

FiRe - Ein Werkzeug für die Datenmigration

Christian Becker, Uwe Erdmenger, Uwe Kaiser, Denis Uhlig

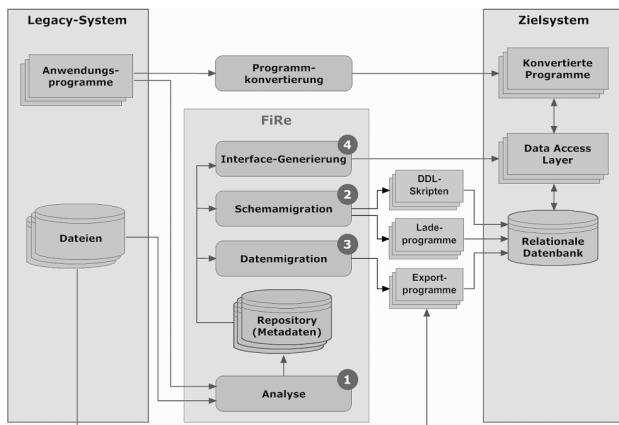
pro et con Innovative Informatikanwendungen GmbH, Dittesstraße 15, 09125 Chemnitz
{christian.becker, uwe.erdmenger, uwe.kaiser, denis.uhlig}@proetcon.de

Abstract

Eine wesentliche Komponente der Softwaremigration ist die Migration von Dateien aus dem Legacy- in das Zielsystem. Dabei werden die Aufgaben der Schemamigration, der eigentlichen Datenmigration und die Migration der Programme, welche im Zielsystem mit den Daten arbeiten, unterschieden. Die Autoren beschäftigen sich schon seit vielen Jahren mit der Entwicklung von Migrationswerkzeugen [1]. In diesem Beitrag wird die aktuelle Entwicklung *FiRe* (**F**iles to **R**elational Database) vorgestellt. FiRe ist eine Toolbox zur Unterstützung der Datenmigration.

1 Überblick

FiRe bietet einen durchgehend automatisierten Prozess, um Dateien aus dem Legacy-System in relationale Datenbanken im Zielsystem zu migrieren. Die folgende Abbildung zeigt eine schematische Prozessdarstellung.



Der erste Schritt ist die *Analyse* der zu migrierenden Dateien. Es werden die Dateibeschreibungen in den Anwendungsprogrammen einschließlich der Satzstrukturen mit einem Compiler-Frontend analysiert und in einem definierten Format (Metadaten) in einem *Repository* gespeichert. Mit diesen Informationen arbeitet die *Schemamigration*, bei der ein Mapping der Dateien des Legacy-Systems auf Datenbanktabellen des Zielsystems erfolgt. Im Ergebnis werden DDL- und Ladeprogramme für das Einrichten der relationalen Zieldatenbank generiert. Im darauf folgenden Schritt der eigentlichen *Datenmigration* gilt es, die relationale Zieldatenbank zu füllen. Auf Basis der Metadaten des Repositories werden Exportprogramme in der Pro-

grammiersprache des Legacy-Systems generiert. Diese exportieren die Dateien des Legacy-Systems in ein neutrales Format, welches für notwendige Datenkonvertierungen (z.B. EBCDIC - ASCII) benötigt wird. Das Ergebnis wird von den in der Schemamigration generierten Ladeprogrammen schließlich in die Datenbank übertragen. Die Migration der Datenbestände ist damit abgeschlossen. Die Anwendungsprogramme des Zielsystems benötigen ein Interface, mit dem auf die Daten in der Datenbank analog zum Legacy-System satzorientiert zugegriffen werden kann. Der dazu notwendige *Data Access Layer* wird ebenfalls komplett aus den Metadaten des Repositories generiert. Der gesamte Prozess ist in FiRe standardisiert und wird über eine graphische Oberfläche (GUI) gesteuert. Die folgenden Ausführungen beschreiben ausgewählte Eigenschaften dieses Prozesses. Als Beispiel dient eine konkrete BS2000-Migration nach Linux. In diesem Projekt wurden die COBOL-Programme werkzeuggesteuert nach Java konvertiert und die BS2000-typische Datenhaltung (SAM-, ISAM-, LEASY-Files) mit FiRe in das relationale Datenbanksystem Oracle migriert.

2 Schemamigration

Struktur und Aufbau der Dateien des Legacy-Systems sind im Repository als Metadaten gespeichert. Auf der Basis der Metadaten erfolgt initial eine automatische 1:1-Abbildung von Dateien (SAM, ISAM, LEASY) auf Datenbanktabellen, wobei jede Datei als Tabelle und die (COBOL)-Datenfelder als Tabellenspalten definiert werden. Dies kann natürlich nur der Ausgangspunkt für weitere Mapping-Manipulationen sein, denn gerade bei historisch gewachsenen Systemen würden dadurch Tabellen mit mehreren hundert Spalten sowie langen und schwer lesbaren Bezeichnern im Zielsystem entstehen. Ebenso müssen Limitierungen der Zieldatenbank beachtet werden. In Oracle sind z.B. maximal 1000 Spalten pro Tabelle und 30 Zeichen pro Bezeichner zulässig. Weiterhin ist dem Anwender die zukünftige Wartung solcher großer Tabellen nicht zuzumuten. Aus diesem Grund bietet FiRe die Funktionalität eines Redesigns der Zieldatenbank. Dabei handelt es sich nicht um ein relationales Redesign, sondern um ein Redesign im Sinne der Datenstrukturierung. FiRe unterstützt dieses durch ein Datenfeld-/Datei-Mapping. Z.B. kann eine große Datenbanktabelle auf mehrere, kleinere

abgebildet werden. Diese Zerteilung wird vertikal an Datenfeldgrenzen durchgeführt. Weiterhin erlaubt COBOL z.B. Dateien mit multiplen Satzstrukturen, d.h., es existieren innerhalb einer Datei Datensätze, die sich in Struktur und Aufbau unterscheiden. Mit FiRe werden diese Probleme durch die Auslagerung solcher Satzstrukturen in eigene Untertabellen im Zielsystem gelöst. Die Datenkonsistenz und -integrität wird dabei über Primär- und Fremdschlüsselbeziehungen in der DDL-Generierung sichergestellt.

3 Datenmigration

Diese Komponente beinhaltet die eigentliche Übertragung der Daten aus den Dateien des Legacy-Systems in die Datenbanktabellen des Zielsystems. Eine wichtige Funktionalität dabei ist u.a. die Abbildung von COBOL-Typen auf Typen des Datenbanksystems Oracle. Auch hier wird in FiRe zuerst ein Standard-Mapping vorgegeben:

	COBOL-Typ	Oracle-Typ
alphanumerisch	PIC XXX	Varchar2(3)
numerisch integer	PIC 99	Number(2)
numerisch dezimal	PIC 99V9	Number(3,1)

Danach kann dieses Standardmapping in FiRe verfeinert werden. Wurde z.B. in der Schemamigration eine große Datei in mehrere Zielsystemtabellen aufgeteilt, dann muss auch die Datenkonvertierung im Ergebnis zerteilte, separat ladbare und konsistente Datenbestände liefern. Häufig müssen auch Änderungen des Typs und der Bezeichner in der Datenbank vorgenommen werden. Festlegungen zur Abbildung von Dateien des Legacy-Systems in Tabellen des Zielsystems in der Schemamigration haben auch Auswirkungen auf die Konvertierung der Daten während der Migration der eigentlichen Datenbestände. In historisch gewachsenen Legacy-Systemen leidet bedingt durch die langen Wartungszeiträume die Datenqualität. Es treten Inkonsistenzen wie z.B. Texte in numerischen Datenfeldern, defekte Umlaute in Textfeldern und fehlerhafte, numerische Kodierungen auf. Während der COBOL-Compiler tolerant mit solchen Ungereimtheiten umgeht, reagieren Datenbanksysteme mit Fehlermeldungen. In diesem Fall ist in der Schemamigration eine Änderung des Datenfeld-Mappings notwendig.

4 Interface-Generierung

Diese FiRe-Komponente realisiert den Zugriff der Anwendungsprogramme auf die relationalen Datenbanken im Zielsystem. Im Legacy-System wird auf die Files satzorientiert zugegriffen, nach der Migration liegen diese Informationen aber als relationales Schema vor. Daher ist eine bidirektionale Abbildung von Datensätzen in Relationen in Form eines Data Access Layers erforderlich. Dieser wird automatisch generiert und ist in allen verwendeten

Sprachen des Zielsystems nutzbar (z.B. Java, Perl, ...). Ausgangspunkt für seine Generierung sind die durch die Schemamigration bereitgestellten Metadaten des Repositories. Sie beschreiben den Aufbau der Daten vor und nach der Migration. Die Interface-Generierung generiert daraus ein C-Programm (Data Access Layer), welches den Zugriff auf die Datenbank mit dynamischem SQL, die Transaktionssteuerung und die Abbildung der Datensätze in Relationen (und umgekehrt) zur Laufzeit übernimmt. Der Data Access Layer arbeitet in einem separaten Prozess. Im Migrationsprojekt erfolgte eine werkzeuggestützte Konvertierung der Programme von COBOL nach Java mit dem Werkzeug CoJaC [2]. In die Anwendungsprogramme des Zielsystems wurde dabei eine Zugriffsschicht integriert, welche mit Mitteln der Interprozesskommunikation (IPC) auf die Funktionen des Data Access Layers zurückgreift. Das hat den Vorteil, dass nicht für jede Programmiersprache des Zielsystems ein eigenes Interface zu entwickeln ist. Die durch die IPC bedingte etwas höhere Laufzeit wird durch effiziente Implementierung in C, optionale Caches bei wiederholtem Lesen identischer Datensätze sowie vorausschauendes Lesen des nächsten Satzes im parallel zu den Anwendungsprogrammen laufenden Data Access Layer (*Prefetching*) ausgeglichen. Diese Optimierungen müssen nicht manuell für einzelne Programme vorgenommen werden. Die Laufzeit der Programme wird protokolliert und aus den Statistikdaten wird automatisch für jedes Programm eine individuelle Optimierungsstrategie abgeleitet. Die erreichte Performance ist dadurch mit der Performance der Filezugriffe im Legacy-System vergleichbar.

5 Zusammenfassung

Erfahrungen aus erfolgreichen Migrationsprojekten flossen in das Werkzeug FiRe ein ([3]). Der Fokus bei der Datenmigration mit FiRe liegt durch den hohen Automatisierungsgrad und einer komfortablen Entwicklungsumgebung nicht mehr auf den eigentlichen Migrationsarbeiten (Übertragung und Konvertierung von Dateien etc.), sondern im strukturellen Redesign der Datenhaltung des Zielsystems. Neben der Reduzierung des manuellen Aufwandes bei der Datenmigration führt dies zu einer verbesserten Datenqualität im Zielsystem.

References

- [1] Erdmenger, U.; Kaiser, U.; Loos, A.; Uhlig, D.: Methoden u. Werkzeuge für die Software Migration. Lecture Notes in Informatics, (LNI)-Proceedings, Volume P-126, S. 83-97.
- [2] Kaiser, U.: Aus alt mach neu - automatisch. In: Business Technology 03/2014, Heft 18, S. 41-46.
- [3] Kaiser, U.; Uhlig, D.: Erfolgreiche BS2000-Migration. In: it-dayli.net, Oktober 2016.