

Ambientes de Desenvolvimento Avançados

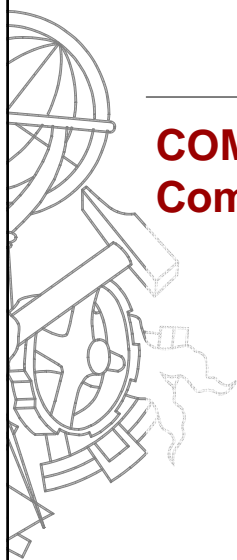
<http://www.dei.isep.ipp.pt/~jtavares/ADAV/ADAV.htm>

Aula 12 Engenharia Informática

2004/2005

José António Tavares
jrt@isep.ipp.pt

1



COM/OLE: Componentes Base



2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

2



COM/OLE: Componentes Base

- GUID = Globally Unique Identifier
- Interfaces
- IUnknown
- Qual o 'aspecto' de um Interface?
- Implementação de Objectos
- Dispatch
- Polimorfismo no COM
- Mecanismos de Reutilização no COM
 - Delegação
 - Agregação
- Interface IClassFactory

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

3



Polimorfismo no COM

Objectos diferentes (de diferentes classes) podem partilhar os mesmos interfaces. Cada objecto pode implementar os interfaces de forma diferente.

Exemplo:

```
interface IAnimal : IUnknown
{
    HRESULT Eat(...);
    HRESULT Sleep(...);
    HRESULT Procreate(...);
}
```

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

4

Polimorfismo no COM

Para criar novos interfaces poderíamos fazer...

```
interface IRabbit : IAnimal
{
    HRESULT RaidGardens(...);
    HRESULT Hop(...);
    HRESULT DigWarrens(...);
}

interface IKoala : IAnimal
{
    HRESULT ClimbEucalyptusTrees(...);
    HRESULT PouchOpensDown(...);
    HRESULT SleepForHoursAfterEating(...);
}
```

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

5

Polimorfismo no COM

Ou poderíamos fazer...

```
interface IRabbit : IUnknown
{
    HRESULT RaidGardens(...);
    HRESULT Hop(...);
    HRESULT DigWarrens(...);
}

interface IKoala : IUnknown
{
    HRESULT ClimbEucalyptusTrees(...);
    HRESULT PouchOpensDown(...);
    HRESULT SleepForHoursAfterEating(...);
}
```

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

6



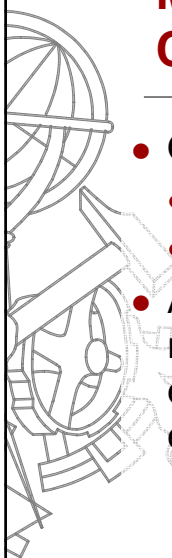
Polimorfismo no COM

Um objecto que implemente **I Animal** e **I Rabbit** de forma separada é funcionalmente equivalente a um que implemente o interface **I Rabbit** derivado do **I Animal**.

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

7



Mecanismos de Reutilização no COM

- O COM tem dois mecanismos de reutilização:
 - **Agregação**
 - **Delegação**
- Ambos são mecanismos que permitem reutilizar componentes em tempo de execução (ao contrário da herança, possível em tempo de compilação).

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

8

Mecanismos de Reutilização no COM

- É possível implementar um objecto **Koala** com os interfaces **I Koala** e **I Animal**.
- Se no sistema já existir um objecto **Animal** com o interface **I Animal** é possível reutilizar esse objecto na implementação do objecto **Koala**.



2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

9

Mecanismos de Reutilização no COM

- A Delegação é a forma mais simples e mais frequente de reutilização no COM.
- Na delegação o objecto que está a ser reutilizado não necessita de nenhuma implementação especial.
- O objecto 'exterior' utiliza o objecto 'interior' como qualquer aplicação cliente que tenha que aceder a um objecto.
- O objecto 'interior' não tem conhecimento que está a ser utilizado por outro objecto...

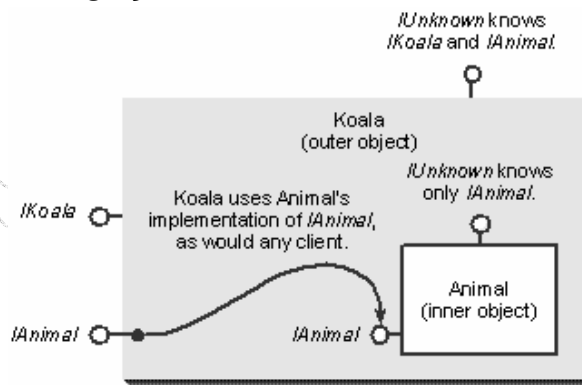
2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

10

Mecanismos de Reutilização no COM

- Delegação:



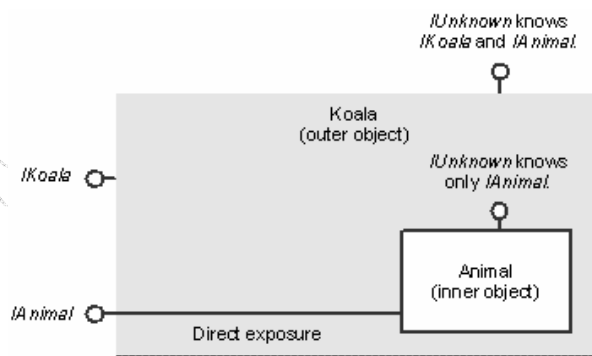
2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

11

Mecanismos de Reutilização no COM

- Agregação



2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

12



Mecanismos de Reutilização no COM

- Existem situações em que seria desejável reutilizar completamente interfaces de um objecto existente sem que para tal fosse necessária nenhuma alteração nem a necessidade de codificar funções que a única coisa que fazem é delegar noutro objecto.
- Suponhamos que para o objecto **Koala** desejamos reutilizar completamente a implementação do interface **IAnimal** do objecto **Animal** sem qualquer modificação.
- Basicamente o que se quer fazer é expor directamente o interface **IAnimal** do objecto **Animal** agregado do objecto **Koala** (tal como se fosse um interface normal do Koala).

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

13



Mecanismos de Reutilização no COM

- **Mas existe um problema...**
 - Os clientes do objecto **Koala** contam com as propriedades reflexiva e transitiva de **IKoala : QueryInterface** e **IAnimal : QueryInterface**.
 - Estes também esperam que os métodos **AddRef** e **Release** chamados de qualquer interface controlem a 'vida' do objecto **Koala**.
 - No entanto apesar da invocação de **IKoala : QueryInterface(IID_IAnimal)** retornar o interface **IAnimal** do objecto **Animal**, a invocação de **IAnimal : QueryInterface(IID_IKoala)** falha, não satisfazendo desta forma os princípios enumerados do **QueryInterface**.

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

14



Mecanismos de Reutilização no COM

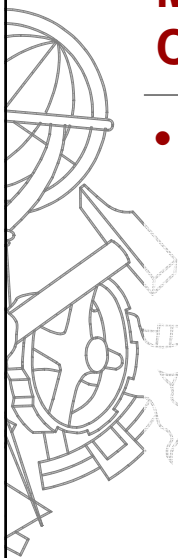
• Justificação

- A implementação desta função, assim como do **AddRef** e do **Release** reside no objecto **Animal** quando invocada pelo interface **IAnimal** e este não tem qualquer conhecimento do objecto externo **Koala**.

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

15



Mecanismos de Reutilização no COM

• Solução

- O objecto **Animal** tem que implementar o comportamento correcto das funções do interface **IUnknown** para que o **QueryInterface** funcione e o contador de referências afecte o objecto **Koala** como um todo único.
- Para que isto seja possível o **Koala** tem que passar um ponteiro do seu interface **IUnknown** para o objecto **Animal** em tempo de criação.
- ...

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

16

Mecanismos de Reutilização no COM

- **Solução (cont.)**

- Para que isto aconteça não existem grandes problemas uma vez que o objecto **Koal a** existe antes do objecto **Ani mal** (criação *top-down*) e assim terá o seu interface **I Unknown** pronto para passar na criação do objecto **Ani mal** agregado.
- Para que um objecto suporte a agregação (tal como o **Ani mal**) este deve implementar uma função de criação que aceite como parâmetro um ponteiro para o interface **I Unknown** do objecto exterior ou controlador.

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

17

Mecanismos de Reutilização no COM

- **Solução (cont.)**

- Assim, quando o objecto **Ani mal** é criado este recebe um ponteiro para um interface **I Unknown** exterior. Se receber um ponteiro NULO este sabe que não está a ser 'agregado', caso contrario este deve delegar todas as chamadas a metodos do seu **I Unknown** nos métodos equivalentes do **I Unknown** exterior.
- Desta forma, qualquer cliente que invoque os métodos **AddRef**, **Release**, e **QueryInterface** do **I Ani mal** estará a chamar as correspondentes do interface **I Unknown** do objecto **Koal a**.

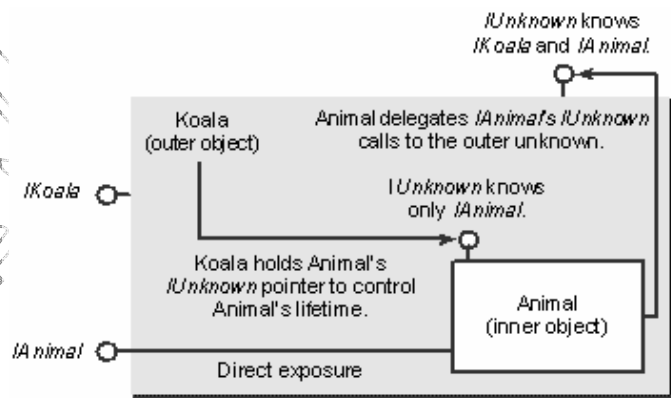
2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

18

Mecanismos de Reutilização no COM

- Agregação ...



2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

19

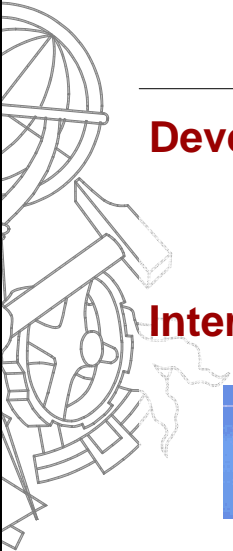
Questões




2004/2005

ADAV
Ambientes de Desenvolvimento Avançados


20



Developing with the .NET Framework




Interoperability



2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

21



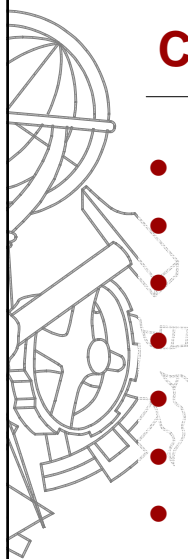
Developing with the .NET Framework

By
Rob Howard
Program Manager
.NET Framework Team
Microsoft Corporation

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

22



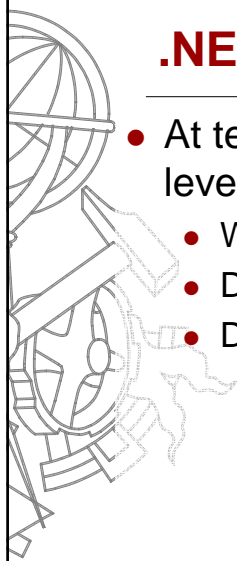
Conteúdo

- Overview
- .NET Evolution
- Design points and Architecture
- Common Language Runtime
- Class Libraries
- Windows Forms
- ASP.NET



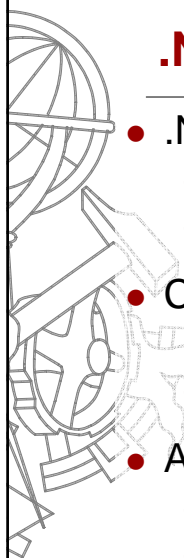
.NET Overview

- The .NET Framework is part of larger .NET space.
- It comprises:
 - the Common Language Runtime (CLR)
 - a large number of partially interfaced, partially class-based frameworks, packaged into assemblies
 - number of tools



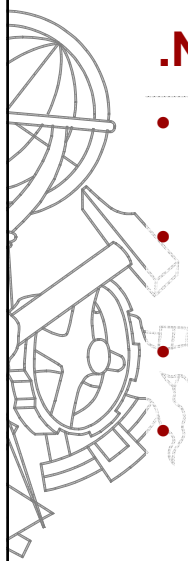
.NET Overview

- At technical level, .NET targets three levels:
 - Web services
 - Deployment platforms (server and clients)
 - Developer platform.



.NET Overview

- .NET Framework
 - Simple, secure, modern, multi-language
 - Great XML and Web Services support
- Comprehensive class libraries
 - Consistent across languages
 - Object oriented - inheritance, etc.
- ASP.NET is a incredible web platform
 - Controls, web services
 - Fast, scalable, available, reliable



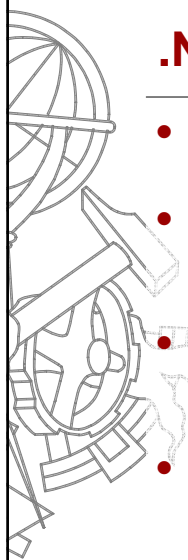
.NET Overview

- Assemblies are units of development, versioning, and management in .NET – that is, they are the .NET software components;
- Assemblies contain metadata, modules, and resources, all of which are expressed in a platform-independent way;
- Code in modules is expressed in CIL (common intermediate language);
- MSIL (Microsoft intermediate language) is a CIL-compliant superset, with instructions added to enable CLR interoperations features that go beyond the CLI (common language infrastructure) specification;

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

27



.NET Overview

- CLR either compiles at install- or at load-time, always executing native code;
- CLR reflection and other type-based concepts cover a large type system space called CTS (Common Type System);
- The Common Language Infrastructure (CLI) specification establishes a language-neutral platform, something like CORBA;
- CLI was jointly submitted to ECMA by Microsoft, Intel, and Hewlett-Packard;

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

28



.NET Overview

- **Ecma International** is a not-for-profit association under Swiss Law/Geneva
- Purpose of Ecma: Development and publication of standards/TRs for Information & Communication Technology and Consumer Electronics
- Formed in **1961** by major multinational computer hardware manufacturers present in Europe.
- Originally "**ECMA**" stood for "**European Computer Manufacturers' Association**".
- In 1994 the name was changed to simply "Ecma".
- "**International**" was added because membership is from around the world, in Asia, Australia, Europe and North America.

www.ecma-international.org

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

29



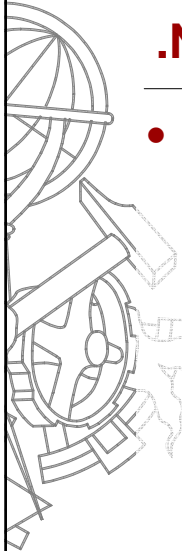
.NET Overview

- Unlike CORBA, CLI also defines an intermediate language (IL) and deployment format (**assemblies**);
- Unlike CORBA and Java, CLI includes support for extensible metadata;
- CLR is the Microsoft implementation of the CLI specification;
- CLR goes beyond CLI compliance and includes support to COM and platform interoperation.

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

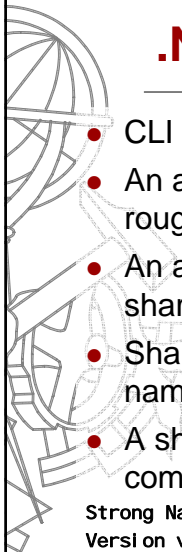
30



.NET Overview

- CLI comprises the specification of the
 - execution engine services
 - loader,
 - JIT compiler and
 - garbage-collecting memory manager)
 - common type system (CTS)
 - and common language specification (CLS)

2004/2005 ADAV 31
Ambientes de Desenvolvimento Avançados



.NET Overview

- CLI deployment units are called assemblies;
- An assembly is a set of files in a directory hierarchy, roughly equivalent to the contents of JAR file.
- An assembly is either private to a single application or shared among multiple applications;
- Shared assemblies are subject to rigorous naming and name resolution scheme;
- A shared assembly is uniquely named by a strong name composed as follows:
Strong Name = (publisher token, assembly name, version vector, culture)
Version vector = (major, minor, build, patch)

2004/2005 ADAV 32
Ambientes de Desenvolvimento Avançados

.NET Overview

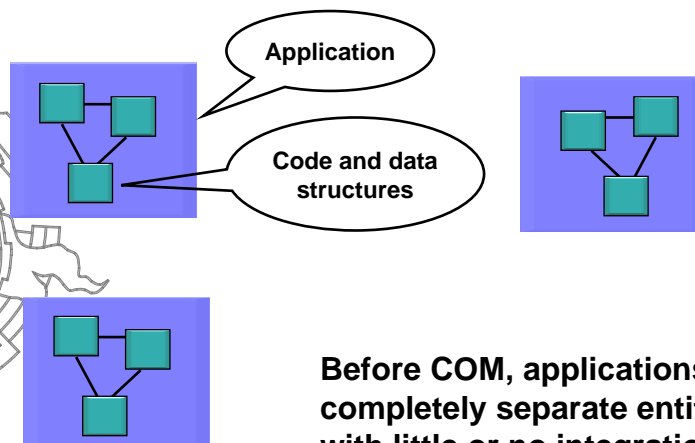
- Shared assemblies are stored in the global assembly cache (GAC) – a facility provided by Windows, starting with XP;
- GAC is effectively a database of assemblies, keyed by their strong names and equipped with policies:
 - **Publisher policy** – states that the assembly is backwards compatible with some older one;
 - **Application policy** – used by the provider of an application to express overriding requirements, such as the need to receive a particular version of an assembly, even if a newer version with its publisher policy claims compatibility;
 - **Machine policy** – used by a machine's administrator to override both publisher and application policy where deemed necessary.

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

33

The .NET Evolution

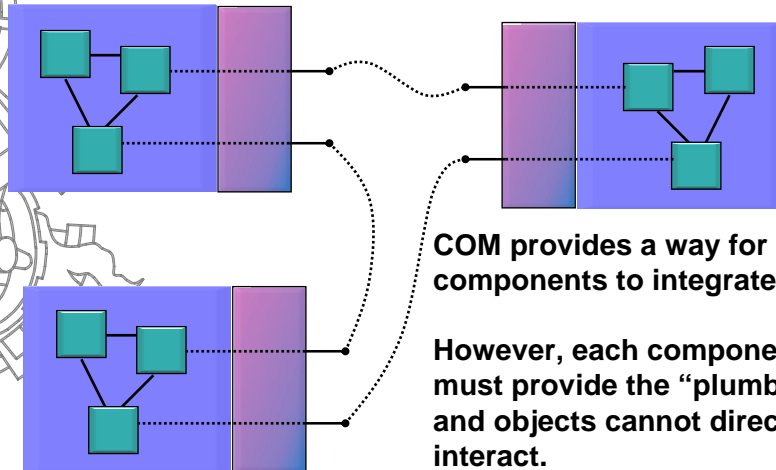


2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

34

The .NET Evolution

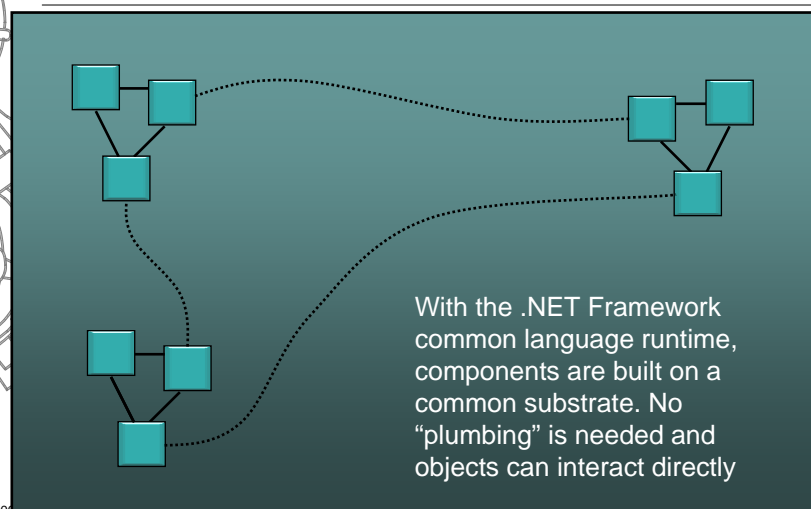


2004/2005

ADAV
Ambientes de Desenvolvimento Avançados


35

The .NET Evolution



2004/2005

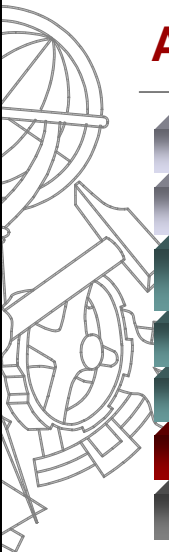
Ambientes de Desenvolvimento Avançados




Design Points

- Simplify development
 - More power, less plumbing
- Unify programming models
 - Across languages and types
- Web standards and best practices
 - Rich XML, standard protocols, stateless
- Easier to deploy, run, & maintain
 - Versioning and availability

2004/2005 ADAV 37
Ambientes de Desenvolvimento Avançados



Architecture

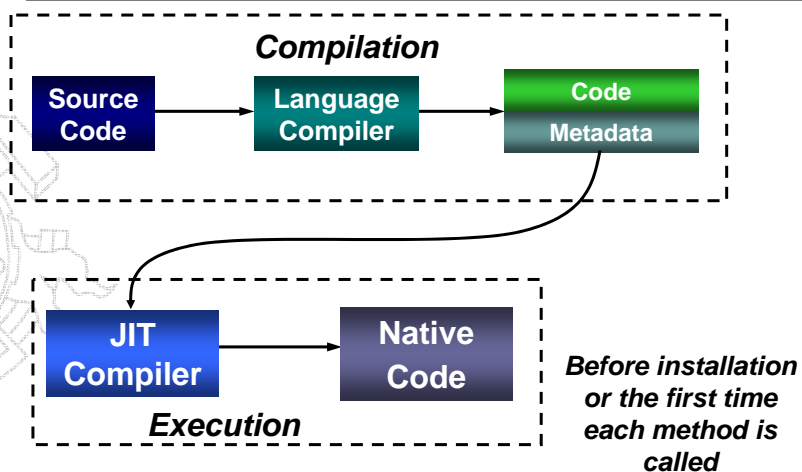


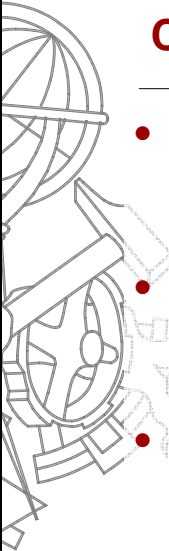
2004/2005 ADAV 38
Ambientes de Desenvolvimento Avançados

Multi-Language

- Over 20 languages supported
 - VB, C++, C#, JScript, Perl, COBOL
- Advanced multi-language features
 - Cross language inheritance, exceptions
- Highly leveraged tools
 - Debuggers, etc work with all languages

Compilation & Execution






Common Language Runtime

- **Manages running code**
 - Type safety and code access security
 - Garbage collection and error handling
- **Provides common type system**
 - Value types (integer, float, etc)
 - Objects, Interfaces
- **Provides access to system resources**
 - Native APIs through P/Invoke
 - COM integration

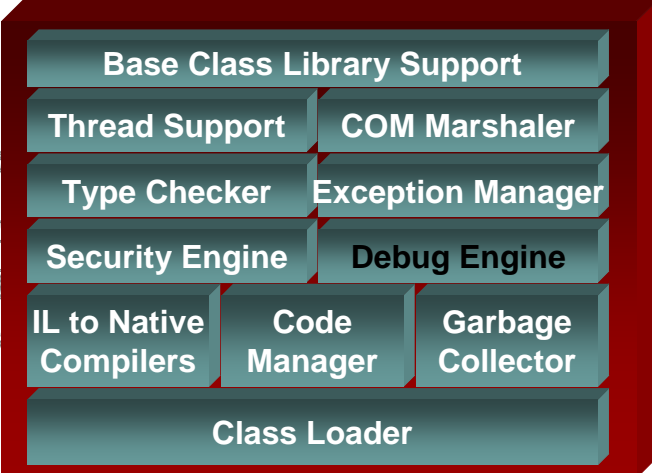
2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

41



Common Language Runtime




Base Class Library Support		
Thread Support	COM Marshaler	
Type Checker	Exception Manager	
Security Engine	Debug Engine	
IL to Native Compilers	Code Manager	Garbage Collector
Class Loader		

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados


42



Simplifies Development

- Eliminates COM plumbing
- No more...
 - Registration => self describing components
 - GUIDs => hierarchical namespaces
 - .IDL files => source code to metadata
 - HRESULTs => structured exceptions
 - IUnknown => root object class
 - AddRef/Release => garbage collector
 - CoCreateInstance => "new" operator

2004/2005 ADAV
Ambientes de Desenvolvimento Avançados 43



Simplified Deployment

- No registration required
 - Code is completely self-describing
 - Simply copy components with app
- Zero-impact installation
 - Installing one app will not affect another
 - Global and non-global components
- Side-by-side execution
 - Multiple component versions co-exist
 - Multiple app versions co-exist

2004/2005 ADAV
Ambientes de Desenvolvimento Avançados 44

Seamless Integration

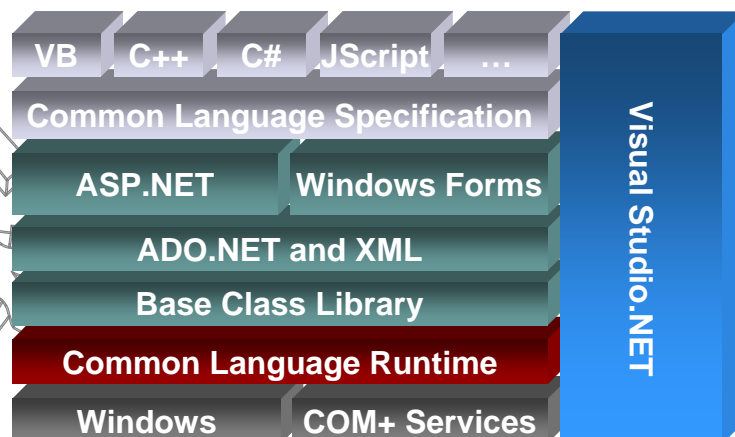
- Use .NET classes as COM objects
 - No extra development work required
- COM objects can be imported
 - No ref counting or COM API needed
- .NET classes utilize COM+ services
 - Transactions, Object pooling, etc...

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

45

Architecture



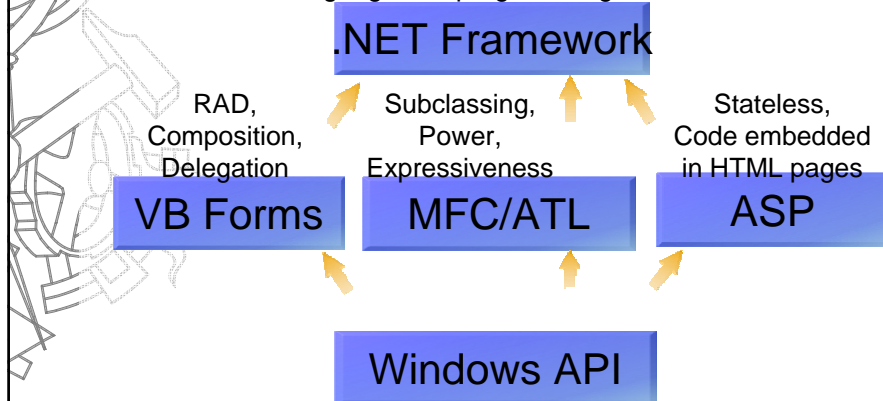
2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

46

A Unified Model

Consistent API availability regardless of language and programming model

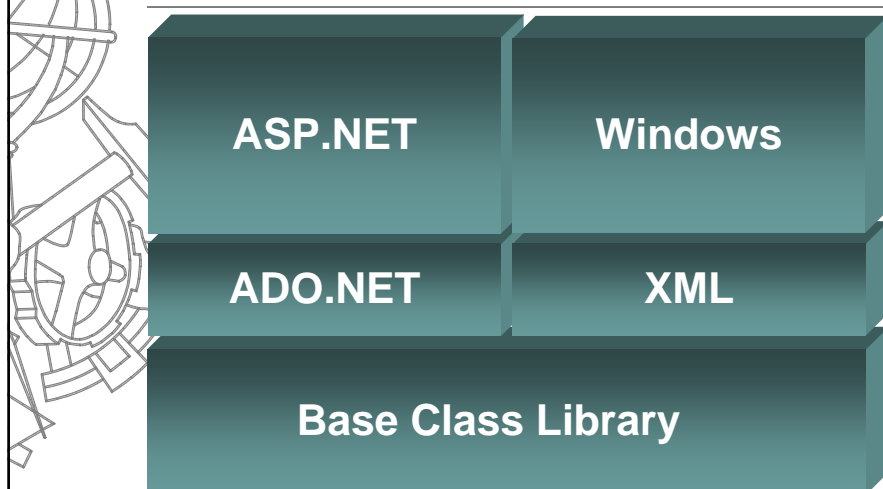


2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

47

.NET Framework Namespaces

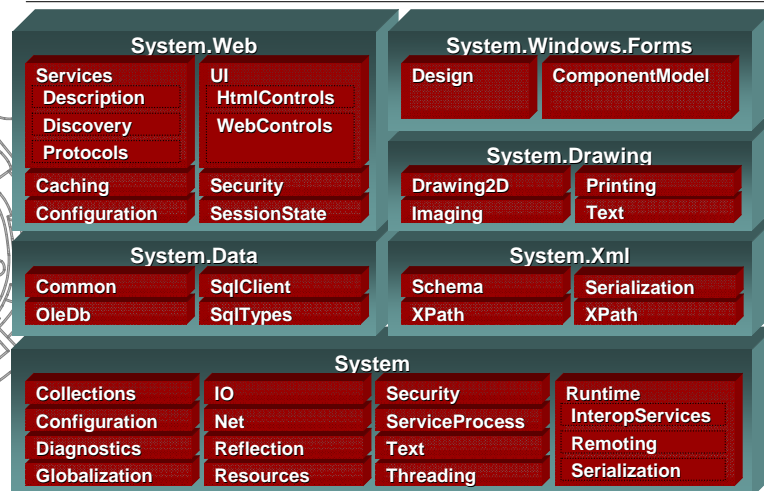


2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

48

.NET Framework Namespaces



2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

49

Data and XML

- Managed Providers
 - Stream-level access to data sources
 - Fastest way to get data out of SQL
- ADO.NET
 - **DataSet** – In-memory buffer
 - **DataReader** – Synonymous with RecordSet
- System.Xml - Great Standards support
 - 100% OASIS compliance
 - XSLT 1.0, XPath 1.0, XSD 1.0, DOM2

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

50



Windows Forms

- Rich Windows applications
 - Combines best of VB Forms, MFC and WFC
 - Easy access to Win32 ® API
- Provides advanced features
 - Visual forms inheritance, automatic layout
 - Advanced graphics support – GDI+
- Secure
 - Code access security prevents harm
 - No Internet Explorer security dialogs!

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

51

How Much Simpler?

Windows API

```
HWND hwndMain = CreateWindowEx(  
    0, "MainWClass", "Main Window",  
    WS_OVERLAPPEDWINDOW | WS_HSCROLL | WS_VSCROLL,  
    CW_USEDEFAULT, CW_USEDEFAULT,  
    CW_USEDEFAULT, CW_USEDEFAULT,  
    (HWND)NULL, (HMENU)NULL, hInstance, NULL);  
ShowWindow(hwndMain, SW_SHOWDEFAULT);  
UpdateWindow(hwndMain);
```

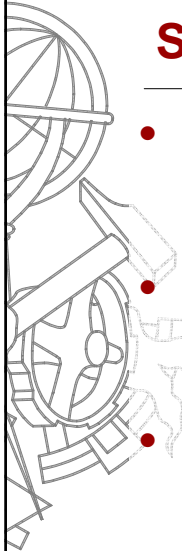
.NET Framework

```
Form form = new Form();  
form.Text = "Main Window";  
form.Show();
```

2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

52



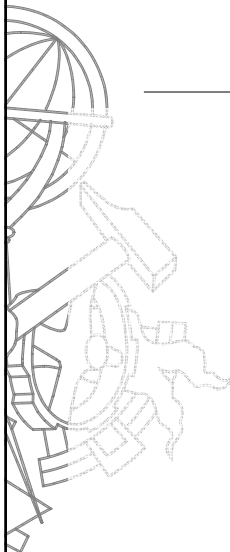
Summary

- .NET Framework
 - Simple, secure, modern, multi-language
 - Great XML and Web Services support
- Comprehensive class libraries
 - Consistent across languages
 - Object oriented - inheritance, etc.
- ASP.NET is a incredible web platform
 - Controls, web services
 - Fast, scalable, available, reliable


2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

53



Questões



2004/2005

ADAV
Ambientes de Desenvolvimento Avançados

54