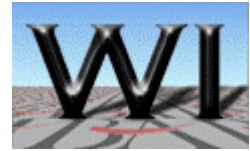




**Berufsakademie Lörrach**  
- Staatliche Studienakademie -



## **Studienarbeit**

### **Microsoft .NET**



### **Einblick und Überblick**

Verfasser: Martin Schaub

Kurs: WWI98

Firma: IBITECH GmbH

Betreuer in der Firma: Dipl. Inf. (FH) Reinhold Klingert

Abgabetermin: 2000-12-08

## Inhaltsverzeichnis

<b>INHALTSVERZEICHNIS</b> .....	<b>2</b>
<b>ABKÜRZUNGSVERZEICHNIS</b> .....	<b>4</b>
<b>ABBILDUNGSVERZEICHNIS</b> .....	<b>5</b>
<b>ANLAGENVERZEICHNIS</b> .....	<b>6</b>
<b>0. PROBLEMSTELLUNG</b> .....	<b>7</b>
<b>1. STRATEGISCHE GEDANKEN</b> .....	<b>9</b>
<b>1.1. Ausgangssituation</b> .....	<b>9</b>
<b>1.2. Zielsetzung und Lösungsansätze</b> .....	<b>10</b>
1.2.1. Abwärtskompatibilität .....	12
1.2.2. Isolierte Komponenten.....	12
1.2.3. Anwendungsportierung ins Web.....	13
<b>2. DAS .NET FRAMEWORK</b> .....	<b>15</b>
<b>2.1. Die Hauptbestandteile des Framework</b> .....	<b>15</b>
2.1.1. Win Forms .....	18
2.1.2. Web Forms .....	20
2.1.3. Die Programmiersprache C# .....	23
<b>3. .NET PRODUKTE UND SERVICES</b> .....	<b>26</b>
<b>3.1. .NET-Building-Blocks</b> .....	<b>26</b>
<b>3.2. Office.NET</b> .....	<b>28</b>
<b>3.3. Die .NET-Enterprise-Server Familie</b> .....	<b>28</b>

<b>3.4. Microsoft .NET-Dienste .....</b>	<b>29</b>
<b>3.5. Ausblick.....</b>	<b>31</b>
<b>QUELLENVERZEICHNIS .....</b>	<b>33</b>

## Abkürzungsverzeichnis

API	Application Programming Interface
ASP	- Application Service Providing - Active Server Pages
bzw.	beziehungsweise
ca.	circa
d.h.	das heißt
f.	folgende (Seiten)
ff.	fortfolgende (Seiten)
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
incl.	inklusive
IT	Informationstechnologie
JIT	"Just - In - Time"
s.o.	siehe oben
SOAP	Simple Object Access Protocol
u.a.	unter anderem
UI	User Interface
VM	Virtual Machine
XML	eXtensible Markup Language
z.T.	zum Teil

## Abbildungsverzeichnis

Abbildung 1 - .NET Framework .....	15
Abbildung 2 - Das Prinzip der Virtuellen Maschine.....	16
Abbildung 3 - Auszug aus dem Win Forms Klassendiagramm.....	19
Abbildung 4 - Auszug aus dem Web Forms Klassendiagramm.....	21
Abbildung 5 - Die Enterprise-Server und ihre .NET Integration .....	29
Abbildung 6 - Der neue MSN- Dienst .....	30

## Anlagenverzeichnis

Anlage 1    Internetquellen (CD- ROM)

Anlage 2    MSDN Show (CD- ROM)

## 0. Problemstellung

Die Microsoft Corporation mit Sitz in Redmond, USA ist gemessen am Umsatz das erfolgreichste Unternehmen in der heutigen Softwarebranche<sup>1</sup>. In den Bereichen der Betriebssysteme, Büroapplikationen und Internetbrowser hält der Softwarehersteller weltweit die Position des Markführers inne.

Auch nach einer möglichen Aufspaltung des Unternehmens durch die US-amerikanische Regierung<sup>2</sup> wäre Microsoft noch so stark am Markt vertreten, dass die Einführung einer neuen „weit reichenden Technologie- Initiative“<sup>3</sup> erhebliche Auswirkungen auf die internationale Informations- und Telekommunikationslandschaft hätte. Die ersten Überlegungen zur Ausarbeitung dieser Initiative begannen bereits vor über zwei Jahren, wobei ein Großteil der Initiative für den Endverbraucher erst im Laufe des kommenden Jahres sichtbar wird.

Im Juni 2000 hat Microsoft das Konzept der *Next Generation Web Services* (NGWS) der Öffentlichkeit vorgestellt und die Ausmaße des damit verbundenen Umstiegs mit dem Wechsel von DOS auf Windows verglichen.

Wenige Tage später auf der *Professional Developer Conference*<sup>4</sup> (PDC) am 11. Juli wurde der endgültige Name für die Plattform bekannt gegeben: Microsoft .NET (gesprochen dot NET). Hierbei wurde deutlich, wie viele unterschiedliche Technologien und Bereiche von der Initiative betroffen sind. „*Es handelt sich um eine Plattform, die die Schnittstelle des Anwenders zum Computer neu definieren wird. Es wird kein Microsoft-Produkt geben, das davon nicht betroffen sein wird.*“<sup>5</sup>

Zum Teil werden die Bereiche, in denen Microsoft tätig ist bzw. ein Microsoft Produkt zum Einsatz kommt, vollständig umstrukturiert, wenn das aus der Initiative entstandene Konzept in die Tat umgesetzt wird.

---

<sup>1</sup> vgl. [Fortune 2000]

<sup>2</sup> vgl. [Heise 2000] <http://www.heise.de/newsticker/data/jk-28.11.00-001/>

<sup>3</sup> vgl. [tecchannel 2000] [www.tecchannel.de/news/20000622/thema20000622-1791.html](http://www.tecchannel.de/news/20000622/thema20000622-1791.html)

<sup>4</sup> vgl. [Microsoft 2000-1] <http://www.microsoft.com/presspass/features/2000/jul00/07-11.NETframework.asp>, 3. Absatz

<sup>5</sup> vgl. [intec-is 1] <http://www.intec-is.de/Service/News/dotnet.htm>

Zukünftig wird die Firmenstrategie von Microsoft durch folgende 3 Ausrichtungen gekennzeichnet sein:

- Das Microsoft .NET Framework,
- die .NET Produkten und Services von Microsoft und
- die .NET Services von Drittanbietern.

Die .NET Strategie ist darauf ausgerichtet die heutige IT- Landschaft langfristig zu verändern. Die Realisierung der .NET Vision folgte jedoch einem evolutionären Pfad. Sie baut auf heute verfügbaren Plattformen und Technologien auf. Die ersten Schritte in die .NET Welt können auf der Basis der gegenwärtigen Windows-Plattform bereits getan werden.

Somit ist eine frühzeitige Auseinandersetzung mit dem Konzept und eine konsequente Verfolgung des Entwicklungsprozesses für den erfolgreichen Einsatz der Neuerungen von großem Vorteil.

Einen Überblick über das .NET Konzept zu geben, sowie dessen Bestandteile und Zusammenhänge aufzuzeigen soll Ziel dieser Studienarbeit sein.



# 1. Strategische Gedanken

Laut Bill Gates, Chairman und Chief Software Architect<sup>6</sup> von Microsoft, dreht sich in Zukunft alles um Software-Komponenten, die so aufeinander abgestimmt sind, dass die einzelnen Anwendungen auf komplexe Daten aus verschiedensten Quellen zugreifen und dem Anwender daraus ein einheitliches Bild bieten. Anlässlich der Eröffnungsrede zur COMDEX/Herbst 2000 gab Gates vor rund 12.000 Fachbesuchern am 13. November 2000 in Las Vegas einen Ausblick auf die neuesten, maßgeblichen Entwicklungen aus seinem Unternehmen. Dieses Kapitel soll einige der Schritte aufzeigen, die zur Erreichung dieser neuen „IT- Landschaft“ nötig sind. Dabei werden auch die zu beseitigenden Probleme auszugsweise erläutert.

## 1.1. AUSGANGSSITUATION

Permanente Innovationen und verbindliche Standards waren die treibenden Kräfte hinter der explosiven Entwicklung des Internet. IP (Internet Protokol), HTML (Hypertext Markup Language) und http (Hypertext Transfer Protocol) sind die Basistechnologien, die aus dem Internet ein Globales Netz machen.

Unternehmen entdecken das Internet und streben eine Präsenz im World Wide Web an. Meist handelte es sich dabei um einen passiven Auftritt mit minimalen Interaktionsmöglichkeiten für die Kunden.

Technologien wie ASP (Active Server Pages) und neue Programmiermöglichkeiten ebnen den Weg für erste E- Commerce- Anwendungen im „Business to Business“ (B2B) und Business to Consumer Bereich (B2C). Mangelnde Integration der Internet Systeme in traditionelle IT- Umgebungen sowie ungenügende Anpassung der Geschäftsprozesse sind die Ursache dafür, dass im Internet zwar Umsätze erzielt werden, dass das E- Business aber nur in wenigen Fällen Gewinn abwirft.

---

<sup>6</sup> vgl. [Gates 2000] <http://www.microsoft.com/billgates/bio.htm>

## 1.2. ZIELSETZUNG UND LÖSUNGSANSÄTZE

Die Applikationslandschaft ist heute nach wie vor vielfach von Anwendungen geprägt, die nicht komponentenbasiert entwickelt wurden. Änderungen und Anpassungen können bei solchen Anwendungen auf Grund ihres starren Aufbaus und der fehlenden Flexibilität<sup>7</sup> nur schwer vorgenommen werden.

Das .NET- Konzept hingegen ist auf die Entwicklung vollständig komponentenorientierter Applikationen ausgerichtet, wobei die zur Programmausführung benötigten Komponenten nicht ortsgebunden sind. Durch diese dezentrale Verfügbarkeit der über das Inter- und Intranet verteilten Module entsteht eine neue Art der Anwendungsentwicklung. Die Programmfunktionalität wird nicht mehr nur durch eine große Applikation zur Verfügung gestellt, sondern durch mehrere Module, die auf bestimmte Funktionalitäten spezialisiert sind, ersetzt. Im Browser, der als Oberfläche und zentrale Ablaufumgebung dient, werden diese Module zu personalisierten Anwendungen zusammengefügt. Der Austausch von Daten erfolgt über die standardisierten Internet- Protokolle und Markup Sprachen<sup>8</sup>, die Microsoft als alleinige Basis für das gesamte .NET- Konzept vorsieht.

Das gemeinsame Datenformat XML<sup>9</sup> (eXtensible Markup Language) spielt hierbei innerhalb des .NET- Konzepts eine herausragend wichtige Rolle. Durch in SOAP<sup>10</sup> (Simple Object Access Protocol) realisierte Verknüpfungen können die unterschiedlichsten Anwendungen und Dienste XML- Daten miteinander austauschen.

Microsoft tritt durch das Anbieten solcher Dienste als Produzent und Host von web-basierenden Basismodulen auf. Auch als Publisher von Web-Diensten und - Anwendungen wie Office.NET will das Softwarehaus tätig werden. Somit beschränkt sich Microsoft nicht mehr nur auf das Anbieten von Entwicklungswerkzeugen, Middleware- Technologie und Betriebssystemen sondern nutzt dieses neu geschaffene Geschäftsfeld als ASP (Application Service Providing) und Anbieter von ASP- Verwaltungssoftware<sup>11</sup>.

---

<sup>7</sup> vgl. [Kruglinski 1998], S. 493

<sup>8</sup> vgl [Kahlisch 2000], <http://elvis.inf.tu-dresden.de/sgml/litkon94.html#a21>

<sup>9</sup> vgl. [Yi 2000] [http://www.fh-wedel.de/~si/seminare/ws98/Ausarbeitung/2.Yi/abgabe\\_1.htm#wasXML](http://www.fh-wedel.de/~si/seminare/ws98/Ausarbeitung/2.Yi/abgabe_1.htm#wasXML)

<sup>10</sup> vgl. [MSDN Library 2000-7] <http://msdn.microsoft.com/library/techart/Hawksoap.htm>

<sup>11</sup> vgl. [Microsoft 2000-2] <http://www.microsoft.com/germany/asp/>

So auch das Betriebssystem Windows, welches in der .NET Version viel modularer und enger mit diversen Komponenten aus dem Internet, sogenannten „Web- Services“ (siehe Kapitel 2.1.2), verknüpft sein wird. Windows wird nicht mehr wie bisher ein monolithisches System sein, sondern vielmehr kann sich jeder Anwender sein Windows nach Bedarf zusammensetzen.

Mit dieser Modularität entkräftet Microsoft auch Argumente im noch immer offenen Kartellrechtsverfahren, da jeder die einzelnen Komponenten von einer anderen Firma verwenden kann<sup>12</sup>.

Aller Wahrscheinlichkeit nach werden all diese .NET- Dienste unter Windows auch performanter sein als unter anderen Plattformen, da Microsoft den Quellcode nicht freigibt und somit als einziger die internen Zusammenhänge und Strukturen kennt. Dennoch ist die Plattformunabhängigkeit und die damit unweigerlich verbundene Öffnung für andere Betriebssysteme ein Ziel, welches sich Microsoft auf die Fahne geschrieben hat. So deutet u.a. die Beteiligung am „Linux- Newcomer“ Corel auf eine mögliche Portierung von .NET auf Linux hin<sup>13</sup>. Linux wird für .NET jedoch nur eine Plattform neben anderen sein, zum Beispiel neben Web- Pads und Internet- fähigen Telefonen, da auch diese von den Vorteilen profitieren können. So könnten diese Geräte zum Beispiel automatisch die neuesten Nachrichten mittels eines .NET Web Service erhalten. Auch MacOS oder andere Unix-Betriebssysteme sind denkbar, da .NET auf offenen Standards basiert.

Darüber hinaus hat Microsoft gemeinsam mit Hewlett-Packard und Intel die neue, objektorientierte Programmiersprache C# (siehe 2.1.3) sowie die Common Language Infrastructure (CLI) der ECMA (European Computer Manufacturers Association) zur Standardisierung vorgelegt<sup>14</sup> und möchte somit C# zur Standardsprache im .NET Umfeld machen.

Die .NET- Strategie kann nur dann von Erfolg gekrönt sein, wenn das Entwickeln einer Applikation oder eines Services einfacher und schneller möglich ist, als dies bisher der Fall war. Deshalb hat sich Microsoft zum Ziel gesetzt, eine hochkomfortable und sprachneutrale Entwicklungsumgebung zu schaffen, die auf einfachste Weise performante Entwicklungen ermöglicht. Um diese Einfachheit zu gewährleisten, greift Microsoft auf die bewährte und bekannte Entwicklungsumgebung von Visual Basic zurück und hat diese sowohl optisch wie auch funktional noch weiter verbessert. Eine kürzere Entwicklungszeit erhofft sich Microsoft durch die konsequente Einhaltung der .NET- Prinzipien, wie z.B.

---

<sup>12</sup> vgl. [Managermagazin 1] <http://www.managermagazin.de/news/artikel/0,1113,82057,00.html>

<sup>13</sup> vgl. [ZDNet 2] Microsoft.NET for Linux

<sup>14</sup> vgl. [ECMA 1] <http://127.0.0.1:800/Default/www.ecma.ch/ecma1/NEWS/NEWS.HTM>

mögliche Zuweisungen per Drag and Drop, ähnliche Klassendiagramme bei Win Forms (siehe 2.1.1) und Web Forms (siehe 2.1.2) und dem Einsatz von visuellen Assistenten. Durch die Ausnutzung der Produktivitätsfeatures der gemeinsamen Sprachlaufzeit wie sprachübergreifende Vererbung<sup>15</sup>, Garbagecollection<sup>16</sup> und Sicherheit werden Entwickler von den Vorteilen des .NET Framework profitieren können.

### 1.2.1. Abwärtskompatibilität

Innerhalb der .NET Architektur bezieht sich die Abwärtskompatibilität hauptsächlich auf die Kompatibilität zum bisherigen Windows- Standard für Komponenten COM (Component Object Model). .NET Komponenten können somit aus bestehenden Entwicklungsumgebungen und Programmiersprachen heraus wie COM- Komponenten angesprochen werden<sup>17</sup>. Umgekehrt ist es auch möglich, COM- Komponenten wie .NET- Komponenten zu behandeln und auch von ihnen abzuleiten, so das die bisher getätigten Investitionen in COM erhalten bleiben.

Durch die Möglichkeit, bestehende COM- Objekte weiterhin zu nutzen und eine problemlose Parallelinstallation von COM und .NET Komponenten zu erlauben ist kein Entwickler gezwungen, auf .NET zu migrieren. Er hat die Möglichkeit zwischen einer kompletten oder schrittweisen Umstellung auf .NET zu wählen. Er kann aber auch seine bestehenden Programme weitgehend unberührt lassen und dennoch, z.B. die neusten .NET Services nutzen.

Um dennoch den notwendigen Migrationenaufwand möglichst gering zu halten, wird Microsoft für die Übernahme von Code aus bestehenden Programmiersprachen in die entsprechende .NET- Version einen Konverter bereitstellen<sup>18</sup>.

### 1.2.2. Isolierte Komponenten

In den derzeitigen komponentenbasierten Betriebssystemen, wie Windows 9x und Windows NT, werden für die gemeinsame Nutzung einzelner Komponenten DLLs (Dynamic Linked Libraries) verwendet<sup>19</sup>. Dabei werden einzelne Softwarekomponenten standardmäßig von mehreren Anwendungen auf dem Computer gemeinsam genutzt. So hat, z.B. jedes Mal, wenn ein Installationsprogramm eine DLL in das Systemverzeichnis kopiert

---

<sup>15</sup> siehe 2.1

<sup>16</sup> vgl. [Richter 2000] <http://msdn.microsoft.com/msdnmag/issues/1100/GCI/GCI.asp>

<sup>17</sup> vgl. [Microsoft 2000-3] <http://msdn.microsoft.com/vstudio/nextgen/technology/csharpintro.asp>, Absatz: Extensive interoperability

<sup>18</sup> vgl. [MSDN Library 2000-8] <http://msdn.microsoft.com/library/techart/vb6tovbdotnet.htm>

<sup>19</sup> vgl. [Kruglinski 1998] S. 533

oder eine Klasse in der COM- Registrierung registriert, der entsprechende Code potentielle Auswirkungen auf andere Anwendungen, die auf demselben Computer ausgeführt werden. Insbesondere wenn eine vorhandene Anwendung zuvor eine ältere Version dieser gemeinsam genutzten Komponente verwendet hat, wird sie nun automatisch unter Verwendung der neuen Version gestartet. Wenn die gemeinsam genutzte Komponente uneingeschränkt abwärtskompatibel ist, stellt dies kein Problem dar. Leider ist in vielen Fällen eine uneingeschränkte Abwärtskompatibilität nur schwierig, oder gar überhaupt nicht realisierbar. Ist diese Abwärtskompatibilität im Einzelfall nicht gegeben, so können bereits auf dem System installierte Anwendungen bei der Installation neuer Applikationen beschädigt werden, vollständig versagen, oder schlimmstenfalls instabil werden. Dieses Problem wird in der Literatur auch als sogenannte "DLL Hell"<sup>20</sup> bezeichnet.

Aus diesem Grund haben innerhalb des .NET Konzeptes isolierte Komponenten (Baugruppen) eine herausragende Bedeutung bei der Anwendungsentwicklung. Isolierte Komponenten haben die Eigenschaft, dass lediglich eine Anwendung auf eine Komponente zugreifen kann. Dies bedeutet, dass die Komponente nicht von mehreren Anwendungen auf dem Computer gemeinsam genutzt wird und Systemänderungen durch andere Anwendungen keine Auswirkungen haben.

Diese Isolation verleiht Entwicklern absolute Kontrolle über den von ihren Anwendungen verwendeten Code. Isolierte bzw. private Anwendungsbaugruppen stellen in .NET- Anwendungen den Standard dar. Der Trend hierzu begann bereits mit Windows 2000 und der Einführung des sogenannten „*local file*“- Verfahrens<sup>21</sup>. Das Verfahren beschreibt das Verhalten des Betriebssystem- Ladeprogramms und der COM- Komponenten bei der Suche nach Bibliotheken. Hierbei wird zunächst das Anwendungsverzeichnis konsultiert bevor auf die entsprechenden Bibliotheken in den Windows System- Verzeichnissen zugegriffen wird.

### 1.2.3. Anwendungsportierung ins Web

Die Entwicklung webbasierter Benutzerschnittstellen wird mit den Web Forms (siehe Kapitel 2.1.2) der Erstellung von Windows- Oberflächen gleichgestellt. Es soll von der Bedienung und Codierung her nur wenige Unterschiede zwischen Windows und Web- Programmierung geben. Die Web Forms ermöglichen es, Anwendungen zu schreiben, die

---

<sup>20</sup> vgl. [MSDN Library 2000-4] <http://msdn.microsoft.com/library/techart/dlldanger1.htm>

<sup>21</sup> vgl. [MSDN Library 2000-3] [http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/winbase/dll\\_53ji.htm](http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/winbase/dll_53ji.htm)

untereinander, auch durch Firewalls<sup>22</sup> hindurch, Informationen austauschen können. Hierbei muss eine hohe Sicherheit gewährleistet sein, sowohl im Bezug auf die sichere Übertragung (Transaktionssicherheit) als auch auf das Sicherstellen des richtigen Adressaten bzw. Absenders mittels digitaler Signaturen.

---

<sup>22</sup> vgl. [MSDN Library 2000-5] <http://msdn.microsoft.com/library/backgrnd/html/cis.htm>

## 2. Das .NET Framework

### 2.1. DIE HAUPTBESTANDTEILE DES FRAMEWORK

Das .NET Framework ist die Laufzeitumgebung für die .NET Anwendungen. Es beinhaltet sämtliche Funktionen, die das Win32 API zur Verfügung stellte, und geht sogar weit darüber hinaus wie Abbildung 1 zeigt.

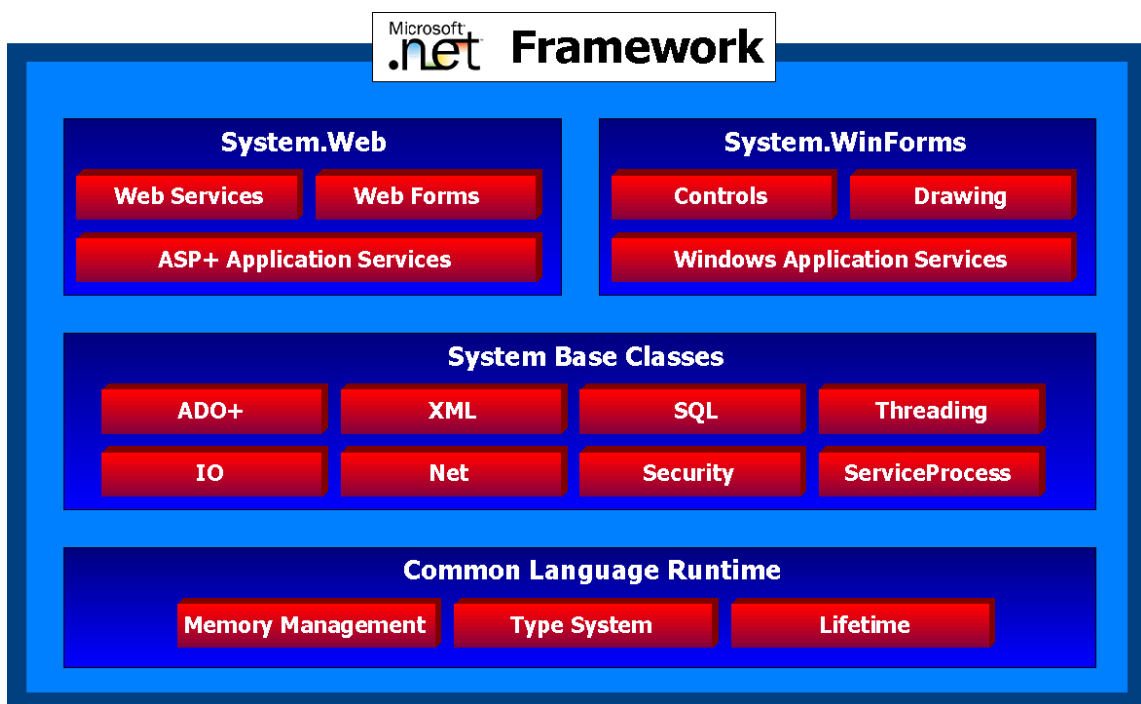


Abbildung 1 - .NET Framework<sup>23</sup>

Dem gesamten Framework liegt die Common Language Runtime (CLR) zugrunde, die eine Virtuelle Maschine (VM), ähnlich der JAVA- VM, darstellt. Eine Virtuelle Maschine „simuliert“ für alle darüberliegenden Ebenen einen kompletten Rechner (siehe Abbildung 2) und bildet eine virtuelle Laufzeitumgebung (Runtime Engine) für die Applikationen. Die CLR erwartet kompilierten Bytecode<sup>24</sup>, der in der sogenannten Intermediate Language (IL) formuliert sein muss.

Diese CPU- unabhängige Maschinensprache, die Microsoft mit anderen Unternehmen und Universitäten entwickelt hat, ist auf einer höheren Ebene als „echte“ Maschinenspra-

<sup>23</sup> Quelle: Eigene Abbildung nach [DNJ 2000-2], S. 17

<sup>24</sup> vgl. [DNJ 2000-1], S. 17

chen einzuordnen. Sie kann unter anderem mit virtuellen Methoden umgehen und kann Objekte erzeugen und initialisieren<sup>25</sup>. Des Weiteren ist sie in der Lage, Fehler als Exceptions (Ausnahmen) zu behandeln. Hierdurch ist es dem Entwickler möglich, gezielte Ausnahmebehandlungen für einzelne Exceptions zu schreiben und die Behandlung aller für ihn nicht relevanten Exceptions der CLR zu überlassen.

Der IL- Code wird auch als Managed Code bezeichnet<sup>26</sup>, da in ihm alle Operationen kontrolliert ablaufen. Das betrifft sowohl den Umgang mit Objekten, die per Garbage Collector entsorgt werden, wie auch Sicherheitsaspekte und Typensicherheit, da Managed Code z.B. keine Pointer und somit auch keine direkte Win32- API- Programmierung kennt.

Die Übersetzung des IL- Codes in echte Maschinensprache erfolgt entweder in einer dynamischen Übersetzung durch den „Just- in- time- Compiler“ (JIT- Compiler, siehe Abbildung 2) während der Programmausführung oder vor dem ersten Programmstart bei der Installation.

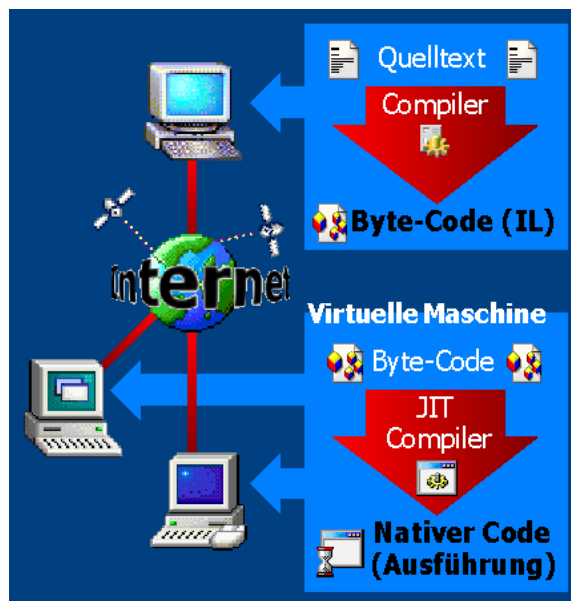


Abbildung 2 - Das Prinzip der Virtuellen Maschine<sup>27</sup>

Da die CLR allen .NET- Sprachen dieselben Klassenbibliotheken zur Verfügung stellt, sind alle .NET- Sprachen einander gleichgesetzt und erzeugen somit auch annähernd denselben IL- Code. Wie das folgende Codebeispiel (Tabelle 1) zeigt, existieren Unter-



<sup>25</sup> vgl. [DNJ 2000-2], S. 16

<sup>26</sup> [VSJ 2000] <http://www.vsj.co.uk/archive/oct2000/outlook1-1000.asp>, Abschnitt „CLR“

<sup>27</sup> Quelle: eigene Abbildung



schiede zwischen den einzelnen Codefragmenten, da die Spezialitäten und Feinheiten der unterschiedlichen Sprachen berücksichtigt werden müssen.

 -IL- Compiler
<pre>.method family hidebysig instance void  button1_Click(class System.Object sender,   class [mscorlib]System.EventArgs e) il managed {   // Code size      12 (0xc)   .maxstack 8   IL_0000: ldstr      "Hello World"   IL_0005: call       value class [System.Windows]System.Windows.DialogResult [Sys- tem.Windows]System.Windows.MessageBox::Show(class System.String)   IL_000a: pop   IL_000b: ret } // end of method Form1::button1_Click</pre>
 Visual Basic .NET- IL- Compiler
<pre>.method family instance void  Button1_Click(class System.Object sender,   class [mscorlib]System.EventArgs e) il managed {   // Code size      16 (0x10)   .maxstack 8   IL_0000: nop   IL_0001: ldstr      "Hello World"   IL_0006: ldc.i4.0   IL_0007: ldnull   IL_0008: call       value class [Microsoft.VisualBasic]Microsoft.VisualBasic.MsgBoxResult [Microsoft.VisualBasic]Microsoft.VisualBasic.Interaction::MsgBox(class System.Object, value class [Microsoft.VisualBasic]Microsoft.VisualBasic.MsgBoxStyle, class System.Object)   IL_000d: pop   IL_000e: nop   IL_000f: ret } // end of method Form1::Button1_Click</pre>

**Tabelle 1 - IL- Code im Vergleich: Aufruf einer MessageBox (Hello World) mittels WinForms<sup>28</sup>**

Mittlerweile konnte Microsoft 17 Sprachen<sup>29</sup> für .NET gewinnen, darunter sind u.a.APL, Haskell, Mercury, ML, Oberon, Perl, Python, Scheme und Smalltalk<sup>30</sup>. Bereits bei der Vorstellung des .NET- Framework im Juli 2000 konnte Microsoft mit Eiffel<sup>31</sup> und COBOL<sup>32</sup> zwei Sprachportierungen von Drittanbietern, neben den hauseigenen Sprachen C#, VB.NET und Jscript, präsentieren.

In Bezug auf die Sprachenneutralität ist interessant, dass eine der früheren Hauptfiguren bei der Entwicklung von Objective Pascal und Delphi, Anders Hejlsberg, die technische

<sup>28</sup> Quelle: Eigene Abbildung

<sup>29</sup> vgl. [MSDN 2000]

<sup>30</sup> vgl. [SDMagazine 2000]

<sup>31</sup> vgl. [MSDN Library 2000-1]

<sup>32</sup> vgl. [MSDN Library 2000-2]

Leitung der C#- Entwicklung bei Microsoft übernommen hat und mittlerweile eine führende Rolle im Bereich der Öffentlichkeitsarbeit für .NET inne hat.<sup>33</sup>

### 2.1.1. Win Forms

Win Forms ist ein Formularpaket mit dem Entwickler in der Lage sind, Windows-basierte Anwendungen zu erstellen, die die Features der Benutzeroberfläche des Windows Betriebssystems voll ausschöpfen.

Win Forms nutzt viele der neuen Technologien, u.a.

- ein gemeinsames Anwendungsframework,
- eine verwaltete Ausführungsumgebung,
- integrierte Sicherheit und
- objektorientierte Entwurfsprinzipien.

Darüber hinaus bietet Win Forms eine Komplettunterstützung für das schnelle und einfache Verbinden mit Webdiensten und dem Erstellen von umfangreichen datenorientierten Anwendungen, basierend auf dem ADO.NET- Datenmodell. Mit der neuen, gemeinsam genutzten Entwicklungsumgebung in Visual Studio können Entwickler Win Forms- Anwendungen in jeder .NET- Sprache schreiben. Dies wird durch die spezielle Architektur des Framework möglich, das allen .NET- Sprachen einheitliche Klassenbibliotheken zur Verfügung stellt. So kann innerhalb eines Projektes eine in C# geschriebene Klasse auf die Eigenschaften, Methoden und Typen einer VB- Klasse zugreifen.<sup>34</sup>

Win Forms ist ein sogenannter Namespace<sup>35</sup> des .NET Framework. Durch Namespaces wird die Klassenhierarchie übersichtlicher, da darin Klassen mit einem ähnlichen Einsatzgebiet zusammengefasst werden<sup>36</sup>. Innerhalb eines Namespaces müssen alle Klassennamen eindeutig sein.

Der Win Forms Namespace ist ausschließlich der Windows-Client- UI- Programmierung dediziert. Das bedeutet, dass er den Zugriff auf sämtliche Klassen ermöglicht, die zur Gestaltung der Benutzeroberfläche zur Verfügung stehen. Er baut auf den gleichen Entwurfsprinzipien auf, die auch vom ASP.NET- UI- Paket (den sog. Web Forms, siehe 2.1.2) ge-

---

<sup>33</sup> siehe Anlage 2: Video [MSDN Show 2000]

<sup>34</sup> vgl. [MSDN News 2000] S.3

<sup>35</sup> vgl. [VSJ 2000-2] <http://www.vsj.co.uk/archive/oct2000/outlook2-1000.asp>

<sup>36</sup> vgl [MSDN Library 2000-6]

<http://msdn.microsoft.com/library/dotnet/cpguide/cpconthenetframeworkclasslibrary.htm>

nutzt werden, jedoch mit Unterschieden hinsichtlich der Implementierung. Es gibt keine Klassen, die eine Konvertierung einer Win32- Komponente zu den Webkomponenten durchführen. Allerdings hat die Konsistenz wie in allen Bereichen des .NET Framework Priorität, dem wird durch ein ähnliches Klassendiagramm Rechnung getragen.

Aufgrund dieser Konsistenz der Klassendiagramme sollte der Win Forms- Entwickler möglichst schnell und problemlos mit dem Schreiben von Code in den Web Forms vertraut sein und umgekehrt. Beide Namespaces besitzen z.B. eine *Button*-Klasse, die *Text*, ein standardmäßiges *OnClick*- Ereignis und die Eigenschaften *ForeColor*, *BackColor* und *Font* enthält. Win Forms- Steuerelemente basieren alle auf der *System.WinForms*. *Control*- Klasse (siehe Abbildung 3).

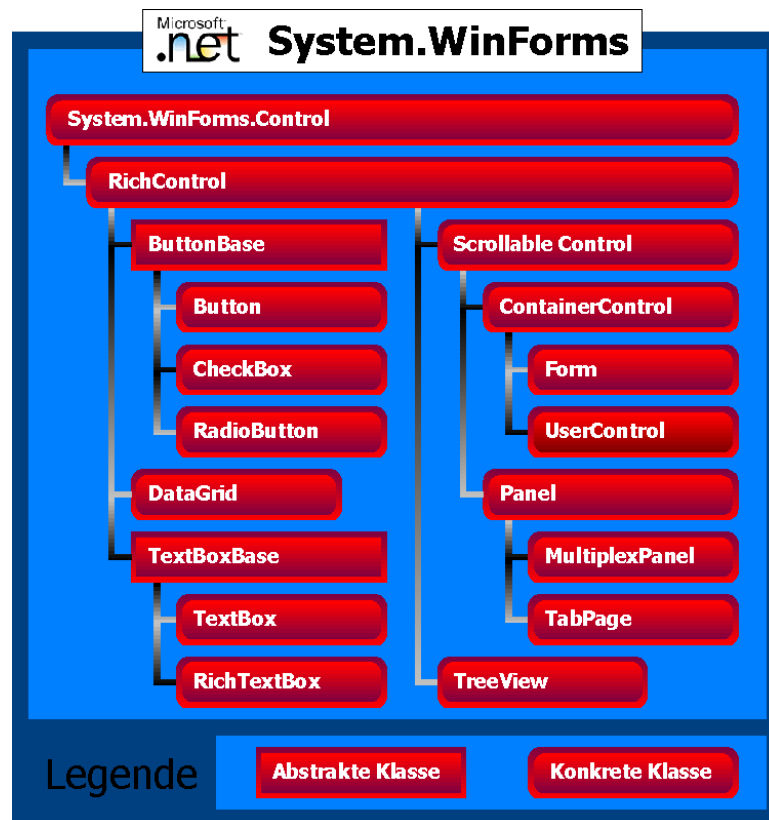


Abbildung 3 - Auszug aus dem Win Forms Klassendiagramm<sup>37</sup>

In *Control* ist die gesamte Windows- Fenster- Standardfunktionalität integriert. Abgeleitet von *Control* ist *RichControl*, über das eine Layoutlogik und die entsprechenden Anweisungen zum Zeichnen des Steuerelements hinzugefügt werden. Die meisten der Steuerelemente im *System.WinForms*-Namespace werden von *RichControl* abgeleitet. Über

<sup>37</sup> Quelle: Eigene Abbildung nach [MSDN News 2000], S. 11

*ScrollableControl* wird die Unterstützung zum Scrollen im Clientbereich eines Fensters hinzugefügt. In der Regel wird auf diese Scrollunterstützung über *ContainerControl* zugegriffen, das sich aus *ScrollableControl* ableitet und unterstützenden Code zum Verwalten von untergeordneten Steuerelementen, Fokusaspekten und Registerkarten hinzufügt. Aus *ContainerControl* leitet sich *Form* ab, das Steuerelement der obersten Ebene in Win Forms, in dem Eigenschaften zum Steuern der Beschriftungsleisten, der Systemmenüs, der nicht rechteckigen Fenster und der Standardsteuerelemente enthalten sind. Von *ContainerControl* leitet sich auch *UserControl* ab, die Basisklasse für Steuerelemente, die Entwickler selbst erstellen können. *UserControl* dient als Host für andere untergeordnete Steuerelemente, wird aber für externe Clients als einzelne Einheit offen gelegt. *UserControl* und *Form* haben beide visuelle Designer in Visual Studio.NET.

Auch wenn das Hauptaugenmerk momentan auf der internetorientierten Entwicklung liegt, ist das Objektprogrammieren der Win32-Plattform nach wie vor ein elementarer Bestandteil der heutigen IT- Welt. Win Forms ist ein komfortables Werkzeug für die Windows- Programmierung, mit dem sich ausgefeilte Anwendungen mit optimaler Benutzeroberfläche erstellen lassen, die hervorragend mit den internet- und datenorientierten Technologien des .NET Framework zusammenarbeiten.

### 2.1.2. Web Forms

Die Web Forms werden auch als Active Server Pages+ (ASP+, ASP.NET) Framework bezeichnet und erlauben die Erstellung von programmierbaren Webseiten als Bestandteil einer Web Applikation.

Eine Vereinfachung der Entwicklung erfolgt unter anderem durch ein ereignisbasierendes serverseitiges Programmiermodell, ähnlich dem Win Forms Modell (siehe 2.1.1). Das „Code Behind“<sup>38</sup> Feature von ASP.NET erlaubt, Design (Präsentation) und Anwendungslogik (Code) zu trennen - eine Datei hält den ASP.NET Code, die zweite Datei ist nur HTML. Damit können Designer und Programmierer unabhängig voneinander an ASP.NET Seiten arbeiten.

Eine Art der "Komponentisierung" von Webseiten wird unterstützt: Sogenannte Pagelets erlauben ASP.NET Seiten in andere ASP.NET Seiten als Komponente importieren zu

---

<sup>38</sup> vgl. [MSDN News 2000], S. 16

werden<sup>39</sup>. Damit können oft benötigte Teile, wie zum Beispiel Benutzeroberflächenfunktionalität, allen betroffenen Seiten einer oder mehrerer Websites zur Verfügung gestellt werden.

Die Anwendungslogik bzw. der Sourcecode wird zur Steigerung der Performance kompiliert, wobei auch hier wie bei den Win Forms der Gedanke der Sprachenunabhängigkeit zur Geltung kommt. Weitere teils clientseitige Features sind Caching, die Unterscheidung zwischen verschlüsselter und unverschlüsselter Übertragungen und Personalisierung<sup>40</sup>.

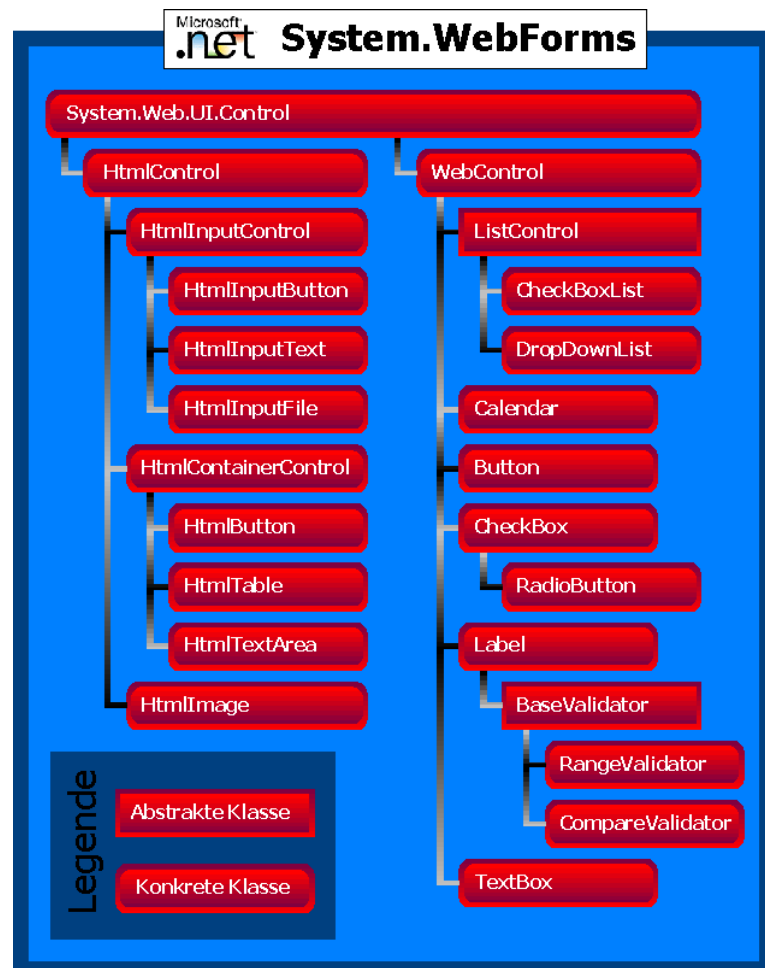


Abbildung 4 - Auszug aus dem Web Forms Klassendiagramm<sup>41</sup>

<sup>39</sup> vgl. [MSDN News 2000], S. 12ff

<sup>40</sup> vgl. [MSDN News 2000] S.13

<sup>41</sup> Quelle: Eigene Abbildung nach [MSDN News 2000], S. 10

Das Klassenmodell (siehe Abbildung 4) unterteilt sich in zwei große Blöcke:

- HTML- controls und
- Web- controls.

Beide Blöcke dienen der Erstellung der Benutzeroberfläche von Webapplikation. Sie stellen beide einen ähnlichen Funktionsumfang zur Verfügung, unterscheiden sich jedoch vor allem in der Entwicklerzielgruppe und im Anwendungsgebiet.

### 2.1.2.1. HTML controls

Aus dem Klassendiagramm wird ersichtlich, dass die HTML controls eine flache, einfache Struktur haben, in der keine abstrakten Klassen vorkommen. Die HTML- controls kommen vor allem bei der Migration von bestehenden ASP- Seiten nach ASP.NET- Seiten zum Einsatz, da die einzelnen Controls jeweils einem HTML Tag entsprechen, wie die Tabelle 2 auszugsweise zeigt.

Control	Entsprechender Tag
HtmlAnchor	<a>
HtmlButton	<button>
HtmlTextArea	<textarea>
HtmlImage	<img>
HtmlTable	<table>

Tabelle 2 - HTML Control Tags<sup>42</sup>

Eine Instanz einer HTML controls Klasse wird durch das Attribut `runat="server"` Attribut erzeugt, wie das folgende Beispiel zeigt. Hierbei wird ein einfaches Eingabefeld (Instanz von *HtmlInputText*) in der HTML- Seite dargestellt

```
<input type="text" runat="server"
id="textBox1" value="Texteingabe">
```

Die HTML controls sind nicht in der Lage automatisch zu erkennen, in was für einem Browser die erstellte HTML Seite dargestellt wird. Der HTML- Code wird nicht automatisch an die Browser angepasst, so dass der Entwickler sich um die Browserunabhängigkeit kümmern muss. Dafür hat er die volle Kontrolle über jedes HTML control und kann genau bestimmen was zum Browser an den Client gesendet und dort dargestellt wird.

---

<sup>42</sup> Quelle: Eigene Abbildung

### 2.1.2.2. Web controls

Die Web controls beinhalten sowohl "traditionelle" Controls (z.B. TextBox und Button) wie auch Controls, die auf einem höheren Abstraktionslevel (z.B. Calendar und DataGrid) angesiedelt sind<sup>43</sup>. Das Klassendiagramm weist eine komplexere, verschachtelte Struktur gegenüber den HTML controls auf. Dies ist nötig, um die deutlich erweiterte Funktionalität abbilden zu können.

In der Basisklasse *WebControl* sind viele Eigenschaften definiert, z.B. ForeColor, BackColor, Font, Enabled, usw., die alle Controls aufweisen<sup>44</sup>. Die Eigenschaften und Methoden tragen absichtlich den gleichen Namen wie ihre Win Forms Pedanten, um eine Konsistenz innerhalb des .NET Framework zu wahren. Gleichzeitig wird so den Programmierern der Einstieg in die Webprogrammierung erheblich vereinfacht und die Gefahr der Programmierfehler vermindert.<sup>45</sup>

Web controls sind in der Lage, automatisch die Version und Art des Browsers, in dem die angeforderte Seite dargestellt werden soll, zu erkennen. Mit diesen zu Grunde liegenden Informationen wird der HTML- Code so angepasst bzw. generiert, dass er die Darstellungsmöglichkeiten des Browsers voll ausnutzt<sup>46</sup>. So enthalten zum Beispiel die Controls zur Wertepfung die Option Scripts zu generieren, die auf dem Client ausgeführt werden, um Interaktivität zu ermöglichen.

Jede Eigenschaft eines Controls kann aus einer Datenbank geladen werden, einige Controls sind sogar in der Lage, ihr Aussehen dem Inhalt einer Datenbank anzupassen. So kann, z.B. eine *ListBox* bestimmte Einträge einfärben (negative Beträge in rot).<sup>47</sup>

### 2.1.3. Die Programmiersprache C#

Ein wichtiger Bestandteil des Microsoft .NET Framework ist die Programmiersprache C# (sprich "C Sharp"). Diese Programmiersprache wurde von Grund auf für das Microsoft .NET Framework designed, und kommt von der Windows Anwendungsentwicklung bis hin zur ASP.NET Programmierung zum Einsatz. Große Teile des .NET Frameworks wurden sogar mit C# entwickelt.

Zunächst fällt auf, dass sich die Syntax eher an der Struktur von C++ anlehnt, die Objektorientierung hingegen auf Smalltalk und Java zurückgeht, wie Tabelle 3 beispielhaft zeigt.

---

<sup>43</sup> vgl. [MSDN News 2000] S.13

<sup>44</sup> vgl. [MSDN News 2000] S.13

<sup>45</sup> vgl. [MSDN News 2000] S. 2

<sup>46</sup> vgl. [MSDN News 2000] S. 13

<sup>47</sup> vgl. [MSDN News 2000] S. 16



	
<pre>using System; class Hello {     static void Main() {         Console.WriteLine("Hello World");     } }</pre>	<pre>class Hello {     public static void main(String[] args) {         System.out.println("Hello World");     } }</pre>

Tabelle 3 - Vergleich zwischen C#- und Java Code<sup>48</sup>

Bei genauerer Betrachtung erinnert C# mehr an das Konzept von Java als an andere Compiler-Sprachen. Merkmale hierfür sind u.a. die Virtuelle Maschine, der JIT- Compiler<sup>49</sup>, der Garbage- Collector als Ersatz für fehlerträchtiges Speichermanagement, die nicht unterstützte Mehrfachvererbung von Klassen und die sichere Typprüfung<sup>50</sup>.

Trotz der großen Gemeinsamkeiten propagieren sowohl Microsoft als auch Sun, dass C# kein Java- Clone sei<sup>51</sup>. Viele Entwickler sind deshalb skeptisch, warum Microsoft die neue Programmiersprache ins Leben gerufen hat, da vergleichbare Systeme schon auf dem Markt sind. Der Grund hierfür ist, dass die Vorteile der verschiedenen Sprachen kombiniert wurden, um eine schnelle und effiziente Entwicklung zu ermöglichen.

Aus diesem Grund wurde C# mit speziellen Features zur Komponentenerstellung ausgestattet. Eigenschaften, Methoden und Ereignisse können direkt ohne Umwege über „Get“ und „Set“- Blöcke codiert werden. Durch die Verwendung sogenannter Attribute wird C#-Code „selbstbeschreibend“, d.h. der Compiler kann direkt aus der C#- Klasse eine XML-Datei generieren, die das Interface der Klasse beschreibt.<sup>52</sup>

Die Zukunft von C# wird sich unter anderem an den Unterschieden zu Java messen lassen müssen (siehe Tabelle 4). Das Überladen von Operatoren, das in C++ seine Wurzeln hat<sup>53</sup>, wurde in die Spezifikation von C# aufgenommen. Insgesamt verspricht C# so eine hohe Flexibilität hinsichtlich der Erzeugung und Verwaltung benutzerdefinierter Datentypen, da der Entwickler in der Lage ist, eigene, auf seine Bedürfnisse zugeschnittene Datentypen zu erzeugen<sup>54</sup>.

<sup>48</sup> Quelle: Eigene Abbildung

<sup>49</sup> vgl. [O'Reilly 2000-1] [http://java.oreilly.com/news/farley\\_0800.html](http://java.oreilly.com/news/farley_0800.html)

<sup>50</sup> vgl. [Microsoft 2000-4], S. 13

<sup>51</sup> vgl. [Entwickler 2000] <http://www.entwickler.com/news/2000/07/14>

<sup>52</sup> vgl. [O'Reilly 2000-2], Deep Inside C#, 1. Antwort

<sup>53</sup> vgl. [Kruglinski 1998] S.9

<sup>54</sup> vgl. [Microsoft 2000-4] S. 35






Eigenschaft			
Virtuelle Maschine	✓	✓	✗
Garbage- Collector	✓	✓	✗
Gemeinsame Basisklasse	✓	✓	✗
Sichere Typprüfung	✓	✓	✗
Mehrfachvererbung von Klassen	✗	✗	✓
Mehrfachvererbung von Schnittstellen	✓	✓	✓
Überladen von Methoden (virtuelle Funktionen)	✓	✓	✓
Überladen von Operatoren	✓	✗	✓
Template- Klassen	✗	✗	✓
Standardisierung	ECMA (in Vorbereitung)	Java Community Process	ISO

Tabelle 4 - Vergleich einiger Eigenschaften zwischen C#, Java und C++<sup>55</sup>

C# besitzt keine eigene Klassenbibliothek, da die Bibliotheken von der CLR (siehe 2.1) bereitgestellt werden. Aus diesem Grund wird die Sprache sehr windowsorientiert. Um jedoch C# wie Java, plattformunabhängig positionieren zu können<sup>56</sup>, müssen die .NET-Bibliotheken auch für die jeweilige angestrebte Plattform bereitgestellt werden.

Mit dem Konzept der Sprachenunabhängigkeit überlässt Microsoft den Softwareentwicklern die Entscheidung welche Sprache sie zur Anwendungsentwicklung einsetzen wollen. Für Java-Programmierer ist dieses Konzept nichts Neues, gibt es doch schon für Cobol, Smalltalk, Python und viele andere Sprachen- Compiler, die Java- Bytecode erzeugen können<sup>57</sup>. Es ist nur eine Frage der Zeit, bis ein Java-Compiler für IL auf den Markt kommt, so dass Java-Applikationen für .NET entwickelt werden können.

<sup>55</sup> Quelle: Eigene Abbildung

<sup>56</sup> vgl. [MSDN Library 2000-9] <http://msdn.microsoft.com/library/techart/InteropdotNET.htm>, Abschnitt "Common Language Runtime"

<sup>57</sup> vgl. [Sun 1999] <http://java.sun.com/docs/books/vmspec/2nd-edition/html/Concepts.doc.html>

### 3. .NET Produkte und Services

Unter Betrachtung der Tatsache, dass sich die .NET Strategie momentan noch eher in der Entwurfsphase als in einer sehr frühen Phase der Umsetzung befindet, ist es doch erstaunlich wie viele Firmen Microsoft bereits zur Herbst COMDEX 2000<sup>58</sup> für die Entwicklung von *Web Services* und damit für die Unterstützung der .NET Strategie gewinnen konnte.

Partnerfirmen	
Allegis Corp.	Made2Manage Systems Inc.
apBirch Co.	OpenTable.com
ASPFree	Phalyanx Systems
Blueskyfrog.com	Rivio Inc.
Maatra (part of Boxspoon)	Rubicon Development
Dell Computer Corp.	State Farm
DMR Consulting Group Inc.	STLNet
Dollar Rent a Car	Street Fusion
e*BIS	ThinQ
Galileo International	WorldStreet
Gazelle Systems	

Tabelle 5 - Partnerfirmen<sup>59</sup>

#### 3.1. .NET-BUILDING-BLOCKS

Neben dem Framework will Microsoft eine Reihe sogenannter "Building- Blocks" entwickeln. Dabei handelt es sich um verteilte, Web- basierte Dienste, die von Microsoft gehostet werden, und die von Service-Providern und Web-Sites in die eigenen Anwendungen integriert werden können.

Die Building- Blocks sollen grundsätzlich von jeder XML- fähigen Plattform unterstützt werden, wobei firmenpolitisch bedingt Windows als optimale Umgebung für die Entwicklung von Web-Diensten zur Verfügung steht.

<sup>58</sup> vgl. [Microsoft 2] <http://www.microsoft.com/presspass/press/2000/nov00/vsfwbeta1PR.asp>

<sup>59</sup> vgl. [MSDN Magazin 2000] <http://207.46.130.166/msdnmag/issues/0900/WebPlatform/WebPlatform.asp>

Microsoft kündigte sieben dieser Building- Blocks an: Neben dem bereits bestehenden Authentifizierungsdienst Passport<sup>60</sup>, der um Funktionen wie Benutzeridentifizierung via Smartcards oder Biometrik-Geräte erweitert werden soll, sind ein Unified- Messaging- Dienst auf Basis von Hotmail<sup>61</sup>, Exchange und Instant- Messenger geplant. Außerdem will Microsoft ein Personalisierungs-Service, einen XML- Web- Speicherdienst, einen Arbeitsgruppen-Kalender sowie einen Directory- und Suchdienst anbieten. Dazu kommt ein Software- Update- Service, der nicht nur für Microsoft-Produkte, sondern auch für andere Hersteller und Plattformen offen stehen soll.

Ein Beispiel für ein bereits realisiertes Projekt stellt der am 13. November 2000 gestartete Web Service „TerraService“<sup>62</sup> dar. Dieser Service stellt jedem Internetbenutzer Satelliten- aufnahmen von fast jedem Ort der Erde zur Verfügung, mit einer Genauigkeit von bis zu einem Meter pro Bildschirmpunkt. Er entstand bereits 1998 aus einer Zusammenarbeit von Compaq, der Geologischen Behördenaufsicht und Microsoft und ist die größte Datenbank des Internets<sup>63</sup>.

Entwickler, die Visual Studio.NET und das .NET Framework einsetzen, haben die Möglichkeit, diesen Service in ihre Applikationen und Services einzubinden. So kann, z.B. ein Maklerbüro, welches sich auf abgelegene Wochenendhäuser spezialisiert hat, jedem Inserenten eine Luftaufnahme des Objektes präsentieren, ohne jemals ein Flugzeug betreten zu haben oder eine Firma mit der Herstellung solche Aufnahmen zu beauftragen. Das Büro muss sich auch nicht mit Schnittstellen- oder Konvertierungsproblemen beschäftigen, da dieser Service in der eigenen Applikation genauso einfach per Drag and Drop benutzt werden kann, wie z.B. die Kalenderfunktionalität von Outlook.

---

<sup>60</sup> vgl. [Microsoft 2000-6] <http://www.passport.com/Consumer/default.asp?PPId=1031>

<sup>61</sup> vgl. [Microsoft 2000-7] <http://www.hotmail.com>

<sup>62</sup> vgl. [TerraServer 1] <http://www.terra-server.com>

<sup>63</sup> vgl. [Microsoft 2] <http://www.microsoft.com/presspass/press/2000/nov00/vsfwbeta1PR.asp>

### 3.2. OFFICE.NET

Microsoft Office wird zu einem .NET- Dienst. Microsoft betritt mit einem seiner kommerziell erfolgreichsten Produkte<sup>64</sup> den Application Service Providing- Markt. Office.NET wird aus einer über das Web verfügbaren Sammlung von Browser-basierten Komponenten bestehen, die sich automatisch auf unterschiedlichen Geräten (PC, Handheld, etc.) installieren. Sollte Microsoft aus den Fehlern der Konkurrenten (Lotus eSuite) lernen, wird es auch mittelfristig möglich sein, das Office als „Fat Client“ zu installieren.

### 3.3. DIE .NET-ENTERPRISE-SERVER FAMILIE

Die Enterprise-Server von Microsoft sind eine wichtige Grundlage für den Erfolg des gesamten Konzepts. Schon die 2000er- Versionen der NET- Enterprise- Server sind durchgängig mit nativem XML- Support ausgestattet<sup>65</sup>. Alle kommenden Versionen werden sich noch stärker an herstellerunabhängigen Standards wie XML und SOAP ausrichten bis sie vollständig auf das .NET Framework aufsetzen und somit zumindest eine theoretische Plattformunabhängigkeit gegeben ist. Auch hier wird sicherlich die Kombination von Microsoft Betriebssystem und Enterprise Server die performanteste sein.

Enterprise Server	Einsatzgebiet
SQL Server 2000	Datenbank- und Analyseserver zur schnellen Erstellung skalierbarer Webanwendungen
Application Center 2000	Bereitstellungs- und Verwaltungstool für Webanwendungen mit hoher Verfügbarkeit, setzt auf Windows 2000 auf
BizTalk Server 2000	Koordinieren von Geschäftsprozessen und Webdiensten innerhalb und zwischen Organisationen.
Exchange Server 2000	Bedienerfreundliche Messaging- und Zusammenarbeitslösung, die Benutzer und Wissen "zusammenbringt".
Host Integration Server 2000	Integrationskomponenten für Hostsysteme
Commerce Server 2000	Erstellung effizienter E-Commerce-Lösungen
Internet Security and Acceleration Server 2000	Integrierter Firewall- und Webcacheserver, der webbasierte Transaktionen sicherer, schneller und leichter verwaltbar machen soll

Tabelle 6 - Die Microsoft Enterprise Server 2000 Familie<sup>66</sup>

<sup>64</sup> vgl. [ASC- Computer 1] [http://www.asc-computer.at/Berichte/2000\\_09/bericht...htm](http://www.asc-computer.at/Berichte/2000_09/bericht...htm)

<sup>65</sup> vgl. [Microsoft 2000-5]

<sup>66</sup> vgl. [Microsoft 2000-5]

Ihre Einsatzgebiete werden sich nicht nur auf das Netzwerk innerhalb einer Firma beschränken, da sie in höchstem Maße auf B2B und B2C ausgerichtet sind. Das nachstehende Diagramm zeigt die Aufgaben der bisherigen Server-Produkte (grün) sowie der neuen .NET- Server in der Weblösungsplattform von Microsoft<sup>67</sup>

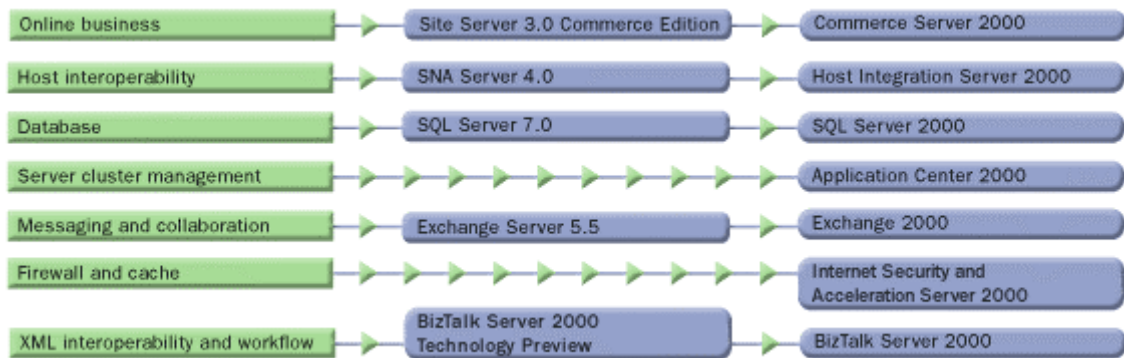


Abbildung 5 - Die Enterprise-Server und ihre .NET Integration<sup>68</sup>

Eine wichtige Rolle in der XML- Strategie spielt Microsoft Biztalk- Server 2000. Mit der integrierten High- Level- Sprache XLANG (gesprochen Slang) ist es auf einfache Weise möglich, Business-Prozesse zu automatisieren, Workflows zu implementieren und dabei selbst lokal verteilte Web- Services auf der Basis von XML zu integrieren. Die grafische Entwicklungsumgebung Biztalk Application Designer generiert automatisch XLANG- Code, der zur Laufzeit ausgeführt wird. Im SQL Server 2000 können die XML- Daten direkt ohne vorherige Konvertierung gespeichert werden, da dieser XML als Input- und Output-Format unterstützt.

### 3.4. MICROSOFT .NET-DIENSTE

Neben diesen Basisdiensten plant Microsoft umfangreiche .NET- Dienste, die die Grundüberzeugung widerspiegeln, dass längerfristig das heute übliche Software-Geschäft von einem Internet- basierten Miet- oder Service-Modell abgelöst werden wird<sup>69</sup>. Microsoft ist davon überzeugt, dass in Zukunft die Anwendungen nicht mehr mittels mehrerer CD's installiert werden, vielmehr werden nur noch die Komponenten aus dem Netz geladen, die auch wirklich benötigt werden. Diese Trennung kann bei Bedarf soweit ausgeweitet werden, bis der Anwender nur noch einen Browser benötigt, um bestehende und zukünftige Funktionen der Applikationen und Services zu nutzen.

<sup>67</sup> vgl. [Microsoft 2000-5]

<sup>68</sup> entnommen aus <http://www.microsoft.com/germany/dna/>

<sup>69</sup> vgl. [MSDN Magazin 2000] <http://207.46.130.166/msdnmag/issues/0900/WebPlatform/WebPlatform.asp>

Zu einem Musterbeispiel für die .NET- Welt soll Microsofts Internet- Dienst MSN werden. Eine erste Vorschau auf die neuen Funktionen bietet der neue integrierte MSN- Client, den man derzeit als Preview- Version von der MSN- Website herunterladen kann<sup>70</sup>.

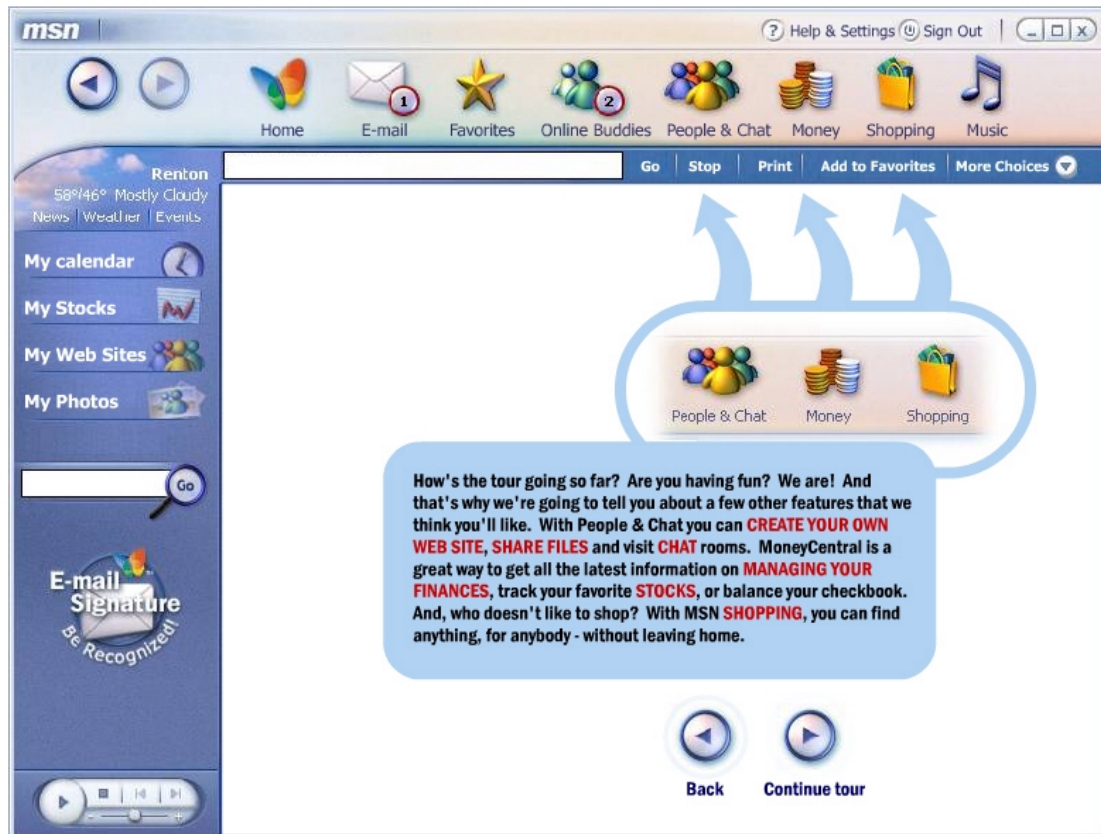


Abbildung 6 - Der neue MSN- Dienst<sup>71</sup>

Ebenfalls im Web zu besichtigen ist „bCentral for .NET“<sup>72</sup>, ein Angebot, mit dem sich kleine und mittlere Unternehmen verschiedene Anwendungen nach Bedarf zusammenstellen und personalisieren können. Neben dem Hosting von Messaging und E-Mail sowie diversen E- Commerce- Modulen wird beispielsweise auch ein Customer- Relationship- Management- Service angeboten.

Die Firma Vertigo<sup>73</sup> hat in Zusammenarbeit mit DataReturn<sup>74</sup> und Microsoft eine fiktive Einkaufsseite mit dem Namen IBuySpy<sup>75</sup> erstellt. Der komplette Webshop wurde in ASP.NET entwickelt und ist aktuell der Visual Studio.NET Beta angepasst. Er soll deutlich

<sup>70</sup> vgl. [MSN 1] <http://explorer.msn.com/home.htm>

<sup>71</sup> entnommen aus <http://explorer.msn.com/features.htm#>

<sup>72</sup> vgl. [bCentral 2000] <http://bcentral.com>

<sup>73</sup> vgl. <http://www.vertigosoftware.com/default.htm>

<sup>74</sup> vgl. <http://www.datareturn.com/default.htm>

<sup>75</sup> vgl. [ibuyspy 2000] <http://www.ibuyspy.com>

machen, wie sich mittels des .NET Framework und ASP.NET ein typischer Webshop mit Produktverwaltung, Warenkorb, Benutzerverwaltung und Produktkommentierung auf einfache Art und Weise entwickeln lässt.

Ebenfalls als .NET- Service ist ein Unterhaltungs- und Bildungsangebot geplant.

Informationen darüber sind bislang jedoch noch Mangelware.

### **3.5. AUSBLICK**

In den vorangegangenen Kapiteln wurde ausführlich auf die neue Produktstrategie von Microsoft eingegangen. Dabei wurde deutlich, dass .NET nicht nur ein neues Produkt ist, sondern auch eine vollständige Neuausrichtung der Firmenstrategie des größten Softwareherstellers der Welt mit sich bringt

Hält man sich diese Tatsache vor Augen, so wird einem die ganze Tragweite dieses Konzeptes bewusst. Nach wie vor haben andere Betriebssysteme und Office Applikationen nicht annähernd die Verbreitung wie sie die Microsoft Produktreihe hat. Wollen die Anwender nicht auf eine vollständig neues Produkt eines anderen Herstellers umschwenken, so werden sie nicht umhinkommen, der Microsoftschen Produktstrategie zu folgen.

Diese negativ angehauchte Sichtweise ist jedoch bei .NET keinesfalls angebracht, denn im Gegensatz zu bisherigen Produktentwicklungen bei Microsoft basieren diese vollständig auf offenen, weltweit anerkannte und zertifizierten Internetstandards. Daher hat Microsoft .NET das Potential auch über die Windows Plattform hinaus eine herausragende Position einzunehmen und über Systemgrenzen hinweg Brücken für Anwendungen und Dienste zu schlagen, was bisher in diesem Umfang nicht möglich ist.

Die Entwicklung plattformunabhängiger Anwendung wird erheblich vereinfacht, da die komfortablen Entwicklungsumgebungen Code für .NET und nicht für ein Betriebssystem erzeugen. Ebenso wird durch die Möglichkeit der viel tiefer als bisher gehenden Modularisierung die Entwicklung von komponentenbasierten Systemen erleichtert und der Entwicklungsaufwand erheblich reduziert. Die Zeit in der ein Softwareentwickler alle Funktionen selbst programmieren musste dürfte daher endgültig zu Ende gehen. In Zukunft wird die Devise sein, die Eigen- und Fremdentwicklungen zu noch besseren Anwendungen zu kombinieren.

Trotz all dieser Möglichkeiten sollte man sich jedoch auch bewusst sein, dass die Zeit von .NET noch lange nicht angefangen hat. Lediglich das Konzept ist bisher entwickelt, von den benötigten Entwicklungsumgebungen sind bisher gerade einmal Beta-Versionen erhältlich. Es wird daher noch mindestens 2 –3 Jahre dauern bis in großem Stil Anwendungen auf .NET Basis für den Endanwender zur Verfügung stehen und dieser auch konkreten Nutzern aus diesem Konzept ziehen kann.



## Quellenverzeichnis

- [bCentral 2000] <http://www.bcentral.com>
- [DNJ 2000-1] Developer Network Journal, Issue 21  
Matt Publishing, Bristol, UK
- [DNJ 2000-2] Developer Network Journal, Issue 20  
Matt Publishing, Bristol, UK
- [Fortune 2000] [http://cgi.fortune.com/cgi-bin/fortune/fortune500/f500rank.cgi?HOWMANY=100&industry=189&sales=6&profits=6&assets=0&common\\_equity=0&market\\_value=0&eps=0&tri=0&employees=0](http://cgi.fortune.com/cgi-bin/fortune/fortune500/f500rank.cgi?HOWMANY=100&industry=189&sales=6&profits=6&assets=0&common_equity=0&market_value=0&eps=0&tri=0&employees=0)
- [Gates 2000] <http://www.microsoft.com/billgates/bio.htm>
- [Heise 2000] <http://www.heise.de/newsticker/data/jk-28.11.00-001/>
- [Ibuyspy 2000] <http://www.ibuyspy.com>
- [Kahlisch 2000] <http://elvis.inf.tu-dresden.de/sgml/litkon94.html#a21>
- [Kruglinski 1998] Inside Visual C++ 6.0, Microsoft Press, Unterschleißheim
- [Microsoft 2000-1] <http://www.microsoft.com/presspass/features/2000/jul00/07-11.NETframework.asp>
- [Microsoft 2000-2] <http://www.microsoft.com/germany/asp/>
- [Microsoft 2000-3] <http://msdn.microsoft.com/vstudio/nextgen/technology/csharpintro.asp>  
Stand 2000-11-29
- [Microsoft 2000-4] C# Language Specification  
<http://msdn.microsoft.com/vstudio/nextgen/technology/clangspec.doc>  
Stand: 2000-11-02
- [Microsoft 2000-5] <http://www.microsoft.com/germany/dna/>  
Stand: 2000-10-13
- [Microsoft 2000-6] <http://www.passport.com/Consumer/default.asp?PPId=1031>
- [Microsoft 2000-7] <http://www.hotmail.com>

- [MSDN 2000] <http://msdn.microsoft.com/theshow/Episode008/Transcripttext.asp>
- [MSDN Library 2000-1] [http://msdn.microsoft.com/library/techart/PDC\\_eiffel.htm](http://msdn.microsoft.com/library/techart/PDC_eiffel.htm)
- [MSDN Library 2000-2] [http://msdn.microsoft.com/library/techart/PDC\\_cobol.htm](http://msdn.microsoft.com/library/techart/PDC_cobol.htm)
- [MSDN Library 2000-3] [http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/winbase/dll\\_53ji.htm](http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/winbase/dll_53ji.htm)  
Stand: 2000-08-14
- [MSDN Library 2000-4] <http://msdn.microsoft.com/library/techart/dlldanger1.htm>  
Stand: Januar 2000
- [MSDN Library 2000-5] <http://msdn.microsoft.com/library/backgrnd/html/cis.htm>  
Stand: 2000-04-13
- [MSDN Library 2000-6] <http://msdn.microsoft.com/library/dotnet/cpguide/cpconthenetframeworkclasslibrary.htm>
- [MSDN Library 2000-7] <http://msdn.microsoft.com/library/techart/Hawksoap.htm>
- [MSDN Library 2000-8] <http://msdn.microsoft.com/library/techart/vb6tovbdotnet.htm>
- [MSDN Library 2000-9] <http://msdn.microsoft.com/library/techart/InteropdotNET.htm>
- [MSDN Magazin 2000] <http://207.46.130.166/msdnmag/issues/0900/WebPlatform/WebPlatform.asp>
- [MSDN News 2000] Volume 9 - Number 5  
MSDN Information Centre, Dublin
- [O'Reilly 2000-1] Microsoft .NET vs. J2EE.  
[http://java.oreilly.com/news/farley\\_0800.html](http://java.oreilly.com/news/farley_0800.html)
- [O'Reilly 2000-2] Deep Inside C#  
An Interview with Microsoft Chief Architect Anders Hejlsberg  
[http://windows.oreilly.com/news/hejlsberg\\_0800.html](http://windows.oreilly.com/news/hejlsberg_0800.html)
- [Richter 2000] <http://msdn.microsoft.com/msdnmag/issues/1100/GCI/GCI.asp>
- [SDMagazine 2000] <http://www.sdmagazine.com/articles/2000/0011/0011/00111.htm?topic=uml>
- [Sun 1999] <http://java.sun.com/docs/books/vmspec/2nd-edition/html/Concepts.doc.html>
- [tecchannel 2000] [www.tecchannel.de/news/20000622/thema20000622-](http://www.tecchannel.de/news/20000622/thema20000622-)

- 
- 1791.html
- [VSJ 2000-1] Visual System Journal  
<http://www.vsj.co.uk/archive/oct2000/outlook1-1000.asp>  
Stand: Oktober 2000
- [VSJ 2000-2] Visual System Journal  
<http://www.vsj.co.uk/archive/oct2000/outlook2-1000.asp>  
Stand: Oktober 2000
- [Yi 2000] [http://www.fh-wedel.de/~si/seminare/ws98/Ausarbeitung/2.Yi/abgabe\\_1.htm#wasXML](http://www.fh-wedel.de/~si/seminare/ws98/Ausarbeitung/2.Yi/abgabe_1.htm#wasXML)

## Ehrenwörtliche Erklärung

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema

### **Microsoft .NET - Einblick und Überblick**

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

\_\_\_\_\_  
Ort

\_\_\_\_\_  
Datum

\_\_\_\_\_  
Unterschrift

