

PROJEKTARBEIT

Data Reloc

Im Rahmen der Abschlussprüfung
zum Fachinformatiker für Anwendungsentwicklung
Sommer 2004

Projekt von:

Kai-Sven Bunk

Sandstrasse 2
92353 Postbauer-Heng

Ausbildungsstätte:

KAGO S&C AG

Ausbilder: Pascal Weimerskirch

Kago-Platz 5
92353 Postbauer-Heng

Inhaltsverzeichnis

I. PROJEKTBSCHREIBUNG	3
1. AUSGANGSLAGE.....	3
2. PROJEKTZIEL.....	3
3. PROJEKTSCHNITTSTELLEN.....	3
II. PROJEKTPLANUNG	4
1. ZEIT- UND ABLAUFPLANUNG.....	4
2. RESSOURCENPLANUNG.....	4
3. KOSTENPLANUNG.....	5
III. PROJEKTDURCHFÜHRUNG	6
1. ANALYSE.....	6
a. Ist-Zustand Analyse.....	6
b. Soll-Zustand Analyse.....	8
2. DESIGNPHASE.....	8
a. Auswahl einer Entwicklungsumgebung.....	8
b. Planung der Programmstruktur.....	8
3. CODIERUNG DER ANWENDUNG.....	8
4. TESTPHASE.....	10
5. OPTIMIERUNG.....	10
6. DOKUMENTATION.....	11
IV. PROJEKTABSCHLUSS	11
1. INBETRIEBNAHME.....	11
2. SOLL-IST VERGLEICH.....	11
3. AUSBLICK.....	12
V. SONSTIGE VERZEICHNISSE	13
1. TABELLENVERZEICHNIS.....	13
2. ANHANGVERZEICHNIS.....	13

I. Projektbeschreibung

1. Ausgangslage

Für das Speichern der Daten aus dem Warenwirtschaftsprogramm der Kago-Kamine-Kachelofen GmbH & Co KG werden mehrere rationale Datenbanken auf verschiedene Datenbanksysteme eingesetzt. Die Firma KAGO S&C AG (KAGO Software and Consulting AG) ist beauftragt mit der Instandhaltung und Verwaltung des Warenwirtschaftsprogramms und der vorhandenen Datenbanksysteme. Eine wichtige Rolle spielt dabei der schnelle und reibungslose Import/Export von großen Datenmengen in verschiedenen Formaten bzw. Datenbanksystemen.

Nur mit viel Aufwand lassen sich zufriedenstellende Importe und Exporte durchführen. Importe und Exporte sind derzeit auf einer ORACLE Umgebung begrenzt. Ein datenbanksystemübergreifendes Import/Export ist nicht möglich. Zudem entsteht bei fehlerhaften Daten durch mehrmalige Import/Exportversuche ein hoher Zeitverlust, da die Software vor der Ausführung die Daten nicht auf Vollständigkeit und Richtigkeit prüft.

2. Projektziel

Durch eine neu entwickelte Applikation soll das Problem gelöst und sichergestellt werden, dass von einer beliebigen Datenquelle bzw. Datenbanksystem schnell, einfach und sicher Daten importiert bzw. exportiert werden können.

Ein automatisch erstelltes Protokoll soll sämtliche Operationen dokumentieren, um mögliche Fehler und Warnungen leicht ersichtlich machen zu können. Die nicht vollständig importierten Daten können so nachbearbeitet und neu importiert werden. Die Anwendung wird somit an die betrieblichen Vorgänge exakt angepasst.

3. Projektschnittstellen

Die komplette Entwicklung und Dokumentation der Anwendung mit den Hauptfunktionen wird im Rahmen dieses Projekts durchgeführt. Eine Weiterentwicklung der Anwendung mit Zusatzfunktionen ist nach Bedarf möglich.

II. Projektplanung

1. Zeit- und Ablaufplanung

Für die Durchführung dieses Projekts wurde ein Zeitaufwand von 72 Stunden vorgegeben. Eine Projektarbeitszeit von täglich fünf bis sechs Stunden wurde angenommen, wobei eine Stunde täglich für Tätigkeiten außerhalb des Rahmen dieses Projektes reserviert wurde. Daraus ergibt sich eine Projektlaufzeit von zwölf bis fünfzehn Tagen, die nach Genehmigung des Projektes stattfinden soll.

Der Projektablauf wurde in Projektphasen und Teilaufgaben gegliedert und ist wie folgt geplant:

Projektphase	Teilaufgabe	geschätzte Dauer
Analyse	Ist-Aufnahme	2 Std.
	Soll-Zustand Analyse	1 Std.
Designphase	Auswahl einer Entwicklungsumgebung	1 Std.
	Planung der Programmstruktur	10 Std.
Durchführung	Codierung der Anwendung	23 Std.
	Testen der Anwendung	8 Std.
	Optimierung der Anwendung	6 Std.
	Projektdokumentation erstellen	18 Std.
Inbetriebnahme	Präsentation und Schulung	2 Std.
	Übergabe	1 Std.
Gesamtzeitaufwand:		72 Std.

[Tabelle 2.1 Zeit- und Ablaufplanung]

2. Ressourcenplanung

Zur Entwicklung der Anwendung wird ein Arbeitsplatzrechner mit einer Entwicklungsumgebung und eine Testumgebung mit verschiedenen Datenbanksystemen benötigt.

Folgende Entwicklungsumgebungen stehen zur Verfügung:

- Delphi 7 Professional
- Microsoft Visual Studio .NET
- ORACLE Forms and Reports 6i

Als Arbeitsplatzrechner wird ein „Pentium IV“ mit dem Betriebssystem „Windows XP“ eingesetzt. Zur Dokumentation wird Microsoft Word 2002 und Visio benutzt. Zum Testen der Anwendung stehen folgende Datenbanksysteme zur Verfügung:

- Oracle 9i und 8i
- BDE (Borland Database Engine) 5.11
- PostgreSQL 7.3
- My SQL 4.0.18

Das Projekt wird komplett von einer Arbeitskraft realisiert.

3. Kostenplanung

Außer den Lohnkosten fallen keine zusätzlichen Kosten für dieses Projekt an. Die benötigten Ressourcen sind bereits vorhanden. Damit erweist sich das Projekt als sehr kostengünstig.

III. Projektdurchführung

1. Analyse

a. Ist-Zustand Analyse

Um ein sehr gutes Projektergebnis zu erzielen wird die momentane Situation in der KAGO S&C AG genau analysiert.

Dazu müssen folgende Punkte ermittelt werden:

- Welche Datenbanksysteme werden eingesetzt und verwaltet?
- Welche Betriebssysteme werden eingesetzt?
- Welche Software wird momentan benutzt, die Daten der verschiedenen Datenbanksysteme importiert und exportiert?
- Welche Datentypen werden in den verschiedenen Datenbanksystemen verwendet?

Nach einer genauen Analyse der oben genannten Punkten wurden die gesammelten Informationen wie folgt ermittelt:

Datenbanksysteme

Datenbanksystem	Betriebssystem	Version
ORACLE	Windows NT, Sun Solaris	9i / 8i
PostgreSQL	Linux RedHat Enterprise	7.3
My SQL	Linux RedHat Enterprise	4.0.18
BDE (Borland Database Engine)	Windows 98, NT, 2000, XP	5.11

[Tabelle 3.1.a Ist-Zustand Datenbanksysteme]

Datentypen

Datentyp	Beschreibung
String	Zeichenkette
Smallint	16-bit Ganzzahl
Integer	32-bit Ganzzahl
Word	16-bit unsignierte Ganzzahl
Boolean	Boolesch (wahr oder falsch)
Float	Dezimalzahl

Currency	Wahrung
BCD	Binarcodierte Dezimalzahl die ohne Verlust der Genauigkeit in Wahrung umgewandelt werden kann
Date	Datum
Time	Zeit
DateTime	Datum und Zeit
AutoInc	Selbst inkrementierende 32-bit lange Ganzzahl
Memo	Text
WideString	Lange Zeichenkette
Largeint	Groe Ganzzahl
TimeStamp	Zeitstempel
FMTBcd	Binarcodierte Dezimalzahl zu gro fur BCD

[Tabelle 3.1.b Ist-Zustand Datentypen]

Software

Software	Datenbanksystem	Betriebssystem	Version
SQL Loader	ORACLE	Windows 2000 Windows XP	9.2.0.1.0/ 8.1.7.0.0
SQL Navigator	ORACLE	Windows 2000 Windows XP	4.1
My PHP Admin	My SQL	Web-Anwendung (Linux Webserver)	2.5.6
PDAdmin	PostgreSQL	Windows 2000 Windows XP	1.3.0
SQL Explorer	BDE	Windows 98 Windows NT Windows 2000 Windows XP	4

[Tabelle 3.1.c Ist-Zustand Software]

Mit der eingesetzten Software konnen Importe/Exporte durchgefuhrt werden, es ist jedoch umstandlich und zeitaufwandig. Es ist nicht moglich Daten direkt von einem Datenbanksystem zum anderen zu importieren bzw. exportieren. Es ist notig Daten zuerst in ein allgemein unterstutztes Format zu exportieren, um es anschlieend in ein anderes Datenbanksystem einfugen zu konnen.

Diese standigen Konvertierungen fuhren zu einem hohen Datenverlustrisiko, da Konvertierungsfehler entstehen konnen.

b. Soll-Zustand Analyse

Import/Export Verfahren sollen einfacher, schneller und sicherer durchgeführt werden können. Eine Anwendung soll es möglich machen Daten direkt von einem Datenbanksystem zum anderen zu importieren bzw. exportieren. Import/Export Prozeduren sollen gespeichert werden, um Datensätze schneller synchronisieren zu können.

2. Designphase

a. Auswahl einer Entwicklungsumgebung

Zur Auswahl der Entwicklungsumgebung wurden folgende Kriterien berücksichtigt:

- Kurze Entwicklungszeit
- Hohe Entwicklungs- und Erweiterungsmöglichkeiten
- Kompatibilität mit den vorhandenen Datenbank- und Betriebssystemen

Als Entwicklungsumgebung wurde Delphi Professional 7 ausgewählt, da dies den Auswahlkriterien am besten entspricht.

b. Planung der Programmstruktur

Nach Ermittlung des Soll-Zustandes und Auswahl einer Entwicklungsumgebung wurden Anforderungen und benötigte Funktionen der Anwendung festgelegt. Da eine objektorientierte Programmierung zu realisieren war, wurden die erforderlichen Klassen geplant und benötigte Funktionen den Klassen zugeordnet. Programmablaufpläne wichtiger Prozeduren und Funktionen wurden anschließend erstellt.

Die Planung der Programmstruktur wurde in einer Vorstudie dokumentiert und befindet sich im Anhang dieser Projektdokumentation (Anhang A).

3. Codierung der Anwendung

Nach Planung der Programmstruktur wurde mit dem Codieren der Anwendung in der ausgewählten Entwicklungsumgebung begonnen. Als Vorlage diente die Planung der Programmstruktur, jedoch sind geringe Abweichungen vorgenommen worden um eine effizientere Anwendung zu produzieren.

Codieren der Klasse „TDBSession“

Begonnen wurde mit der Codierung der Klasse „TDBSession“. Sie stellt die Verbindung zu einer beliebigen Datenbank zur Verfügung. Um die verschiedenen Datenbanksystemen unterstützen zu können, wurde die Klasse „TADOConnection“, die bereits in Delphi vorhanden ist, genutzt. Die Klasse „TADOConnection“ wurde als geschütztes Attribut in „TDBSession“ eingefügt und kann ausschließlich von „TReloc“ und „TRepFile“ benutzt werden. Veröffentlicht wurden nur Methoden zur Verbindung und Trennung einer Datenbank sowie zur Erstellung der Verbindungsparameter.

Codieren der Klasse „TReloc“

Die Klasse enthält Methoden zur Prüfung und Durchführung von Import/Export Verfahren. „TReloc“ enthält eine Verbindung zu einer Quelldatenbank und einer Verbindung zur Zieldatenbank. Die Verbindung zu den Datenbanken wird von „TDBSession“ gesteuert. Protokollnachrichten und nicht importierte Datensätze werden anhand von Parameter in Ereignissen ausgegeben.

Für erfolgreiche Import/Export Verfahren unterstützt „TReloc“ 37 verschiedene Datentypen. Dies ergibt eine Anzahl von 1369 verschiedenen Kombinationen Daten zu konvertieren. „TReloc“ ermöglicht es für jede dieser Kombinationen eine bestimmte Konvertierungsfunktion einzusetzen. Des Weiteren ermittelt „TReloc“, ob eine Konvertierung durchführbar ist und gibt Warnungen bei möglichem Datenverlust oder sonstigen kritischen Operationen.

Da die Entwicklung von dieser Anzahl von Funktionen im Zeitrahmen dieses Projektes nicht möglich ist, wurde erstmals für alle Kombinationen eine einfache Kopierfunktion eingesetzt. Anschließend wurden die Funktionen für die momentan erforderlichen Datentypkonvertierungskombinationen entwickelt und eingesetzt. Die Weiterentwicklung dieser Funktionen wird nach Bedarf fortgesetzt.

Codieren der Klasse „TRepFile“

Zur Verwaltung mehrerer Datenbanksitzungen und Relocs (Import/Export Verfahren) wurde die Klasse „TRepFile“ programmiert. Sie enthält Methoden zum Laden und Speichern der eingegebenen Datenbanksitzungen und Relocs. Datenbanksitzungen und Relocs können beliebig hinzugefügt und entfernt werden.

Um Datenbanksitzungen eindeutig zu machen, werden von den Relocs nur die Namen der Quell- und Zieldatenbanken gespeichert. „TRepFile“ verknüpft die Datenbanksitzungen mit den Quell- und Zieldatenbanken der Relocs durch deren Namen.

Erstellen der Anwendung

Zuletzt wurden die codierten Klassen in einer Anwendung integriert. Es wurde eine MDI Umgebung (wie z.B. Microsoft Office) als Programmoberfläche ausgewählt. Entwickelt wurde die Anwendung für Benutzer die Grundkenntnisse von SQL und relationalen Datenbanksystemen haben.

Um die Benutzerfreundlichkeit der Anwendung zu erhöhen, wurden außer der in den Klassen vorhandenen Methoden, weitere Funktionen eingebaut. So ist es z.B. für den Benutzer möglich, vorhandene Tabellen und Daten einer Datenbank zu sehen und SQL Abfragen auszuführen.

Protokollnachrichten werden sofort sichtbar gemacht und können in verschiedenen Formaten gespeichert werden. Benutzer können Import/Export Operationen in Echtzeit verfolgen. Beliebige SQL Abfragen, sowie nicht importierte Datensätze von Import/Export Verfahren werden als CSV Dateien exportiert.

4. Testphase

Während der Codierung wurde die Anwendung auf einzelne Funktionen getestet. Um ein Gesamtergebnis der Anwendung zu erhalten, wurde in der Testphase eine Testumgebung vorbereitet in dem alle verwendeten Datenbanksysteme eingerichtet wurden. Anschließend wurden mehrere Import/Export Operationen durchgeführt.

Die Testergebnisse befinden sich im Anhang dieser Projektdokumentation (Anhang B).

5. Optimierung

Anhand der Testergebnisse wurde die Anwendung überarbeitet und optimiert. Folgende Änderungen wurden vorgenommen:

- Zur Ermittlung der vorhandenen Tabellen in ein ORACLE Datenbanksystem wurde eine SQL Abfrage benutzt. Überflüssige Systemtabellen werden nicht mehr angezeigt.
- Bei Import/Export Operationen wird das Einfügen eines Datensatzes nur bei Fehlern oder Warnungen protokolliert.
- Quelle und Ziel sind in einem einzigen Formular sichtbar und können direkt aus der „Reloc“ Maske bearbeitet werden.

6. Dokumentation

Nach Fertigstellung der Anwendung wurden die entwickelten Klassen ausführlich dokumentiert. Die Klassendokumentation befindet sich im Anhang dieser Projektdokumentation (Anhang C).

Die Projektdokumentation wurde im Laufe der Projektlaufzeit erstellt und gemäß den Projektablauf erweitert.

IV. Projektabschluss

1. Inbetriebnahme

Nach Abschluss der Entwicklungsarbeiten wird „Data Reloc“ den Benutzern vorgestellt. In einer Präsentation werden die Benutzer mit der Anwendung vertraut gemacht.

Abschließend wird „Data Reloc“ auf den Arbeitsplatzrechner der Benutzer installiert. Die Inbetriebnahme verläuft erfolgreich, da alle Anforderungen umgesetzt wurden.

2. Soll-Ist Vergleich

Das Projekt konnte erfolgreich umgesetzt werden, da alle Projektziele erfüllt wurden. „Data Reloc“ ermöglicht es Daten mit höherer Effizienz direkt von einer Datenbank zur anderen importieren/exportieren. Dies ist auch möglich, wenn die Quell- und Zieldatenbank auf verschiedene Datenbanksystemen laufen. Protokolle dokumentieren sämtliche Import/Export Verfahren und fehlerhafte Importe können ausgefiltert und reimportiert werden.

Das Projekt konnte innerhalb der geplanten Zeit durchgeführt werden. Eine Zeitabweichung zur Zeitplanung von $-2,78\%$ wurde festgestellt. In der folgenden Tabelle wird die geplante und tatsächliche Dauer der Projektphasen gegenübergestellt.

Projektphase	Teilaufgabe	geplante Dauer	tatsächliche Dauer
Analyse	Ist-Aufnahme	2 Std.	1 Std.
	Soll-Zustand Analyse	1 Std.	1 Std.
Designphase	Auswahl einer Entwicklungsumgebung	1 Std.	1 Std.
	Planung der Programmstruktur	10 Std.	8 Std.
Durchführung	Codierung der Anwendung	23 Std.	24 Std.
	Testen der Anwendung	8 Std.	11 Std.
	Optimierung der Anwendung	6 Std.	6 Std.
	Projektdokumentation erstellen	18 Std.	16 Std.
Inbetriebnahme	Präsentation und Schulung	2 Std.	1 Std.
	Übergabe	1 Std.	1 Std.
Gesamtzeitaufwand:		72 Std.	70 Std.

[Tabelle 4.2 Soll-Ist Vergleich]

3. Ausblick

„Data Reloc“ wurde nach Abschluss der Entwicklung erfolgreich eingesetzt. Nach Bedarf werden die Programmfunktionen erweitert. Die Entwicklung der Funktionen für die verschiedene Konvertierungsmöglichkeiten sind vorgesehen. Weitere Funktionen zur Datenverarbeitung wie Datenaktualisierung werden in Betracht gezogen.

Zudem wird auch die Möglichkeit bestehen, einer Demoversion (25 Datensätze) sowie einer kostenpflichtigen Vollversion zum Download online zur Verfügung zu stellen.

V. Sonstige Verzeichnisse

1. Tabellenverzeichnis

Tabelle 2.1 Zeit- und Ablaufplanung.....	4
Tabelle 3.1.a Ist-Zustand Datenbanksysteme.....	6
Tabelle 3.1.b Ist-Zustand Datentypen.....	6
Tabelle 3.1.c Ist-Zustand Software.....	7
Tabelle 4.2 Soll-Ist Vergleich.....	12

2. Anhangverzeichnis

Vorstudie	Anhang A
Testergebnisse	Anhang B
Klassendokumentation	Anhang C
Einverständniserklärung	Anhang D