



von Alexander Büch,
Chief Technology Officer
EZE Technologies Ltd.

Zukunftsperspektiven für Integrationsprojekte

Informatik ist ein schnelllebiges Geschäft. In den vergangenen Jahren haben mehr als einmal fundamentale Veränderungen stattgefunden. Weil bestehende Systeme jedoch nur schwer zu ersetzen sind oder die Ablösung exorbitante Kosten verursacht, bleiben viele Systeme und Komponenten trotz der Neuerungen über Jahre und Jahrzehnte hinweg parallel zu neuen Systemen und Anwendungen bestehen.

Bei beinahe der Hälfte der mittleren und grossen Schweizer Unternehmen stehen noch IT-Anwendungen im Einsatz, die mehr als 20 Jahre alt sind und mehr als 70 Prozent betreiben Lösungen, die über 15 Jahre alt sind. Solches geht aus einer im März 2005 veröffentlichten Studie der Schweizer Unternehmensberatung Bosshard & Partner hervor. Diese überlieferten Lösungen (Legacy-Systeme) wurden zumeist in den altherwürdigen Programmiersprachen Cobol und PLI entwickelt. Neue Architekturansätze in der Informatik erfordern aber eine grundlegende Veränderung der Integration und Entwicklung. Dabei gilt es unter allen Umständen zu vermeiden, dass erneut proprietäre Komponenten verwendet werden. Hierbei bietet die im Folgenden beschriebene, durchgängig transparente, komplett Modell-basierte Vision die Chance, den aus proprietären Teilen bestehenden, teuren und kaum mehr wartbaren «Gesamtkunstwerken» zu entkommen.

Unternehmen sitzen auf Legacy-Anwendungen

Da überlieferte (Legacy-)Systeme in ihrer monolithischen Bauweise heutigen Anforderungen kaum mehr genügen, gewinnen Software-Sanierung und -Modernisierung bei mittelgrossen und grossen Unternehmungen hohe Aktualität. Laut Marktforschungsinstitut Forrester Research werden heute mehr als 40 Prozent der gesamten IT-Ausgaben für Integrationsaufgaben aufgewendet – Tendenz steigend. Nach Einschätzung anderer Branchenexperten ist dieser Wert

eher vorsichtig angesetzt und könnte sogar noch um einiges höher liegen.

Die effizienteste Lösung für das Problem wäre wahrscheinlich ein vollständiger Kahlschlag, gefolgt von einer grundlegenden Neuprogrammierung der gesamten Applikationslandschaft. Damit liessen sich zweifellos homogene, modulare Systeme aufbauen, die auf aktuellen und zukunfts-sicheren Technologiestandards und Entwicklungstools basierten.

Dass diese Vision letzten Endes eine Illusion ist, führte Andrew Watson,

SOFTWAREPROJEKTE IN DER KRISE

- 75% der Business Manager beklagen das Scheitern von Projekten oder die mangelhafte Zuverlässigkeit der IT (Meta Group)
- 90% aller IT-Projekte sind verspätet (Aberdeen Group)
- 50% aller IT-Projekte überschreiten das Budget (Gartner)
- 50% aller Applikationsprojekte verfehlen die Geschäftsziele (Gartner)
- 40% der IT-Projekte erreichen den angestrebten ROI nicht (Harvard Business School Press)

Vizepräsident und Technischer Leiter des Standardisierungsgremiums Object Management Group (OMG) an einem Pressegespräch im Vorfeld der OMG Information Days in Zürich aus. Laut Watson ist ein Neuanfang finanziell unter keinen Umständen zu bewältigen, da viel zu viele Mittel bereits in den Aufbau der bestehenden Infrastrukturen gesteckt wurden.

Viel zu viele Informatikprojekte scheitern kläglich

Es ist leider eine allgemein bekannte Tatsache, dass weit mehr als die Hälfte aller Informatikprojekte nicht erfolgreich abgeschlossen werden können oder vollständig scheitern. Oft werden die Unternehmungen nicht rechtzeitig oder überhaupt nie fertig, überschreiten das Budget gravierend oder verfehlen die vereinbarten Geschäftsziele klar.

Die Gründe für die Misserfolge sind in den meisten Fällen dem Faktor Mensch zuzuschreiben. Oft ergeben sich bereits bei der Zielvereinbarung zwischen Fach- und Entwicklungsabteilung Kommunikationsprobleme, die sich mangels geeigneter Mittel nur schwer kompensieren lassen.

Bei der Umsetzung sorgen nicht selten mangelnde Erfahrung im Umgang mit neuen Technologien und eine im permanenten Zeitdruck begründete

Nachlässigkeit dafür, dass sich der grösste Teil der Softwareentwicklung im Ausmerzen von Fehlern erschöpft und nur wenig Raum für innovative Entwicklung bleibt.

Der neue Lösungsansatz:

Integration und Automatisierung

Genau an dieser Stelle setzt das Konzept der Modell-basierten Softwareentwicklung (Model-Driven Architecture; MDA) an. Anstatt komplizierte Softwareanwendungen von Hand auszuprogrammieren (wie das bisher zumeist der Fall war), wird die Arbeit auf den Architekturteil verlagert. Aus dem erarbeiteten Architekturmodell wird der zur Umsetzung der Applikation nötige Programmcode automatisch generiert. Daraus ergibt sich eine weit höhere Produktivität des einzelnen Programmierers, gleichzeitig steigt die Qualität des Softwareproduktes erheblich.

Die Object Management Group (OMG) erkennt in der Modell-basierten Softwareentwicklung einen zentralen Faktor zukünftiger Software-Architekturen (siehe auch Interview nächste Seite). Nebst der bereits diskutierten Effizienzsteigerung lassen sich mit dem neuen Ansatz bereits bestehende Softwarekomponenten auch bedeutend öfter wiederverwenden. Dieser willkommene Nebeneffekt ist

vor allem auf eine konsequente Trennung der technologie-unabhängigen Geschäftslogik von systemabhängigen Komponenten zurückzuführen. Davon verspricht sich die OMG stark reduzierte Unterhaltskosten (siehe Abb. 1).

Im Gegensatz zur Modell-basierten Softwareentwicklung, die mit der Einführung neuer Technologie verbundenen ist und sicherlich noch in den Kinderschuhen steckt, handelt es sich beim zweiten wichtigen Trend, der Service-orientierten Architektur (Service Oriented Architecture, SOA), weniger um eine neue Technologie als vielmehr um eine radikale Umkehrung der Sichtweise, wie Software zukünftig organisiert werden soll. Mit Service-orientierten Ansätzen verabschieden sich Softwareentwickler definitiv von einer computerzentrierten Sicht, bei der Softwareentwicklung ein fundiertes Verständnis der Systemvorgänge voraussetzt. An ihre Stelle rückt eine fachliche, geschäftsorientierte Sichtweise.

Nach der strukturierten, der objekt-orientierten und der komponenten-basierten Methode findet hier in der Softwareentwicklung scheinbar ein Rückschritt statt in eine einfachere, strukturierte Art des Programmierens (Abb. 2). Da dieser jedoch auf einem weit höheren Abstraktionsniveau stattfindet, bedeutet dies, dass zukünftig ganze Funktionen erzeugt werden, etwa «Buch bestellen» (Kunde, Titel, Preis) oder «Prüfe Lieferfrist» (Titel, Lieferant, Menge). Da solche Services die zu ihrer Ausführung notwendige Systemtechnologie kapseln, wird der Alltag des Programmierers immer weniger von plattformabhängigen Arbeiten dominiert. Stattdessen steht die Integration einzelner Services zu komplexen, geschäftsoptimierenden Prozessen im Vordergrund.

Serviceorientierte Softwareentwicklung ist nicht in erster Linie eine neue technologische Herausforderung. So gibt es einzelne Unternehmen, etwa Schweizer Grossbanken, die ihre Informatikprozesse schon seit fast zehn Jahren in effizienten, klar fokussierten Services strukturieren. Serviceorientiertes Denken setzt jedoch einen funktionierenden Dialog zwischen der IT-Abteilung und der operativen Geschäftsleitung voraus.

Ist MDA die Alternative zu Offshoring?

Sicherlich, wenn man davon ausgeht, dass sich mit dem Einsatz wieder verwendbarer Modelle in der Softwareentwicklung und -integration erheblich mehr Kosten einsparen lassen, als wenn man diese Arbeit einfach in Länder mit tieferen Lohnkosten verlagert. Die Zahlen, die in verschiedenen Un-



Abb. 1: Modell-basierte Softwareentwicklung (MDA) als Grundlage für erhöhte Qualität und Wiederverwendbarkeit von Softwarekomponenten ist mit grossem Einsparungspotenzial verbunden.



Evolution der Softwareentwicklungsmethodik vom strukturierten zum serviceorientierten Programmieren. Mit jeder Evolutionsstufe erhöht sich der geschäftsrelevante Abstraktionsgrad.



Andrew Watson

«Die Regeln sind das Design»

Interview mit Andrew Watson, Vizepräsident der Object Management Group (OMG), dem Standardisierungsgremium der Modell-basierten Software-Architektur.

ICT kommunikation

Herr Watson, worin besteht die Besonderheit der Modell-basierten Software-Entwicklung?

Andrew Watson: Model-Driven Architecture (MDA) macht das Design eines Softwaresystem zum aktiven Bestandteil des Endproduktes, im Gegensatz zu herkömmlicher Softwareentwicklung, bei der das Design lediglich dokumentiert wird. Herkömmliche Ansätze behandeln Entwicklung als einen Prozess, in dem Entwickler ihr Fachwissen anwenden, um elegante Lösungen von oftmals unzureichend definierten Problemen zu erarbeiten. Das Ergebnis besteht in Source-Code. Beim MDA-Ansatz ist das Design selbst ein Produkt, das einen Satz formaler Abbildungen definiert.

Wie macht sich dieser Ansatz in der Praxis bemerkbar?

MDA basiert auf der Idee, dass für die Konstruktion eines Softwaresystems ein genaues Modell gebaut werden soll, aus dem dann die eigentlichen ausführbaren Softwarekomponenten durch Transformation generiert werden können. Der Rahmen für die Umwandlung wird gegenwärtig als QVT (Query, Views, and Transforms) spezifiziert. In MDA wird das Applikationsmodell verallgemeinernd als plattformunabhängiges Modell dargestellt. Das Ergebnis einer Umformung kann entweder ein plattformspezifisches Modell sein, oder Komponenten der Zielanwendung, beispielsweise Datenbankschemata oder ausführbarer Code, aber auch Test-Scripts und Konfigurationsfiles.

Welche Auswirkungen hat MDA auf die Arbeit der Entwickler?

Ausgebildete Analytiker und Programmierer werden ihren Beruf

ausüben wie bisher, jedoch mit zwei wesentlichen Unterschieden. Zum einen werden sie, anstelle von Java, C++ oder Smalltalk zu schreiben, ihre Abstraktionen in einer Sprache ausdrücken, die präzise Modelldefinitionen unterstützt, wie zum Beispiel UML (Unified Modelling Language). Zum anderen wird die Fähigkeit zur Abstraktion eine noch mehr gesuchte Qualifikation werden. Es wird wichtig sein, Geschäfts- und Arbeitsprozesse in einer implementierungsunabhängigen Form zu modellieren.

Welche weiteren Veränderungen bringt MDA mit sich?

Plattformunabhängige Modellkomponenten sind in hohem Masse wiederverwendbar. Das reduziert Entwicklungskosten und verringert die Wartungskosten, da automatische Modelltransformationen schnell und zuverlässig kleine Modelländerung mit weit reichenden Auswirkungen implementieren können.

Welche Schwierigkeiten ergeben sich bei der Einführung von MDA im Markt?

Die Angst vor dem Unbekannten ist ein mächtiger Bremsklotz bei der Verbreitung jeder neuen Technologie. Der Wechsel zu MDA bietet jedoch Systemdesignern, Programmierern, Testern und Personen, die für den Unterhalt zuständig sind, die Chance, ihre Fähigkeiten auf produktivere Art und Weise einzubringen und einige frustrierende Aspekte ihrer Arbeit zu eliminieren. Damit erhöht MDA die Produktivität von Softwareentwicklern – ein weiterer Vorteil, der den Erfolg dieses Ansatzes begründet. ■

Interview: Jürg Buob

tersuchungen zum Einparungspotenzial durch Offshoring genannt werden, liegen zwischen 15 und weniger glaubhaften 47 Prozent. Kann da MDA noch günstiger sein? Ja, bei konsequenter Umsetzung bestimmt.

Es sind nicht nur die direkten potenziellen Kosteneinsparungen in der Entwicklung, die für MDA sprechen, sondern vor allem die Vision, die hinter MDA steht. Der standardbasierte Ansatz für eine automatische Übersetzung der Businesslogik in funktionierende Anwendungen hat zur Folge, dass die Business-Dokumentation in wieder verwendbare und wandelbare Modelle umgesetzt wird und damit Betrieb, Unterhalt, Erweiterung und Integration mit anderen Lösungen immense Einsparungen bringen und für nie da gewesene Transparenz zwischen «Business-Owner» und IT-Abteilung sorgt.

Mit MDA ist definitiv eine deutlich kosteneffizientere und qualitativ höherwertige Software-Entwicklung möglich. Da gibt es schon zu denken, dass der Modell-basierte Ansatz in den Köpfen vieler CIOs noch nicht angekommen zu sein scheint. Braucht es noch mehr Hiobsbotschaften bevor ein Umdenken stattfindet, wie diejenige von Marktanalyst Forrester, der voraussagt, dass infolge Offshoring in den nächsten fünf Jahren ein Viertel aller IT-Jobs in Europa verloren gehen und bis im Jahr 2015 1,2 Mio IT-Arbeitsplätze aus Europa in andere Länder verlagert werden?

Zeit zum Handeln

Auch wenn Europa – besonders der deutschsprachige Raum – bei der Entwicklungen der Modell-basierten Softwareentwicklung vorne mit dabei ist, besteht wenig Anlass, sich auf den Lorbeeren auszuruhen. Die neueren technologischen Tendenzen im Umfeld der Softwareentwicklung bergen Gefahren, denen rechtzeitig begegnet werden muss, um in diesem Umfeld auch in Zukunft noch eine Rolle spielen zu können.

Beide Entwicklungen, SOA wie auch MDA, verstärken den Bedarf an konzeptionellem Denken, einem erhöhten Abstraktionsvermögen sowie methodischem Vorgehen. Es reicht nicht mehr, nur die technischen Aspekte der Softwareentwicklung zu verstehen, kommunikative Aspekte spielen eine ebenso bedeutende Rolle.

Es ist absehbar, dass die neueren Entwicklungen im Umfeld von MDA und SOA nicht über Nacht Einzug in den Informatikalltag halten werden. Bei der Umsetzung sollten sich Software-Entwickler eine für unsere Region typische Stärke zu Nutze machen: überlegen, erkennen, aber dann auch wirklich handeln. ■