

Rückblick

- Einfache, aus 3 Komponenten bestehende, Anwendung
- Was ist eine Komponente?
 - Eine einzelne Klasse?
 - Eine ganze Anwendung?



(c) 2004 AG Sysoft - University of Trier

Komponente

- In sich abgeschlossener Baustein
 - Klar definierte Schnittstelle (Interface)
 - Interne Strukturen unsichtbar
 - Verwendete Programmiersprache unsichtbar
 - Verwendete Bibliotheken etc. unsichtbar
 - Verwendete Laufzeitplattform unsichtbar
(Verteilte Komponenten \Rightarrow Kommunikation)
- Größe
 - Kann Klasse sein, kann ganze Anwendung sein
 - Typische Größe liegt zwischendrin
- Architekturaspekte innerhalb einer Komponente
- Beispiel: Rechenknecht auf Interfaces umstellen



(c) 2004 AG Sysoft - University of Trier

Höhere Entwurfsmuster?

- Design Pattern
 - Musterkataloge
 - Gamma et al. Liefert Standardkatalog
 - Diverse domänenspezifische Kataloge
- Beispiel: Komplexe Komponente
 - Facade
 - Spezielle Klasse(n) delegiert Funktionen des Interface an implementierende Klassen
 - Factory / Factory Method
 - Erzeugung von Objekten bzw. Objektgruppen, wenn die Instanziierung über einen Konstruktor nicht klappt



(c) 2004 AG Sysoft - University of Trier

Wie kommunizieren Entwickler ihre Ideen?

- Welche Bereiche sind im Beispiel aufgefallen
 - Wie stelle ich Komponenten dar?
 - Aufbau einer Komponente?
 - Beziehungen zwischen Klassen, ...?
 - Wer ist aktiv?
 - ...
- Einheitliche Syntax
 - Graphisch: UML
- Aber vorher ...



(c) 2004 AG Sysoft - University of Trier

Software Reuse

3. .NET Framework

Peter Sturm
Universität Trier

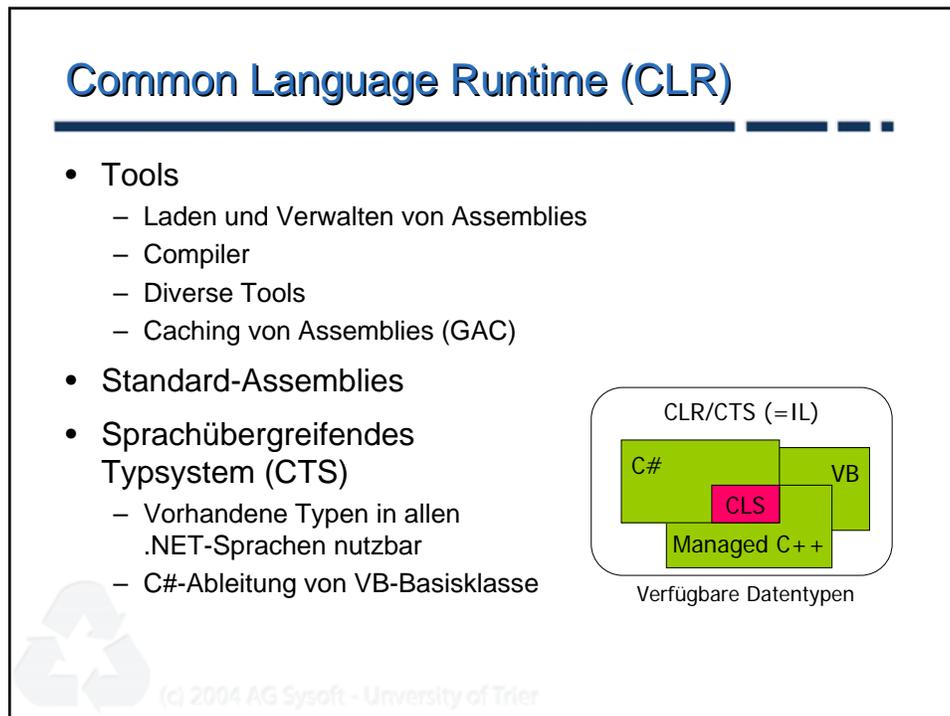
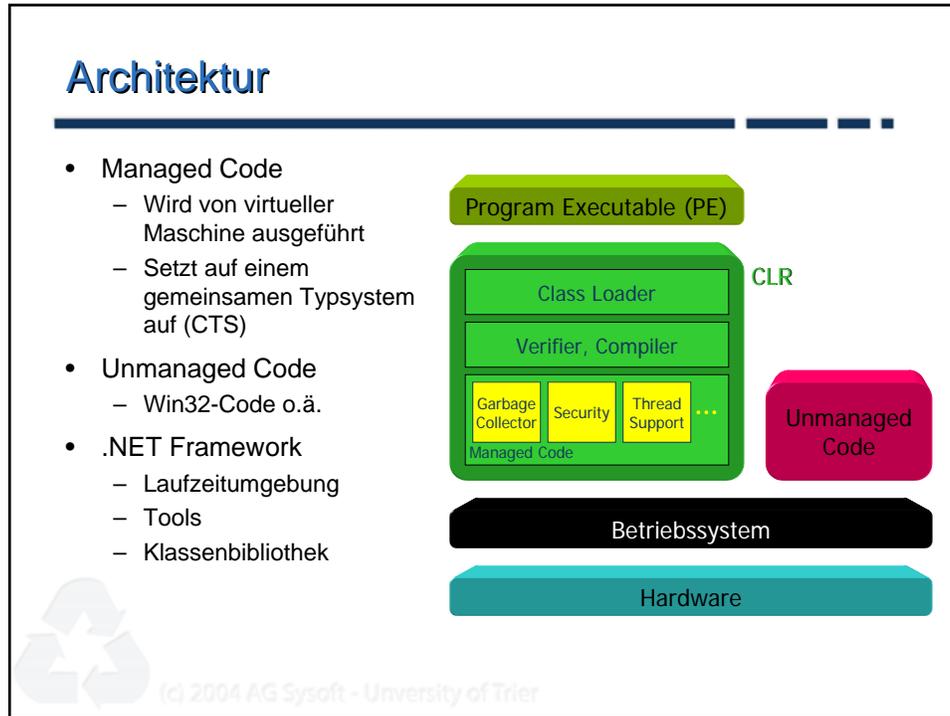
(c) 2004 AG SYSOFT - UNIVERSITY OF TRIER

.NET

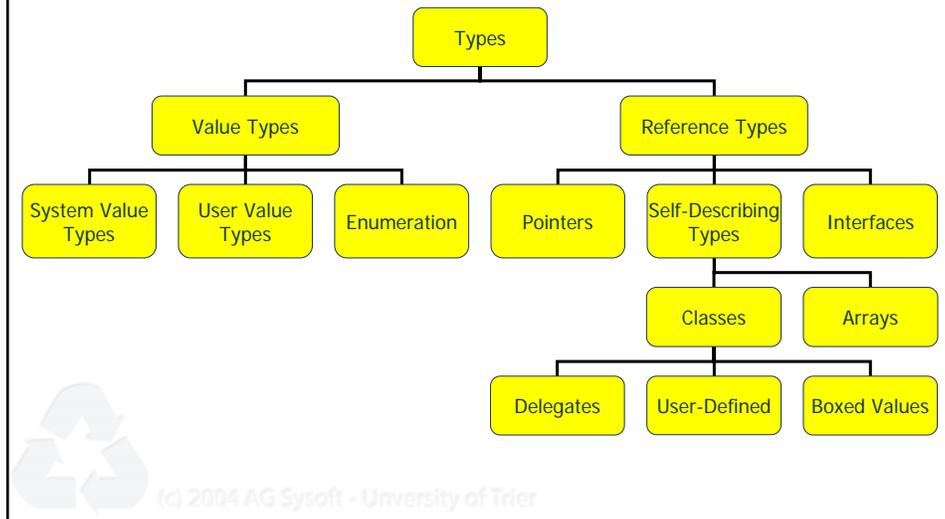
- Völlig neue Systemstruktur als Antwort auf Java
 - Assembly = Komponente
 - Zwischensprache (IL = Intermediate Language)
 - Übersetzung einer Methoden vor erstem Aufruf in nativen und für Zielmaschine optimierten Code (JIT)
 - Einfach Verteilbar
- Weitgehender Wegfall der Windows Registry
 - Assembly enthält alle notwendigen Metadaten
 - Systemweiter Cache benutzter Assemblies
 - Unterschiedliche Versionen einer Assembly auf einem System möglich



(c) 2004 AG Sysoft - University of Trier

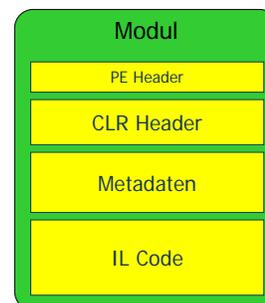


Common Type System (CTS)



Managed Modul

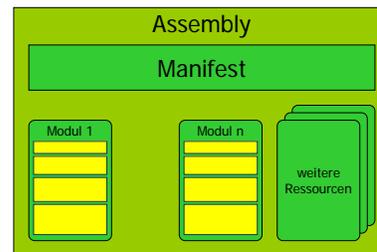
- Übersetzungseinheit in .NET
- Bestandteile
 - PE Header
 - Dateityp (GUI, CUI, DLL, ...)
 - Zeitstempel
 - Primär für unmanaged Code
 - CLR Header
 - Benötigte CLR Version
 - Ort der Metadaten etc.
 - Einstiegspunkt
 - ...
 - Metadaten
 - Definierte Typen und Member
 - Referenzierte Typen und Member
 - IL
 - Der übersetzte Code



(c) 2004 AG Sysoft - University of Trier

Assembly

- Zusammenfassung von
 - 1 oder mehreren managed Modulen
 - Weitere Ressourcen (Graphiken, HTML-Dateien, etc.)
- Struktur wird durch ein Manifest beschrieben
 - Name
 - Shared Name
 - Version
 - Hash
 - Referenzierte Assemblies
 - ...
- Assembly Linker (AL.exe)



Rückblick „Rechenknecht“

- 3 Komponenten
 - Generator
 - Erzeugt Rechenaufgaben
 - GUI
 - Frontend gibt Aufgaben aus und liest Antwort ein
 - Tutor
 - Prüft Ergebnis und „betreut“ den Schüler
- Einfachstkomponenten
 - Jeweils eine instanziierte Klasse
- Was steht in den Assemblies?
 - Werkzeug [ildasm.exe](#)



Assembly Cache

Global Assembly Name	Type	Version	Culture	Public Key Token
Microsoft.Vsa		7.0.3300.0		b03f5f7f11d50a3a
Microsoft.Vsa.Vb.CodeDOMProcessor		7.0.3300.0		b03f5f7f11d50a3a
Microsoft.VSDesigner	Native Images	7.0.3300.0		b03f5f7f11d50a3a
Microsoft_VsaVb		7.0.3300.0		b03f5f7f11d50a3a
msatinterop		7.0.3300.0		b03f5f7f11d50a3a
mscorcfg		1.0.3300.0		b03f5f7f11d50a3a
mscorlib	Native Images	1.0.3300.0		b77a5c561934e089
MSDATASRC		7.0.3300.0		b03f5f7f11d50a3a
MSDOSLHP		7.0.3300.0		b03f5f7f11d50a3a
MSDOSP		7.0.3300.0		b03f5f7f11d50a3a
Office		7.0.3300.0		b03f5f7f11d50a3a
Regguide		1.0.3300.0		b03f5f7f11d50a3a
SoasSudsCode		1.0.3300.0		b03f5f7f11d50a3a
tdiole		7.0.3300.0		b03f5f7f11d50a3a
System	Native Images	1.0.3300.0		b77a5c561934e089
System		1.0.3300.0		b77a5c561934e089
System.Configuration.Install		1.0.3300.0		b03f5f7f11d50a3a
System.Data		1.0.3300.0		b77a5c561934e089
System.Design	Native Images	1.0.3300.0		b03f5f7f11d50a3a
System.Design		1.0.3300.0		b03f5f7f11d50a3a
System.DirectoryServices		1.0.3300.0		b03f5f7f11d50a3a
System.Drawing	Native Images	1.0.3300.0		b03f5f7f11d50a3a
System.Drawing		1.0.3300.0		b03f5f7f11d50a3a
System.Drawing.Design	Native Images	1.0.3300.0		b03f5f7f11d50a3a
System.Drawing.Design		1.0.3300.0		b03f5f7f11d50a3a
System.EnterpriseServices		1.0.3300.0		b03f5f7f11d50a3a
System.Management		1.0.3300.0		b03f5f7f11d50a3a
System.Messaging		1.0.3300.0		b03f5f7f11d50a3a
System.Runtime.Remoting		1.0.3300.0		b77a5c561934e089
System.Runtime.Serialization.Formatters.Soap		1.0.3300.0		b03f5f7f11d50a3a
System.Security		1.0.3300.0		b03f5f7f11d50a3a
System.ServiceProcess		1.0.3300.0		b03f5f7f11d50a3a

- Über Explorer
 - siehe Bild
- Tool
 - gacutil

.NET FCL

- .NET Framework Class Library
 - Teile sind ECMA-Standard
- Größenordnung
 - mehr als 7000 Typen
 - ca. 100 Namespaces
- Alle gängige Funktionalität vorhanden
 - ... zumindest bisher nichts vermißt
- Windows Forms
 - „MFC.NET“

.NET und Open Source

- Microsoft läßt Teile von .NET standardisieren
 - C# und CLR sind ECMA-Standard
 - C# und CLR sollen ISO-Standard werden
- Außen vor bleiben wesentliche Teile der .NET FCL
- Bekannte “Open Source”-Ansätze
 - Mono: <http://www.go-mono.com/>
 - DotGNU: <http://www.dotgnu.org/>
 - Rotor:
<http://msdn.microsoft.com/msdnmag/issues/02/07/sharedsourcecli/default.aspx>



(c) 2004 AG Sysoft - University of Trier

mono::

- Initiator Miguel de Icaza, Ximian
- “Open Source”-Implementierung
 - C#-Compiler
 - CLR
 - Class Libraries
- Class Libraries
 - ECMA-definierten Klassen sind “kein” Problem
 - ASP.NET: Web Forms und Web Services, Controls
 - ADO.NET: Providers für verschiedene DBs fertig
 - Windows Forms: Win32-Implementierung gegen WinLib



(c) 2004 AG Sysoft - University of Trier

DotGNU

- Meta Projekt in der GNU Tradition
- DotGNU Portable.NET (ECMA-basiert)
 - C# Compiler
 - CLR
 - Class Libraries
- Mehr Weltanschauung als bei Mono
- Aktuell geringer Reifegrad
 - Arbeiten an C#-Compiler und CLR
 - Ziele insgesamt etwas diffus



(c) 2004 AG Sysoft - University of Trier

Rotor

- Rotor = Common Language Infrastructure (CLI)
- Microsoft-kontrollierte Öffnung zu Forschungszwecken
 - Shared Source License
 - <http://msdn.microsoft.com/msdnmag/issues/02/07/sharedsourcecli/default.aspx>
- Inhalt:
 - Implementierung der CLI-Laufzeitplattform (ECMA-335) für Windows XP und FreeBSD
 - Compiler für C# (ECMA-334) und Jscript
 - Werkzeuge: assembler/disassemblers (ilasm, ildasm), debugger (cordbg), metadata introspection (metainfo), ...
 - Platform Adaptation Layer (PAL) für die Portierung der CLI von Windows XP nach FreeBSD
 - Entwicklungswerkzeuge: nmake, build, ...
 - Dokumentation und Testsuiten



(c) 2004 AG Sysoft - University of Trier