



Ein objektorientierter Ansatz zur Pflege relationaler Datenbanken

Thomas Nindel

Betreuer: Hans J. Günther



Bibliografische Beschreibung und Autorenreferat

NINDEL, Thomas, Berufsakademie Sachsen, Staatliche Studienakademie Dresden, Studienrichtung Informationstechnik, Praxisarbeit 3. Semester, 17 Seiten, 2 Literaturquellen, 2 Anlagen.

Anhand einer XML-basierten Lösung für die Pflege von Produktdaten eines Onlineshops wird die hervorragende Eignung objektorientierter Datenspeicherung für Datenpflege - auch für die Pflege von relationalen Datenbanken allgemein - gezeigt.



1.	Einführung	5
1.1.	Abstract	6
1.2.	Aufgabenstellung	6
2.	Aufgabenanalyse	6
2.1.	Technische Grundlagen	6
2.1.1.	Objektorientierte Datenspeicherung (OOS)	6
2.1.2.	XML als Anwendung der OOS	7
2.1.3.	Relationale Datenspeicherung	8
2.1.4.	SQL als Interface zu Relationalen Datenbanken	8
2.1.5.	Mapping XML <-> SQL	8
2.2.	Zielsetzung	9
2.3.	Zielgruppen	9
2.4.	Anforderungen Allgemein	10
2.4.1.	Performance	10
2.4.2.	Benutzbarkeit	10
2.4.3.	Datenkonsistenz	10
2.5.	Ist-Analyse	10
2.6.	Soll-Analyse	11
2.7.	Materialien	11
2.8.	To-Do Liste	11
3.	Projektrealisierung	12
3.1.	Wahl des Ansatzes	12
3.1.1.	Pflege durch Tabellen	12
3.1.2.	Pflege durch XML - Zwischenformat	12
3.2.	Entscheidung	13
3.2.1.	Argumentation aus Technologischer Sicht	13
3.2.2.	Argumentation aus organisatorisch - wirtschaftlicher Sicht (Make or Buy)	13
3.3.	Implementation	13
4.	Diskussion	14
4.1.	Einsatz objektorientierter Datenrepräsentation zur Pflege	14
4.2.	Fazit	15





Abkürzungsverzeichnis

OOS	Objektorientierte Datenspeicherung
RDB	Relationale Datenbank
SQL	Structured Query Language
XML	Extensible Markup Language
XQL	Extensible Query Language



1. Einführung

1.1. Abstract

Anhand einer XML-basierten Lösung für die Pflege von Produktdaten eines Onlineshops wird die hervorragende Eignung objektorientierter Datenspeicherung für Datenpflege - auch für die Pflege von relationalen Datenbanken allgemein - gezeigt.

1.2. Aufgabenstellung

Für die Pflege der Datenbank eines Onlineshops soll eine effektive und nutzerfreundliche Lösung geschaffen werden. Dabei soll die Verwendung von XML als Datenformat für die Datenpflege auf seine Verwendbarkeit hin geprüft und gegebenenfalls eine entsprechende Lösung implementiert werden.

2. Aufgabenanalyse

2.1. Technische Grundlagen

2.1.1. Objektorientierte Datenspeicherung (OOS)

Die mathematische Grundlage der objektorientierten Datenspeicherung (nachfolgend kurz OOS genannt) ist die Speicherung von Daten als gerichteter Graph, genauer als Baum. Daten werden als Objekte und deren Attribute aufgefasst, wobei Attribute wiederum Objekte oder Objektlisten sein können. Auf diese Weise lassen sich 1:n Beziehungen von Entitäten realisieren. Mehrfachrelationen (n:m) können durch Datenredundanz oder einen Linkmechanismus dargestellt werden.

Durch die Implizierte Darstellung von Relationen als Baum erreicht man so eine implizierte semantische Integrität.

2.1.2. XML als Anwendung der OOS

XML ist eine Anwendung der OOS. Am folgenden Beispiel will ich kurz die Implementation des objektorientierten Datenmodells in XML erläutern.

```
<shop>
<produktgruppe name="lampen">
<produkt name="Stehlampe" preis="10.00€" farbe="beige">
</produkt>
<produkt name="Tischlampe" preis="15.00€" farbe="rot">
</produkt>
</produktgruppe>
</shop>
```

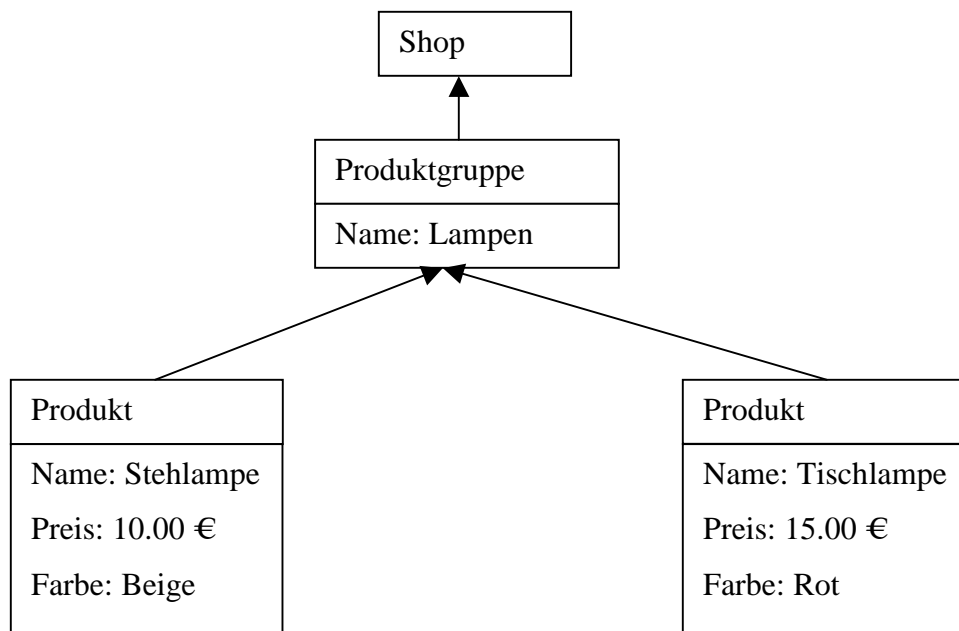


Abb. 1: Beispieldokument als Objektbaum

Abfragen im Dokument und eine Navigation im Objektbaum sind über die Abfragesprache XQL möglich. Um z.B. das Produkt „Tischlampe“ im Beispiel zu erreichen formuliert man folgende XQL – Abfrage:



```
/Produktgruppe[ @Name="Lampen" ]/Produkt[ @Name="Tischlampe" ]
```

XML als Implementation der OOS ist auf vielen Systemen und für viele Programmiersprachen als DOM – Implementationen (Level 1 und 2) verfügbar. XML eröffnet außerdem eine einfache Variante des Publishing (vgl. Praxisarbeit 1. Semester). Serversysteme und XML-kompatible Datenbanksysteme sind in großer Anzahl verfügbar. XML ermöglicht auch dezentrale Datenspeicherung (Data Warehouse).

2.1.3. Relationale Datenspeicherung

Die mathematische Grundlage der relationalen Datenspeicherung (im folgenden RDS genannt) sind Relationen auf einer Menge von Tupeln. Diese Relationen werden durch explizite Zuordnung durch Primär-, Sekundär- und Fremdschlüssel realisiert. Somit wird die semantische Integrität manuell hergestellt.

Relationale Datenbanksysteme sind in allen Preis- und Größenklassen für viele Systeme verfügbar.

2.1.4. SQL als Interface zu Relationalen Datenbanken

Die Abfrage von Informationen aus Relationalen Datenbanken (RDB) ist mit der Sprache SQL Standardisiert worden. Grundlegende Befehle dieser Sprache beinhalten das Auslesen von Daten („SELECT“), das Einfügen von Daten („INSERT“), das Aktualisieren von Daten („UPDATE“) und das Löschen von Daten („DELETE“). Eine gezielte Auswahl erfolgt dabei durch Spezifizierung einzelner Attribute („WHERE“).

2.1.5. Mapping XML <-> SQL

Die folgenden Tabellen Repräsentieren die Darstellung der Beispieldaten in einer relationalen Datenbank.

ID	Name
1	Lampen

Tabelle 1: Produktgruppen



ID	Produktgruppe	Name	Preis	Farbe
1	1	Stehlampe	10.00 €	Beige
2	1	Tischlampe	15.00 €	Rot

Tabelle 2: Produkte

Die oben beispielhaft genannte XQL - Abfrage müsste in SQL so formuliert werden:

```
SELECT * FROM Produkte WHERE Name LIKE "Tischlampe" AND  
Produktgruppe=1
```

Eine Äquivalenz von OOS und RDS ist durch so genanntes Mapping möglich. Dabei werden 1:n Relationen durch Objekthierarchien abgebildet – eine Tabelle entspricht dann einer Ebene im XML-Baum. Durch die implizierten Relationen entfällt die Zuordnung durch Primär- und Fremdschlüssel und damit die Notwendigkeit von IDs. Diese können bei der Rücktransformation automatisch generiert werden.

Eine Abbildung von n:m-Relationen ist durch Einführung von Datenredundanz möglich.

2.2. Zielsetzung

Ziel des Projektes ist die Realisierung einer intuitiven und einfachen Pflegemöglichkeit einer relationalen Datenbank. Die Datendarstellung und das Editieren sollen transparent sein – die zugrunde liegende Datenbank soll nicht sichtbar sein.

2.3. Zielgruppen

Zielgruppe dieser Pflegeapplikation sind Betreiber eines Onlineshops, die die Datenpflege vereinfachen und effektiveren wollen. Für die Datenpflege sollen keine Spezialkenntnisse, wie die Theorie relationaler Datenbanken, nötig sein. Allerdings wird die Kenntnis der Struktur der zugrunde liegenden Daten vorausgesetzt (Hierarchie der Produktstruktur des Shops).



2.4. Anforderungen Allgemein

Das Pflegesystem soll die Möglichkeit der Erstellung und Wartung mehrerer Gliederungsebenen von Produkten bieten. Produktvariationen, z.B. Größen und Farben, sollen unterstützt werden. Des Weiteren sollen externe Ressourcen wie z.B. Bilder unterstützt werden. Dateninkonsistenz darf bei der Datenpflege nicht auftreten. Eine Möglichkeit, Backups der Datenbasis auf einfache Art und Weise zu Erstellen, ist durch das System zu realisieren.

2.4.1. Performance

Das Pflegesystem muss beim Aktualisieren von Daten sowie bei Backups und Restores eine ausreichende Performance bieten, eine entsprechend schnelle Datenanbindung zum Datenbankserver wird vorausgesetzt.

2.4.2. Benutzbarkeit

Das Pflegesystem soll eine die Produktstruktur repräsentierende Datensicht, am besten baumförmig, bieten. Die Benutzung soll intuitiv sein und darf keine Spezialkenntnisse erfordern.

2.4.3. Datenkonsistenz

Bei jedem Pflegeschritt der Shopdaten muss eine Konsistenz der Daten gewährleistet werden. Auch während der Pflege muss die produktive Datenbank des Shops verfügbar sein. Deshalb wird eine Offline-Variante bevorzugt. Probleme des Multiuserzugriffs auf die Daten sollen gelöst oder umgangen werden.

2.5. Ist-Analyse

Vorhandene Pflegemöglichkeiten beschränken sich auf



- Die manuelle Pflege durch Einsatz von SQL
- Die Pflege durch Einsatz eines Datenbankfrontends
- Die Pflege durch ein Online–Pflegeinterface welches für jeden Shop separat erstellt werden muss
- Die Pflege durch ein vom eingesetzten Webshop-System im Lieferumfang enthaltenem Pflgetool

2.6. Soll-Analyse

Das Pflegesystem soll einen Editor mit geeigneter Datenansicht, die gegebenenfalls erforderliche Software zum Auslesen und Rückschreiben der Daten aus der relationalen Datenbank und eine Backupmöglichkeit enthalten. Des Weiteren sollen Maßnahmen zur automatischen Erhaltung der Datenkonsistenz getroffen werden.

2.7. Materialien

- Borland Delphi 5 Professional
- Implementation des DOM Level 2 für Delphi
- mySQL – Zugriffskomponenten für Delphi
- Vorhandener Webshop „Magic-Dessous“
- Shopdatenbank als mySQL
- Webserver mit PHP-Unterstützung
- Literatur

2.8. To-Do Liste

- Erstellen eines Pflegesystems für einen Onlineshop
- Wahl eines geeigneten Ansatz
- Editieren der Daten in geeigneter Ansicht – favorisiert wird eine Baumansicht
- Möglichkeit von Information-Hiding im View



- Offlinepflege realisieren
- Konsistenzwahrung während der Pflege / Updates
- Beachtung der Probleme eines Multiuserzugriff
- Backupmöglichkeit
- Komfort durch Drag & Drop, Zwischenablage
- Trennung in Datenkonvertierung und Dateneditieren
- Beliebige Strukturierung der Daten
- Möglichkeit der Erweiterung der Datenstruktur

3. Projektrealisierung

3.1. Wahl des Ansatzes

3.1.1. Pflege durch Tabellen

Eine Pflege durch ein tabellenbasiertes Programm oder direktes Ausführen von SQL-Kommandos ist in vielerlei Hinsicht problematisch:

Eine Möglichkeit für die Offline-Datenpflege ist nur schwierig realisierbar, da eine komplette Instanz der Datenbank offline verfügbar gemacht werden muss. Eine versehentliche Änderung eines Primär- oder Sekundärschlüssels kann die Konsistenz der gesamten Datenbank verletzen und diese somit unbrauchbar machen. Außerdem repräsentiert die Darstellung der Daten in Tabellenform nicht die dem Shop zugrunde liegende Produktstruktur.

Vorteilhaft ist der mit Einsatz eines Datenbankfrontend mit Formularen verbundene Komfort der Datenänderung und -suche.

3.1.2. Pflege durch XML - Zwischenformat

Die Pflege durch ein XML-Zwischenformat bietet die Vorteile von objektorientierter Datenspeicherung:

Wahrung der Datenkonsistenz durch Implizieren der Relationsdarstellung, eine die Produktstruktur des Shops repräsentierende Ansicht durch einen Baum, eine intuitive und



nutzerfreundliche Navigation durch die Baumansicht - inklusive Information Hiding. Die dargestellten Daten sind vollständig transparent und von der Datenspeicherung entkoppelt - somit ist keine Kenntnis über diese erforderlich. Offlinepflege ist durch die Verfügbarkeit aller Daten in einem XML-File leicht möglich.

3.2. Entscheidung

3.2.1. Argumentation aus Technologischer Sicht

Die entscheidenden Vorteile der Lösung über ein XML-Zwischenformat, wie die Erhaltung der Datenkonsistenz, Transparenz, Möglichkeit einer Offlinelösung und die Verwendbarkeit für Backups rechtfertigen die Implementation dieser Lösung. Deshalb wird sie realisiert.

3.2.2. Argumentation aus organisatorisch - wirtschaftlicher Sicht (Make or Buy)

Für das Erstellen der Lösung spricht:

Es existiert kein vergleichbares Pflegesystem für relationale Datenbanken. Das System, richtig angelegt und modularisiert, ist vielfach verwendbar. Das Portfolio der Firma fairnet medienagentur kann durch die Software erweitert werden und die Software ist als Entwicklerversion komplett weiterverkaufbar.

Gegen die Entwicklung spricht der relativ hohe Zeitaufwand von ca. 60 Mannstunden.

Ein Einkaufen der Lösung entfällt wegen der Nichtverfügbarkeit einer vergleichbaren Software.

3.3. Implementation

Besonderheiten

Das Pflegesystem mit XML-Zwischenformat wird in 3 Teile aufgespalten. Eine Applikation liest die SQL-Datenbank aus und generiert die Datenbasis im XML-Format. Ein XML-Editor



realisiert die Datenpflege. Eine weitere Applikation erledigt die Rücksynchronisierung der XML-Daten mit der SQL-Datenbank.

Der XML-Editor ist hinsichtlich Information-Hiding, Informationsrepräsentation und Strukturdefinition durch eine spezielle Regeldatei anpassbar. Sie erlaubt die Anpassung der Informationsdarstellung im Baum durch Übersetzen der XML-Syntax in menschliche Sprache, limitiert das Erstellen neuer Zweige auf die durch das Shopformat vorgegebenen Grenzen, hilft bei der Strukturerstellung durch Auswahl der erlaubten Unterknoten und kann die XML-Daten validieren.

Der Editor enthält komfortable Features wie Drag-and-Drop und eine Zwischenablage.

Die Applikationen für die Datensynchronisierung kümmern sich um die Erhaltung der Datenkonsistenz der RDB durch das automatische Kreieren und Aktualisieren von Primär- und Fremdschlüsseln.

Probleme

Das Löschen von Information aus der XML-Struktur muss aufgezeichnet werden, um so die Daten auch in der SQL-Datenbank löschen zu können. Die mit Multiuserzugriff verbundenen Probleme wurden durch das Sperren der Datenbank nach Datenexport und das Lösen der Sperre bei Rücksynchronisierung umgangen.

4. Diskussion

4.1. Einsatz objektorientierter Datenrepräsentation zur Pflege

Der Komfort der Pflege von Daten steht und fällt mit dem Abstraktionsgrad der Informationsdarstellung in digitaler Form. Objekte als unmittelbare Abstraktion bieten eine gute Verständlichkeit und einen direkten Bezug zu den Daten. Die Darstellung von Daten in



objektorientierter Art und Weise im XML-Format bietet durch die an die Realität angelehnte Darstellungsweise eine sehr gute Möglichkeit der Dateneinsicht und -pflege.

Das XML-Format glänzt durch Vielseitigkeit hinsichtlich Kompatibilität, Portierbarkeit, Erweiterbarkeit und als Datenbasis für medienübergreifendes Publishing.

4.2. Fazit

Bei der Realisierung der Pflegeapplikation zeigte sich, wie oben erwähnt, die hervorragende Eignung des XML-Formates für diesen Zweck. Aus theoretischer Sicht auf die Form Datenspeicherung ist die Konsistenzerhaltung der Daten ein fulminantes Feature. Aus Sicht von System- und Datenbankadministratoren gefällt die Eignung als Backuplösung. Der Softwareentwickler freut sich über Erweiterbarkeit und Wiederverwendbarkeit der Lösung und die Nutzer des Editors erfreuen sich an der Einfachheit des Editierens komplexer Daten.

Nachteilig allerdings ist die umständliche Umsetzung von n:m - Relationen, was aber durch eine intelligente Datenstrukturierung vor Implementierung der relationalen Datenbasis umgangen werden kann.



Anlage 1: Literaturverzeichnis

[1] Benedikt von Treskow: XML-Datenbanken,

<http://www.ipd.ira.uka.de/~oosem/WebDB0001/ausarbeitung/BenediktVonTreskow.pdf>

[2] Andreas Heuer: Objektorientierte Datenbanken Addison Wesley ISBN 3-89319-800-8

[3] Site des World Wide Web Consortium <http://www.w3c.org>



Anlage 2: Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die Zustimmung der Firma zur Verwendung betrieblicher Unterlagen habe ich eingeholt. Die Arbeit wurde bisher in gleicher oder ähnlicher Form weder veröffentlicht noch einer anderen Prüfungsbehörde vorgelegt.

Dresden,

Thomas Nindel