

Migration der Daten von Legacy-Anwendungen in Oracle-Umgebungen

Erfahrungen mit den G7-Migrations-Werkzeugen

Zielsetzung

Durch den erfolgreichen Einsatz von relationalen Datenbanken für die Speicherung der Unternehmensdaten vergrößert sich der Druck auf solche Unternehmen, die noch mit proprietären Dateisystemen oder nichtrelationalen Datenbanken arbeiten, ihre Datenhaltung zu reformieren. Die Unternehmen können außerdem durch Konsolidierung der Datenbestände IT-Kosten senken und die Flexibilität der Anwendungen verbessern.

Um bestehende und bewährte Mainframe-Anwendungen (ohne Veränderung der Funktionalität für die Benutzer) zu migrieren und dabei die nichtrelationalen Datenstrukturen (indexsequentielle, hierarchische und Netzwerk- bzw. CODASYL-Strukturen) in relationale Strukturen zu konvertieren, gibt es die G7-Migrations-Werkzeuge, die einen hohen Automationsgrad gewährleisten.

In diesem Beitrag wird auf Grund erfolgreicher Projekte mit den G7-Migrations-Werkzeugen und deren Konvertern und Generatoren sowie dem in der Zielumgebung eingesetzten G7-Framework gezeigt, wie mit nur geringem manuellen Aufwand eine sanfte, risikofreie und smarte Transformation von proprietären Dateisystemen (ISAM, VSAM, LEASY, KLDS, UFAS, DATACOM), sowie nichtrelationalen Datenbanken (IMS, DL/1, IDMS, IDS2, DMS, UDS) in eine moderne relationale Datenbank-Umgebung (SOA) ermöglicht wird. In Abbildung 1 sind die möglichen Datenstrukturen schematisch dargestellt.

Werkzeuge und Datenbankschnittstellen (Adapter)

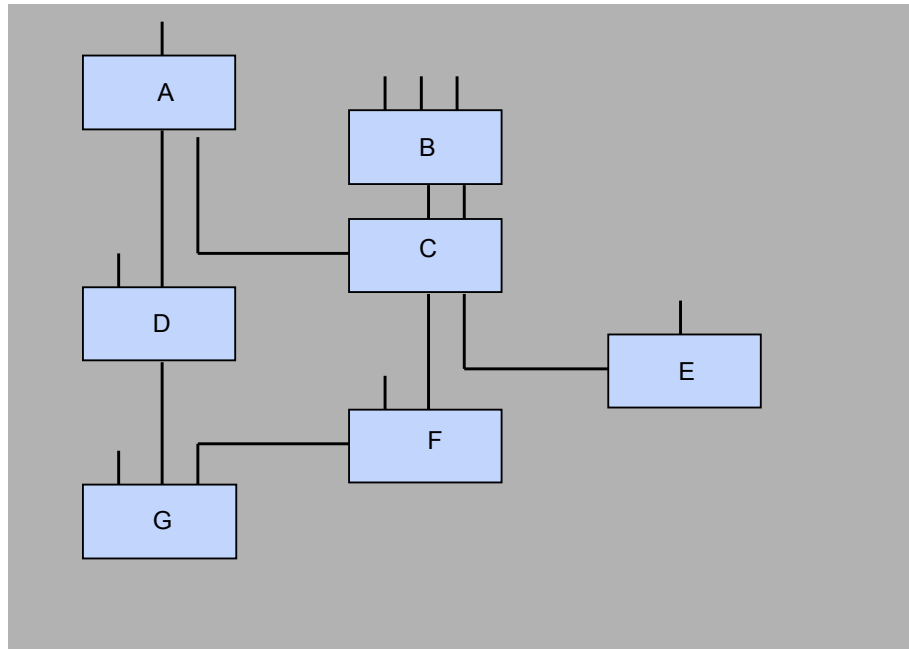
Bei der Daten-Migration werden alle relevanten Software-Komponenten, wie Datenstrukturen, Datenzugriffe (in den Programmen) und Daten, durch intelligente **G7-Konverter** transformiert. Auf Basis der Datenstrukturen (Schemas) werden die SQL-Skripte für die Definition der Datentabellen, Indexe Trigger und Grants, die Control-Skripte für den Oracle-Loader und die Zugriffs-Module (**Adapter**) mit Definition der Common-Currency-Bereiche und die Copy-Elemente (Includes) mit den Datenstrukturen (Programm-Sicht und DB-Sicht) durch **G7-Generatoren** automatisch generiert (siehe Abb.2).

Die generierten **Adapter** werden aus den Anwendungs-Programmen bei jedem Datenbankzugriff (Dateizugriff) per CALL aufgerufen. Durch Einbindung der betreffenden Copy-Elemente kennen sie die Eigenschaften der Datenbankstrukturen und sie kontrollieren sessionsbezogen die aktuelle Cursor-Position in der Datenbank, bezogen auf die Abhängigkeiten der Tabellen und Daten voneinander. Sie übernehmen auch die Konvertierung der Datensichten zwischen Programm und Datenbank. Zusätzlich wird durch die Adapter der Zugriff auf die Datenbank in Bezug auf Performanz optimiert.

Der Einsatz der G7-Migrations-Werkzeuge führt bei den bestehenden Legacy-Anwendungen zu einer deutlichen Modernisierung. Meistens ist die Zielumgebung ein offenes System (Unix, Linux oder Windows). GSE hat bereits mehr als 50 Migrations-Projekte gemeinsam mit seinen Kunden durchgeführt und konnte dabei selbst bei großen Projekten nachweislich einen **ROI von weniger als 6 Monaten** erzielen.

In der Abbildung 2 sind die für eine Datenmigration erforderlichen Konverter und Generatoren schematisch dargestellt:

Abb.1 Schematische Darstellung verschiedener Datenstrukturen



Erläuterungen:

In dieser Abbildung stehen die Kästchen für die Records (Datensätze) und die Striche für die Verbindungen zwischen den Records bzw. die Einstiegswege über die Sekundär-/Alternativ-Schlüssel (Key).

Auf den Record A kann nur über einen Key zugegriffen werden, auf Record B über drei Keys, auf Record C auf zwei Wegen über Record B und einen Weg über Record A, auf Record D über einen Key und über Record A, auf Record E über einen Key und über Record C, auf Record F über einen Key und über Record C, auf Record G über einen Key sowie über Record D und Record F.

Der Record C hat zwei Owner (A und B), der Record D einen Owner (A), die Records E und F jeweils einen Owner (C), der Record G zwei Owner (D und F). Der Record C hat Record B als Owner in zwei Owner-Member-Relationen, die sich voneinander beispielsweise durch die Sortierung unterscheiden (z.B. Order First und Order Last).

Diese prinzipielle Darstellung gilt nicht nur für die Codasyl-Datenbank, sondern als Sonderfall auch für die hierarchische Datenbank, bei der es eben nur einen Owner pro Member gibt, und für die index-sequentielle Datei, bei der es keine Verbindung mit anderen Dateien gibt, aber durchaus mehrere Einstiegswege.

Für jeden Record und jeden Einstiegsweg bzw. jede Verbindung sind im Schema Eintragungen in folgender Weise vorhanden:

Record: Recordname, Datenfelder mit Format-Beschreibung (Programmsicht und evtl. DB-Sicht)

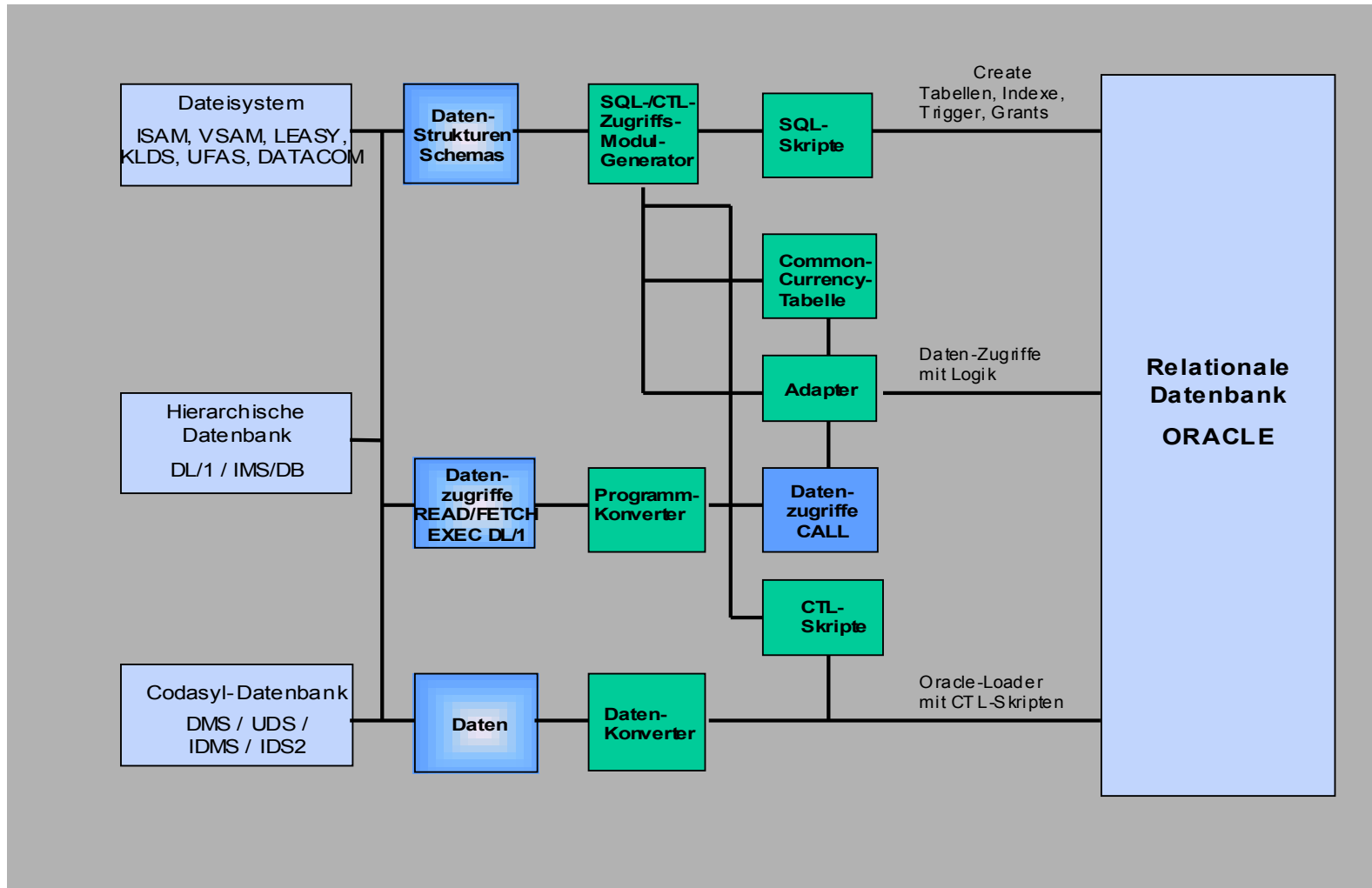
Verbindung: Setname, Ownername, Membername, Manual oder Automatic, Ownerkey-Name, Order (First, Last, Sorted)

Einstiegsweg: Keyname, Keybestandteile (Felder)

Aus den Records werden bei der Konvertierung die Tabellen in der Datenbank, aus den Einstiegswegen und Verbindungen die Indexe.

Die Datenzugriffe sind aber ebenfalls abhängig von diesen drei Eintragungsarten, die neben der Zugriffsart und der Performanz-Optimierung bei der Generierung der Datenzugriffsmodule mit dem G7-Generator Berücksichtigung finden.

Abb.2 Schematische Darstellung der für die Daten-Migration benötigten G7-Konverter und G7-Generatoren



Verfahren für die Datenkonvertierung

1. Indexsequentielles Dateisystem (ISAM, VSAM, LEASY, KLDS, UFAS, DATACOM)

In der Regel wird jede indexsequentielle Datei als eine relationale Tabelle in der Datenbank abgebildet. Mehrere Tabellen pro Datei können dann sinnvoll sein, wenn die ursprüngliche Datei Satzart- oder Überlagerungs-Strukturen aufweist und die Daten SQL-fähig gehalten werden sollen. In diesem Fall wird durch die intelligenten Adapter bei jedem Datenzugriff zur Ablaufzeit auf Grund der Satzart entschieden, welche Tabelle für diesen Zugriff relevant ist. Dies funktioniert auch beim sequentiellen Lesen. Bei der Programm-Konvertierung genügt es – wie bei den einfachen Tabellen – im wesentlichen, die READ-/WRITE-Statements in CALL-Statements auf die Adapter zu konvertieren.

Daneben werden die Dateidefinitionen im Programm (DCL- bzw. SELECT-Statements) entfernt und die Dateibeschreibungen (Satzdefinition in der File Section) in die Working-Storage Section übernommen. Die G7-Programm-Konverter und G7-Standard-Adapter für die Realisierung dieser Technik funktionieren für **COBOL- und PL/1-**Programme. Die Programmier-Sprache und die Programm-Struktur bleiben erhalten.

Sofern die Zugriffe bereits über CALLs (LEASY, VSAMIO, DATACOM) erfolgen, können die kompatiblen G7-Standard-Adapter (G7LEASY, G7VSAM, G7DATACOM) verwendet werden, die bereits bei vielen Unternehmen erfolgreich im Einsatz sind und alle solchen Zugriffe auf die relationale Datenbank umlenken.

Aus den Copy-Elementen (Includes) mit den Satzbeschreibungen werden (mit Hilfe eines G7-Generators und manueller Konfiguration) die Datenstrukturen der zu konvertierenden Dateien in das G7-Schema übernommen. Es bildet die Basis für die Generierung von SQL-Skripten, CTL-Skripten und Datenzugriffs-Modulen (Adapter) mit Hilfe des G7-SQL-/CTL-/Zugriffsmodul-Generators und enthält alle Informationen über Datenfelder, Schlüsselbegriffe (Primär- und Sekundär-Keys), Satzlängen und Satzartenfelder (für den Fall, dass in einer Datei unterschiedliche inkompatible Satzarten existieren, die in unterschiedliche Tabellen der Datenbank abzubilden sind).

Die G7-Standard-Adapter werden aus den Anwendungs-Programmen aufgerufen und greifen über die generierten Datenzugriffsmodule auf die relationale Datenbank zu.

Auch das G7-Datenkonvertierungs-Programm, das für die Übernahme der Daten aus dem indexsequentiellem Dateisystem in die Oracle-Datenbank sorgt, berücksichtigt die Informationen im Schema für die strukturabhängige Konvertierung der Daten in ein Format, das vom Oracle-Loader mit Hilfe der generierten CTL-Skripts in die Datenbank geladen werden. Dieses Datenkonvertierungs-Programm übernimmt gegebenenfalls auch die bei Plattformwechsel erforderliche EBCDIC-ASCII-Code-Konvertierung.

2. Hierarchische (DL/1, IMS/DB) und Codasyl-Datenbanken (UDS, DMS, IDS2, IDMS)

Für jedes Segment bzw. für jeden Record der Datenbank wird eine Tabelle in der relationalen Datenbank aufgebaut. Um die Hierarchie (bzw. das Netzwerk) abzubilden, werden in jeder einzelnen Daten-Tabelle die Schlüsselbegriffe aller direkt übergeordneten Segmente der hierarchischen bzw. Codasyl-Datenbank ergänzt, da beim Zugriff auf eine Tabelle der relationalen Datenbank alle übergeordneten Schlüsselbegriffe angegeben werden müssen (siehe Abb.2). Ist ein Schlüssel nicht eindeutig, dann wird er durch Ergänzung der automatisch vergebenen Satznummer eindeutig gemacht. Dies hat den Sinn, dass auch beim Lesen über einen mit ORDER FIRST oder ORDER LAST in der Codasyl-Datenbank definierten Schlüsselbegriff die richtige Reihenfolge eingehalten werden kann.

Um diese Schlüsselbegriffe automatisch versorgen zu können, werden die Daten aus der ursprünglichen Datenbank in einen sequentiellen File entladen und dabei für jedes Segment dessen Typ (Kennzeichnung) ergänzt. Die Reihenfolge der Entladung spiegelt die Hierarchie der einzelnen Datensätze im Verhältnis zu allen übergeordneten wider.

Unterschiedliche hierarchische oder Codasyl-Datenbanken können in unterschiedlichen Domains in der relationalen Datenbank abgebildet werden.

Bei hierarchischen und Codasyl-Datenbanken, die teilweise nicht hierarchisch organisiert sind, sondern beliebige Netzwerk-Strukturen aufweisen können, werden die Parent-Child-Relationen (hierarchische Datenbank) bzw. die SET-Relationen oder Owner-Member-Relationen (Codasyl Datenbank) bei der Generierung der Adapter berücksichtigt, sie werden nicht als eigenständige Tabellen gespeichert, weil der Pflegeaufwand dafür zu groß und die Performanz des Systems darunter leiden würde. Bei der Generierung der Adapter werden diese außerdem bezüglich der Performanz und zur Vermeidung von Deadlocks automatisch optimiert. Dies betrifft hauptsächlich die sequentiellen Lesevorgänge und die Modifikation von Datensätzen.

Bei den Codasyl Datenbanken gibt es bereits ein Schema, das direkt als Basis für die Generierung benutzt werden kann. Bei anderen Datenbanken müssen die dort verwendeten Strukturbeschreibungen durch einen G7-Konverter zuerst in das Schema-Format konvertiert werden.

Im übrigen gelten die gleichen Aussagen wie für die indexsequentiellen Dateisysteme, da in allen Fällen die G7-Generatoren und G7-Konverter auf der Basis der Schemas aufsetzen können.

Da die Records (Segmente) teilweise keinen oder keinen eindeutigen Schlüsselbegriff haben, die Adapter aber einen solchen für den Zugriff in der relationalen Datenbank benötigen, wird in jedem Datensatz durch automatische Vergabe beim Laden der Datenbank und beim Einfügen ein eindeutiger Schlüssel (Satznummer 18-stellig) zusätzlich vergeben. Dieser wird für spezielle Zugriffe benötigt.

Während der Programm-Ausführung werden durch die Adapter alle aktuellen Schlüsselbegriffe in der Common-Currency-Tabelle hinterlegt und gepflegt. Dadurch ist allen Adaptern, die in einer Session aufgerufen werden, die aktuelle Position jeder Tabelle in der Datenbank zu jeder Zeit bekannt und kann von ihnen bei allen Datenzugriffen abgefragt und automatisch ergänzt werden. Diese Common-Currency-Tabelle wird im Common-Bereich abgelegt, der vom TP-Monitor bzw. von G7 zur Verfügung gestellt wird, und steht damit allen Programmen zur Verfügung. Bei Batchprogrammen wird sie im Batch-Kontroll-Programm definiert, das Bestandteil von G7 ist. Bei der Programm-Konvertierung wird in allen Programmen diese Currency-Tabelle automatisch als Parameter eingeführt.

Konvertierung der Datenzugriffe mit dem G7-Programm-Konverter

Die Programmkonvertierung ist sehr einfach, wenn die Datenbank über CALL-Befehle (SESAM, CBLTDLI, PLITDLI) aus den Programmen aufgerufen wird. Für diese Aufrufe existieren bereits G7-Schnittstellen, die dem Programm die Illusion geben, daß sie mit den originalen Datenzugriffen und der ursprünglichen Datenbankstruktur arbeiten. Mit den Adaptionern werden dann sowohl die dezimal gepackten Felder als auch die Integer-Felder umgesetzt. Für dieses Verfahren sind in den Original-Programmen die Procedure Statements und die Call-Aufrufe zu modifizieren. Dies erfolgt mit dem G7-Programm-Konverter automatisch.

Für die Umstellung von SESAM und DL/1 bzw. IMS/DB wird neben diesen geringfügigen Modifikationen im Programm lediglich der entsprechende G7-Adapter mit dem Namen G7SESAM bzw. G7TDLI als Unterprogramm eingesetzt.

Etwas schwieriger gestaltet sich die Programmkonvertierung, wenn der Aufruf über spezielle Befehle einer DML erfolgt. Beispiele dafür sind: FETCH, FIND, ERASE, STORE, MODIFY, READY, FINISH bei den CODASYL-Datenbanken (DMS, UDS, IDS2, IDMS). Diese Befehle werden mit dem G7-Programmkonverter in CALL-Aufrufe auf G7-Adapter konvertiert, die über Common-Tabellen (g7-link-param, g7-uds-crs) jederzeit Kenntnis über die aktuellen Datenbankpositionen haben und über die SQL-Schnittstelle von ORACLE bzw. über ODBC den Datenbankzugriff durchführen lassen.

Beispiele für die Umsetzung der UDS-DML (Cobol):

1. Direktes Lesen eines untergeordneten Segments

```
G70017*   FETCH ARTIKEL WITHIN ARTKEY1
G70018*           USING ARTNR

G70019           move "ARTIKEL"      to g7-record
G70020           move "ARTKEY1"     to g7-key
G70021           move "FETCH7"      to g7-op
G70022           move ARTNR         to g7-using
G70023           call "G7-UDS" using ARTIKEL-REC
G70024                                   g7-link-param
G70025                                   g7-uds-cru
```

2. Modifizieren des zuletzt gelesenen Segments

```
G70051*   MODIFY ARTIKEL.

G70052           move "ARTIKEL"      to g7-record
G70053           move "MODIFY"      to g7-op
G70054           call "G7-UDS" using ARTIKEL-REC
G70055                                   g7-link-param
G70056                                   g7-uds-cru.
```

Ergebnisse der Daten-Migration mit den G7-Migrations-Werkzeugen

- **Durch Einsatz moderner Technologie wird eine Konsolidierung der Datenhaltung und damit eine Verbesserung der Kostenstruktur erreicht**
- **Die Anwendung wird durch automatisches Reengineering modernisiert**
- **Für die Speicherung aller wichtigen Unternehmensdaten wird nur ein relationales Datenbanksystem (Oracle) benötigt**
- **Durch die Transaktionssteuerung der Datenbank wird sowohl die Datenintegrität, als auch die Netzwerkfähigkeit der Daten sichergestellt**
- **Durch Einsatz von SQL und Datenbank-Tools (z.B. APEX) werden zusätzliche Datenauswertungen und Datenbearbeitungen ermöglicht**
- **Die bewährten Anwendungen können weiterhin auf dem gewohnten Mainframe-System oder in einer offenen Client/Server-Umgebung laufen**
- **Die Datenbank ermöglicht eine Datenreplikation und eine Daten-Archivierung im laufenden Betrieb**
- **Die Projektdauer wird durch den Einsatz von Werkzeugen für die Migration erheblich verkürzt und die Fehlerrisiken minimiert.**

Verfasser: Dr. Karl-Albrecht Graeber

GSE Graeber Software-Entwicklung

In den Siefen 51

D - 66346 Püttlingen / Saarbrücken

Telefon: +49 (0) 6806 / 306 00 70

Internet: www.g7-migration.de

