

# VON COBOL ZU SOA

In vielen Unternehmen laufen die zentralen Anwendungen unter COBOL und niemand will es riskieren, funktionierende Applikationen abzulösen. Dieser Artikel versucht aufzuzeigen, dass solche Umgebungen dennoch zukunftsfähig sind: Produktive COBOL-Transaktionen lassen sich in Business-Services transformieren und in neuen SOA-Umgebungen einsetzen.

Die Wiederverwendbarkeit von erfolgreich im Einsatz befindlichen Softwaremodulen und -funktionen gehört zu den wichtigsten Zielen einer effektiven Softwareentwicklung. Kurz: Man sollte das Rad nicht immer wieder neu erfinden. Wer hier an Komponenten, objektorientierte Programmierung, Klassen und Methoden denkt, liegt genau richtig. Allerdings waren die produktspezifischen Implementierungen viel zu speziell angelegt, zu detailbesessen und zu technikfixiert. Der fachliche Bezug zu Geschäftsprozessen fehlte.

Ein wieder verwendbarer Business-Service dagegen muss allgemeiner angelegt sein. Typische Beispiele für Business-Services sind: „Ermittle aus allen bei dieser Bank geführten Konten die Summe der gesamten Barmittel, über die ein Kunde verfügen kann.“ „Überprüfe die Kreditwürdigkeit eines Kunden.“ „Führe eine Online-Bestellung aus.“

Der Unterschied zu früheren Komponentenarchitekturen: Bei *serviceorientierten Architekturen (SOAs)* geht es darum, über Methoden zu verfügen, um auf Business-Services zugreifen und sie in anderen Applikationszusammenhängen wie J2EE oder .NET wiederverwenden zu können.

Der Wert einer derartigen SOA ist vielfach dokumentiert. Die Marktforscher von Gartner schätzen, dass sich in Folge einer solchen Vorgehensweise die langfristigen

IT-Kosten um bis zu 20% verglichen mit traditionellen Verfahren senken lassen. Die Einsparungen können sogar noch größer werden, je mehr die Business-Services-Bibliothek wächst und der Grad der Wiederverwendbarkeit ansteigt.

## COBOL-Business-Services und SOA

Zu den wichtigsten Vorteilen wieder verwendbarer Business-Services zählen die Aspekte Schutz und weitere Nutzung vorhandener Investitionen. Eines der besten Beispiele dafür sind die weltweit mehr als 200 Milliarden Zeilen COBOL-Code, die in vielen Unternehmen die Kernsysteme ausmachen.

Unbestritten ist, dass sich einige Programmiersprachen für bestimmte Aufgaben besser eignen als andere. Das Metier von Java und C-ähnlichen Sprachen ist die Steuerung der Benutzungsoberflächen und die Präsentationsschicht einer Applikation. Die Stärke von COBOL dagegen ist die Anwendung von Geschäftsregeln und den damit verbundenen Datentransaktionen. Um wiederverwendbare Business-Services für eine SOA zu erstellen, eignet sich aus meiner Sicht COBOL sehr gut.

COBOL-Transaktionen erfüllen die Anforderungen von Fachabteilungen recht präzise. Sie bewerten beispielsweise den Versi-



Rolf Becking  
 (E-Mail: [Rolf.Becking@microfocus.com](mailto:Rolf.Becking@microfocus.com))  
 ist Senior Consultant bei Micro Focus in Dortmund.

cherungsschutz in Alltagssituationen, sie ermitteln die Krankengeschichte von Patienten oder berechnen aktuelle und prognostizierte Werte von Lebensversicherungen. Alle derartigen Aufgaben bestehen aus einer Reihe von COBOL-Transaktionen, bei denen mehrstufige Datenabfragen und -aggregationen erforderlich sind.

Wer sich für SOA als Softwarearchitektur entscheidet, muss sich nicht weiter darum kümmern, ob die Aktivitätenabfolge aus einer oder mehreren COBOL-Transaktion besteht. Es kommt vielmehr auf den mit COBOL modellierten Geschäftsprozess an – einschließlich der Transaktionen, Screens, Datenzugriffe, Algorithmen, Ablauflogik usw. Der Nutzen einer SOA entsteht durch den in COBOL gespeicherten und als Service zugänglichen Geschäftsprozess (siehe Abb. 1).

Zu klären bleiben abschließend die vier Fragen nach *Service-Discovery*, *-Definition*, *-Generierung* und *-Deployment*. Erst dann kann ein Client die Business-Services effektiv nutzen. So steht den Entwicklern mit „Micro Focus Revolve“ beispielsweise ein leistungsfähiges Analyse-Tool zur Verfügung, das eine umfassende Sicht auf die Funktionen und Interdependenzen von Applikationen ermöglicht. Damit lässt sich sehr schnell feststellen, welche Funktionen zur Erfüllung bestimmter Business-Services bereitstehen.

Sind die transaktionalen und Ein/Ausgabe-Datenanforderungen eines Business-Service ermittelt, stehen Werkzeuge bereit, um die benötigten Business-Services zu erzeugen. Beispielsweise bietet Micro Focus zwei Werkzeuge („SOA Express“ und das „Interface Mapping Toolkit“ in „Net Express“) abhängig davon, wo der Service letztlich installiert und betrieben wird. Sind die Business-Services einmal generiert, lassen sie sich in die unterschiedlichsten Mainframe-Umgebungen (CICS, IMS) oder auch im Umfeld von J2EE und .NET Application Server in SOA-Lösungen integrieren.

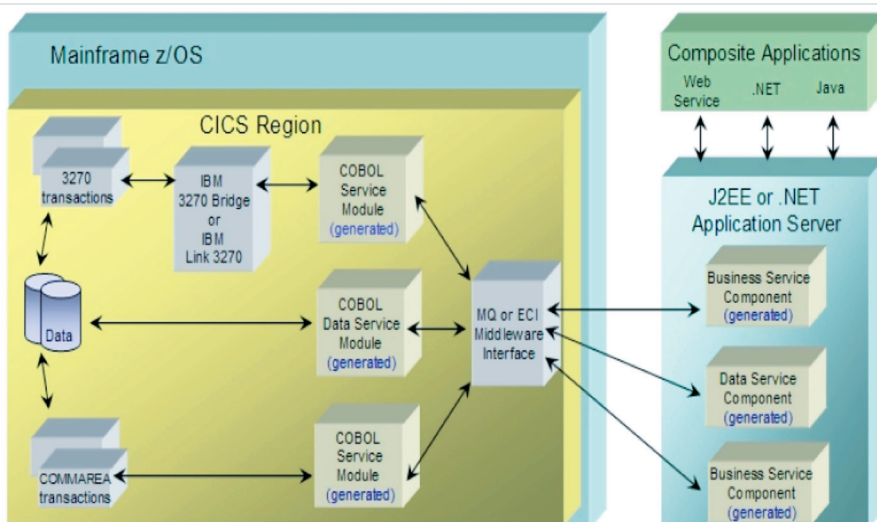


Abb. 1: Die Architektur einer servicefähigen COBOL-Applikation auf dem Mainframe