

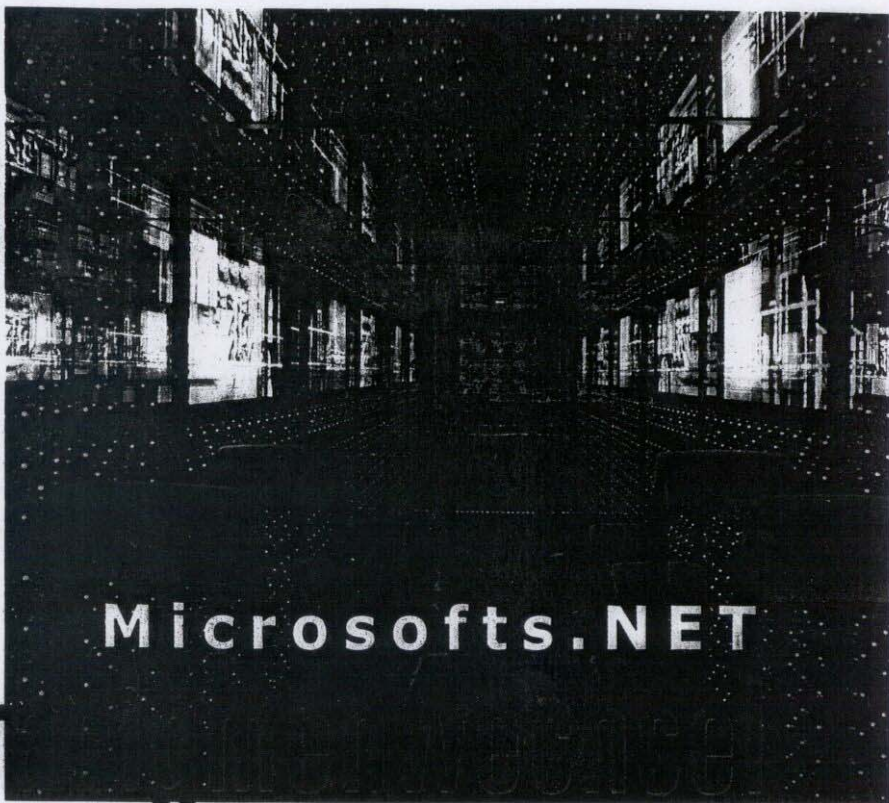
Mit ihrer .Net-Plattform, welche als eine „neue Generation des Computing“ vorgestellt wurde, bricht Microsoft in vielen Bereichen bewusst mit den traditionellen Herangehensweisen bei der Entwicklung von Lösungen.

Text:

Gerhard Rainer und Thomas Mach

Par

Alle Fotos Siemens



Microsofts.NET

Die Informationstechnologie steht vor einem Umbruch. Computer bewerkstelligen mittlerweile Aufgaben, die sich noch vor 20 Jahren niemand vorstellen konnte. Das Zitat des ehemaligen IBM-Vorstands Thomas Watson, der 1943 meinte, er sehe „einen weltweiten Markt für vielleicht fünf Computer“ regt heute nur mehr zum Lächeln an. Die heutigen Rechner werden immer kleiner, schneller und intelligenter. Personal Digital Assistants (PDAs), Laptops und Palmtops beginnen die klassischen „grauen Kästen“ abzulösen, wie selbige einst die Zentralrechner und ihre Terminals verdrängten. Dazu kommt, dass nunmehr jedes Gerät mit jedem anderen kommunizieren und Daten sowie Informationen austauschen will und soll. All dies setzt voraus, dass sich die Entwicklergemeinde auf die neuen Möglichkeiten und Philosophien der Automatisierung einstellt. Eine Basis dafür stellt Microsoft mit der .Net-Plattform zur Verfügung.

Framework aus zwei Komponenten

Das .NET-Framework besteht aus zwei grundlegenden Komponenten: einerseits aus der Common Language Runtime (CLR), welches die Basis darstellt, damit Programme unabhängig von der Programmiersprache zusammenarbeiten können, und andererseits aus der .NET-Klassenbibliothek. Mittels CLR kann eine Applikation aus Teilen von C++, C# (sprich: C-sharp) und Basic erstellt werden, ohne dass die

einzelnen Komponenten vorher in Binärformate, wie beispielsweise DLLs (Dynamic Link Library) oder COM-(Component Object Model-) Container überführt werden müssen. Zudem bietet CLR viele Vorteile, die sich für robusten Industriecode von Vorteil erweisen, etwa automatische Ressource-Freigabe (Garbage Collection), strikte Typen- und Codekontrolle, verschiedene Sicherheitskonzepte und vieles mehr. All diese Dinge sind weder neu, noch bedeutet dies, dass ohne sie nicht auch schon stabile Anwendungen implementiert wurden. Jedoch unterstützt die Umgebung den Entwickler, so dass er sich effektiver auf die Lösung des Problems konzentrieren kann und sich gewisse „Unterlassungssünden“ beziehungsweise „Fehler“ leisten kann. So sind zum Beispiel Memory Leaks nun nicht mehr möglich. Die Klassenbibliothek ihrerseits sorgt für eine durchgängig konsistente und objektorientierte Umgebung, welche gemeinsam alle verfügbaren Programmiersprachen nutzt. Dadurch kann sich ein C#-Programmierer ohne weiteres in den Quellen seines Basic-Kollegen zurechtfinden. Die Neuentwicklung der Bibliothek ermöglichte ein „sauberes Design“, da nicht mehr der Kompatibilität mit all ihren Seiteneffekten Tribut gezollt werden muss. Die Klassenbibliothek wirkt dadurch sehr aufgeräumt und übersichtlich, ohne jedoch an Funktionalität zu leiden. Sie unterstützt nun die Entwicklung der vielfältigsten Arten von Applikationen, ausgehend von klassischen Services, Kommandozeilen- und Windows-Programmen bis hin zu den neuesten und relativ unbekanntesten Technologien wie Webservices und ASP.NET-WebForms.

Erste Kunden-Implementierungen

Einer der ersten Kunden von Microsofts .Net-Framework war die Hausbank München. Das enorme Wachstum ihres Mietkautionsservices mit vergleichsweise geringen Margen pro Kautionskonto veranlasste das Unternehmen, mit MIKAU online ein Entwicklungsprojekt aufzulegen, das den bisherigen Prozess weitgehend automatisiert. Mittels der Mission Critical Anwendung können die Bankkunden seither selbst Mietkautionskonten anlegen und verwalten. Die