

# Technologies *Microsoft*<sup>®</sup> :

## De Visual Basic à .NET 3.0

---

*Avec nos remarques impertinentes pour les experts du monde J2EE*

*Denis Dollfus*

*Responsable Pôle .NET du Groupe SFEIR*

*Florent Dugué*

*Expert .NET du Groupe SFEIR*

*Didier Girard*

*Directeur Technique du Groupe SFEIR*

---

### **Résumé**

Avec la sortie de .NET 3.0, les technologies portées par Microsoft connaissent des évolutions majeures. Ce *white paper* les analyse en les situant dans le contexte actuel, fortement marqué par l'irruption de technologies qui soutiennent les approches SOA et qui brouillent les contours de la typologie des interfaces graphiques qu'on pensait acquise, "client léger" contre "client lourd".

## PREFACE

Disponible depuis fin 2006, la version 3.0 du framework .Net a désormais trouvé son porte parole français pour l'informatique d'entreprise, au travers de cet ouvrage rédigé par la société SFEIR.

Version 3.0 ! Vous êtes en mesure de vous interroger. Qu'en est-il des versions précédentes ? Allez-vous comprendre ce dont il s'agit si vous n'êtes pas aguerri des versions 1.1 et 2.0 ?

Et nous savons combien la mise en perspective des versions précédentes est un exercice délicat en termes de granularité et d'exhaustivité.

Soyez rassurés : en dressant un panorama synthétique des versions précédentes, les rédacteurs ont réussi cette figure –quasi-imposée dans la mesure où le framework .Net 3.0 complète la version 2.0 tout en en restant dans la philosophie d'innovation et d'unification de la plate-forme Microsoft.

Par ailleurs, à un moment où l'entreprise baigne dans un environnement hétérogène et exigeant en terme de nouveaux usages, les remarques impertinentes des experts SFEIR nous rappellent que les technologies .Net et Java apprennent l'une de l'autre, tout en soulignant aussi les différentiateurs de Microsoft.

A ce titre, le présent ouvrage est définitivement emprunt d'originalité : technique sans l'être trop, dans un esprit d'ouverture sans entrer dans la polémique.

Mes félicitations renouvelées à l'équipe SFEIR qui a réalisé un ouvrage de référence pour quiconque souhaite découvrir l'étendue des services rendus par la plate-forme Microsoft et le framework .Net pour l'entreprise

Stève SFARTZ, Architecte en Systèmes d'Information, Microsoft France

## **Remerciements**

Les rédacteurs remercient toute l'équipe SFEIR pour l'aide apportée à la préparation de ce livre blanc.

**TABLE DES MATIERES**

LES INTERFACES UTILISATEURS CLIENTS LÉGERS .....	5
ASP, première vague .....	5
ASP.NET, la refonte.....	5
Langages .....	6
Librairies de classes .....	6
Outils de développement, la simplification .....	6
ASP.NET 2.0, l'aboutissement.....	7
ASP.NET Ajax, le RIA.....	7
LES INTERFACES UTILISATEURS CLIENTS LOURDS.....	8
Les wrappers Win32, la préhistoire.....	8
Windows Forms, la renaissance .....	8
Les services de déploiement des clients RDA.....	9
LES INTERFACES OFFICE RICHES .....	10
Visual Basic for Applications.....	10
VBA permet trois types de développements.....	10
Visual Studio Tools for Office.....	10
Office Business Applications.....	11
Business Data Catalog.....	11
InfoPath.....	12
LES SERVICES WEB AVANT .NET 3.0 .....	13
ASP.NET et les services Web .....	13
Web Services Enhancements.....	14
LE FRAMEWORK .NET 3.0 .....	15
Windows Presentation Foundation .....	16
WPF Everywhere.....	17
Encore plus d'interactions homme-machine .....	17
Windows Communication Foundation .....	19
Workflow Foundation .....	21
CardSpace .....	24
LES OUTILS .....	26
Visual Studio .....	26
Team System .....	27
Les outils Expression .....	28
Et bientôt.....	29
GLOSSAIRE .....	31

## LES INTERFACES UTILISATEURS CLIENTS LÉGERS

---

*ASP, c'est comme les JSP, ni plus ni moins.*




---

*ASP.NET, c'est un peu comme l'arrivée de Struts dans le monde J2EE. Les développeurs prennent conscience qu'il est important de séparer le métier de la présentation : exit le code SQL dans les pages. ASP.Net c'est aussi la gestion d'un mode qui permet de coupler un état et des événements à des composants graphiques HTML comme ce que propose JSF.*

### **ASP, première vague**

La première version des **ASP** (Active Server Pages), dites ASP classiques, est une technologie Microsoft qui permet de développer des pages Web dynamiques. Elles utilisent des scripts insérés aux pages HTML qui sont exécutés sur le serveur Web IIS. Ces scripts peuvent intégrer des composants COM (Component Object Model) et grâce à ADO (ActiveX Data Object) établir des connexions à des bases de données.

La principale avancée des ASP par rapport aux technologies du moment (1996-2000) telles que les scripts CGI, est de fournir des composants prêts à l'emploi facilitant la gestion des sessions, des requêtes, des réponses et des états de l'application Web. Ce modèle de composants a d'ailleurs donné lieu à des développements similaires avec d'autres technologies, notamment les JSP (Java Server Pages).

### **ASP.NET, la refonte**

**ASP.NET** représente l'ensemble d'outils et de bibliothèques du Framework .NET dédié aux applications Web.

Avec l'apparition du Framework .NET en 2002, les Active Server Pages ont été entièrement repensées. ASP.NET 1.0 garde les caractéristiques de base des ASP : il combine HTML, langages de programmation, composants serveur et il s'exécute côté serveur.

En revanche certaines limitations reconnues des ASP classiques sont dépassées avec ASP.NET :

- Séparation présentation/traitements. Le style de programmation des ASP imposait au sein d'une même page un mélange de HTML et de script. La clarté du code et donc la maintenabilité de l'application Web en souffrait, il était difficile de réaliser des sites Web complexes. Pour améliorer ce point, ASP.NET permet une séparation claire entre la présentation (le HTML) et la logique applicative (par exemple le code C#). Cette séparation est rendue possible par l'introduction du concept de page de code "cachée" (code-behind) associée à chaque page HTML.
- Objets. L'orientation objet vient combler les limites de l'aspect procédural des ASP classiques en permettant, voire en imposant une structuration de la logique de traitement.
- Performances. Les pages ASP classiques n'étaient pas compilées mais interprétées ce qui entraînait des performances moindres que les technologies concurrentes. ASP.NET compile le code.
- Réutilisabilité. Une amélioration importante d'ASP.NET est l'introduction de parties de pages Web réutilisables – les contrôles serveurs. Ils permettent de développer une seule fois des parties de pages Web qui seront réutilisées dans de multiples pages.
- Homogénéité. Les ASP classiques avaient l'inconvénient de mélanger les technologies ASP et COM, forçant le développeur à passer d'un environnement de développement à un autre. A l'inverse ASP.NET

repose entièrement sur le Framework .NET, socle homogène contenant le traitement des pages Web comme les traitements métier.

- Ouverture. Les ASP classiques verrouillaient les accès à la mécanique de bas niveau de traitement de requêtes http. En revanche ASP.NET ouvre ces couches de bas niveau à la programmation. Il est ainsi possible de développer son propre code de gestion des requêtes dans le cas d'impératifs spécifiques non couverts par les fonctions par défaut d'ASP.NET.

Tous ces points sont le résultat de l'innovation fondamentale sur laquelle repose ASP.NET : le Framework .NET. Celui-ci amène d'autres ruptures.

### **Langages**

ASP.NET supporte tous les langages développés par Microsoft ou par des tierces parties conformément aux spécifications de la CLI. La CLI, pour Common Language Infrastructure, est une spécification internationale (Ecma-335) de machine virtuelle indépendante de tout langage de développement. Le CLR (Common Language Runtime) est l'implémentation commerciale de Microsoft de cette CLI. Le CLR donne déjà accès à plus de 25 langages, certains développés par Microsoft, d'autres par des éditeurs indépendants ou dans le cadre de projets open source : Cobol, Lisp, Delphi, PHP, Pascal, Ada, Prolog, Python, Ruby, etc.

### **Librairies de classes**

Microsoft fournit un ensemble de librairies (Framework Class Library) qui repose sur le CLR et qui compose le Framework .NET. Accessible depuis tous les langages évoqués plus haut, il tend à couvrir l'ensemble des fonctions et interactions nécessaires au développement d'applications, depuis les accès disque jusqu'aux Web Services. Les appels natifs au système Windows restent cependant nécessaires pour quelques points spécifiques.

ASP.NET est une partie de la Framework Class Library.

Une version "compacte" (8% de la taille du Framework standard) du Framework .NET est destinée aux périphériques mobiles, téléphones ou PDA. Il s'agit de .NET Compact Framework, qui n'implémente qu'une partie de la librairie et optimise certains traitements, notamment ceux liés à la communication distante.

### **Outils de développement, la simplification**

Au lieu d'un environnement de travail par langage, le "multilinguisme" du Framework .NET entraîne une consolidation de la palette d'outils. Visual Studio supporte ainsi les différents langages .NET. D'une manière générale la compilation, l'implémentation et le déploiement sont facilités et plus rapides.

## ASP.NET 2.0, l'aboutissement

Avec ASP.NET 2.0 le temps de développement de sites Web classiques est considérablement réduit. Les fonctionnalités récurrentes sont en effet prises en charge par de nouvelles bibliothèques du Framework .NET 2.0 consacrées :

- Les Pages Maîtres permettent de structurer par un modèle de mise en page le format d'un site. Les zones définies par les pages maîtres sont ensuite remplies dynamiquement par les contenus adéquats, typiquement une bannière d'entête, un menu de navigation, une colonne de news, etc.
- Les Web Parts permettent la personnalisation des pages par l'utilisateur final.
- Des nouvelles bibliothèques de classes facilitent l'implémentation de fonctionnalités souvent réécrites telles que la gestion des menus et la navigation du site, la gestion des utilisateurs et des droits associés et la gestion du cache.

ASP.NET 1.0 suit le cycle de .NET Framework 1.1 et sa maintenance s'arrêtera en 2007. Quant à ASP.NET 2.0, lié à .NET Framework 2.0, il sera supporté jusqu'en 2011.



## ASP.NET Ajax, le RIA

Le Framework Ajax de Microsoft, anciennement dénommé Atlas, permet d'ajouter aux clients légers l'ergonomie et un look & feel que le HTML seul ne permet pas. Comme tous les Frameworks Ajax, ASP.NET Ajax repose sur des bibliothèques JavaScript offrant au minimum le téléchargement asynchrone des données et le rafraîchissement partiel, plus agréable à l'utilisateur que le rechargement total accompagné de la disparition de la page que nous connaissons tous.

L'Ajax de Microsoft n'est pas entièrement lié à ASP.NET. Plus précisément il est composé de trois blocs distincts :

- Microsoft Ajax Library est un ensemble de classes JavaScript indépendant de tout environnement serveur, y compris ASP.NET.
- ASP.NET 2.0 Ajax Extensions, est une bibliothèque de classes .NET destinée à faciliter l'usage d'Ajax lors du développement sous ASP.NET. Ces classes sont chargées côté serveur. En termes de développement, elles s'intègrent facilement à Visual Studio pour simplifier le développement Ajax réputé ardu.
- ASP.NET AJAX Control Toolkit est une bibliothèque de contrôles qui reposent sur les extensions Ajax et permettent de mettre en œuvre rapidement sous Visual Studio des éléments graphiques tels que des boîtes de dialogue modales, des panneaux rétractables, etc.

---

*Les Frameworks Ajax fleurissent dans le monde Java. Il est ainsi difficile de s'y retrouver. Ceci dit, GWT qui semble une bonne solution pour faire de l'Ajax n'a pas d'équivalent soutenu par Microsoft.*

## LES INTERFACES UTILISATEURS CLIENTS LOURDS

Les clients **RDA** (Rich Desktop Application) commencent à avoir une longue histoire. Les avancées de .NET 3.0 laissent à penser qu'il se prépare une rupture.

### **Les wrappers Win32, la préhistoire**

Les systèmes d'exploitation Windows contiennent nativement toutes les bibliothèques nécessaires à l'affichage de fenêtres Windows. Cependant l'accès à ces fonctions de gestion de fenêtre, dites Win32, est extrêmement coûteux en temps de développement. Microsoft fournit plusieurs bibliothèques qui cachent la complexité Win32 derrière des objets qui apportent une mise en œuvre rapide en contrepartie de fonctionnalités restreintes.

Microsoft Foundation Classes (MFC) est la bibliothèque dédiée aux développeurs C++. Elle a été à l'origine de très nombreuses applications Windows depuis sa première version en 1992.

Les VB Forms sont un autre ensemble d'objets pour les développeurs Visual Basic. Reposant sur une construction visuelle des fenêtres par simple glisser-déplacer, la rapidité de mise en œuvre de cette technologie associée à la simplicité de Visual Basic (VB) lui-même a fait de cet environnement de développement l'un des plus utilisés au monde.

### **Windows Forms, la renaissance**

Windows Forms, introduit avec .NET 1.0 et communément appelé WinForms, est une bibliothèque d'interfaces graphiques qui peut être utilisée à partir des langages autorisés par le Framework .NET. Elle comprend tout le nécessaire pour développer des applications Windows classiques. WinForms couplé aux outils de développement rapide de Visual Studio permet d'atteindre depuis la version 2.0 une productivité approchant celle de Visual Basic 6.0.

Un des points clés de la productivité est le degré de facilité avec laquelle le développeur lie les contrôles graphiques – tableaux de valeurs, listes déroulantes, etc. – avec les données : c'est le Data Binding. .NET 2.0 apporte des améliorations au data binding grâce à de nouveaux objets dédiés exploitables aussi bien en client lourd qu'en client léger. De plus le data binding de .NET 2.0 offre des accès non plus seulement aux sources de données orientées SGBD (les Datasets) mais aux objets métier via une nouvelle classe ObjectDataSource. Que ce soit en mode WinForms ou Web Forms, ce binding est bidirectionnel depuis .NET 2.0. Il reste d'ailleurs exploitable depuis les extensions serveur AJAX.

---

*Pas vraiment d'équivalent dans le monde de Java, c'est plus simple à utiliser que du Swing et plus puissant que du SWT.*

---

*Dans le monde Java il y a  
Java Web Start et Eclipse  
Update qui adresse le même  
domaine que ClickOnce*

## **Les services de déploiement des clients RDA**

Windows Installer est le service Windows dédié à la gestion des déploiements de clients RDA. Robuste et validé par un historique avantageux, on lui reconnaît des limites que vient combler **ClickOnce**.

ClickOnce est une technologie de déploiement incluse avec .NET 2.0 qui lève certaines limites de Windows Installer et apporte des nouvelles fonctionnalités :

- La gestion des mises à jour. Une application déployée par ClickOnce après téléchargement depuis un site Web ou depuis un disque réseau partagé peut vérifier à chaque lancement (ou à d'autres moments au choix du concepteur) si une mise à jour est disponible. Cette vérification n'implique aucun développement particulier. Cette fonctionnalité lève intégralement l'handicap habituellement reconnu des RDA par rapport aux RIA, c'est-à-dire les coûts induits par le mode de distribution.
- La sécurité des applications. Windows Installer impacte de nombreux sous-systèmes de l'OS, et requiert donc la plupart du temps des privilèges élevés pour installer les logiciels. ClickOnce apporte la possibilité de ne demander à l'installation que les privilèges nécessaires.
- Les déploiements Windows Installer reposent souvent sur des composants partagés dont la gestion des versions relève parfois du casse-tête. L'approche de ClickOnce résout ce problème en forçant une distribution à être entièrement autonome, sans possibilité d'interagir avec les autres composants du système.

## LES INTERFACES OFFICE RICHES



L'intégration des applications avec les outils bureautiques est un facteur essentiel de la productivité du poste client. Les clients riches, et à plus forte raison les clients légers, offrent peu de possibilité d'interaction avec ces applications, parmi lesquelles la suite Microsoft Office est la plus répandue. Microsoft propose plusieurs approches pour faciliter l'intégration des applications métier à Office. Office 2007 amène quelques changements majeurs qui visent à améliorer encore l'accès au développement d'applications riches de type **ROA** (Rich Office Applications).

### Visual Basic for Applications



VBA est une déclinaison de Visual Basic dédié à l'automatisation des applications bureautiques. VBA, sorti avec Office 97, a été le support de nombreux développements autour des applications de la suite Office tels qu'Excel et Word, mais aussi PowerPoint, MapPoint, Visio et Outlook.

### VBA permet trois types de développements.

- Les développements orientés documents. Le code VBA est attaché au document, il se manifeste lorsqu'un utilisateur ouvre le document.
- Les développements orientés modèles. Le code VBA est intégré aux modèles disponibles, typiquement réunis dans un répertoire partagé accessible depuis tous les postes.
- Les développements orientés applications. Dans ce cas le code VBA est chargé par l'application au démarrage sans qu'un document soit nécessaire. Dans ce cas on parle d'add-in à l'application. Par exemple pour Excel, l'add-in VBA se présente sous forme d'un fichier d'extension .xla. D'autres langages peuvent servir à générer des Add-In. En C++ notamment, on peut développer pour Excel des fichiers d'extension .xll.

VBA est un bon outil pour personnaliser l'environnement de travail Office. Office 2007 conserve cette technologie. Ses limitations apparaissent dès que l'industrialisation des développements et de la maintenance deviennent nécessaires, ce qui est vite le cas lorsque le volume et les origines (différentes équipes, développeurs isolés, etc.) du code VBA vont croissant.

### Visual Studio Tools for Office



Visual Studio Tools for Office (VSTO) vient combler les limites de VBA. Il offre, comme VBA, l'accès aux applications Office via un modèle objet. Mais il offre cet accès dans l'environnement .NET, ce qui amène l'ergonomie et les fonctionnalités dirigées vers le développement d'applications de qualité (tests unitaires, contrôle des sources, programmation orientée objet) de Visual Studio et des langages .NET. En somme, le développement d'applications Office intégrées à d'autres systèmes tels que bases de données et services web a tout intérêt à se reposer sur VSTO plutôt que VBA.

La personnalisation d'Office via VSTO couvre la quasi-totalité de l'interface graphique des applications Office : menus, ruban d'Office 2007, fenêtre des

tâches, etc. Ces éléments de l'interface graphique communiquent avec des composants .Net qui accèdent aux services de l'entreprise. Office devient ainsi le support IHM d'applications intégrées.

Au-delà des fonctionnalités de personnalisations de l'IHM, les innovations d'Office 2007 amènent encore plus loin les possibilités d'intégration de la suite bureautique dans les applications d'entreprise. C'est à travers le concept des Office Business Applications que Microsoft décrit les architectures qui tirent parti de ces nouveautés.

### **Office Business Applications**

Office System est un des moyens promus par Microsoft pour résoudre la problématique du "last-mile of productivity". A l'image des réseaux de communication dont le dernier kilomètre de connectivité peut être le plus coûteux, les systèmes d'information d'entreprise, même correctement conçus côté back-end, pêchent souvent par un manque d'intégration sur le poste de l'utilisateur final. Ou est le gain en productivité d'une architecture SOA si les processus métier sont au final soumis aux aléas du copier-coller et à l'import-export de fichiers ?

L'approche Office Business Applications est centrée sur Office System. Elle s'applique à toutes les couches de l'architecture depuis les données jusqu'aux interactions homme-machine. L'extensibilité de l'interface graphique dont on a déjà parlé est importante mais ne suffit pas à garantir une intégration "en profondeur". Ce sont les nouveaux composants aptes à s'exécuter au sein des couches business et data qui font la différence :

- L'arrivée d'Open XML, nouveau format XML des documents Office, permet aujourd'hui de manipuler les documents Office sans devoir instancier les exécutables. La voie est donc ouverte pour mettre en place des traitements métiers sur les documents sur les couches non IHM.
- Sharepoint héberge Workflow Foundation (WF, un des composants de .Net 3.0, et décrit plus loin dans ce document). Sharepoint est livré avec un ensemble de workflows prédéfinis qui couvre les situations les plus communes telles que les circuits de validation de documents. La définition de workflow spécifiques au métier de l'entreprise est possible soit directement depuis Sharepoint Designer soit depuis Visual Studio.
- Business Data Catalog fait le lien entre les données métier dans Office SharePoint Server 2007.

### **Business Data Catalog**

Cette nouvelle fonctionnalité introduite avec Office SharePoint Server 2007 fournit un moyen d'intégrer les données métier (SAP, Siebel, etc) dans SharePoint sans écriture de code. Business Data Catalog (BDC) est une couche d'indirection et d'abstraction au-dessus des services métiers. Qu'il s'agisse de services Web internes ou externes, de base de données ou

d'annuaires OLAP, BDC en produit une vue homogène dont la description repose sur un méta modèle.

La mise au point des méta-données de description ne requiert pas d'expertise de développement, mais demande toutefois une bonne connaissance d'XML et des bases de données.

Une fois enregistrés dans le BDC, les données et services peuvent être consommés sans plus de développement. De nouvelles extensions de SharePoint permettent de les inclure dans des Web Parts, listes, formulaires de recherche, etc.



### InfoPath

Office InfoPath répond au besoin bien précis mais omniprésent de la gestion des formulaires de saisie, depuis la collecte au sein une interface graphique ergonomique jusqu'à l'intégration des données collectées dans les processus métier. InfoPath est une application autonome qui fonctionne sous deux modes distincts.

---

*Pas d'équivalent pur en java.*

- En mode « design », l'auteur du formulaire choisit sa ou ses sources de données parmi les sources proposées par InfoPath, SQL Server, fichier XML ou service Web. La version 2007 offre des sources de données supplémentaires comme la connexion à une bibliothèque de sources SharePoint. Le designer compose ensuite son formulaire et travaille la mise en forme par les outils habituels d'Office. Enfin il publie son formulaire. Plusieurs moyens de diffusion sont disponibles, dont la publication vers un site SharePoint, par email ou par simple dépôt sur un dossier partagé.
- En mode « utilisateur », le formulaire est prêt à être rempli et envoyé. La version 2003 nécessitait que l'utilisateur dispose d'InfoPath sur son poste pour pouvoir remplir le formulaire. Ce n'est plus le cas avec Office 2007. Les formulaires peuvent dorénavant être saisis depuis un navigateur Web. C'est un mode de développement RAD pour des applications de saisie de données. Les formulaires peuvent être définis par des utilisateurs finaux, et les équipes techniques peuvent définir des modèles de formulaires réutilisables. Enfin, les données saisies au format XML peuvent être réinjectés dans le SI, qu'il s'agisse d'applications .Net, SharePoint, des orchestrations Biztalk ou WF ou encore des services J2EE.

Les aspects Biztalk dépassent le cadre de cette étude. Retenons que Biztalk complète le framework .Net au travers de fonctionnalités complémentaires. Notamment, l'orchestration de processus distribués, longues durées ou encore transactions, mais propose aussi un moteur de règles métiers, et des aspects intégration (connecteurs aux données et applications du SI), le tout complété de métriques d'activités (Business Activity Monitoring)

Les développeurs peuvent aussi intégrer des formulaires InfoPath directement dans leurs applications WinForms avec VSTO SE. Le XML résultant de la publication peut ensuite être intercepté par la fenêtre hôte et initier des traitements spécifiques.

## LES SERVICES WEB AVANT .NET 3.0

A l'heure où les architectures orientées services (SOA) devraient aborder leur phase de maturité, les standards autour des services Web ne cessent d'évoluer. Il devient difficile pour les architectes de faire le bon choix parmi les différentes spécifications portant sur la communication, la sécurité, la fiabilité et les transactions. Les développeurs sont eux face à une multitude d'implémentations qui proposent leur propre interprétation des standards. Dans ces conditions, le risque est grand d'aboutir à des systèmes à l'interopérabilité faible, dont la maintenance et l'évolution requièrent des manipulations explicites au niveau SOAP, base de la grammaire des services Web, au détriment de la productivité, de la conception et de la maintenabilité du système.

Les éditeurs d'outils de développement ont intérêt à s'entendre pour tenir les promesses d'interopérabilité et de productivité du SOA, car d'autres technologies open-source pourraient émerger, en témoigne le succès naissant des architectures REST.

Microsoft participe à cet effort en proposant des outils qui cachent la complexité des standards derrière des bibliothèques de composants.

### **ASP.NET et les services Web**



Depuis la première version de .NET, Visual Studio permet de consommer un service Web et d'en créer en quelques clics (ASMX1.1). Au-delà de cette facilité, la mise au point des caractéristiques étendues du service est en ASMX1.1 à la charge du développeur, avec le risque de produire un service Web peu interopérable car muni d'une configuration "exotique" des différentes spécifications de sécurité et d'encodage. Côté consommation des services Web, des points d'extension rendent possible les traitements spécifiques sur les messages, garantissant l'exploitation d'un service aux caractéristiques imprévues.

La diversité des technologies de transport, encodage, cryptographie, et l'évolution des standards pour chacune d'elles a justifié la création d'un organisme, WS-I (Web Service Interoperability), chargé d'identifier et de conseiller des profils de standards. Un profil est essentiellement un ensemble de spécifications versionnées. WS-I Basic Profil 1.1, par exemple, définit un standard de conformité basé sur SOAP 1.1, WSDL 1.1 et UDDI V2, standards faisant eux-mêmes référence à XML1.0, http 1.1, SSL 3.0, etc.

Avec ASMX 2.0, sous-partie de .NET 2.0, la pile de composants Web Services s'enrichit d'un moyen de rendre un service Web conforme à WS-I Basic Profil 1.1 en une instruction précédant l'interface du service. .NET 2.0 supporte aussi SOAP 1.2.

Pour rester en phase avec l'évolution des standards et étendre la déclaration de conformité à la sécurité et à la fiabilité des échanges, Microsoft délivre aussi des compléments sous la forme d'une librairie qui étend ASMX via les points d'extension prévus à cet effet, WS Enhancements.

### **Web Services Enhancements**

**WSE** (Web Service Enhancements) est librement téléchargeable depuis le site de Microsoft. Une fois installé, le développeur peut déclarer la conformité des services Web qu'il conçoit en prenant en compte les spécifications des différentes entités de standardisation, WS-I ou l'OASIS.

WSE fournit ainsi une implémentation de spécifications WS-\*. La version actuelle, WSE 3.0, propose par exemple WS-Security 1.0 and 1.1, WS-ReliableMessaging, WS-Trust, WS-SecureConversation, WS-Federation, WS-Addressing, SOAP Message Transmission Optimization Mechanism (MTOM), SOAP 1.1 et SOAP 1.2.

Nous verrons plus loin que .NET 3.0 apporte une nouvelle façon d'aborder la communication inter-applicative qui englobe et dépasse les services Web.

## LE FRAMEWORK .NET 3.0



Par rapport à .NET 2.0, .NET 3.0 est un simple ajout de bibliothèques de classes : ce n'est donc pas une nouvelle version du Framework, comme l'est le 2.0 par rapport au 1.0/1.1. En conséquence il n'y aura pas de problème de migration puisque les fichiers .dll de .NET 2.0 font toujours partie de la version 3.0 sans aucun changement de version.

Le Framework .NET 3.0 est essentiellement composé de quatre sous-ensembles :

- Windows Presentation Foundation (WPF, anciennement « Avalon »)
- Windows Communication Foundation (WCF, anciennement « Indigo »)
- Windows Workflow Foundation (WF)
- Windows CardSpace (nom de code « InfoCard »)

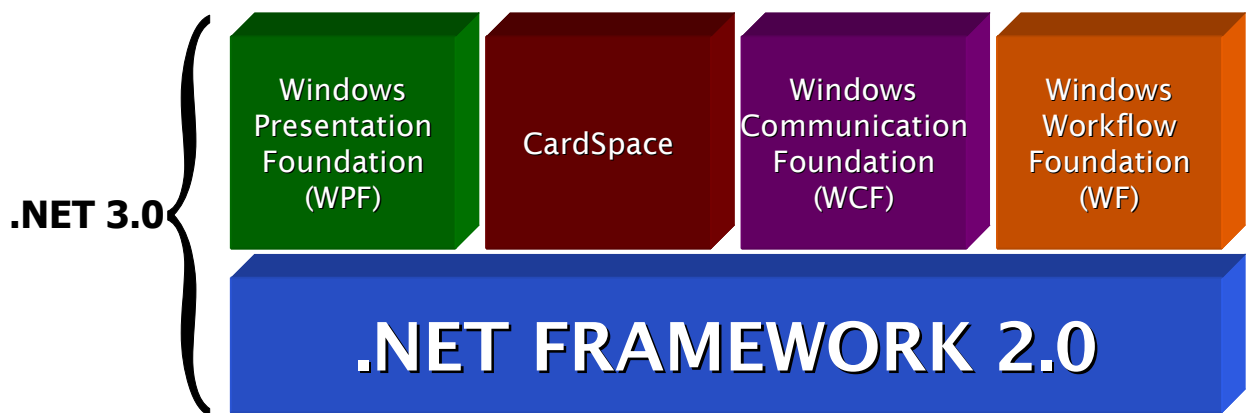


Figure 1. Les nouveaux composants s'appuient sur .NET 2.0

## Windows Presentation Foundation

*WPF c'est un peu une révolution pour les développeurs d'application desktop de nouvelle génération. Cette technologie ouvre un monde nouveau dans le domaine du développement des IHM. La technologie est là, les patterns d'utilisation restent à définir.*

Windows Presentation Foundation est une nouvelle technologie Microsoft de gestion des interfaces graphiques. Composant essentielle de .NET 3.0, elle amène des ruptures à plusieurs points de vue :

- Le look & feel et la qualité des graphiques changent la relation entre l'utilisateur et l'application. WPF permet en effet de présenter des interfaces qui comportent des animations, des effets de transition, de transparence et de 3D.
- La conception et le développement des applications WPF permettent voire imposent une nouvelle organisation du développement. Le travail de design peut être conduit indépendamment du travail de codage de la logique applicative et de l'accès aux données. L'élément clé dans cette séparation des tâches est le XAML (Cf. ci-dessous).
- La technologie elle-même fait rupture. Côté matériel, au lieu de s'appuyer sur le processeur 2D de la carte graphique, l'affichage des visuels WPF tire parti de la puissance de traitement des processeurs 3D habituellement inexploitée par les applications d'entreprise. Côté logiciel, DirectX est le point d'accès à cette puissance. Les API Win32 d'accès aux fonctions de dessin 2D ne sont plus employées.



Figure 2. Prototype d'agenda développé en WPF

---

*XAML, plus qu'un langage de description d'IHM, c'est d'abord un langage de sérialisation d'instances. // n'y a pas vraiment d'équivalent dans d'autres langages ou technologies. Certains comparent XUL ou XForms avec XAML, mais ces technologies sont très différentes : XAML est un format de sérialisation d'instances de classes, alors que XUL et XForms sont indépendants de toute implémentation. Le même XAML sert d'ailleurs à persister les workflows de WF ou tout autre objet .NET en mémoire.*

**XAML** est l'élément central de description des interfaces graphiques WPF. Indépendamment de WPF, XAML est une grammaire XML qui permet de définir une arborescence d'objets. .NET 3.0 fournit les composants nécessaires pour lire un document XAML et obtenir l'objet .NET racine du document et inversement pour sauvegarder un objet (WPF ou non) et ses objets rattachés sous forme de fichier XAML.

Appliqué à la logique de Windows Presentation Foundation, cela signifie que les styles, modèles de comportement, animations, contrôles utilisateur et sources de données peuvent être décrits de manière déclarative dans un fichier XML. XAML permet de distinguer la conception de l'interface et son code sous-jacent, facilitant la collaboration entre développeurs et designers.

XAML browser application (**XBAP**) désigne un mode d'exécution d'une application XAML dans lequel l'interface graphique est captive d'Internet Explorer. Une application XBAP est typiquement **RIA** puisque contrainte par des limites de sécurité qui l'empêche d'interagir avec le bureau de l'utilisateur, le système de fichier ou la base de registre. Elle va donc obtenir ses données de machines distantes, par exemple depuis des services Web.

### **WPF Everywhere**

XBAP se restreint à Internet Explorer et ne doit donc pas être confondu avec **WPF/E**. WPF/E, pour WPF Everywhere, est le nom de code d'une technologie qui amènera WPF sur tous les navigateurs. Il s'agit, comme XBAP, d'un sous-ensemble de WPF. WPF/E ne fait pas partie du .NET Framework 3.0 et est attendu pour la première moitié de 2007. Il sera pris en charge par un grand nombre de navigateurs et fonctionnera sous plusieurs systèmes d'exploitation.

### **Encore plus d'interactions homme-machine**

WPF nous pousse à redécouvrir l'étendue des interactions homme-machine. WPF inclut en effet plusieurs ensembles de composants dédiés à enrichir les moyens d'interactions :

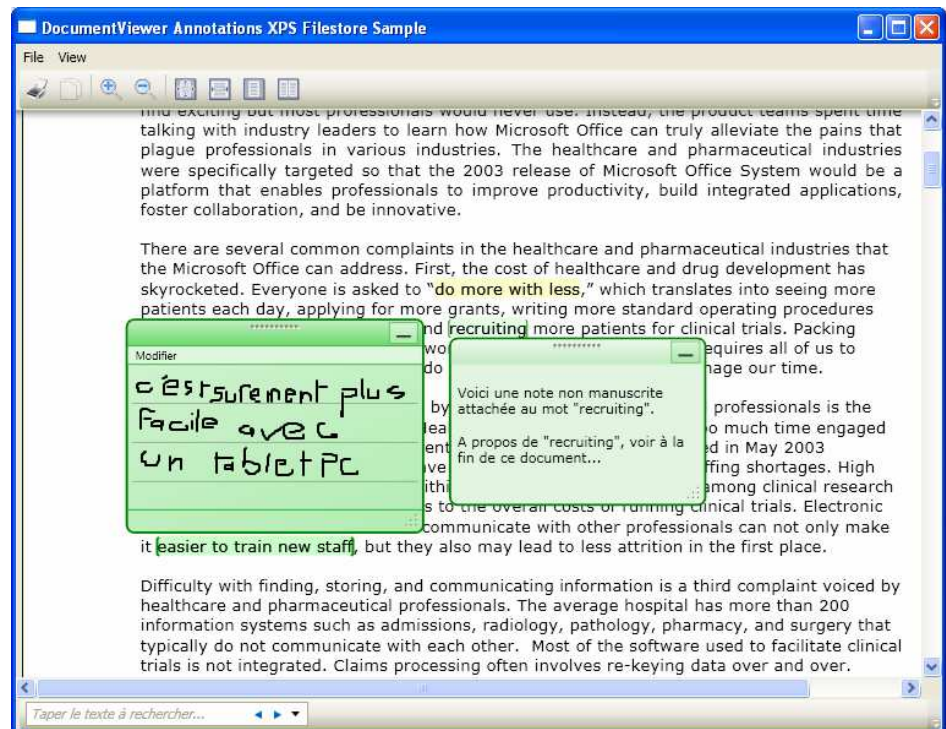
- Les vidéos peuvent être très simplement incluses dans une interface. Les transformations applicables aux éléments visuels classiques s'appliquent de la même façon aux vidéos. C'est donc une nouvelle possibilité qui est offerte aux concepteurs de logiciels pour, par exemple, présenter l'aide d'un logiciel en mode multimédia.
- WPF propose un ensemble de classes dédiées à la reconnaissance de l'écriture manuscrite temps réel. Les moyens de saisie de textes s'étendent ainsi au périphérique de saisie manuscrite sans surcoût de développement.
- Un synthétiseur vocal est inclus dans WPF. Sa mise en œuvre à travers la classe `SpeechSynthesizer` est extrêmement simple : il suffit d'appeler la méthode `Speak` en lui passant le texte à faire entendre.
- La reconnaissance de la parole est aussi fournie clé-en-main dans WPF via les nouvelles classes de `System.Speech.Recognition`.

---

*WPF/E, en autorisant l'accès aux objets WPF via JavaScript, apparaît comme un sérieux concurrent de la technologie Flex d'Adobe.*

- WPF concrétise le concept de document au travers d'un ensemble de fonctionnalités et services qui couvrent la présentation des documents dans l'interface graphique leur annotation par l'utilisateur, leur persistance et leur impression.

Les nouveautés qui concernent les documents débordent largement du cadre de l'IHM pour atteindre le back-end. La première est un format de description de package (System.IO.Packaging) qui favorise la modélisation et manipulation des relations entre le document et les éléments qui le composent (images, styles, chapitres, annotations, etc.).



**Figure 3. Exemple de document XPS annoté par une note manuscrite et une note en mode texte, fonctionnalités intégrées à .Net 3.0.**

---

*XPS se positionne clairement  
comme concurrent de PDF*

La deuxième est que ce format de packaging est déjà exploité par Office 2007 avec Open XML et par Vista avec Xml Paper Specification (XPS). XPS est un format de description de documents mis en page. Un des atouts d'XPS est qu'il s'appuie sur XML, ce qui autorise la génération et l'exploitation de ce format côté serveur avec des composants standards de gestion d'XML. Autre atout, les interfaces graphiques WPF et même Win32 peut être rendues au format XPS. A noter que le gestionnaire d'impression de Windows Vista exploite nativement ce format.

## Windows Communication Foundation

*WCF, ça c'est de la technologie ! Une technologie qui masque complètement les aspects middlewares et qui poussent sur le devant de la scène la notion de services. Un des meilleurs alliés pour la diffusion des principes SOA dans l'entreprise. C'est un peu comme si on unifiait le transactionnel des EJB, les services Web JAX-RPC et WSIT, et le messaging JMS tout en intégrant les aspects sécurité et scalabilité dans un framework de développement disponible à la fois côté serveur mais aussi côté client.*

Windows Communication Foundation (**WCF**) est un des composants majeurs de .NET 3.0 en terme de volume de code et donc d'investissement Microsoft. C'est une refonte des services supportés par ASMX et WSE, étendue aux communications hors services Web. Cette nouvelle bibliothèque de classes a été pensée pour unifier l'accès aux différents moyens de communication de .NET. Là où l'architecte devait intégrer les interfaces de .NET Remoting, WSE, des Enterprises Services (p. ex. transactions distribuées) ou MSMQ, WCF fournit une vue homogène pour les besoins de communication.

WCF repose sur le concept de message. Les messages seront échangés entre deux points selon les modalités définies par l'adressage, le mode de liaison et le contrat de service. Ces trois points sont résumés par Microsoft par le triplet A, B et C : Address, Binding et Contract.

- Address. Un point de destination (endpoint) d'une communication WCF doit avoir une adresse représentée sous forme d'un URI.
- Binding. Une liaison comprend trois types d'éléments.
  1. Le protocole définit la sécurité, la fiabilité, le contexte de communication ou un protocole personnalisé pour les messages qui circuleront entre les points de destination. Dans cette catégorie figurent des spécifications telles que WS-ReliableMessaging ou WS-Federation.
  2. Le transport définit la couche réseau qui servira de base au transport des messages, par exemple tcp ou http.
  3. L'encodage définit la méthode de codage du contenu des messages, par exemple text/XML, Binary ou MTOM.
- Contract. Le contrat expose les opérations que le service est capable d'assurer. En plus de la signature des opérations qui décrivent les données échangées dans les messages, la nature de l'échange est aussi déterminée par le contrat (uni- ou bidirectionnelle, synchrone ou asynchrone), ainsi que la mécanique de gestion des erreurs.

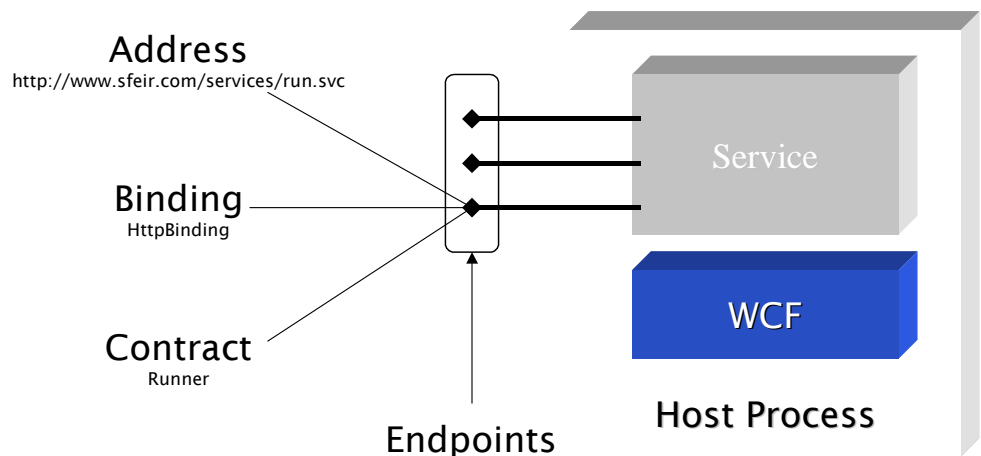


Figure 4. Principe de définition des services exposés par un processus via WCF.

Toutes ces informations sont exposées dans un modèle objet unifié qui vise à assurer la productivité des développements quel que soit le niveau de spécificité de l'application distribuée à mettre en œuvre. Le développeur gagne en temps de codage et évite l'éparpillement et les apprentissages de multiples APIs. De plus WCF est extensible (possibilité d'ajouter ses protocoles, ses formats de compression, ses mécanismes de trace) en venant s'insérer dans le pipeline de traitements des messages.

Autre facteur de productivité, les bindings les plus courants, tels que ceux correspondants aux standards WS-BasicProfil et WS-Federation, sont prédéfinis. Ils peuvent toutefois être paramétrés dans les détails (algorithme de cryptage, types de clés, types de jeton de sécurité, etc.) pour assurer l'interopérabilité avec des services existants. D'autres bindings moins courants sont prédéfinis : peer-to-peer sur tcp, canaux nommés, échanges convoyés par MSMQ, etc. Et si le paramétrage des bindings prédéfinis ne suffit pas, WCF propose la totalité des options dans un binding vierge (Custom Binding).

WCF facilite l'implémentation des services non plus seulement au sein d'IIS mais dans tout exécutable grâce à de nouvelles classes qui jouent le rôle d'hôte du service. C'était déjà possible sous WSE mais la quantité de code à écrire était bien supérieure. L'hôte se charge notamment de l'instanciation du service (Per Call, Per Session, Singleton). Il est par ailleurs possible de gérer l'étanchéité des services au travers des AppDomain du framework .Net

WCF permet de placer les informations de description d'un service hors du code, dans un fichier XML de configuration. Les informations de binding et d'adresse notamment peuvent être changées par édition du fichier correspondant, ce qui permet de tester rapidement plusieurs configuration, de s'adapter aux différents environnements de développement, de recette et de production, et d'accompagner les évolutions du système d'information à moindre coût. Si le système ne doit pas être configuré par des tiers (autres que les développeurs), on favorisera l'utilisation de l'API programmatique.

En résumé, WCF permet de séparer la logique du service de son implémentation technique. Il propose une très grande souplesse dans le paramétrage de l'implémentation tout en garantissant le respect des standards.

---

*Windows a maintenant son moteur de workflow intégré (par opposition aux précédents moteurs de Biztalk et SharePoint). Rien n'empêche de l'utiliser depuis d'autres mondes. Une application Java peut s'appuyer sur ce moteur pour orchestrer son métier.*

## **Workflow Foundation**

Un des éléments de base des applications composites et des architectures orientées service est l'échange de messages en mode asynchrone. Dans de tels systèmes, la complexité des développements est en grande partie liée au suivi des états des systèmes expéditeurs et destinataires des messages, en supposant que la complexité des communications inter-systèmes est prise en charge par WCF. La gestion de la logique d'enchaînement de ces états et de leur persistance peut être confiée à Windows Workflow Foundation (WF).

Windows Workflow Foundation est une technologie dédiée à la définition de workflows et à leur exécution. Un workflow modélise les états d'un système, la logique de transition entre états, les événements extérieurs, les conditions de transition, etc. C'est une approche extrêmement structurante de la modélisation de processus, qu'il s'agisse de processus métiers, de validations de documents ou des enchaînements des écrans d'une interface graphique. La création de processus s'en trouve simplifiée. Cela ouvre la porte à d'autres fonctionnalités telles que la surveillance de l'exécution du processus.

WF est composé d'un ensemble de classes qui couvrent les fonctionnalités suivantes :

- La définition d'un workflow. Les éléments de base d'un workflow WF sont les "activités", qui rendent compte des actions et interactions du workflow. WF est livré avec des activités prédéfinies comme l'appel à des services Web, la réponse à des événements ou des activités de contrôle (boucle while, exécutions parallèles, etc.);
- L'hébergement du workflow. Ces classes incluent le moteur d'exécution et des outils de contrôle de l'exécution des instances d'un même workflow. Elles permettent d'héberger le moteur d'exécution dans tout type d'application, console, WinForms ou ASP.NET;
- La persistance du workflow. En entreprise les processus qui durent plusieurs heures voire plusieurs mois sont courants (support IT, commandes, etc.). Dans ce cas l'état du workflow ne peut se permettre de résider en mémoire : le flot doit pouvoir s'interrompre – éventuellement au milieu d'une activité transactionnelle, et se « réveiller » au bon moment. C'est pourquoi WF propose un service de persistance prédéfini couplé à SQL Server et extensible à d'autres moyens de persistance ; Le workflow est alors sérialisé sous un format XML, il s'agit de XOML de façon complémentaire à XAML.
- La conception en mode graphique hors Visual Studio. Ces classes ouvrent la possibilité de développer des applications dédiées à la conception de workflows métier, composés d'activités personnalisées agencées par simple point & click. Une personne d'un profil fonctionnel aura la possibilité de définir ou de modifier un processus métier sans avoir à écrire de code.
- La définition et l'exécution de règles métier. WF comprend un moteur de règles capable de déduire l'ordre d'exécution des règles à partir de leurs dépendances. Typiquement applicable à des systèmes

d'informatique de gestion pour lesquels les contraintes légales changeantes requièrent une souplesse dans le codage de la logique métier, le moteur n'est cependant pas optimisé pour permettre l'exécution intensive de plusieurs milliers de règles.

Récemment (mars 2007) Microsoft a livré une version "preview" d'une extension de WF destiné à gérer BPEL, le langage XML d'exécution de processus d'entreprise.

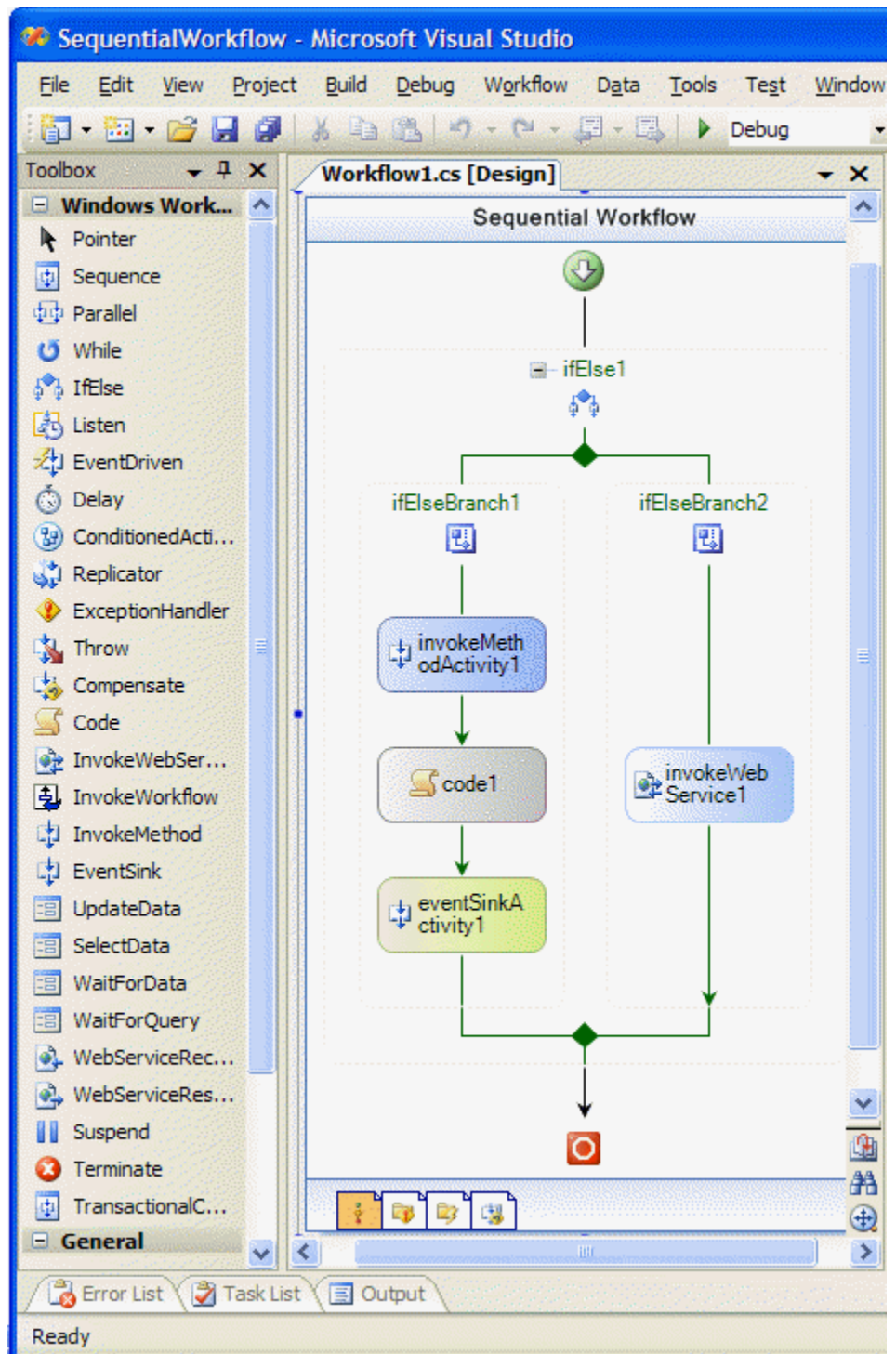


Figure 5. Exemple de workflow dans le concepteur de Visual Studio

## CardSpace

CardSpace, initialement nommé InfoCard, est une partie du Framework 3.0 qui apporte une réponse au besoin de gestion des identités numériques sur Internet et en entreprise. Il ne s'agit plus de s'attaquer à l'enjeu du Single Sign-On (SSO) planétaire, dont on a pu constater l'échec relatif des différentes initiatives, mais de sécuriser l'authentification en se conformant aux standards WS-\*. A ce titre CardSpace est plus proche des dispositifs physiques d'authentification de type smart cards ou clés USB sécurisées que des projets tels que Liberty Alliance ou OpenID, avec lesquels il peut s'interfacer.

Pour schématiser, trois aspects sont abordés par CardSpace.

- Les utilisateurs ont plusieurs identités. Les internautes ne souhaitent pas utiliser les mêmes identifiants pour une connexion à un compte professionnel que pour un site de jeux en ligne. Les adresses emails et autres informations renseignées sont différentes. CardSpace propose aux utilisateurs de définir autant de profils d'identité qu'ils le souhaitent. Certaines de ces identités sont garanties, c'est-à-dire certifiées par un tiers de confiance (entreprise, banque, centrale d'achats, etc.) appelé fournisseur d'identité.
- Un internaute se sert souvent de très peu de mots de passe – et souvent d'un seul – essentiellement parce qu'il est difficile de gérer autant de mots de passe que de sites à accès protégé. C'est pourquoi CardSpace adopte pour support mnémotechnique un visuel (une carte) choisi par l'utilisateur plutôt qu'une chaîne de caractère.
- Les tiers de confiance et fournisseurs d'identités sont et seront multiples, il n'y aura pas de vainqueur dans la lutte pour le SSO global. CardSpace se place au-dessus des fournisseurs d'identité en permettant aux développeurs d'applications sécurisées de relier les visuels CardSpace à n'importe quel fournisseur d'identité.

Concrètement, CardSpace se compose d'un jeu de protocoles de communication WS-\* et d'une implémentation de ces protocoles sous la forme d'une extension du système d'exploitation qui permet à l'utilisateur de décliner son identité de façon sécurisée. Au moment où une application RIA ou RDA demande à l'utilisateur de se signer via CardSpace, Windows présente une interface graphique « protégée », c'est-à-dire qui empêche toute autre application de l'espionner, y compris les keyloggers et autres logiciels malveillants.

L'application qui demande une preuve d'identité spécifie les éléments d'information requis (adresse, email, etc.) et CardSpace propose à l'utilisateur les cartes de visite qui s'y conforment. Si aucune ne correspond, l'utilisateur crée ou installe une nouvelle carte satisfaisant les pré-requis. Deux types de carte existent, les cartes auto générées et les cartes gérées. Les cartes gérées sont validées par un tiers de confiance, tel qu'une banque, un employeur ou un organisme gouvernemental, qui joue alors le rôle de STS (*Security Token Service*). Dans ce dernier cas, le tiers participe à l'authentification en générant un jeton de sécurité.

En termes de compatibilité, les échanges sont réalisés dans le cadre des standards WS-Security, WS-Trust, WS-MetadataExchange et WS-SecurityPolicy. Depuis un client léger, l'interface graphique CardSpace est mise en œuvre par un objet ActiveX instancié au sein du navigateur. Initialement supporté seulement par Internet Explorer 7, Firefox le supporte aujourd'hui via un plug-in à télécharger.



Figure 6. L'interface graphique CardSpace

## LES OUTILS




---

*Avec Eclipse, Netbeans et IntelliJ, l'offre dans le monde Java n'a pas à pâlir de l'offre Microsoft, si l'on omet les fonctionnalités d'automatisation de type Software Factories.*

---

*A noter qu'il existe des interfaces graphiques de création de scripts XML pour MSBuild, comme MSBuild Sidekick, ainsi qu'une alternative open-source en licence GPL, Nant, portage .NET du célèbre Ant.*

### Visual Studio

Visual Studio est l'environnement de développement intégré (IDE) dédié au développement d'applications .NET. Visual Studio 2003 permet de développer pour .NET 1.1. Pour .NET 2.0, Visual Studio 2005 est nécessaire. En revanche cette même version autorise le développement d'applications .NET 3.0 après téléchargement des extensions adéquates.

Les différentes déclinaisons du produit offrent un ensemble de fonctionnalités qui couvrent la totalité du cycle de développement : architecture, analyse de code, tests unitaires et déploiement. Une version gratuite, Visual Studio Express Edition, est disponible en libre téléchargement, accompagnée de SQL Server Express Edition.

La clé de voûte de Visual Studio est la productivité. L'environnement est conçu pour minimiser l'effort de développement lors de la réalisation des tâches les plus communes : accès aux tables et procédures stockées et définition des interfaces graphiques par simple glisser-déplacer, liaison des contrôles graphiques aux données sans écriture de code, exposition de services Web par apposition d'attributs, etc. Les extensions .NET 3.0 pour WPF et WF se conforment à ce schéma en intégrant des assistants visuels de conception.

Au-delà du mode RAD (Rapid Application Development), l'IDE amène aussi la productivité sans perte de qualité lors de l'écriture du code via des raccourcis rapides pour insérer des patterns de code (snippets) et des outils de refactoring à portée de clic.

Le deuxième point essentiel de Visual Studio est son extensibilité. Les macros représentent le moyen le plus simple de personnaliser l'environnement pour gérer les tâches répétitives. Les Add-In vont au-delà en donnant accès aux menus, barres d'outils et aux fenêtres d'outils tout en permettant de faire appel à des composants externes. Enfin le toolkit de développement Visual Studio 2005 SDK fournit aux éditeurs d'extension les moyens de développer un nouveau langage et de nouveaux types de projets intégrés à l'IDE.

Cette extensibilité rend possible la mise au point d'usines logicielle (Software Factories) métier. Les modules GAT (Guidance Automation Toolkit), GAX (Guidance Automation Extensions) et DSL (Domain Specific Languages) permettent de personnaliser Visual Studio via des assistants contextuels qui guideront le développeur dans la démarche méthodologique de l'entreprise et lui simplifieront l'utilisation des frameworks ou socles techniques adaptés aux spécificités du métier de l'entreprise. Les exemples de Software Factory proposés par l'équipe Patterns & Practices (code source inclus) apportent une bonne illustration de ces possibilités (Web Client Software Factory, Smart Client Software Factory, Web Service Software Factory).

MSBuild, le moteur de compilation distribué avec .NET 2.0 et exploité par Visual Studio 2005, offre un autre moyen de personnaliser l'environnement de développement en livrant l'accès au processus de compilation. MSBuild s'appuie sur XML pour définir les ressources et actions de la compilation. Il permet par exemple de déclencher l'exécution de modules adaptés à l'environnement du projet tels qu'un transfert FTP ou l'accès à un service Web.

Dans l'écosystème .NET, les périphériques mobiles jouent un rôle de plus en plus important. A l'heure où Windows Mobile est le troisième système d'exploitation sur le marché des smartphones<sup>1</sup> et le premier sur celui des PDA<sup>2</sup>, Visual Studio propose tout le nécessaire pour développer pour ces périphériques en les émulant sous Windows. Le développement s'appuie alors sur .NET Compact Framework, version compressée et allégée du Framework .NET.

## Team System

Visual Studio Team System étend Visual Studio à la gestion de l'ensemble du cycle de vie d'une application :

- Les architectes disposent d'outils de modélisation de l'architecture logicielle, de l'infrastructure et des déploiements. La modélisation graphique des classes est synchronisée avec le code produit.
- Les développeurs peuvent lancer une analyse du code qui délivre des diagnostics sur la qualité du code (méthodes trop longues, appels à des méthodes obsolètes, pratiques recommandées, etc.). Un profiler permet aussi d'identifier les éventuels goulots d'étranglement qui compromettent les temps de réponse d'une application.
- Les ingénieurs qualité disposent d'outils pour les tests unitaires, les tests de montée en charge, les tests manuels, les tests d'application Web et la gestion des campagnes de tests.
- Les spécialistes de bases de données accèdent à des utilitaires de refactoring, de tests unitaires dédiés au SGBD et de génération des scripts de changement de schémas de base de données.

---

*Il existe d'autres outils permettant d'obtenir ces fonctionnalités, par exemple NUnit pour les tests unitaires et CruiseControl pour l'intégration continue. La société JetBrains, propose des outils de refactoring (ReSharper), d'intégration continue (TeamCity), de profiling, etc...*

Ces différents rôles partagent leur travail et se synchronisent grâce à Team Foundation Server. Foundation Server vise à couvrir l'ensemble des besoins de collaboration et de gestion de projet qui mène à l'industrialisation des développements :

- La gestion de configuration ne s'appuie plus sur Visual Source Safe mais sur un serveur entièrement réécrit couplé à SQL Server, et accessible au travers de Web Service
- La gestion des changements et des anomalies est basée sur les rôles des intervenants.
- Les indicateurs sont produits par SQL Server Reporting Services et englobent l'évolution des taux de couverture des tests, les bugs actifs et corrigés, etc.
- Un portail de diffusion SharePoint offre le partage des informations, agendas, rapports d'avancement ou actualités.

---

<sup>1</sup> Canalys, troisième trimestre 2006.

<sup>2</sup> Gartner, communiqué du 7 août 2006.

- L'intégration avec Excel, Project et Visual Studio associée aux rapports, au portail et au suivi des tâches permet à chaque intervenant du projet de connaître et compléter ses activités et priorités.
- Le serveur d'intégration permet de définir quels sont les composants à intégrer pour une version donnée du logiciel et quelles sont les tâches (tests, livraison des exécutables) à exécuter avant ou après la compilation.

Il est aussi possible de manipuler le référentiel Team Foundation Server au travers d'un simple browser Web grâce à la technologie Team Web Plain Access disponible gratuitement depuis son acquisition par Microsoft fin Mars 2007

### Les outils Expression

La conception des interfaces graphiques WPF bénéficie, on l'a vu plus haut, de la séparation du design et du code applicatif. Les outils de la gamme Expression concrétisent cette séparation en permettant aux designers de travailler indépendamment de Visual Studio.

Expression Design est dédié aux créatifs. Il permet de concevoir des visuels sophistiqués et animés, et de les exporter en XAML (en autres formats) pour les exploiter sous Visual Studio et Expression Blend.

Expression Blend fournit tous les outils pour développer des styles, des modèles de comportement et des animations. Expression Blend peut ouvrir un projet Visual Studio, autorisant ainsi les designers à travailler en parallèle avec les développeurs et à accéder aux données et événements du logiciel.

La création de contenu multimédia est prise en charge par Expression Media, et la conception de sites Web par Expression Web. Ces quatre logiciels forment Expression Studio.

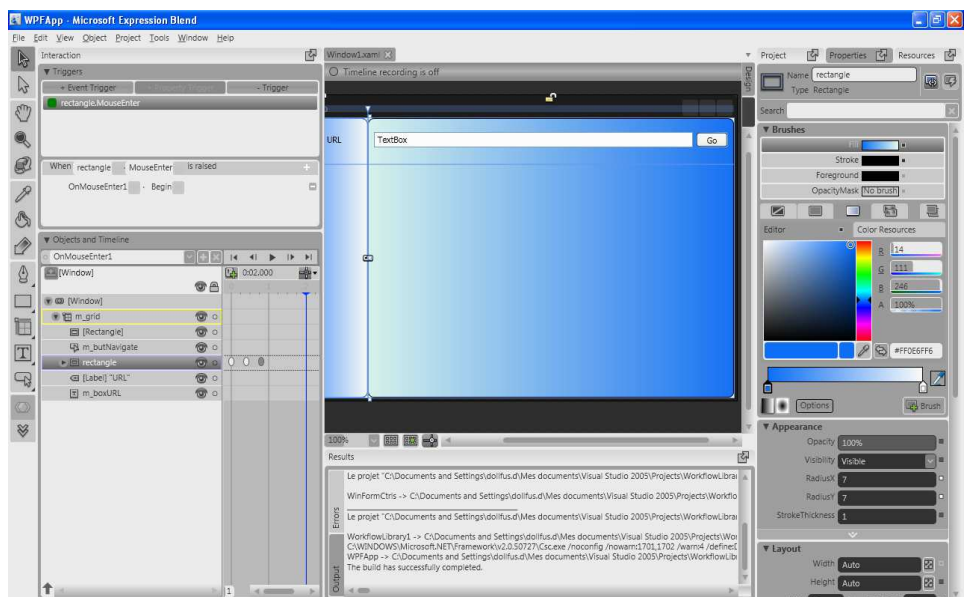


Figure 7. Saisie d'écran du logiciel de design Expression Blend

### Et bientôt...

La prochaine version de Visual Studio dont le nom de code est Orcas, annonce quelques nouveautés marquantes. Jusqu'ici essentiellement cantonnée aux assistants graphiques et à l'écriture du code, la productivité promet d'aborder aussi la couche métier des architectures logicielles à travers deux nouveautés, LINQ et C# 3.0.

Microsoft était absent des technologies de correspondance objets/bases de données relationnelles (ou ORM pour Object-Relational Mapping). Celles-ci permettent de définir le lien entre les classes d'objets en mémoire et les tables en base de données sans écriture de code SQL. Hibernate pour Java et NHibernate pour .NET en sont les représentants les plus connus et les plus utilisés. L'impact des ORM sur la productivité est réel puisque le développement des traitements de chargement, de création et de mise à jour des objets n'est plus nécessaire. Et surtout, à l'inverse des gains en productivité qui nuisent à l'évolutivité tels que les outils RAD de binding SGBD/IHM, c'est une productivité compatible avec les architectures multi-couches qui laissent la porte ouverte au SOA.

Linq est essentiellement un framework de requettage. Il comprend plusieurs déclinaisons en fonction des éléments sur lesquels portent les requêtes : Linq to SQL, Linq to XML, Linq to Objects, etc. Le plus proche de Nhibernate est Linq to Entities.

Un atout de Linq to Entities par rapport à Nhibernate est qu'il permet aux développeurs de bénéficier de l'intégration du langage de requête dans C# : l'analyseur syntaxique et le compilateur peuvent indiquer les erreurs avant les tests (Cf. Figure 8).

Peut-être de quoi séduire les architectes et développeurs Java.

```
var query =
    from c in customers
    from o in orders
    where o.CustomerName == c.Name
    select new {
        c.Name, o.OrderID, o.Amount, c.Age
    };
```

**Figure 8. Exemple de requête LINQ écrite en C# 3.0**

---

*Certains ont déjà commencé à développer "Linq to Nhibernate". A suivre...*

Implanté en France, en Espagne et au Luxembourg, le Groupe SFEIR compte aujourd'hui plus de 250 collaborateurs.

Groupe indépendant, SFEIR suit un rythme de croissance organique de 30 % depuis 2004, se traduisant par un chiffre d'affaires 2006 de 22 Millions d'euros.

Nous sommes aujourd'hui reconnus par nos clients pour nos compétences techniques avancées, la maîtrise de leur problématique métier et notre savoir-faire dans le pilotage de projets.

Véritable SSII internationale, nous intervenons depuis plus de 20 ans dans les secteurs suivants :

- Bourse et Marchés de Capitaux,
- Banque et Assurance,
- Industrie et Services.

Afin de répondre entièrement aux attentes de nos clients, SFEIR est doté d'une direction technique. Composée de consultants architectes ayant de solides expériences, elle participe à la réussite des projets. Elle garantit également que les compétences des collaborateurs restent en phase avec les dernières avancées technologiques.

La direction technique est composée de trois pôles, dont un dédié au monde .Net. Ses missions sont multiples – conseil, accompagnement, audit, développement de socle technique – et mettent en œuvre les technologies les plus abouties : Framework 3.0, Team System, Ajax, Nhibernate, Spring.Net, NUnit, Reporting Services, Sharepoint, Biztalk, ...

Vous souhaitez plus de renseignements sur la société, sur les postes que nous pouvons vous proposer ?

- Consultez notre site : [www.sfeir.com](http://www.sfeir.com),
- Ecrivez-nous à l'adresse suivante : [recrutement@sfeir.com](mailto:recrutement@sfeir.com).

## GLOSSAIRE

**ADO.Net** : Evolution de ADO (ActiveX Data Object), bibliothèque d'objets pour l'accès aux bases de données.

**Ajax** : Technologie dédiée à l'enrichissement de l'expérience utilisateur pour les applications Web. Repose sur JavaScript et l'asynchronisme des échanges de données au format XML.

**ASP** : Active Server Pages. Technologie de génération dynamique de documents HTML côté serveur antérieur à .NET.

**ASP.NET** : Version .NET de ASP. Permet de séparer le code HTML de la logique applicative.

**CardSpace** : Librairie du framework 3.0 intégré à IE7 permettant de sécuriser l'authentification des utilisateurs et capable de s'intégrer à différents systèmes standards de gestion des identités.

**CGI** : Common Gateway Interface. Historiquement, premier mécanisme standard capable d'interfacer un serveur Web à des données et processus de l'entreprise.

**CLI** : Common Language Infrastructure. Spécification publiée par Microsoft qui décrit les exécutables et l'environnement qui définissent le cœur de .NET.

**CLR** : Common Language Runtime. Machine virtuelle supportant .NET, implémentation Microsoft de la spécification CLI.

**COM** : Component Object Model. Technologie orientée objet prônant la réutilisabilité au niveau composant grâce à la publication d'interfaces faisant office de contrat.

**MFC** : Microsoft Foundation Classes. Librairie de classes C++ encapsulant la complexité de l'interface de programmation graphique Win32.

**MTOM** : Recommandation W3C pour l'optimisation des échanges de données binaires pour les services Web.

**RAD** : Rapid Application Development. Terme générique identifiant les méthodes et environnements de développement dont l'atout principal est la rapidité des développements.

**RDA** : Rich Desktop Applications. Désigne les applications client "lourd" dont l'ergonomie est aussi riche que les applications RIA. Terme soulignant le fait que l'ergonomie des applications connectées a récemment progressé plus rapidement que celle des applications Windows.

**RIA** : Rich Internet Applications. Applications connectées offrant une ergonomie bien supérieure à ce que le HTML autorise. Ajax, XBAP, WPF/E et Flex sont des technologies compatibles RIA.

**ROA** : Rich Office Applications. Applications intégrées à Office et bénéficiant de son ergonomie et de ses services.

**VSTO** : Visual Studio Tools for Office. Ensemble d'outils et de bibliothèques facilitant le développement d'applications ROA.

**WCF** : Windows Communication Foundation. Librairie .NET 3.0 dédiée à la mise en œuvre des standards de communication des services et plus généralement des applications composites interconnectées.

**WF** : Workflow Foundation. Librairie de .NET 3.0 qui apporte un socle technique pour définir et mettre en œuvre des workflows.

**WPF** : Windows Presentation Foundation. Librairie de .NET 3.0 apportant l'accès à la nouvelle couche graphique de Microsoft reposant sur DirectX.

**WPF/E** : Nom de code d'une variante de WPF capable de s'exécuter dans de nombreux navigateurs et systèmes d'exploitation. Candidat pour servir de socle à la couche de présentation des applications RIA.

**WSE** : Web Service Enhancements. Extensions à .NET permettant d'exposer des services Web conformes aux spécifications des standards du W3C et de l'OASIS.

**XAML** : Extensible Application Markup Language. Dialecte XML de .NET 3.0 permettant de décrire des objets, leurs propriétés, leurs relations et leurs interactions. XAML est par exemple utilisé pour décrire une application WPF et un workflow WF.

**XBAP** : XAML Browser Applications. Une application XBAP est une application WPF qui s'exécute au sein d'Internet Explorer.