DELTA software technology

Technologiewechsel

Automatisch und im laufenden Betrieb

Je länger eine Anwendung lebt, desto größer ist die Wahrscheinlichkeit, dass sie verschiedene Technologien für ein und die selbe Aufgabe enthält. Um die Wartbarkeit und Performance der Anwendung langfristig zu sichern, gilt es, diese Technologien zu konsolidieren. Mit einer regelbasierten Factory kann ein solcher Technologiewechsel erfolgreich im laufenden Betrieb durchgeführt werden.

1 Verschiedene Technologien für eine Aufgabe

Software lebt - und das häufig viel länger als ursprünglich gedacht. Bei Neu- oder Weiterentwicklung von Anwendungsteilen wird oft auf neue Technologien gesetzt. Bestehende und funktionierende Anwendungsteile werden jedoch nicht angepasst. Daraus resultieren Anwendungen, in denen verschiedene Technologien nebeneinander existieren oder gar interagieren müssen.

Eine Koexistenz verschiedener Technologien kann mit der Zeit zu Problemen führen: die Performance der Anwendung nimmt ab, der Aufwand für die Wartung steigt und ggf. wird auch eine Migration auf eine neue Plattform verhindert. Um dem entgegen zu wirken ist es erforderlich Technologien auszutauschen oder zu konsolidieren. Oft sind die auszutauschenden Technologien jedoch fest in den Anwendungen verankert. Und diese Anwendungen sind zumeist groß, komplex und unternehmenskritisch. Ein manueller Technologiewechsel wäre deshalb zu aufwändig und risikoreich, würde einen enormen Testaufwand bedeuten und die reguläre Wartung beeinträchtigen.

Wir zeigen am Beispiel IMS/DB-Ersetzung, wie ein solcher Technologiewechsel dennoch gelingen und der Testaufwand auf ein vertretbares Niveau verringert werden kann.

1.1 Von hierarchisch zu relational

Die Kernanwendungen großer Unternehmen, Banken und Versicherungen sind oft in COBOL oder PL/I implementiert. Als Datenhaltungssystem wurde zunächst IBM IMS/DB eingesetzt, im Laufe der Zeit kamen jedoch auch andere Datenhaltungssysteme hinzu ohne dass die IMS/DB ersetzt wurde. Das Besondere an der IMS/DB ist, dass die Daten hierarchisch organisiert sind, während sie in neueren Datenbanksystemen meist relational organisiert werden. Um die Datenhaltung von hierarchisch nach relational zu überführen ist somit ein Paradigmenwechsel erforderlich, ein 1:1-Austausch ist nicht möglich.

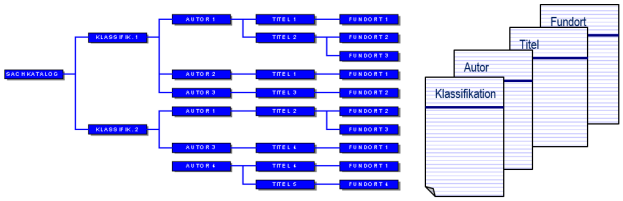

Figure 1: Hierarchische vs. relationale Datenhaltung

Um die Verwendung von IMS/DB zu ersetzen, müssen drei Arten von Code-Änderungen vorgenommen werden: Die Zugriffe auf die Datenbanken müssen ersetzt werden, IMS-spezifische Fehler-Codes gegen neutrale Codes ausgetauscht und die Schnittstellen der Programme bereinigt werden. Die Schnittstellenbereinigung ist notwendig, da bei Programmaufrufen IMS-Verwaltungsstrukturen (PCB) übergeben werden.

2 Automatischer Technologiewechsel im laufenden Betrieb

Wir schlagen für den Austausch der IMS/DB den Einsatz einer modell- und regelbasierten Factory vor. Durch zielgerichtete Analysen erzeugt die Factory für alle Artefakte der Anwendung verschiedene Modelle, wie den Abstract Syntax Tree oder Daten- und Kontrollflussmodelle. Alle weiteren Schritte erfolgen dann auf den Modellen.

Die Ersetzung der IMS/DB erfolgt regelbasiert in drei Schritten. Zunächst werden alle Stellen im Code ermittelt, die evtl. zu transformieren sind. Diese Stellen werden genauer analysiert und ihr Kontext bewertet. Abhängig von der Kontextanalyse erfolgt dann die eigentliche Transformation. Dazu werden die Modelle mittels der Transformationsregeln modifiziert und der neue Code generiert. Für IMS muss bei der Analyse zunächst festgestellt werden, ob es sich tatsächlich um einen Zugriff auf eine Datenbank handelt oder um einen Aufruf des Transaktionsmonitors (IBM IMS/TM). Für einen Austausch des Datenbanksystems ist nur im ersten Fall eine Transformation erforderlich. Mittels Kontroll- und Datenflussanalyse wird bestimmt, wie die Übergabeparameter belegt sind, d.h. auf welche Datenbank wie zugegriffen werden soll.

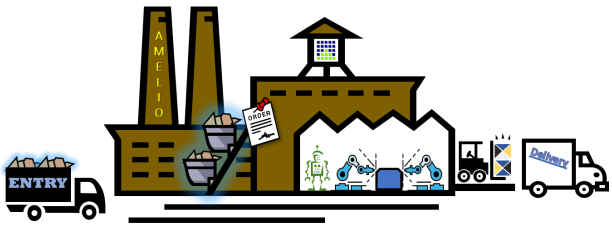

Figure 2: Transformations-Factory

Durch das modell- und regelbasierte Vorgehen sind die Transformationen absolut gleichförmig und reproduzierbar. Damit kann eine Anwendung in Pakete unterteilt und diese einzeln transformiert werden. Ist ein Paket umgestellt, erfolgt der Test. Entspricht das Ergebnis nicht den Anforderungen, wird die verantwortliche Transformationsregel angepasst. Selbst eine Modifikation des zugrunde liegenden Datenmodells stellt kein Problem dar. Bereits transformierte Pakete werden nach Anpassung erneut transformiert, so dass Entscheidungen auch relativ spät im Projekt noch revidiert werden können. Es gibt sehr viele Möglichkeiten IMS/DB zu verwenden und die Zugriffe zu implementieren, deshalb werden die Regeln der Factory immer auf die jeweiligen Projektziele angepasst. In dieser Zeit kann die Wartung der Anwendung ungestört weitergehen. Ist die Factory konfiguriert und getestet, so benötigt es nur wenige Stunden um auch große Anwendungen vollständig zu transformieren. So kann der Technologiewechsel auch im laufenden Betrieb erfolgen.

DELTA
software
technology

2.1 Eine zusätzliche Datenzugriffsschicht

Soll IMS/DB durch ein relationales Datenbanksystem ersetzt werden, so ist eine 1:1-Ersetzung auf Grund des Paradigmenwechsels nicht möglich. Beim Wechsel des Datenbanksystems soll die Anwendung so wenig wie möglich modifiziert werden. Die Daten, die aus der relationalen Datenbank geliefert werden, müssen deshalb so aufbereitet werden, als würden sie aus der IMS/DB stammen. Gleiches gilt für das Schreiben der Daten. Das bedeutet auch, dass es in vielen Fällen notwendig ist, eine Zeile Code durch einen ganzen Code-Block zu ersetzen. Deshalb bietet es sich stattdessen an, eine neue Zugriffsschicht in Form von Datenservices einzuführen.

Die Datenservices können aus den vorliegenden Modellen automatisch generiert werden. Innerhalb der Services werden die Zugriffe auf die Datenbank realisiert, so wie die Aufbereitung der Daten. In der Anwendung muss dann nur der Zugriff auf die IMS/DB durch einen passenden Aufruf des Datenservices ersetzt werden. Dies gilt dann auch für Fehlercodes: IMS-spezifische Fehler-Codes werden durch die Factory durch neutrale Fehler-Codes ersetzt. Der Datenservice bildet die Fehler-Codes des neuen Datenbanksystems auf die neutralen Codes ab. Durch die Einführung der Datenservices können somit die Eingriffe in den bestehenden Code reduziert werden.

3 Regeln statt Änderungen testen

Gleichförmigkeit und Reproduzierbarkeit des regelbasierten Vorgehens lassen sich auch beim Testen ausnutzen. Denn: ist eine Regel einmal korrekt, so ist sie dies auch bei mehrmaliger Anwendung. Es ist also nicht notwendig alle geänderten Programme der Anwendung zu testen, sondern alle Regeln. Durch die Auswahl eines geeigneten Testsets, das alle Regeln abdeckt, aber nur einen vergleichsweise kleinen Ausschnitt der gesamten Anwendung darstellt, kann der Testaufwand signifikant reduziert werden.

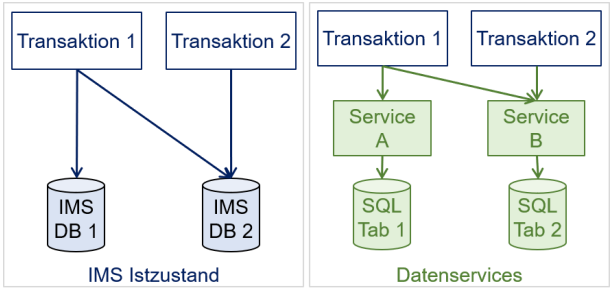

Figure 3: Einführung einer Datenserviceschicht

4 Automatischer Technologiewechsel

Ein Technologiewechsel ist auch für große und komplexe Anwendungen im laufenden Betrieb möglich, wenn ein automatisierter Ansatz gewählt wird. Die Einführung einer Datenzugriffsschicht kann die notwendigen Eingriffe in die bestehende Anwendung beim Tausch des Datenhaltungssystems minimieren. Durch Ausnutzen des regelbasierten Vorgehens werden die notwendigen Tests auf ein Minimum beschränkt. Anschließend lässt sich die modernisierte Anwendung wieder mit weniger Aufwand warten und kann effizient ihre Aufgaben erfüllen.

Bestellnummer: MT 11091.01 – April 2019

Delta Software Technology GmbH · Eichenweg 16 · 57392 Schmallenberg, · Tel +49 2972 9719-0 · E-Mail: info@delta-software.com · www.delta-software.com