

Migration des Debeka-Software-Repositorys auf ein RDBMS

Johannes Bach, Martin Schulze

Debeka-Hauptverwaltung, Abteilung IE/Q, D-56058 Koblenz
{johannes.bach,martin.schulze}@debeka.de

1 Zusammenfassung

Dieser Beitrag stellt das *Software-Repository* der Debeka sowie seinen Einsatz in der Softwareentwicklung vor. Er beschreibt weiterhin die *Migration* des Repositorys von Rochade auf eine *Oracle-Datenbank* und die Neuentwicklung der Repository-Anwendungen. Schließlich werden Funktionen eines neuen *Repository-Browsers* vorgestellt, die auf Basis des neuen Systems möglich geworden sind.

2 Das Software-Repository der Debeka

In der Anwendungsentwicklung der Debeka-Versicherungsgruppe wird ein *zentrales Repository* eingesetzt, das die Softwarelandschaft abbildet. Es dient unter anderem als Informationsquelle für die Software-Entwicklerinnen und -Entwickler und als Grundlage für das Auftragswesen und die Produktionsübernahme von Softwarebausteinen. Im Repository werden Informationen über die Programme, Jobs, Skripte, Datenbanken und Masken der Fachanwendungen auf dem Bull-Großrechner und den dezentralen Unix-Systemen sowie über Entwicklungsaufträge abgelegt. In ihrer *täglichen Arbeit* ermitteln die Entwickler ausgehend von den Anforderungen aus den Fachabteilungen mit Hilfe des Repositorys, welche Softwarebausteine von gewünschten Änderungen betroffen sind, und analysieren Abhängigkeiten zwischen den Bausteinen. Nach ihren Änderungen und Erweiterungen wird die Übernahme der geänderten Software in die Produktion ebenfalls durch das Repository-System gesteuert. Neben diesen Routinearbeiten wurde und wird das Repository in *Projekten* wie dem Jahr-2000-Projekt, der Euro-Umstellung oder der Migration der Großrechner-Software auf einen Unix-Cluster eingesetzt.

Basis des Repositorys ist das Produkt *Rochade* der Firma Allen Systems Group, Inc. (ASG), das die Debeka seit 1995 im Einsatz hat. Das Metamodell des Repositorys ist eine Eigenentwicklung der Debeka, es umfasst Debeka- und Bull-Spezifika. Der Zugriff auf die Inhalte des Repositorys erfolgt über selbst entwickelte Software, die in Visual Basic, COBOL und einer proprietären Programmiersprache von Rochade (RPL) geschrieben ist. Mit einem modellunabhängigen Repository-Browser navigieren die Entwickler in den Inhalten des Repositorys. Ein Werkzeug ZEPRA („Zentrales Projektunterstützungs- und Auftragsystem“) dient zur Erfassung von Entwicklungsaufträgen und zum Anstoßen der Produktionsübernahme. Die Datenbankadministration verwendet ein Werkzeug DBis („Datenbasis-Import-/Export-System“) zur Über-

nahme von Datenbankschemata aus einem CASE-Werkzeug und zur Propagation von Datenbankänderungen vom Test- auf das Produktionssystem. Bei der Produktionsübernahme befüllen Quelltext-Scanner das Repository mit Hilfe eines Importers.

3 Migration des Repositorys auf ein relationales DBMS

Trotz (oder wegen?) der stetigen Weiterentwicklung des Repository-Systems wurde es immer schwieriger, auf neue Anforderungen aus anderen Projekten und Vorhaben der Debeka flexibel und ausreichend schnell zu reagieren. Die Wartung und Administration gestaltete sich zunehmend aufwändiger, so dass es *Zeit* für eine *Neuentwicklung* wurde. Dabei war es ein wichtiges Ziel, proprietäre Lösungen abzuschaffen und eine Neuausrichtung auf Technologien vorzunehmen, die auch in anderen Bereichen der Debeka im Einsatz sind. Deshalb wurden für das neue System *Oracle* als relationale Datenbank und *Java* als Programmiersprache für die Repository-Anwendungen ausgewählt.

Das neu entwickelte System wurde „*Zentrales Debeka-Repository*“ (ZEDER) genannt. Das Metamodell des alten Systems konnte grundsätzlich übernommen werden, wurde aber überarbeitet und insbesondere um ein Vererbungskonzept erweitert. Zum Zugriff auf die Datenbank wurde ein gemeinsames Repository-API für alle Anwendungen implementiert. Auf Basis des APIs wurden der Browser und ZEPRA neu entwickelt. Die Scanner konnten unverändert weiterverwendet werden, lediglich der Importer wurde neu geschrieben. DBis wird durch ein Werkzeug zur Synchronisation zwischen Rochade und ZEDER angebunden. Die Migration der Daten erfolgt durch Lade-Programme, welche die manuell erstellten Informationen (Stammdaten, Entwickleraufträge, Dokumentationen, ...) aus Rochade extrahieren und nach ZEDER übertragen, sowie durch Neuscannen der Quelltexte aller Softwarebausteine.

Das *Repository-API* als Kern aller Anwendungen setzt die Java-Klassenhierarchie der Repository-Objekte auf getrennte Datenbanktabellen für Basis- und abgeleitete Attribute und Beziehungen um. Zum effizienten Zugriff auf die Objekte des Repositorys dient eine Pfadsprache, deren Ausdrücke direkt in SQL umgesetzt und auf dem DB-Server ausgewertet werden, ohne dass Objekte in den Client geladen werden müssen. Weiterhin stellt das API die Konsistenz des Repositorys sicher. Es wird deklarativ an-

gegeben (durch sog. „Link Deletion Strategien“), wie beim Deaktivieren und Löschen von Objekten abhängige Objekte ebenfalls automatisch zu deaktivieren oder zu löschen sind oder dass die Löschung verhindert wird, wenn noch Verweise bestehen. Für jede Klasse können weitergehende Konsistenzüberprüfungsmethoden angegeben werden, die beim Abschluss einer Transaktion ausgeführt werden und verhindern, dass inkonsistente Objekte geschrieben werden.

4 Der Repository-Browser

Der *ZEDER-Repository-Browser* (Abb. 1) wurde als *Eclipse*-Anwendung implementiert, wodurch eine Integration mit den anderen auf Eclipse basierenden Entwicklungswerkzeugen der Debeka leicht möglich ist. Der Browser erlaubt den Entwicklern zunächst eine Auswahl von Repository-Objekten anhand ihres Typs sowie über einen Attributfilter. Die ausgewählten Objekte werden in einer Objektliste dargestellt, in der die für eine Fragestellung relevanten Objekte selektiert werden. Durch die Angabe eines Linktyps werden in einer weiteren Liste alle Objekte dargestellt, die zu einem Objekt aus der ersten Liste benachbart sind. Diese Liste kann wiederum gefiltert und selektiert werden und dient als Ausgangspunkt für den nächsten Linkverfolgungsschritt. Dieses Vorgehen wird solange wiederholt, bis ein gewünschter Pfad verfolgt ist. Zu allen Objekten in einer der Listen können die Attribute, Verweise und Dokumentationen angezeigt werden.

Gegenüber dem Vorgänger erlaubt der für ZEDER neu entwickelte Browser eine *einfachere Navigation* und eine *übersichtlichere Darstellung* der Repository-Inhalte durch Abkürzungs-Links und die Auswahl von Detaillierungsgraden. Abkürzungs-Links sind durch die Pfadsprache definierte Pfade, die häufig benötigte Abfragen direkt zugänglich machen. Durch eine vorgegebene Zuordnung von Klassen und Attributen zu Detaillierungsgraden kann ein Benutzer des Browsers einstellen, welche Informationen er sehen und welche er ausblenden möchte.

Eine weitere Verbesserung ergibt sich durch die Einführung der *Vererbungshierarchie* auf Repository-Objekte. Bisher war es in manchen Fällen notwendig, mehrere Anfragen durchzuführen, weil Pfade von unterschiedlichen Objekten ausgingen, die nun unter einer gemeinsamen Oberklasse zusammengefasst werden konnten. Da nun Objektlisten nicht mehr notwendigerweise homogen sind und je nach Selektion der Objekte unterschiedliche Attribute dargestellt werden können und unterschiedliche Verweise gültig sind, wird der Benutzer durch eine Sortierung der Attribute und Verweise eines Objekts oder einer Objektmenge anhand der Vererbungshierarchie unterstützt.

5 Fazit

Die Migration des zentralen Software-Repositorys der Debeka auf ein RDBMS und die Neuimplementierung der Anwendungen in Java ermöglicht es, auf *Anforderungen* aus Projekten *schneller reagieren* zu können und weniger Aufwand für Administration und Wartung des Systems betreiben zu müssen. Darüber hinaus stellt es einen wichtigen Beitrag zur *Abschaffung proprietärer Systeme* und zur Konsolidierung und Neuausrichtung auf moderne, in anderen Bereich des Unternehmens eingesetzte Technologien dar. Die im Vorhaben gesammelten *Erfahrungen* und das aufgebaute *Know-How* werden in weiteren Projekten der Debeka zum Einsatz kommen. Die Entwicklerinnen und Entwickler der Debeka werden schließlich durch ein *leistungsfähiges System* mit einer ansprechend und ergonomisch gestalteten Oberfläche bei ihrer täglichen Arbeit unterstützt.

Die Autoren danken dem Repository-Team der Abteilung IE/Q: Maria Defrancesco, Daniel Höh, Boris Klug, Beatrix Reppert und Tanja Sopoth.

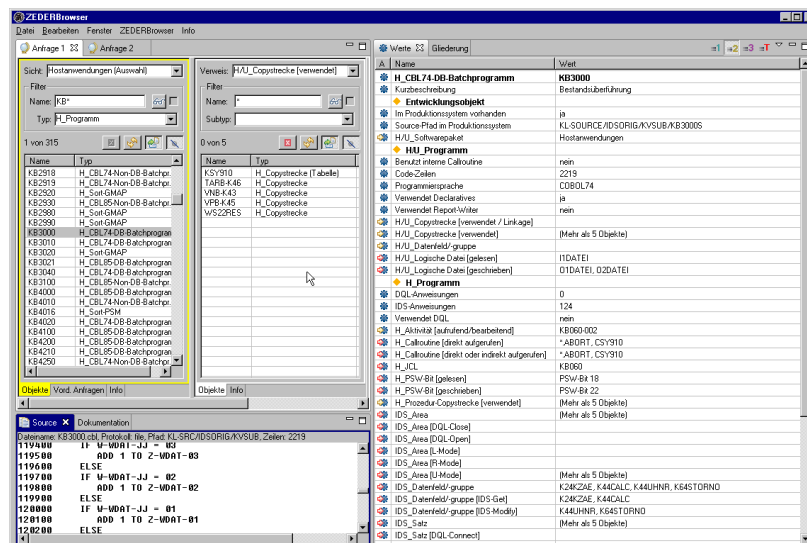


Abbildung 1: ZEDER-Browser