



CENTRALE
L Y O N

L'architecture de .NET

Bertrand DAVID : Génie Logiciel

- **Les composants de .NET s'organisent de la manière suivante :**
 - C#, un nouveau langage orienté objet destiné à faciliter la programmation dans .NET, notamment les composants, qui intègre des éléments de C, C++ et Java en apportant quelques innovations comme les méta-données.
 - Un environnement d'exécution commun (*Common Language Runtime - CLR*) qui exécute un *bytecode* écrit dans langage intermédiaire (*Microsoft Intermediate Language - MSIL* ou IL). Du code et des objets écrits dans un langage quelconque peuvent être compilés en IL et exécutés par le CLR si un compilateur IL existe pour ce dernier.
 - Une grande bibliothèque de composants et d'objets de base accessibles par le CLR, qui fournissent les fondations pour écrire rapidement un programme (accès réseau, graphisme, accès aux bases de données).
 - ASP.NET, une nouvelle version d'ASP (Active Server Pages) qui supporte une véritable compilation en IL, alors qu'ASP était interprété auparavant. On peut également écrire les pages ASP dans n'importe quel langage disposant d'un compilateur IL.
 - Visual Studio .NET, une refonte de l'environnement Visual Studio et de Visual InterDev permettant aussi bien le développement d'applications et de composants classiques que Web.
 - WinForms et WebForms, un ensemble de composants graphiques accessibles dans Visual Studio .NET.
 - ADO.NET, une nouvelle génération de composants d'accès aux bases de données ADO qui utilise XML et SOAP pour l'échange de données.

3

BTD/GL/.NET

CENTRALE
L Y O N

L'environnement d'exécution commun

Bertrand DAVID : Génie Logiciel

- Tout comme Java, MS .NET est muni :
 - d'une machine virtuelle nommée CLR (Common Language Runtime)
 - d'un code intermédiaire, le MSIL (MicroSoft Intermediate Language).
- Ainsi, ce code généré par compilation est indépendant de l'architecture sous-jacente et peut s'exécuter sur n'importe quelle machine comportant la machine virtuelle .NET.
- Pour l'instant, seul les OS de Windows le supportent.
- Le CLR comporte aussi un ramasse-miettes ainsi qu'un système de droits d'accès au code (sécurité).

Le diagramme illustre l'architecture de .NET en couches. À la base se trouvent Windows et COM+. Au-dessus se trouvent les Bibliothèques de Classes de Base et le CLR (Common Language Runtime). Ensuite viennent les services ASP.NET et WinForms, ainsi que les Bibliothèques de Classes de Base. Au-dessus de ces services se trouvent les langages de programmation VB.NET, C++, C#, et d'autres. À droite du diagramme se trouve Visual Studio .NET.

4

BTD/GL/.NET

CENTRALE
L Y O N

Common Language Specification (CLS) – Common Type System (CTS)

- Le langage MSIL présente des fonctionnalités intéressantes. Il constitue une base commune, un support à tous les langages compatibles avec l'environnement .NET, entraînant parfois l'ajout ou la suppression de quelques caractéristiques propres à chacun.
- Par exemple, la surcharge d'opérateurs est à présent possible en Eiffel#. En revanche, le debugger n'accepte plus l'héritage multiple du C++ ou le type *unsigned int*.
- Le concept de CLS (Common Language Specification, sous-ensemble commun à tous les langages) est issu de ce principe et offre des fonctionnalités comme un mécanisme commun de débogage, la généricité, la surcharge d'opérateur, des gestions des exceptions ou la manipulation des pointeurs. La base commune contenant également tous les types admis, le transtypage ne pose plus de problèmes : en effet, les types sont communs à tous les langages modifiés (CTS : Common Type System).
- Les fichiers écrits en MSIL sont compilés à nouveau par un compilateur JIT en code natif pour donner un exécutable valide pour la machine.

Bertrand DAVID : Génie Logiciel

BTD/GL/.NET

5

CENTRALE
L Y O N

Les briques de base (Base Class Library)

- Dans l'optique d'aider le développeur et de favoriser la réutilisation de composants, .NET propose une grande bibliothèque de classes, similaire en structure à celle de Java.
- Cette bibliothèque est commune à tous les langages, puisque écrite en MSIL, et rapproche d'autant les différents langages utilisés.
- Ainsi, dans une application, des composants en C++ peuvent faire appel à d'autres créés en Cobol...
- D'autre part, on remarque que dans cette bibliothèque commune, la gestion des fichiers XML est très présente.
- En parallèle à ce système de briques, Microsoft souhaite vendre ou louer des composants plus élaborés réutilisables pour constituer des applications Web complexes.
- L'interopérabilité forte de ces services est fondée sur XML.

Bertrand DAVID : Génie Logiciel

BTD/GL/.NET

6

CENTRALE
L Y O N

ASP.NET

- ASP.NET est une nouvelle version d'ASP, un langage interprété produisant des pages HTML et WML très comparable à PHP en de nombreux points.
- La mise à jour introduit une vraie rupture dans le paradigme d'ASP avec l'ajout de deux nouveaux concepts comme la compilation des pages et les WebForms.
- Comme les Java Server Pages (JSP), le code ASP.NET est toujours compilé (en IL puis code natif avec le JIT) et peut être écrit dans un langage de haut niveau.
- ASP.NET supporte tous les langages du CLR, ce qui inclut C# et VB.NET.

Bertrand DAVID : Génie Logiciel

BTD/GL/.NET

7

CENTRALE
L Y O N

WebForms

- Les WebForms sont une couche d'abstraction ajoutée pour permettre une programmation composite d'interface homme-machine orientée Web.
- Des composants génériques tels les formulaires, tableaux, boutons et zones de textes peuvent être assemblés afin de générer les pages ASP.NET.
- S'ils sont utilisés avec soin (aucun ajout de code HTML en dur par exemple), ces composants nommés WebForms répondent à deux problématiques des développeurs Web :
 - la gestion des différences entre les navigateurs et leurs nombreuses versions
 - et la reconnaissance à l'exécution du terminal et/ou du navigateur client afin d'utiliser un équivalent natif se rapprochant le plus possible de l'effet voulu.

Bertrand DAVID : Génie Logiciel

BTD/GL/.NET

8

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Visual Studio .NET

- Microsoft Visual Studio .NET est l'outil de référence des développements en .NET.
- Cette nouvelle version de Visual Studio apporte réellement de nouvelles fonctionnalités intéressantes (comme UML) pour les développeurs Windows.
- En particulier, il est possible d'écrire et de déployer des applications Web en des temps très courts.

9

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Les services Web et ADO.NET (Data & XML)

- Les services Web tels qu'ils sont introduits dans .NET sont une petite révolution dans l'informatique, non pas dans l'avance technologique, mais dans la simplicité de son approche, ce qui est intéressant dans un domaine réputé complexe :
 - Un élément important de .NET appelé à être utilisé au sein des services Web est ADO.NET, un modèle orienté réseau pour les bases de données, qui permettra des prouesses comme la consultation et la mise à jour par HTTP et XML.
 - Conformément à l'esprit de .NET, cette évolution d'ADO (ActiveX Data Objects) met XML au cœur des bases de données qu'elles soient relationnelles ou pas.
 - L'innovation centrale d'ADO.NET est le DataSet, un modèle XML déconnecté des données.
 - Un objet DataSet peut ainsi effectuer des requêtes sur la base et traduire les résultats en XML.
 - Le grand intérêt réside dans le fait que les manipulations ultérieures sur le DataSet s'effectuent sans connexion à la base. L'API est très riche, elle permet notamment d'exprimer des contraintes, des relations entre tables, des insertions, des mises à jour...
 - L'objectif d'ADO.NET est à proprement parler de libérer les données en leur permettant d'être manipulées et échangées sur le Web avec une grande simplicité.

10

CENTRALE
L Y O N

L'architecture de .NET Framework selon Microsoft

VB.NET C++ C# ... Visual Studio .NET

CLS (Common Language Specification)

ASP.NET WinForms

ADO.NET XML

Bibliothèques de Classes de Base

CLR (Common Language Runtime)

Windows COM+

Bertrand DAVID : Génie Logiciel

BTD/GL/.NET

11

CENTRALE
L Y O N

Essai de comparaison avec J2EE

- L'un des objectifs évidents de Microsoft avec ce nouveau produit est de concurrencer Sun sur le marché des serveurs.
- C'est tout naturellement que .NET s'inspire largement de la technologie J2EE, référence dans le domaine.
- Il est intéressant d'établir un parallèle entre les deux architectures pour voir si .NET a su reprendre les points forts de J2EE et palier à ses défauts.
- La comparaison que nous effectuons ici concerne tous les domaines, du langage de programmation à l'agencement des différents éléments d'architecture.

Bertrand DAVID : Génie Logiciel

BTD/GL/.NET

12

CENTRALE L Y O N	Le modèle Objet : Les langages C# et Java (1/3)
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> • Les langages Java et C# sont respectivement les langages de référence de J2EE et .NET. Le tout nouveau C# s'appuie en partie sur Java, en partie sur C++, ce qui explique les ressemblances. • Ce que C# a repris à Java : <ul style="list-style-type: none"> → les interfaces sont des classes totalement abstraites, → l'héritage est simple, mais plusieurs interfaces peuvent être implémentées pour la même classe, → les variables doivent être initialisées par défaut, → le concept de packages / espaces de nom est très présent pour regrouper et classer les classes dépendantes, → on retrouve également les mêmes attributs de visibilité (public, private), → les classes intérieures, classes définies à l'intérieur des classes, → la gestion des exceptions, → la génération de documentation... → en ce qui concerne les méta-données, les deux langages proposent une API permettant de manipuler des attributs sur la classe en cours (capacité d'introspection), mais il est possible dans .NET d'ajouter d'autres attributs de méta-informations, → le multithreading est présent des deux côtés mais bien mieux intégré dans Java (communication, synchronisation),
BTD/GL/.NET	13

CENTRALE L Y O N	Le modèle Objet : Les langages C# et Java (2/3)
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> • En revanche, C# innove dans les aspects suivants : <ul style="list-style-type: none"> → Les événements : Chaque classe peut recevoir ou exporter des événements. → Système de type unifié : A la différence de Java, les types primitifs comme int et float héritent de Object, ce qui évite l'encapsulation en Java par les classes comme Int et Float. → Les fonctions déléguées (<i>delegates</i>) : Elles sont un équivalent au typage sûr des références aux fonctions de C ou de C++. → Les propriétés (<i>properties</i>) : Elles contribuent à simplifier la syntaxe en encapsulant les accesseurs (les get et set parfois désagréables de Java) dans la déclaration d'un membre. → Les structures (<i>struct</i>) : elles sont similaires aux classes, sans héritage, et sont allouées sur la pile. Elles contiennent directement leurs valeurs (<i>value type</i>) à la différence des classes classiques dont on ne manipule en réalité que le pointeur vers une instance. L'usage des structures permet d'améliorer la performance du code produit. → Les types énumérés (<i>enum</i>) : Restaurés de C++, ils permettent d'associer des noms avec des valeurs numériques, ce qui évite les static final int FIELD de Java.
BTD/GL/.NET	14

CENTRALE L Y O N	Le modèle Objet : Les langages C# et Java (3/3)
Bertrand DAVID : Génie Logiciel	<p>→ La surcharge d'opérateur (<i>operator overloading</i>) : Héritée de C++, elle simplifie également la syntaxe pour la manipulation des objets.</p> <p>→ La généricité (<i>templates</i>) : C# reprend le principe des <i>templates</i> de C++. Pour rendre cela possible, le code MSIL n'a pas, comme Java, de fonction AddInt ou AddFloat mais une fonction Add générique (<i>type neutral</i>) qui est remplacée par du code natif approprié en fonction du contexte.</p> <p>→ Les méthodes virtuelles : Comme en C++, les méthodes ne sont pas virtuelles par défaut, c'est-à-dire qu'il faut déclarer une méthode virtuelle dans la classe mère et ajouter le mot clé override dans la méthode redéfinie pour pouvoir surcharger – c'est exactement le contraire en Java. Outre un gain faible en performance, ceci permet de voir clairement quelles méthodes peuvent et sont redéfinies.</p> <p>→ Les indexers : Les objets implémentant l'interface IEnumerable peuvent voir leur contenu parcouru par le mot clé foreach, ce qui simplifie également la syntaxe.</p> <p>→ Syntaxe de passage des paramètres riche : Les mots clés in, out, ref et params permettent un interfaçage aisé avec les autres langages.</p> <p>→ Les espaces de nommage (<i>namespaces</i>) : Les paquetages (<i>packages</i>) de Java qui étaient très fortement liés à une structure sous forme de répertoire sont remplacés par une notion plus souple d'espace de nommage.</p>
BTD/GL/.NET	15

CENTRALE L Y O N	Les machines virtuelles : CLR et JVM
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> • Les composants de base sont les mêmes. • Le langage de programmation est compilé en un langage intermédiaire (<i>bytecode / MSIL</i>), et sont destinés à fonctionner dans une machine virtuelle. • On note également l'existence d'un ramasse-miettes, d'un gestionnaire de sécurité, de multithreading. • Mais le Java bytecode est interprété alors que MSIL est recompilé. • Le CLR supporte plusieurs langages précompilés, dont ceux qui ne sont pas orientés objet, et permet la création de composants partagés. • Il faut également insister sur la portabilité qui est valable surtout chez Java.
BTD/GL/.NET	16

CENTRALE
L Y O N

Les bibliothèques de données

Bertrand DAVID : Génie Logiciel

- Les API proposés pour les deux langages (On part du principe que .NET correspond surtout à C#) sont très semblables.
- Il faut reconnaître que .NET propose un panel de fonctions de manipulation de SOAP et XML très étendu.

BTD/GL/.NET 17

CENTRALE
L Y O N

L'archivage, les composants

Bertrand DAVID : Génie Logiciel

- Java et .NET comportent des composants permettant le déploiement des archives.
- La gestion de la propriété et des versions est en revanche visible uniquement du côté de l'outil de Microsoft.
- L'interopérabilité des composants est présente dans les deux camps, mais les EJBs restent un modèle beaucoup plus sûr et plus mûr.

BTD/GL/.NET 18

CENTRALE
L Y O N

Le remoting (accès à distance)

Bertrand DAVID : Génie Logiciel

- Le remoting est suffisamment bien implémenté dans les deux camps pour le rendre simple d'utilisation.
- Mais .NET a l'avantage car il est plus flexible en ce qui concerne son extensibilité.
- Il permet également les invocations asynchrones.

BTD/GL/.NET 19

CENTRALE
L Y O N

Du côté Web : ASP.NET / JSP

Bertrand DAVID : Génie Logiciel

- ASP.NET et JSP sont toutes deux écrites en langage de haut niveau (Java pour JSP, les langages du CLS pour ASP.NET) et sont compilées (en bytecode JVM pour Java, en IL puis code natif pour ASP.NET) pour accélérer l'accès. La syntaxe et les concepts sont similaires.
- Mais, surtout dans ce domaine, on peut dire que ASP.NET est orienté langage car il autorise un très grand nombre de scripts (pour chaque langage compatible .NET) alors que JSP est orienté plate-forme puisqu'il fonctionne sur de nombreux serveurs.
- Les WebForms apportent une couche d'abstraction par rapport au langage cible de présentation (HTML, WML...) dont JSP ne dispose pas.

BTD/GL/.NET 20

CENTRALE
L Y O N

Connexion à la base de données : JDBC / ADO.NET

Bertrand DAVID : Génie Logiciel

- Si dans les deux plates-formes, on dénote bien le découplage entre le modèle de données concret et l'utilisateur (en utilisant DataSet et ResultSet), on s'aperçoit que :
 - ➔ ADO.NET se base sur XML et travaille toujours en non-connecté. A l'aide de HTTP et XML, les services Web et ADO.NET permettent de mettre en ligne des services, de partager et mettre à jour des bases de données simplement et indépendamment de l'architecture cible.
 - ➔ L'architecture J2EE est moins ouverte de par sa dépendance envers Java (RMI, messagerie Java). Un pont CORBA existe, mais n'est pas directement utilisable.
 - ➔ Le modèle de données de JDBC est connecté à la différence d'ADO.NET.

BTD/GL/.NET 21

CENTRALE
L Y O N

WinForms et Java Swing

Bertrand DAVID : Génie Logiciel

- Bien que fortement orientés réseau, .NET et J2EE permettent le développement d'applications graphiques classiques.
- Swing et WinForms sont orientés-objets en respectant le paradigme Modèle-Vue-Contrôleur (MVC).
- Après longue maturation (échec de AWT), Swing fonctionne de manière presque similaire sur toutes les plates-formes cibles.
- WinForms est pour l'instant restreint à Windows uniquement.

BTD/GL/.NET 22

CENTRALE
L Y O N

Les terminaux mobiles

Bertrand DAVID : Génie Logiciel

→ Les deux architectures comportent également un support pour terminaux mobiles

- .NET Mobile SDK pour .NET (limité à Windows CE pour l'instant)
- J2ME (Micro Edition) pour Java.

BTD/GL/.NET 23

CENTRALE
L Y O N

Bref...

Bertrand DAVID : Génie Logiciel

- Malgré la différence d'âge entre les deux technologies, elles ne présentent pas de différences majeures ni de révolution.
- C'est dans une optique d'approche du problème que .NET fait preuve de plus de simplicité. Mais si ce dernier se démarque franchement de ce que Microsoft nous a habitué, il reste à voir si le produit saura se vendre sur le marché.
- Il est clair que, avec ce nouveau produit, Microsoft propose un moyen de construire des Web Services qui s'appuient sur les technologies les plus en vogue comme XML et SOAP et en reprenant certains points forts de J2EE.
- Cette offre se veut avant tout developper-friendly, parce qu'il prétend pouvoir supporter une vingtaine de langages. En revanche, la portabilité laisse à désirer, et reste encore à l'état de promesse.
- Il semble donc que .NET est un produit qui concurrencera Java dans le domaine des serveurs et surtout des Web Services dès qu'il sera complet.
- L'architecture du .NET Framework est semblable à celle de J2EE, et l'outil Visual Studio .NET permet d'agir à tous les niveaux. Il est donc facile et rapide de construire des Web Services, mais la maîtrise du développement en est d'autant plus limitée.
- Mais, laissons le temps montrer de lui-même les qualités et les défauts de Microsoft .NET, ses succès et ses échecs.

BTD/GL/.NET 24

CENTRALE L Y O N	<h2>Bibliographie</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none">→ <i>Jean-François Bobier Microsoft .NET : architecture et services</i> Mémoire d'option IDL, www.developpez.biz/downloads/dotnet/MemoireDotNet.pdf → <i>Markus Völter, CTO, Mathema AG / Michael Stal, SIEMENS : Comparing J2EE with .NET</i> Présentation PowerPoint, www.voelter.de/data/presentations/J2EE_vs_NET_MV.ppt → <i>Kevin Hoffman, Jeff Gabriel, Denise Gosnell, Jeff Hasan, Christian Holm, Ed Musters, Jan Narkiewickz, John Schenken, Thiru Thangarathinam, Scott Wylie, Jonothon Ortiz : .NET Framework Professionnel</i>, Edition Wrox Press France SARL → <i>IBM Competitive Technology Laboratory : J 2EE vs. .NET Comparison</i>, Présentation Powerpoint, ftp://ftp.software.ibm.com/software/websphere/partners/roadshow/scene1-j2ee-dotnet.pdf
BTD/GL/.NET	25