

ERP-Architekten bauen auf .NET oder J2EE

Von CW-Redakteur Volker Schinkel*.

MÜNCHEN (COMPUTERWOCHE) - *Volker Schinkel ist Berater und Vorstand der INFOSOFT AG in Hamburg, die Firmen bei der ERP-Auswahl berät.*

Anbieter von ERP-Software beschäftigen sich intensiv damit, ihre Systeme zu serviceorientierten Architekturen weitzuentwickeln. Zwei Wege führen zum Ziel: Microsofts .NET oder die herstellerunabhängige Plattform J2EE. Die unlängst besiegelte Kooperation zwischen Sun und Microsoft lässt auf bessere interoperabilität von .NET und Java hoffen.

Die lang erwartete Konzentration im ERP-Markt hat mit Macht eingesetzt. Käufe und Übernahmen von namhaften Herstellern spiegeln diese Marktkonzentration wider und stellen Anbieter von ERP-Lösungen vor zwei große Aufgaben: Auf der einen Seite müssen durch Akquisitionen erworbene funktionale Komponenten in bestehende ERP-Systeme integriert werden, zum anderen gilt es, den "gekauften" Kunden schlüssige Migrations- oder Koexistenzszenarien anzubieten. Dies verstärkt die ohnehin präesente Kundenforderung nach Integrationsfähigkeit und Web-Services.

Hier lesen Sie ...

- welche Vor- und Nachteile die Infrastrukturen J2EE und .NET in Bezug auf Business-Software bieten,
- welchen Weg verschiedene Hersteller beschreiten, um ihre Client-Server-Lösungen weiter zu entwickeln,
- warum Anbieter von Windows-basierenden ERP-Systemen nicht umhin kommen, .NET-Lösungen zu entwickeln.

Technik harmonisieren

Es tut sich was im ERP-Markt. SAP [1] wirft die Angel nach den Kunden von DCW Software [2] aus, die J.D.-Edwards-Anwender finden sich unversehens in den Armen von Peoplesoft [3] wieder, und SSA Global [4] kündigt ebenso wie Agilisys lautstark weitere Käufe an. Diese Bewegungen verstärken den Druck auf ERP-Anbieter in der Frage nach

integrationsfähigen Systemarchitekturen. So müssen sich die Hersteller heute die grundsätzliche Frage stellen, wo sie ihre Zukunft sehen: Entweder entscheiden sie sich für die Java 2 Enterprise Edition (J2EE) oder für Microsofts [5] .NET.

Der ERP-Anbieter SAP beispielsweise hat sich der Java 2 Enterprise Edition (J2EE) verschrieben und seine Infrastrukturplattform "Netweaver" daran angepasst. SAP schuf so eine Architektur, in der sowohl die in der eigenen Sprache Abap implementierten Funktionen als auch neue, in Java entwickelte Komponenten laufen können. Der Web Application Server ersetzt SAPs bisherige, rein für Abap konzipierte Ablaufumgebung "SAP Basis". Mit Netweaver verlagern sich die Aufgaben der Integration, der Sicherheit und der Middleware-Funktionen in die zentrale Infrastrukturplattform.



Anbieter von Windows-Software tendieren zu .NET, J2EE verspricht Plattformunabhängigkeit. (Collage: ww)

J2EE als Definition von standardisierten Diensten dient dabei als Basis, um neue Komponenten plattformunabhängig zu entwickeln. Was sich einfach anhört, erweist sich komplexer und von den Kunden schwerer begreifbar als zunächst angenommen. In der Praxis verwirren die vielen Abhängigkeiten zwischen den unterschiedlichen Anwendungsversionen

und der Ablaufumgebung die verantwortlichen Planer und Systemarchitekten auf Kundenseite. Hier zeigt sich, dass bei der Umsetzung der J2EE-Standards in konkrete Produkte spätestens dann Zugeständnisse gemacht werden müssen, wenn es gilt, Vorgängerversionen der eigenen Produkte zu integrieren.

Java und J2EE gilt vielen Herstellern als der einzige Weg zu einer mehrschichtigen, serviceorientierten Architektur. Die Java-Plattform liefert eine Vielzahl von standardisierten Diensten für Datenbankzugriffe, Anbindung von Legacy-Systemen, Sicherheit, Transaktions-Management, Mail-Versand, Messaging und XML-Verarbeitung. Die Anwendungslogik wird dabei in standardisierte, modulare Komponenten eingebettet. Die J2EE-Ablaufumgebung soll für hohe Verfügbarkeit, Sicherheit und Skalierbarkeit sorgen.

Als Alternative zu J2EE führte Microsoft die .NET-Plattform ein, eine im Gegensatz zum hersteller- beziehungsweise plattformübergreifenden Java-System proprietäre und nur an Windows angepasste Architektur. Das ".NET Framework" besteht aus herstellereigenen Komponenten wie ".NET Framework Classes" (Sammlung von Funktionen zur Anwendungsentwicklung), "ADO.NET Data Sets" (sie umfassen Active X Data Objects, in denen zum Beispiel Datenbankinhalte gespeichert werden) und dem ".NET Remoting Protocol" (ein Mechanismus für den Client-Zugriff auf verteilte Server). Die technische Basis für .NET bilden der "Windows Server 2003" und die Entwicklungsumgebung "Visual Studio .NET".

Kein leichter Weg zu .NET

Microsoft forciert mit dem "Empower"-Programm massiv den Einstieg mittelständischer Softwarehäuser in die .NET-Architektur. Eine Reihe deutscher Softwareanbieter wie AP Automation + Productivity, Myfactory und Step Ahead haben bereits ERP-Systeme auf der Grundlage von .NET entwickelt. Aber: Migrationswillige Softwareanbieter werden bei der Berücksichtigung ihrer bestehenden Systeme auch durch .NET vor Probleme gestellt, denn es gilt, die bisher auf dem "Component Object Model" (COM) basierenden Programme an die .NET-Architektur anzupassen, wodurch erhebliche Kosten entstehen. Eine weitere Einschränkung betrifft die Integration: Microsofts "Host Integration Server" dient unter .NET hauptsächlich dazu, auf Mainframes und den Biztalk Server zuzugreifen. J2EE dagegen ermöglicht mit der "Java Connector Architecture" auch die Kommunikation mit existierender Business-Software von Herstellern wie SAP oder Siebel [6] über spezielle Ressourcenadapter.

Herstellern von Windows-basierender Software bleibt jedoch keine Wahl, denn wer auch in Zukunft kostengünstig unter Windows entwickeln will, dem bietet sich kaum eine Alternative zu .NET. Der Grund: Die Kernbestandteile des Windows-XP-Nachfolgers Longhorn werden auf .NET implementiert. Zudem laufen die auf "Windows Distributed Internet Applications Architecture" (Windows DNA) aufsetzenden Produktlinien von Entwicklungs-Tools (zum Beispiel Visual Basic 6.0) 2005 aus. Die neue Umgebung Visual Studio .NET ermöglicht dann Entwicklungen in C#, Visual Basic .NET und C++, wobei die Sprachen durch die Zwischensprache Microsoft Intermediate Language (MSIL) interoperabel gehalten werden.

ERP von und für den Mittelstand

Als Kunden visieren AP, Myfactory und Step Ahead vor allem kleinere und mittlere Unternehmen (KMUs) an. Firmen dieser umworbenen Zielgruppe treffen heute ihre Investitionsentscheidungen aber nur zu einem sehr geringen Teil unter technischen Gesichtspunkten. Ihnen kommt es vielmehr auf die Abdeckung funktionaler Anforderungen durch die Software und die Branchenkompetenz ihrer IT-Partner an. Bisherige Versuche Microsofts, .NET als Basis für ERP-Systeme zu vermarkten, verpufften weitgehend. So hat der an die AP AG verliehene ".NET-Award" dem Anbieter nach eigener Aussage bisher keinen Vorteil beim Endkunden eingebracht. Microsoft möchte Mittelstandskunden von den technologischen und wirtschaftlichen Vorteilen von .NET überzeugen, um so ihr Investitionsverhalten zu beeinflussen.

Zwar bietet .NET den kleineren ERP-Anbietern eine Plattform, um funktionale Integration mit zusätzlichen Services zu verwirklichen. Die Frage aber, ob Microsoft auf Dauer eigenständigen Wettbewerbern im ERP-Markt die nötige Luft zum Atmen lassen wird, lässt sich nicht mit Gewissheit bejahen. Analogien zur Situation im Markt für Office-Software und zum Diktat durch die Marktführerschaft von Windows drängen sich auf.

Für welchen Weg man sich auch entscheidet, fest steht, dass sich die ERP-Anbieter von klassischen Client-Server-Konzepten verabschieden und serviceorientierten Architekturen zuwenden müssen. Bei Client-Server-Lösungen steuert die jeweilige Applikation die Kommunikation beziehungsweise die Transaktionsverarbeitung. In serviceorientierten Systemen übernimmt ein spezieller Server zwischen Client und Backend diese Aufgabe. (fn)

Links in diesem Artikel:

<http://www.softwareberatung.de/>

- [1] <http://www.sap.com/>
- [2] <http://www.dcw-software.com/>
- [3] <http://www.peoplesoft.com/>
- [4] <http://www.baan.com/>
- [5] <http://www.microsoft.com/>
- [6] <http://www.siebel.com/>