmes, on se doit d'en maîtriser la technicité avec des outils et des méthodes adaptées.

Dans le domaine logiciel, les contrats remplissent cette fonction, garantissant la qualité et facilitant la réutilisation des composants. Introduits à la fin des années 80 par

Bertrand Meyer avec son langage Eiffel, ils vont bien au-delà des mécanismes d'assertions et de typage fort disponibles auparavant, sans pour autant atteindre le niveau d'abstraction des langages de spécification et de preuve formelle. Leur intégration parfaite avec le modèle objet et les méthodes de développement associées en fait des outils de choix.

En ingénierie informatique, ainsi que dans d'autres domaines, on a coutume de dire que "la qualité est tout et que tout est qualité". C'est en effet un facteur essentiel du succès d'un produit, il faut donc employer tous les moyens disponibles pour l'accroître et la garantir. De plus, la qualité de l'ensemble découle de celle des différents composants, mais aussi des diverses étapes de la réalisation.

Néanmoins, tout n'exerce pas la même influence sur la qualité, qui ne sera d'ailleurs pas perçue de manière identique par tout le

monde. Définir la qualité comme une combinaison de différents facteurs permet donc de comparer de manière objective l'influence des méthodes ou des composants sur chacun de ces facteurs, mais aussi d'en étudier l'influence sur la perception de la qualité qu'auront les divers acteurs.

Meyer propose de décomposer la qualité à partir des facteurs suivants : correction, robustesse, efficacité, facilité d'utilisation, richesse fonctionnelle, délais, coût, adaptabilité et extensibilité, compatibilité, portabilité, réutilisation, compréhensibilité, facilité de test. On voit bien que certains de ces facteurs n'ont pas le même poids et que les personnes chargées des spécifications, de la programmation, des tests, de la maintenance ou de la vente, pour ne citer que celles-là, auront chacune des priorités qui leur sont propres dans leur approche de la qualité.

A cette première définition de la qualité, ajoutons celle de son contraire, le défaut. Cette fois encore, c'est une formulation générale qui semble la plus satisfaisante, car elle permet d'intégrer les défauts non seulement du code, mais aussi des spécifications, de la documentation et du reste. Un défaut est une propriété d'un logiciel qui peut et doit être améliorée. Accroître et garantir la qualité, c'est donc employer tous les moyens disponibles pour éviter d'introduire des défauts, et détecter puis éliminer ceux que l'on a pu introduire par mégarde.

L'utilisation de méthodes de conception et d'analyse, les formalismes de descriptions, les conventions de documentation, l'identification de bonnes pratiques sont

autant d'outils et de guides pour éviter d'introduire des défauts. Pour les détecter et les supprimer, on peut utiliser la revue (une personne qui n'a pas participé à la réalisation commente et critique les choix et les résultats) lors de la plupart des étapes, mais aussi des techniques et outils particuliers lorsqu'il s'agit du code source : analyse statique (compilateur, lint, métriques), tests (unitaires, intégration, validation), outils CASE (Computer Assisted Software Engineering), langages de preuve formelle, etc.

Les défauts découlant des erreurs humaines peuvent être introduits à tous les stades du développement, lors de l'analyse, des spécifications, de la réalisation, de la documentation, de la validation, ou même par la suite lors de la réutilisation, si l'on adopte une approche composants. La qualité est donc un souci de tous les instants.

Nous allons tenter de montrer en quoi la programmation par contrat se révèle une aide pratique et efficace tout au long de la vie d'un logiciel, de sa conception à sa réutilisation en passant par sa réalisation et sa validation.

Les contrats

Les contrats représentent un moyen de répondre à deux questions essentielles : que doit faire le logiciel ? Fait-il ce qu'il est censé faire ?

Dans la vie courante, les contrats sont des accords liant des parties en décrivant précisément les obligations et bénéfices de chacun. Transposés au domaine du logiciel, ils vont s'appliquer aux classes et aux appels de méthodes pour lesquels ils joueront un rôle identique, et décrire de manière précise plusieurs aspects de leur interaction. Le contrat s'applique à la relation entre l'appelant et l'appelé, grâce aux préconditions, aux postconditions et aux invariants, mais il a aussi des conséquences sur l'héritage et la gestion des exceptions et des cas anormaux, sans compter qu'il facilite la documentation.

Les invariants, définis pour chaque classe, sont les propriétés devant être vérifiées à tout instant et qui le seront avant et après l'exécution de toute méthode. Ils permettent donc de capturer le sens et les caractéristiques de validité de certaines propriétés des objets.

Les préconditions sont des conditions portant sur les arguments en entrée que le client (appelant) a pour devoir de respecter. Si ces conditions ne sont pas

respectées, le fournisseur (appelé) n'a pas à s'engager à exécuter correctement la méthode appelée. Les préconditions doivent être transparentes, c'est-à-dire qu'elles doivent être nécessaires et suffisantes pour que l'appelant puisse être servi et obtienne les garanties associées aux postconditions.

Les postconditions sont des garanties sur les sorties que le fournisseur (appelé) s'engage à respecter si la méthode s'exécute normalement. Si le fournisseur rencontre une erreur impossible à gérer, il doit lancer une exception et peut ne pas respecter les postconditions. Mais il lui est interdit, même en cas d'erreur, de terminer l'exécution et de rendre le contrôle normalement au client, sans respecter ses garanties sur les sorties. Le mot-clé "old" peut être utilisé dans les postconditions ou lors du traitement des exceptions pour faire référence à l'état dans lequel était l'objet avant l'exécution de la méthode. Cela permet de paramétrer les garanties sur les sorties ou de tenter un nouvel essai d'exécution après une exception.

Les assertions classiques se retrouvent sous leur forme habituelle pour vérifier une condition dans le corps d'une méthode ou pour faciliter le traitement des boucles, on peut alors définir des invariants de boucles et des conditions pour les variables utilisées dans les boucles. L'idée est donc toujours de faire apparaître au mieux ce qui relève de l'implémentation et de la spécification.

Les exceptions lancées lors de l'exécution peuvent être attrapées pour tenter de résoudre le problème en local, sans passer le contrôle à l'appelant. Le traitement de ces exceptions peut choisir de modifier l'état de l'objet pour corriger les erreurs et reprendre l'exécution, ou restaurer l'état d'origine grâce à old et faire un nouvel essai, ou bien agir pour que l'objet soit dans un état cohérent avant de rendre le contrôle à l'appelant en lançant une exception.

Quand une classe hérite d'une autre, elle bénéficie aussi des contrats associés aux méthodes, mais peut choisir de relâcher les contraintes portant sur le client et de renforcer ses propres garanties. De cette façon, on peut se reposer sur le contrat d'un parent pour appeler la méthode d'un enfant, car les obligations de l'appelant ne peuvent qu'être identiques ou moindres (qui peut le plus peut le moins) et les garanties de l'appelé ne peuvent qu'être égales ou supérieures (abondance de bien ne nuit pas).

Le contrat est une documentation naturelle, car il répond à la

question : qu'est-ce que cette classe est supposée faire ? Plutôt que de se contenter de l'interface, on peut donc l'utiliser avec le contrat (invariants et pré/postconditions) comme élément de base de la documentation d'une classe. C'est utile lors des spécifications et de la génération automatique de la documentation à partir du code et de ses commentaires, comme le permettent des outils disponibles dans de nombreux langages. Il est important de faire remarquer que le contrat n'est pas une méthode de programmation défensive. Au contraire, il sert à se prémunir de ce genre d'attitude qui consiste à toujours vérifier en début de méthode la validité des arguments reçus, puisqu'il est recommandé qu'une méthode ne cherche pas à vérifier dans son corps ce qui a été garanti par les préconditions du contrat.

Enfin, les contrats vont bien plus loin que les assertions, qui ne sont en comparaison qu'un mécanisme de base, lequel n'a que peu en commun avec les aspects méthodologiques et l'intégration au modèle objet et à l'héritage qu'offrent les contrats. Il est cependant intéressant de noter que plusieurs langages qui ne proposent pas de programmation par contrat dans leur version de base utilisent le mécanisme d'assertion, lorsqu'il est disponible, pour mettre en œuvre les contrats avec des bibliothèques, des préprocesseurs ou d'autres moyens spécialisés.

Langages et contrats

À l'origine, Bertrand Meyer a appliqué ses idées à son langage Eiffel, qui reste le seul à proposer un mécanisme de contrats faisant partie intégrante du langage. Cependant, de nombreux langages modernes offrent une mise en œuvre plus ou moins exhaustive des contrats, rajoutée par le biais de préprocesseurs, de métaprogrammation, ou d'autres techniques faisant usage par exemple de la réflexivité lorsqu'elle est disponible. Parmi eux, on compte Java, C++, Python, Perl, ou d'autres langages spécialisés et moins connus comme Narval, dédié à la mise en œuvre d'assistants personnels intelligents.

Nous allons voir maintenant ce que chacun propose, et comparer le code obtenu en deux langages, Python et Eiffel, pour un même exemple : le modèle d'un réservoir. Sa capacité est mesurée en litres et on le considère plein lorsqu'il l'est à 97 %, et vide à moins de 3 %. Les actions possibles sont :

vider le réservoir, le remplir, ajouter un volume précis ou retirer un volume précis.

Eiffel

Eiffel, en tant que langage d'origine, propose une mise en œuvre assez exhaustive avec les invariants, les pré et postconditions, old, les formes abrégées de classes utilisées comme documentation, et l'intégration avec le mécanisme d'héritage. Eiffel ne propose cependant pas de traitement générique des exceptions et se contente d'un do/rescue/retry : si la méthode décrite par do produit une exception, le contrôle passe à rescue, dans lequel on peut tenter un retry, qui reprend l'exécution au début du do. Notre exemple du réservoir peut être implémenté en Eiffel (listing 1).

Python

Une seule mise en œuvre des contrats est disponible pour Python, mais elle utilise au mieux le caractère dynamique du langage grâce à une approche de métaprogrammation. Le constructeur des objets que l'on désire voir respecter les contrats renvoie un proxy (au sens des design patterns), qui à chaque appel de méthode prendra soin de conserver une copie de l'objet à laquelle faire référence par la suite avec old, et extraira de la chaîne de documentation associée à la méthode et à la classe les conditions nécessaires, qui seront vérifiées en utilisant le mécanisme d'assertions natif.

Cela permet, sans modifier l'interpréteur et en choisissant classe par classe, de faire respecter des invariants, des pré et postconditions, d'utiliser old, mais aussi de rajouter un typage fort des arguments des méthodes, tout en conservant les avantages qu'apportent les contrats en matière de documentation, puisque les conditions sont conservées dans les commentaires associés aux méthodes et aux classes. Cette mise en œuvre ne prévoit en l'état rien de particulier concernant les exceptions, mais l'approche de métaprogrammation choisie permet d'y porter remède très rapidement en modifiant la classe proxy. Notre exemple du réservoir peut être implémenté en Python (listing 2).

Java

Java ne propose dans sa version de base ni contrats ni assertions. Malgré cela, deux mises en œuvre des contrats sont disponibles pour Java.

JContractor utilise le mécanisme de réflexivité de Java et la possibilité de remplacer le ClassLoader

Listing 1

```

class RESERVOIR
-- RESERVOIR modélise un réservoir dont la capacité est
-- déterminée à la création et que l'on peut remplir et vider
-- creation creer
feature
  capacite: INTEGER
  remplissage: INTEGER
  capacite_maximale: INTEGER is 2000
  invariant
    remplissage >= 0
    remplissage <= capacite
    capacite >= 0
    capacite <= capacite_maximale
  creer(capa:INTEGER) is
    -- Créer un réservoir de capacité capa
    require
      capa <= capacite_maximale
    do
      capacite := capa
      remplissage := 0
    ensure
      capacite = capa
      remplissage = 0
    end
  vider(NONE) is
    -- vider le réservoir
    do
      -- contrôle le vidage du réservoir
    ensure
      remplissage*100 < 3*capacite
    end
  remplir(NONE) is
    -- Remplir le réservoir
    do
      -- remplissage du réservoir
    ensure
      remplissage*100 > 97*capacite
    end
  ajouter(volume:INTEGER) is
    -- Ajouter volume donné au réservoir
    require
      remplissage + volume <= capacite
    do
      -- remplissage du réservoir
    ensure
      remplissage = old remplissage + volume
    end
  retirer(volume:INTEGER) is
    -- Retirer volume donné du réservoir
    require
      remplissage - volume >= 0
    do
      -- remplissage du réservoir
    ensure
      remplissage = old remplissage - volume
    end
  contient(NONE):INTEGER is
    -- Renvoie volume dans le réservoir
    do
      Result:= remplissage
    end
  estVide(NONE):BOOLEAN is
    -- Teste si le réservoir est vide
    do
      Result:= (remplissage*100 < 3*capacite)
    end
  estPlein(NONE):BOOLEAN is
    -- Teste si le réservoir est plein
    do
      Result:=
        (remplissage*100 > 97*capacite)
    end
end
-- class RESERVOIR

```

de la machine virtuelle pour substituer à un appel à une méthode une suite d'appels à la méthode des invariants de classe, sa méthode précondition, elle-même, sa méthode postcondition, et à nouveau la méthode des invariants, si elles ont toutes été correctement nommées. En cas d'exception, le contrôle est transmis à la méthode de gestion appropriée. jContractor gère les invariants, les pré et postconditions, old et les exceptions génériques, mais l'utilisation des contrats pour la documentation est difficile, leur expression étant contenue dans des méthodes, elle n'apparaît pas avec les outils classiques comme javadoc.

iContract est un préprocesseur qui extrait les directives inscrites dans les commentaires précédant les méthodes pour générer le code nécessaire, lequel sera ensuite compilé comme à l'habitude. iContract traite les invariants et les pré et postconditions. Les directives étant écrites dans les commentaires, les contrats apparaissent naturellement dans la documentation des classes si l'on recourt aux outils habituels.

C et C++

Deux mises en œuvre des contrats sont disponibles pour le C et le C++.

Assertions.h définit tout sous forme de macros pouvant être activées au moment de la compilation et qui transforment les conditions en des assertions de base du langage C. Cette mise en œuvre est plutôt exhaustive et comprend les invariants, les pré/postconditions, les assertions classiques, l'utilisation de old, mais aussi des connecteurs logiques, l'appartenance à un ensemble, ou encore la validité de bornes et les comparaisons avec zéro. Son défaut : ne fournir aucune solution d'intégration avec le mécanisme d'héritage, ni avec les outils de génération de documentation.

GNU Nana, de son côté, définit tout sous forme de macros C/C++ sur la base des assertions du langage C, et offre, en plus de la possibilité de les désactiver à la compilation, d'autres macros pour faciliter le débogage avec gdb. Cette mise en œuvre est très complète et propose, outre les contrats définis dans Eiffel, des fonctions facilitant le débogage et la mesure de certains paramètres (nombre de cycles consommés par le processeur pour exécuter une fonction, temps d'exécution...). Comme Assertions.h, GNU Nana ne permet pas d'intégrer les contrats avec le mécanisme d'héritage de C++,

mais il fournit en revanche un générateur de documentation.

Narval et XML

Le langage Narval est dédié à la création d'assistants personnels intelligents. Avec Narval, plutôt que des programmes, on écrit des recettes qui décrivent des comportements que pourra avoir l'assistant. Les recettes sont des enchaînements d'étapes liées par des transitions. Les étapes, quant à elles, sont des actions ou des transformations qui acceptent des fragments d'XML en entrée et qui en produisent en sortie. La caractéristique intéressante est que le prototype de ces étapes est constitué de conditions sur les entrées et les sorties exprimées en XPath. Cela permet à la fois d'expliquer le type des arguments et d'imposer des pré/postconditions. On a cherché à unifier ainsi la signature d'une fonction et la notion de pré/postcondition en les formulant comme une extension du typage. Ces prototypes servent de documentation pour les étapes, mais sont aussi employés pour la planification automatique ; leur utilisation se rapproche dans ce cas des prédicats en logique. Dans ce langage, il n'y a cependant pas d'héritage des actions, car il est mis en œuvre à un niveau inférieur dans l'écriture des bibliothèques. De plus, la spécialisation des conditions sur les entrées et les sorties ne se fait que par ajout et renforcement.

Autres langages

Des mises en œuvre des contrats existent aussi dans d'autres langages, n'hésitez donc pas à partir à leur recherche sur Internet. Les amateurs de Perl sont les plus chanceux puisqu'un module ad hoc est disponible sur CPAN.

Avantages

La programmation par contrat, de même que l'approche orientée objet qu'elle vient renforcer, permet de formaliser les problèmes et leur solution à un niveau d'abstraction où ils trouvent une expression claire. C'est en l'occurrence une avancée nette par rapport aux mécanismes d'assertions ou de typage fort disponibles dans plusieurs langages. Bien que les contrats apportent un niveau de formalisme et d'abstraction supérieur, ils ne constituent pas pour autant une marche difficile à franchir. Etant parfaitement intégrés au modèle objet, ils sont facilement abordables et compréhensibles, ce qui les rend rapidement utilisables,

avec l'avantage d'en retirer un bénéfice direct. De plus, comme ils font partie du programme, ils évitent de creuser l'habituel fossé qui sépare langage de spécification ou de preuve formelle et langage de réalisation. Ils facilitent toutes les étapes du développement, des spécifications aux tests et au débogage, en passant par la documentation, car ils permettent de comparer ce que le programme est supposé faire avec ce qu'il fait réellement. On ne peut même pas leur reprocher d'imposer un compromis gênant qui contraindrait à échanger cette sécurité supplémentaire contre une dégradation des performances, puisque la plupart des mises en œuvre permettent de supprimer la vérification des contrats. Le même code source peut donc être utilisé avec les contrats activés lors du développement et des tests, puis sans les contrats lorsqu'il est déployé et que l'on souhaite améliorer les temps d'exécution.

Lorsqu'on désire réutiliser un composant logiciel, les contrats servent à la fois de documentation et de garantie que le comportement attendu sera obtenu. Leur avantage est d'être indissociables du code source et facilement activables pour vérifier qu'ils sont correctement utilisés. On se reportera à ce sujet aux mésaventures du lanceur européen Ariane 5.

Comparativement aux autres méthodes disponibles dans la panoplie de l'ingénieur logiciel pour éviter d'introduire des défauts ou pour détecter et supprimer ceux que l'on aurait introduits, les contrats sont particulièrement rentables et offrent un rapport élevé entre le bénéfice qualitatif et l'effort consenti tout au long du développement.

Au final, les contrats présentent des avantages certains dans le développement logiciel, qu'il s'agisse de l'analyse et des spécifications, de la documentation, des tests et du débogage, de la robustesse, ou de la réutilisation de composants. Ces avantages sont autant de facteurs pour améliorer la qualité, qui doit être présente dès l'origine et non simplement vérifiée à la fin du développement. Pourquoi les contrats ne sont-ils pas plus largement répandus ? Probablement par manque d'éducation et de publicité, mais aussi parce qu'ils ne font pas partie des versions de base de la plupart des langages modernes. En ayant tenté de vous faire connaître les différentes mises en œuvre disponibles, nous apportons ainsi notre modeste pierre à l'édifice... ●

ALLER PLUS LOIN

Object-Oriented Software Construction, Bertrand Meyer. Prentice Hall. 2nd Edition, 1997.

www.eiffel.com/doc/oosc.html

Design Patterns : Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson et John Vlissides. Addison-Wesley. 1995.

www.hillside.net/patterns/DPBook/DPBook.html

Design by contract : The Lessons of The Ariane, Jean-Marc Jezequel et Bertrand Meyer.

IEEE Computer. 1997. 30. 2. 129-130.

www.eiffel.com/doc/manuals/technology/contract/ariane/page.html

Design By Contract : A Missing Link In The Quest For Quality Software, Lockheed Martin, US EPA. 1998.

www.elj.com/eiffel/dbc/

Eiffel : www.eiffel.com/doc/manuals/technology/contract/

iContract : www.reliable-systems.com/tools/iContract/iContract.htm

jContractor : Reflective Java Library to Support Design-by-Contract, Murat Karaorman, Urs Hölzle et John Bruno.

Technical Report TRCS98-31, University of California, Santa Barbara.

1998. www.cs.ucsb.edu/oocsb/papers/TRCS98-31.html

Assertions.h : www.elj.com/eiffel/dbc/Assertions.h

Gnu Nana : Improved Support for Assertions and Logging in Cand C++, P. J. Maker.

www.cs.ntu.edu.au/homepages/pjm/nana-home/

Design by Contract for Python, Reinhold Plösch. IEEE Proceedings of the Joint Asia Pacific Software Engineering Conference (APSEC97/ICSC97), Hongkong. 1997.

www.swe.uni-linz.ac.at/publications/abstract/TR-SE-97.24.html

Narval : www.logilab.org/narval/doc.html

Perl : `Class::Contract` disponible sur CPAN.

<http://search.cpan.org/search?module=Class::Contract>

Listing 2

```
class Reservoir :
    """Réservoir modélise un réservoir dont
    """la capacité est déterminée à la création
    """et que l'on peut remplir et vider.
    inv: 0 <= self.remplissage
        <= self.capacite;
    inv: 0 <= self.capacite
        <= self.capacite_maximale;
    """
    capacite_maximale = 2000
    def __init__(self, capa) :
        """Créer un réservoir de capacité capa
        req: capa <= self.capacite_maximale;
        ensure: self.capacite = capa;
        ensure: self.remplissage = 0;
        """
        capacite = capa
        remplissage = 0
    def vider(self) :
        """Vider le réservoir
        ensure: self.remplissage*100
            < 3*self.capacite;
        """
        pass # vidage du réservoir
    def remplir(self) :
        """Remplir le réservoir
        ensure: self.remplissage*100
            > 97*self.capacite;
        """
        pass # remplissage du réservoir
    def ajouter(self,volume) :
        """Ajouter volume donné au réservoir
        olds.remplissage: self.remplissage;
        req: self.remplissage + volume
            <= self.capacite;
        ensure: self.remplissage =
            self.olds.remplissage + volume;
        """
        pass # remplissage du réservoir
    def retirer(self,volume) :
        """Retirer volume donné du reservoir
        olds.remplissage: self.remplissage;
        req: self.remplissage - volume >= 0;
        ensure: self.remplissage =
            self.olds.remplissage - volume;
        """
        pass # vidage du reservoir
    def contient(self) :
        """Renvoie volume dans le réservoir"""
        return self.remplissage
    def estVide(self) :
        """Test si le réservoir est vide"""
        return (self.remplissage*100 < 3*self.capacite)
    def estPlein(self) :
        """Teste si le réservoir est plein"""
        return (self.remplissage*100 > 97*self.capacite)
```


Lotus
.....

Domino Rnext bêta 2



Denis Le Guiff est chef de projet dans une SSII et Certified Lotus Professional Principal V4 et V5.

APRÈS PLUS DE DEUX ANS DE BONS ET LOYAUX SERVICES, LA R5 DE LOTUS NOTES/DOMINO DEVRAIT ÊTRE REMPLACÉE DANS QUELQUES MOIS PAR UNE NOUVELLE VERSION, VRAISEMBLABLEMENT LA R6, NOM DE CODE RNEXT.

NOTRE VERDICT

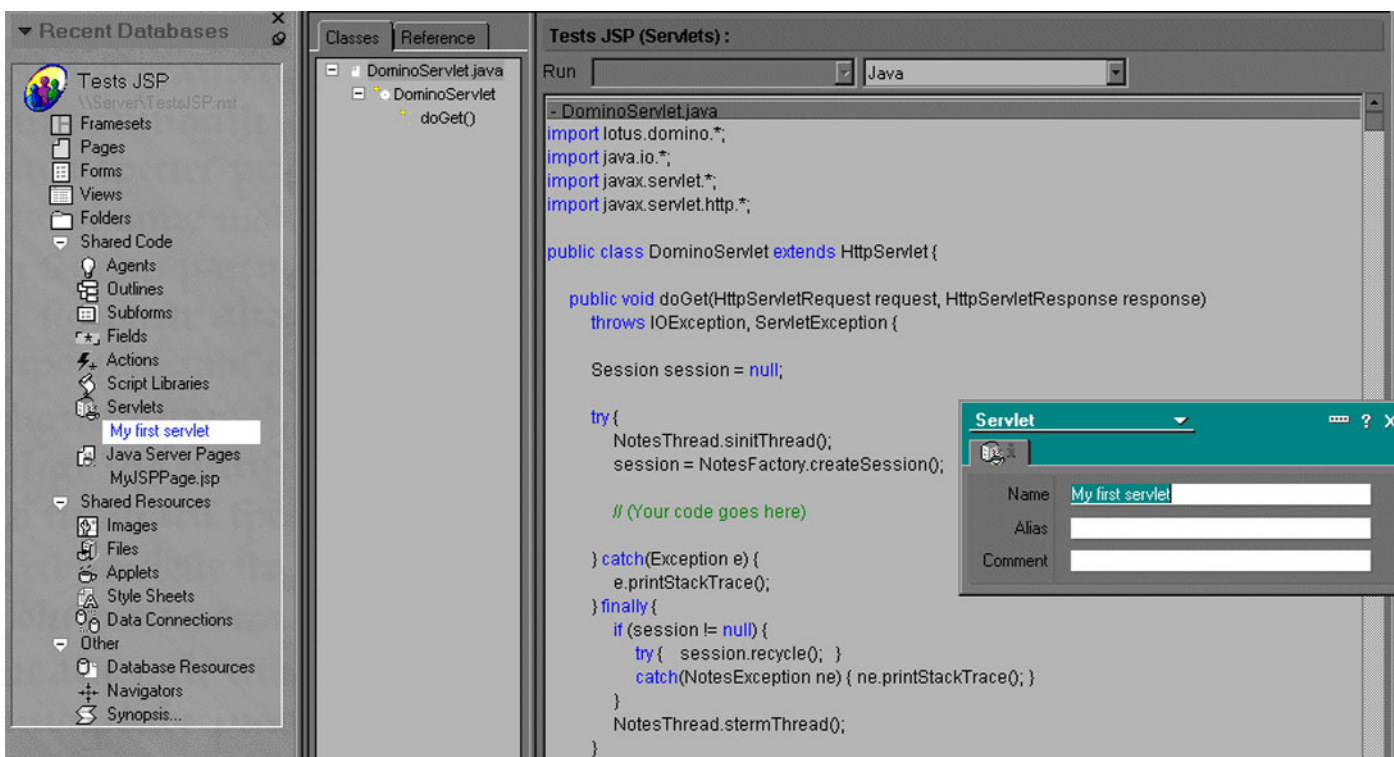
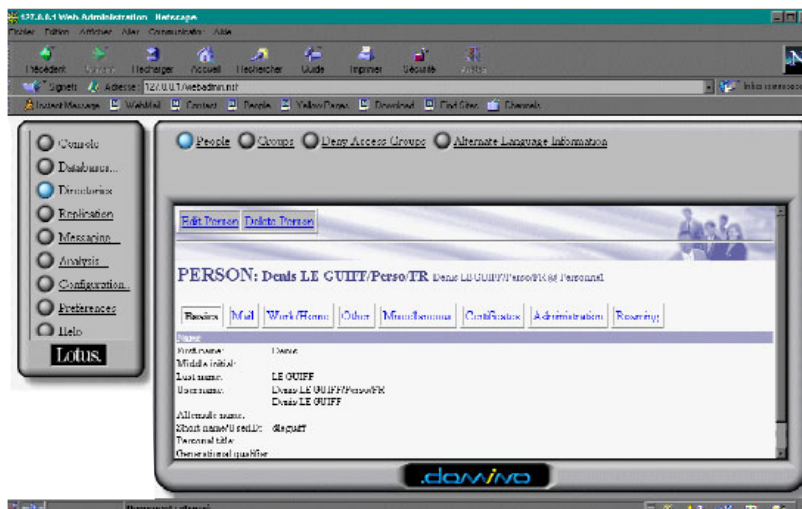
Pour

- Le support du mode ASP pour la mutualisation d'un serveur
- Le support des JSP, des servlets, et ultérieurement des langages PHP, Python, Perl
- La possibilité de personnalisation très avancée pour les utilisateurs
- Le développement à l'aide d'autres outils compatibles WebDAV

Contre

- Les bugs
- Les fonctions non implémentées
- L'occupation disque sur les postes clients

L'administrateur dispose d'une interface Web pour piloter le serveur Domino.



La création de servlets et de JSP au sein des bases Rnext se fait dans la zone Shared Code. Il est possible soit d'importer des servlets et des JSP créés par d'autres outils, soit de coder directement dans Designer le servlet ou le JSP. Un squelette de code Java est fourni en standard lors de la création d'un nouveau servlet. Un squelette de code HTML contenant des instructions dédiées aux JSP est proposé en standard lors de la création d'un JSP.

Lotus met plutôt l'accent sur la facilité d'utilisation que sur une refonte complète de l'environnement. Passons en revue les nouveautés à destination des administrateurs, des développeurs, mais aussi des utilisateurs.

Pour l'administrateur, Lotus fournit de nombreux outils optimisés en vue de simplifier l'administration des serveurs Rnext. Tout d'abord, l'administrateur dispose d'une interface Web pour administrer son serveur. Afin de surveiller l'état de ce dernier, il est désormais possible de définir des seuils statistiques que l'on souhaite suivre en temps réel. Ces rapports temps réel, affichés sous forme graphique, peuvent aussi être sauvegardés et faire ultérieurement l'objet d'une étude détaillée. La console serveur peut également être paramétrée de façon à faire plus rapidement ressortir les messages importants grâce à des codes couleur. Les commandes utilisables au niveau de la console disposent désormais d'une aide en ligne. Des profils d'administration permettent de répartir des droits d'administration différents aux administrateurs. Avec la Rnext, à chaque connexion d'un client, le serveur récupère des informations sur la plate-forme et la version d'OS utilisées et dresse ainsi un inventaire exhaustif de l'ensemble des postes clients.

Outils d'administration

Lors de l'installation d'un serveur Rnext, il est possible de mettre automatiquement à jour les listes de contrôle d'accès des applications système et des modèles associés. Lotus a mis en place de nouvelles règles de gestion des quotas des boîtes aux lettres. Ces paramètres permettent de définir la politique à tenir lorsqu'un utilisateur dépasse son quota ou son seuil.

Les applications Notes/Domino accessibles avec un browser en mode sécurisé se généralisant, Lotus a revu son mode de gestion des mots de passe http en permettant de synchroniser le mot de passe http avec celui stocké dans l'ID Notes. Un système de changement de mot de passe à intervalles réguliers avec contrôle de la complexité du mot de passe a été implémenté. Après installation du matériel et du logiciel spécifiques au lecteur de cartes à puce, on peut configurer le client Notes afin qu'il recoure à une carte à puce pour authentifier l'utilisateur en lieu et place de la traditionnelle fenêtre de login/mot de passe.

Au niveau du protocole SMTP utilisé pour l'envoi et la réception de mails sur Internet, la sécurité n'a pas été non plus oubliée, puisqu'une gestion de listes d'exclusions a été implémentée. De nombreuses applications développées font appel à des workflows basés sur l'organisation de l'entreprise. Avec la Rnext, des champs supplémentaires ont été ajoutés dans le carnet d'adresses, permettant de lister six niveaux de hiérarchie et jusqu'à quatre personnes par niveau de hiérarchie pour chaque utilisateur. Pour ceux qui se connectent à différents postes de travail, certaines bases habituellement stockées localement le sont désormais sur le serveur. Il s'agit de l'option de roaming. Pour une gestion plus fine des profils des utilisateurs, la Rnext introduit la notion de police, qui est une règle de gestion appliquée aux utilisateurs. Ces polices concernent l'archivage, l'environnement

de travail, la sécurité, les options d'enregistrement, la configuration et la police maîtresse. La liste des options configurables à l'aide des polices est très importante car l'ensemble des paramètres utilisateur peuvent être gérés par ce biais.

Dernière optimisation notable du fonctionnement d'un serveur Notes/Domino : la gestion centralisée des annuaires. On peut désormais définir un serveur d'administration contenant l'annuaire complet, et des serveurs de configuration qui disposent d'une version allégée de cet annuaire. Les serveurs de configuration effectuant des requêtes auprès du serveur d'administration pour obtenir les informations dont ils ont besoin, et dont ils ne disposent pas en local. Ce mode de fonctionnement permet d'alléger la taille des carnets d'adresses répliquant entre les serveurs.

Services ASP et architecture Web

Le serveur Rnext supporte les principaux langages et technologies d'Internet : HTTP, Java, JavaScript, PHP, JSP, Perl, Python, servlets... Au niveau des JSP, c'est la version 1.1 qui a été implémentée et étendue à l'aide de bibliothèques de tags personnalisés autorisant l'accès aux bases de documents Notes/Domino ainsi qu'aux objets qui les constituent (vues, champs...). Ces bibliothèques permettent de s'affranchir de l'écriture de code Java de bas niveau. Le moteur de servlets, quant à lui, permet de compiler en mémoire des JSP et de fournir aux navigateurs la réponse attendue.

L'orientation ASP (Application Service Provider) se révèle être un axe important d'évolution de la Rnext, car de nombreuses entreprises mutualisent leurs ressources. Les mécanismes de refacturation et de contrôle et gestion des accès ont donc été améliorés. Dans ce contexte d'hébergement Web, le serveur Domino assure la gestion du déploiement des applications. Ainsi, une application ayant été configurée pour être déployable contiendra toutes les informations nécessaires pour que le serveur Domino puisse en extraire les composants devant être installés sur le système de fichiers du serveur.

Au niveau de l'architecture Web, la Rnext permet de supporter un nombre virtuellement illimité de sites Web. Le carnet d'adresses du domaine a été remanié pour que les informations relatives aux sites Web soient stockées dans une vue dédiée. L'environnement de gestion d'annuaire LDAP a été optimisé et une nouvelle base contient le schéma de définition de l'annuaire Rnext. La gestion de la traçabilité et du logging a été améliorée de façon à remonter plus d'informations sur les sessions utilisateurs, les activités des agents et les protocoles SMTP, HTTP, POP3, IMAP, NNTP et LDAP.

Une nouvelle tâche sur le serveur assure la mise en œuvre d'un mécanisme d'autorité de certification qui autorise la gestion des clés Notes et Internet affectées aux utilisateurs. Cette tâche permet aussi de pouvoir gérer des listes de révocation de certificats pour vérifier la validité d'un certificat.

Support des langages et protocoles Internet

Pour les développeurs, la stratégie de Lotus est de faire de Notes/Domino une plate-forme intranet/Internet complète. De gros efforts ont été

accomplis pour intégrer tous les protocoles significatifs d'Internet au sein de la Rnext.

Le support des JSP dans leur version 1.1, d'abord. Les bases applicatives Notes/Domino disposent désormais dans la rubrique Shared Code d'une zone de stockage des JSP. Il est possible d'importer des JSP ou d'en créer nativement dans l'outil Designer. Une fois présentes dans la base, les pages JSP doivent être déployées sur le serveur Domino. Or, dans la bêta 2 de la Rnext, les bases d'aide font état d'une tâche serveur dédiée à cette fonction qui, malheureusement, ne fait pas partie des exécutables livrés avec le serveur Rnext.

Les servlets, dans leur version 2.2, sont aussi supportés. Il existe, comme pour les JSP, une zone dans les bases applicatives pour les stocker. Ils peuvent être soit importés, soit développés dans Designer. Tout comme les JSP, ils doivent être déployés sur le système de fichiers du serveur Domino. La tâche dédiée à cette fonction n'est pas encore disponible, comme nous l'avons vu précédemment. La création de servlets et de JSP au sein des bases Rnext se fait dans la zone Shared Code.

Il est possible soit d'importer des servlets et des JSP créés par d'autres outils, soit de coder directement dans Designer le servlet ou le JSP. Un squelette de code Java est fourni en standard lors de la création d'un nouveau servlet. Un squelette de code HTML contenant des instructions dédiées aux JSP est proposé en standard lors de la création d'un JSP (figure 1 page précédente).

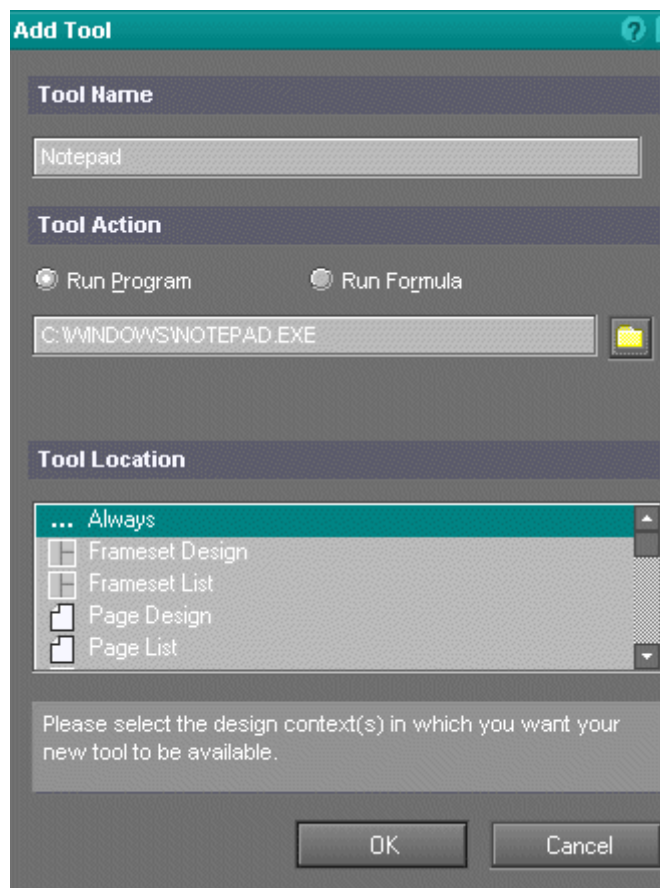
Le langage JavaScript, supporté déjà depuis plusieurs versions dans Notes/Domino, l'est maintenant dans sa version 1.4. Concernant Java, c'est le JRE 1.3.0_02 qui est supporté. Il a été aussi annoncé le support des langages Perl, Python et PHP. Si c'était déjà le cas pour le langage Perl depuis la version 4.6 de Notes/Domino, pour Python et PHP, c'est une nouveauté. Hélas, la documentation de la bêta 2 de la Rnext n'indique pas comment configurer le serveur Domino pour faire fonctionner les programmes Python et PHP.

Le dernier axe de travail des équipes de développement concerne le langage XML. Avec la Rnext, Lotus augmente encore les potentialités XML de Notes/Domino en ajoutant plusieurs classes pour exporter ou importer des applications Notes/Domino (fichiers ayant l'extension .nsf) vers des données DXL (Domino XML Language).

Le langage DXL, créé à partir d'XML, est utilisé pour décrire la structure et les données d'une application Notes/Domino. Une nouvelle classe permet de recourir à XSL pour transformer (via XSLT) des données brutes XML et leur appliquer une feuille de style XSL. Enfin, une classe SAX Parser permet de parser les données XML en entrée suivant le langage SAX (Simple Api for XML).

Ouverture et développement en équipe

La norme WebDAV (Web Distributed Authoring and Version, RFC2518 de l'IETF) décrit les extensions HTTP pour des développements distribués (HTTP Extensions for Distributed Authoring-WebDAV). Ce protocole décrit comment un outil peut modifier et gérer des fichiers sur un serveur Web distant. Rnext supporte WebDAV



En plus de l'intégration via WebDAV et des connecteurs en cours de développement vers d'autres outils de développement, Lotus propose dans la Rnext de faire de Designer la boîte à outils du développeur d'applications Web en permettant d'appeler tous les outils nécessaires au développement d'une application Web (éditeur HTML, outil de traitement d'image, compilateur Java...).

données. Ces nouvelles ressources permettent de définir des feuilles de style qui seront mises en œuvre pour des masques ou des pages, et des descriptions de connexions à des bases de données. Les feuilles de style sont des ressources externes que l'on intègre dans la base, comme des images ou des fichiers. Pour ce qui est des connexions de données, ce sont des fiches descriptives de connexion à une base de données, dans lesquelles on indique à quelle base on se connecte, quel login et quel mot de passe doivent être utilisés, et à quel type d'objet de la base on souhaite accéder (table, vue ou procédure).

L'interface de gestion des agents a été revue pour permettre plus facilement d'activer ou de désactiver un agent et de pouvoir le signer, le tout grâce à des boutons. Les barres d'action remplacent en fait les anciennes actions et leur aspect "bouton". Il existe maintenant deux types d'actions : les boutons d'action et les cases à cocher. Les traitements associés se font dans l'événement OnClick. Les boutons d'action peuvent aussi être regroupés sous forme de liste déroulante. A ces nouvelles actions peuvent être associées des images fournies par l'utilisateur, qui peut aussi définir précisément le positionnement de ces boutons et leurs caractéristiques (taille, couleur, bordure...). Grâce à toutes ces nouveautés, on obtient un contrôle esthétique complet sur le rendu des boutons d'action, qui sont mieux intégrés dans les applications Web (figure 3 page suivante). La personnalisation des boutons d'action permet un meilleur contrôle esthétique du rendu de l'application.

Les layers, enfin, viennent compléter cette gestion du rendu de l'application en définissant des "calques" dans les masques sur lesquels on peut disposer des objets (champs, textes, images...). Le positionnement de ces calques s'effectue en coordonnées absolues ou par déplacement des éléments.

Côté client

La première évolution visible de l'environnement utilisateur de la Rnext est la fenêtre de connexion, qui a été complètement redéveloppée. Ainsi, l'utilisateur peut sélectionner quels identifiant et document site choisir. La page d'accueil devient entièrement paramétrable en autorisant le changement de couleurs, textes, liens, l'accès au système de fichiers... Sur la partie droite de l'écran (masquée par défaut) apparaît la section Quick Notes, qui regroupe les tâches que l'utilisateur effectue le plus fréquemment.

Autre nouveauté, la possibilité de changer dynamiquement la taille des fenêtres de dialogue, les rendant plus lisibles. Les anciens "smarticons" ont été remplacés par des barres d'outils se conformant plus à l'aspect des applications Microsoft.

Il est désormais possible de "bookmarker" plusieurs bases en même temps puis de les mettre dans un dossier par glisser-déposer. Deux menus supplémentaires, Tri et Vue, optimisent le classement des signets. Un nouveau dossier intitulé Démarrage permet de définir les bases qui seront systématiquement ouvertes lors du démarrage du client Notes. Enfin, au niveau des signets, le dossier historique garde une trace des bases consultées dans les sept jours qui précèdent.

pour les éléments susceptibles d'être déployés sur le système de fichiers du serveur Web distant : les JSP, les servlets, les images, les fichiers et les feuilles de style.

Des outils comme IE5, Macromedia Dreamweaver supportent aussi WebDAV. Il devrait donc bientôt être possible de développer avec Dreamweaver dans Notes/Domino. En complément des spécifications du modèle JavaServlet 2.2, Rnext supporte les fichiers WAR (Web Application Archive), qui sont des JAR (Java Archive) adaptés aux contraintes de déploiement des applications.

Autre option intéressante : la possibilité d'ajouter des outils au sein des menus de Designer. Ce mode de fonctionnement permet donc de pouvoir appeler tous les outils de développement dont le développeur peut avoir besoin à un instant donné sans avoir à quitter Designer. Il est aussi possible de rendre contextuel ce menu en définissant à quel moment les outils sont accessibles (figure 2 ci-dessus).

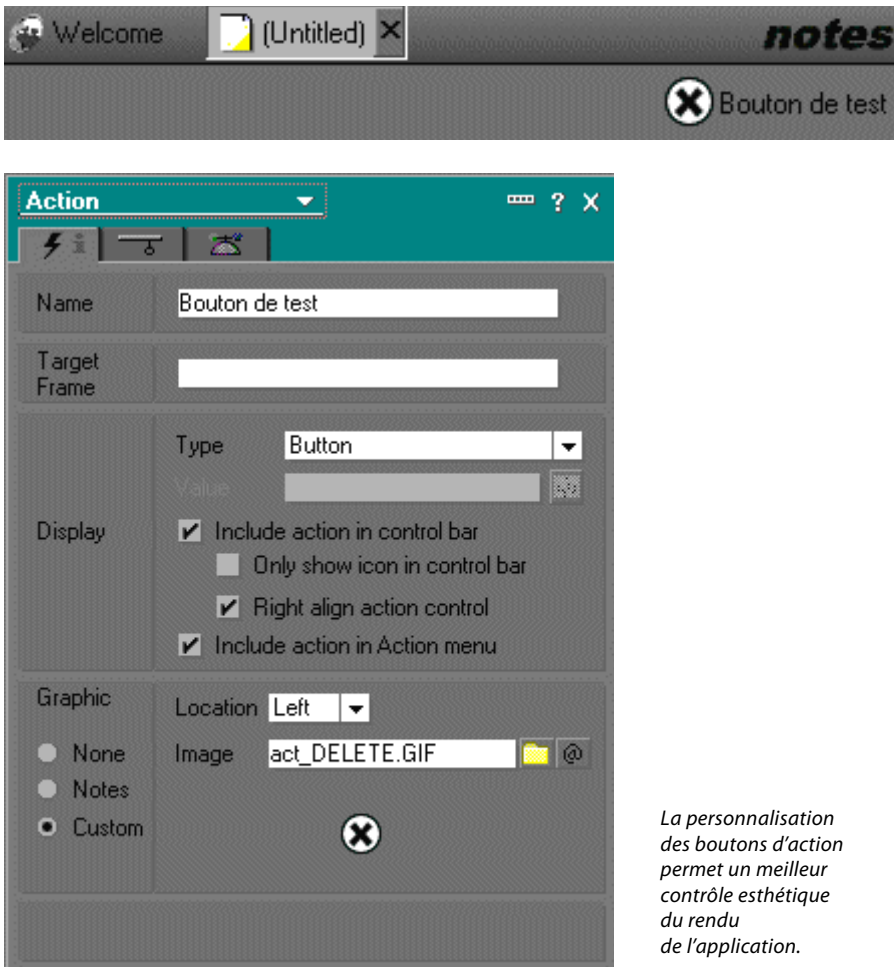
En plus de l'intégration via WebDAV et des connecteurs en cours de développement vers d'autres outils de développement, Lotus propose dans la Rnext de faire de Designer la boîte à outils du développeur d'applications Web en permettant d'appeler tous les outils nécessaires au développement d'une application Web (éditeur HTML, outil de traitement d'image, compilateur Java...). La Rnext propose un mécanisme de travail collaboratif afin de verrouiller des éléments de design. Ainsi, un développeur intervenant sur une vue peut bloquer celle-ci en modification, de façon à ce que lui seul puisse la modifier.

Langages et objets Notes

Pour redonner vie au langage de formule encore utilisé pour sa simplicité et sa rapidité, Lotus a ajouté un certain nombre de nouvelles fonctions qui lui faisaient cruellement défaut, telles que la gestion de boucles (@For, @While, @DoWhile), le traitement des chaînes (@Compare, @Sort, @Transform), les manipulations de champs (@GetValue, @ThisName et @ThisValue), et des fonctions diverses @Nothing, @Count, @IfError. L'éditeur de fonctions présente désormais au développeur un pop-up avec les différents mots clés utilisables. Lorsque le mot clé est choisi, un nouveau pop-up apparaît pour présenter la syntaxe de la fonction désirée. De plus, un analyseur syntaxique effectue une coloration syntaxique, et lorsque toute la formule est terminée, il est possible de la "compiler" pour valider sa cohérence. Le langage de formules reste malgré tout un langage interprété.

Il apparaît que JavaScript et LotusScript peuvent intervenir sur des événements similaires (entrée et sortie de champ par exemple). Lotus a donc préconisé avec la Rnext de ne plus utiliser les événements LotusScript Entering et Exiting, mais les événements JavaScript OnBlur et OnMouseOut.

Nous avons évoqué précédemment les deux ressources communes (Shared Code) de type JSP et servlets qui avaient été introduites avec la Rnext. Il en existe deux autres, qui sont les feuilles de style et les connexions de



La personnalisation des boutons d'action permet un meilleur contrôle esthétique du rendu de l'application.

Messagerie et agenda

La messagerie et l'agenda bénéficient aussi de leur lot de nouveautés. Il est désormais possible de paramétrer à volonté le dossier Courriers en arrivée. On peut inverser l'ordre des colonnes par simple glisser-déplacer du titre de la colonne. On peut aussi masquer ou afficher des colonnes, redimensionner leur largeur et indiquer l'ordre de classement des documents. Tous ces paramétrages effectués à la volée sont ensuite conservés et seront disponibles à la connexion suivante. Enfin, l'utilisateur peut décider de demander à ce que le dossier Courriers en arrivée se rafraîchisse automatiquement lorsqu'un nouveau mail lui est distribué. Il existe un nouveau bouton servant à répondre à un message, avec un historique à la mode Internet.

FICHE TECHNIQUE

Plate-forme :

- client : MS Windows 95/98/ME/NT/2000, Macintosh
 - outil de développement : MS Windows 95/98/ME/NT/2000
 - serveur : MS Windows NT/2000, IBM AIX, Linux, Sun Solaris/Sparc

URL : www.notes.net/rnext. La bêta courante y est téléchargeable (environ 125 Mo pour les clients et 150 Mo pour le serveur), un didacticiel est disponible ainsi que de nombreux articles du Webzine Iris Today

Prix : non communiqué, sortie courant 2002

La gestion des attachements a nettement changé. En cliquant sur un attachement, une nouvelle fenêtre avec cinq boutons apparaît et permet de lancer le visualiseur, d'enregistrer l'attachement sur le disque, d'ouvrir l'attachement en lançant l'application qui a servi à le créer, de supprimer l'attachement du corps du document, et d'ouvrir le document en modification et enregistrer les modifications dans l'attachement.

Pour la partie Agenda, de nombreuses améliorations ont été apportées. Les principales visent à rendre plus lisible l'agenda. Pour cela, une coloration par défaut des types d'entrées d'agenda a été implémentée. En plus des traditionnelles possibilités de visualisation de l'agenda, on peut maintenant obtenir une liste de rendez-vous classés par date, grâce à l'option Résumé. Le document d'entrée d'agenda a été redessiné pour offrir une plus grande lisibilité.

L'intégration de Lotus Notes/Domino avec Sametime (le produit de réunion instantanée de Lotus : chat, tableau blanc, partage d'applications) a également été prise en compte. Il est ainsi possible de préremplir le tableau blanc, qui sera partagé via Sametime lors de la réunion. La gestion de la délégation de l'agenda a été aussi améliorée en permettant à une secrétaire de gérer l'agenda de son patron sans avoir à utiliser la boîte aux lettres de ce dernier.

Avec les nouvelles options de vues d'agenda, il est facile de déplacer un rendez-vous : il suffit de faire un glisser-déposer qui replanifiera le rendez-vous. De même, en "tirant" le contour de l'entrée d'agenda dans la vue, on peut modifier la durée

du rendez-vous. Enfin, un simple clic de souris suffit pour éditer les données de l'entrée d'agenda affichées dans la vue.

Quelques améliorations ont été apportées à la page Réplicateur, qui peut désormais présenter les bases comme des signets. La réplique sélective, qui permet de ne répliquer qu'un ensemble de documents d'une base, s'effectue en sélectionnant un ensemble de documents – ou une vue entière, un dossier – et en le glissant-déposant sur l'icône du dossier Réplicateur. Enfin, les bases peuvent aussi être regroupées au sein de dossiers, afin de répliquer un ensemble de bases simultanément.

Grâce au multithreading, les opérations longues se font en tâche de fond et ne bloquent plus l'utilisateur. A l'aide de la barre d'avancement, ce dernier sait à tout instant où en est la tâche qu'il a lancée. En bloquant les ressources, il peut interdire l'édition d'un document déjà en cours de modification. Ainsi, les risques de conflits d'enregistrement peuvent être supprimés.

Comme on vient de le voir, de nombreuses améliorations ont été ajoutées au code de la Rnext. L'ensemble de ces modifications semble répondre à deux mots d'ordre : personnalisation et optimisation. Personnalisation d'abord, car de nombreuses options tant au niveau du serveur que des différents clients permettent aux administrateurs, aux développeurs et aux utilisateurs de personnaliser leur environnement Notes/Domino, pour le rendre plus pratique à utiliser et plus intuitif. Optimisation ensuite, car d'autres améliorations rendent plus efficaces certains traitements (mécanisme de réplique, multithreading, supervision du système...). Certes, ces nouvelles fonctions restent encore partielles ou boguées dans la bêta 2, mais elles seront toutes opérationnelles lors de la sortie définitive du produit l'année prochaine.

Il est important de noter que Lotus, en mettant en place ce processus de bêtas, tient à récupérer le maximum de retours d'expériences de la part de ses clients (par le biais du forum dédié à la Rnext sur www.notes.net/rnext et à la base de collecte des bugs) pour permettre d'améliorer son produit phare, et mieux répondre ainsi aux attentes de ses clients. Déjà particulièrement intéressante, la bêta 2 est une ébauche en devenir, qui ne demande qu'à évoluer. N'hésitez pas à télécharger les futures versions bêta de la Rnext, car de nombreuses autres améliorations sont encore dans les cartons des développeurs de Lotus et devraient prochainement voir le jour. ●

ALLER PLUS LOIN

www.notes.net/today.nsf

De très bons articles du Webzine Iris Today sont disponibles sur ce site. On pourra y lire les articles suivants :

Enhancements of the formula language in Rnext, par Michelle Mahoney et Robert Perron

Domino Designer Rnext technical overview, par David DeJean
Domino Rnext technical overview, par Jane Hosie-Boumar et Christie Williams

A tutorial on Web site addressing, par John Chamberlain

Notes Rnext technical overview, par Craig Lordan

L'interview de Jeff Calow sur les nouvelles technologies du Web dans Domino Rnext, par Christie Williams

et Laura Rutherford

Overview of the Domino Rnext Web application server, par John Chamberlain