

TRENDS BEI MICROSOFT: .NET AUF DEM VORMARSCH

Im Jahr 2002 hat Microsoft sein Programmiermodell „Managed Code“ im Markt eingeführt. Mit den kommenden Versionen .NET 2.0 und 3.0 beweist Microsoft nun, dass man auch für die interne Entwicklung auf dieses Modell setzt. In den nächsten Versionen von Betriebssystemen, Servern- und Desktop-Anwendungen wird .NET eine zentrale Rolle spielen.

Ursprünglich ist Microsoft mit einer großen Marketing-Blase gestartet: Bei der Einführung des „.NET Framework 1.0“ wurde das Etikett „.NET“ auch auf das Windows-Betriebssystem, die BackOffice-Server und sogar Microsoft Office geklebt – obwohl keines der Produkte auf *Managed Code* basierte. Entsprechend groß war die Verwirrung auf dem Markt. Inzwischen hat Microsoft diese Marketing-Fauxpas eingesehen und die .NET-Logos wieder heruntergerissen. Der ursprünglich angekündigte „Windows .NET Server“ ist als „Windows Server 2003“ erschienen und wurde damit der Tatsache gerecht, dass er außer dem .NET Framework selbst nur einen einzigen in *Managed Code* geschriebenen Dienst (UDDI¹⁾) anbot. Sogar die kommende Version der Entwicklungsumgebung Visual Studio wird kein .NET mehr im Namen tragen, obwohl sie fast ausschließlich der .NET-Entwicklung dient. Die integrierte Entwicklungsumgebung selbst ist aber zum größten Teil in *Unmanaged C++* geschrieben.

Der Begriff .NET hat sich also auf das .NET Framework fokussiert und Microsoft konzentriert sich nun darauf, den .NET-Begriff zu leben, statt zu ihn verkleben. Mit .NET 2.0 (Codename „Whidbey“), das Ende März 2005 in die Beta 2-Phase eintritt, wird Microsoft die Klassenbibliothek ungefähr verdoppeln. Eines der Ziele von .NET 2.0 ist es, die Programmierbarkeit des Betriebssystems durch *Managed Code* zu erhöhen. Die dritte Version von .NET wird

Teil des nächsten Betriebssystems („Windows 2006“, Codename „Longhorn“) sein. „WinFX“ nennt Microsoft die in *Managed Code* geschriebene Klassenbibliothek für das Betriebssystem, die das alte, auf den C-Bibliotheken basierende „Win32“ ablösen soll. Zentrale Elemente von WinFX werden

- das XML-basierte Benutzungsschnittstellen-Framework „Avalon“,
- die Kommunikationsplattform „Indigo“ und
- das datenbankbasierte Dateisystem „WinFS“

sein, wobei Microsoft letzteres schon wieder auf das Jahr 2007 verschoben hat. Die permanente Verschiebung von Projekten gehört auch zu den Entwicklungstrends bei Microsoft.

Mit Avalon übernimmt Microsoft die aus dem Web-Entwicklungsframework „ASP.NET“ bekannte Trennung von Benutzungsschnittstellen-Layout durch XML und Benutzungsschnittstellen-Steuerung durch Code auch für die Entwicklung von Windows-Desktop-Anwendungen. Bereits mit ASP.NET 2.0 liefert Microsoft eine Vielzahl von Softwarekomponenten aus, deren vordefiniertes Verhalten durch XML-Tags gesteuert werden kann. Eigener Programmcode zum punktuellen Überschreiben des Verhaltens ist möglich, soll aber die Ausnahme bleiben. Mit Avalon und ASP.NET 3.0 wird Microsoft den Trend „XML statt Programmcode“ fortsetzen.

Das Kommunikationsframework Indigo soll ein einheitliches Programmiermodell

für Fernaufrufe und Applikationsserver bieten. Microsoft will hier bestehende Ansätze und Produkte wie „ASMX“ (XML-Web-Services), „.NET Enterprise Services“, „.NET Remoting“, „Message Queuing“ (MSMQ) und auch das alte „Distributed COM“ vereinen, sodass der Entwickler ohne großen Aufwand den Wechsel zwischen verschiedenen Optionen vollziehen kann.

Schuldig bleibt Microsoft den Entwicklern aber weiterhin einen in *Managed Code* geschriebenen Applikationsserver und eine Lösung für Objektpersistenz. Der ursprünglich in der Alpha-Version von .NET 2.0 enthaltene objekt-relationale Mapper „ObjectSpaces“ ist ein Verschiebungsoffer, das frühestens im Jahr 2007 zusammen mit dem WinFS-Dateisystem erscheinen wird. Während die Java-Welt gerade über die Zusammenführung der bestehenden Standards *Enterprise Java Beans* und *Java Data Objects* diskutiert, bleibt in der Microsoft-Welt bislang nur eine unüberschaubare Vielfalt von Drittanbieter-Produkten.

Unzählige Male verschoben wurde auch die nächste Version des „Microsoft SQL Server“. Aufgrund der Tatsache, dass .NET tief in den Kern der Datenbank integriert sein wird und die Erstellung von *Stored Procedures*, *Triggers*, *UDTs* und Funktionen in .NET-Sprachen erlauben wird, wartet das Produkt auf die Fertigstellung des .NET Framework 2.0. Sogar das bislang sehr spartanische Windows-Kommandozeilenfenster wird in von *Managed Code* erobert werden. Die Microsoft-Shell (Codename „Monad“)

¹⁾ Universal Description, Discovery and Integration

erlaubt die Kommunikation so genannter *Commandlets* durch eine objektorientierte Pipeline, die beliebige .NET-Objekte statt nur Zeichenketten befördern kann. *Commandlets* sind *Managed-Code*-Klassen, die durch *Reflection* den Inhalt der Pipeline auswerten und modifizieren können.

Außerdem scheint Microsoft verstanden zu haben, dass zu einer modernen Enterprise-Entwicklungsplattform mehr gehört als nur Klassen und Editoren mit *IntelliSense*. Bei den Entwicklungswerkzeugen will Microsoft zukünftig den gesamten Software-Lebenszyklus adressieren. Die Entwicklungsumgebung „Visual Studio“ fokussierte bisher nur den Entwicklungsprozess im engeren Sinne. Für den Modellierungsprozess gab es mit „Microsoft Visio“ und für die Versionsverwaltung mit „Microsoft SourceSafe“ zwei Zusatzprodukte, die aber jeweils deutliche Schwächen aufwiesen. Andere Bereiche des Software-Lebenszyklus waren entweder gar nicht oder nur mit einzelnen, nicht in die Entwicklungsumgebung integrierten Werkzeugen (z. B. „Webstress“, „CLR Profiler“) abgedeckt. Für Visual Studio 2005 wird Microsoft ein in die Entwicklungsumgebung integrierbares Toolset unter dem Namen „Visual Studio Team System“ (VSTS) anbieten. VSTS wird die Bereiche Modellierung, Projektmanagement, Codeanalyse, Profiling, Test, Quellcodeverwaltung und Aufgabenverfolgung umfassen. VSTS wird voraussichtlich erst im Jahr 2006 erscheinen.

Managed Code breitet sich in Microsoft-Produkten aus. Ohne dieses deutliche Signal

würde Microsoft sein .NET nicht glaubwürdig verkaufen können. Von der Vision, eines Tages Windows fast komplett in *Managed Code* neu zu implementieren, ist Microsoft aber immer noch Lichtjahre entfernt. ■

Dr. Holger Schwichtenberg
(hs@IT-Visions.de)

Literatur & Links

[Cha05] D. Chappel, *Introducing Indigo: An Early Look*, Februar 2005, siehe: msdn.microsoft.com/longhorn/default.aspx?pull=/library/en-us/dnlong/html/introindigov1-0.asp

[Mic-a] Microsoft, *.NET 2.0 (Whidbey) Developer Center*, siehe: lab.msdn.microsoft.com/vs2005/

[Mic-b] Microsoft, *Longhorn Developer Center (erste Teile von .NET 3.0)*, siehe: msdn.microsoft.com/longhorn/

[Mic-c] Microsoft, *Informationen zu Avalon*, siehe: msdn.microsoft.com/longhorn/understanding/pillars/avalon/default.aspx

[Mic-d] Microsoft, *Produkt-Webseite zu Microsoft SQL Server 2005*, siehe: www.microsoft.com/sql/2005/default.asp

[Mic-e] Microsoft, *Informationen zum Visual Studio Team System (VSTS)*, siehe: lab.msdn.microsoft.com/teamsystem/

[Sch] H. Schwichtenberg, *Informationen zur Microsoft Shell*, siehe: www.dotnetframework.de/scripting/msh

[Sch05] H. Schwichtenberg, *.NET 2.0 Crashkurs*, Microsoft Press, April 2005

„Aktuelle Trends der Softwaretechnik“ bilden den Schwerpunkt dieser Ausgabe von OBJEKTSPEKTRUM. Peter Hruschka und Gernot Starke haben in Zusammenarbeit mit Frances Paulisch eine Auswahl besonders praxisrelevanter Themen getroffen und namhafte Experten als Autoren gewinnen können.



Jens Coldewey, freier Berater mit dem Schwerpunkt agile Entwicklung, Vorstandsmitglied der Agile Alliance.



Thilo Frotscher, freiberuflicher Systemarchitekt, Trainer und Autor mit den Schwerpunkten J2EE, Web-Services und Patterns.



Jürgen Hahn, Trainer und Berater für Systemarchitekturen und OO-Vorgehensweisen bei der Sophist Group, Autor von „UML 2 glasklar“.



Dr. Peter Hruschka, Trainer, Berater und Coach für Software- und Systemarchitekturen und IT-Strategie.



Dr. Thorsten Janning, Mitbegründer der Keron AG mit den Schwerpunkten komplexe IT-Architekturen und marktorientierte Organisation von IT-Dienstleistern.



Janos Koppány, Mitgründer und Geschäftsführer der Firma Intland, dort für kollaborative Softwareentwicklungsplattformen zuständig.



Dr. Arne Koschel, Senior Software Architect, Middleware-Experte, Application Integration Team Lead, Dozent und Autor.



Johannes Link, Projektleiter bei der andrena objects ag in Karlsruhe und Autor des Buchs „Softwaretests mit JUnit“.



Mustafa Öztürk, Berater bei der Keron AG, mit dem Schwerpunkt Innovations- und Technologiemanagement in der IT-Strategie.



Dr. Stefan Queins, Trainer und Berater für Systemarchitekturen und OO-Vorgehensweisen bei der Sophist Group, Autor von „UML 2 glasklar“.



Martin Rösch, Spezialist für innovative Softwaretechnik, Berater, Trainer und Coach.



Chris Rupp, Geschäftsführerin der Sophist Group, mit den Schwerpunkten Requirements-Engineering und Objekt-orientierung.



Dr. Holger Schwichtenberg, selbständiger Softwarearchitekt, Dozent und Fachjournalist mit dem Schwerpunkt .NET.



Michael Stal, Senior Principal Engineer bei Siemens Corporate Technology, Chefredakteur JavaSPEKTRUM und Koautor der Buchreihe POSA.



Dr. Gernot Starke, Trainer, Berater und Coach für Software- und Systemarchitekturen und IT-Strategie.



Stefan Tilkov, Geschäftsführer der innoQ Deutschland GmbH, einem auf SOA und MDA spezialisierten Beratungshaus.



Markus Völter, freiberuflicher Berater und Coach mit den Schwerpunkten Softwarearchitektur, modellgetriebene Entwicklung und Komponenten-Middleware.



Tim Weikiens, Trainer und Berater bei der oose.de GmbH, Autor von Buch- und Zeitschriftenartikeln, Mitglied der Arbeitsgruppen zu SysML und UML 2.1



Frank Westphal, freier Softwareentwicklungcoach, Buchautor, Podcaster der agilen Audio-kolumne „Tonabnehmer“.



Ralf Westphal, freier Autor, Berater, Trainer und Sprecher auf Entwicklerveranstaltungen, einer von 130 Microsoft Regional Directors.