Grundstudium

Datentransformation bei XML-basierten Geschäftsdokumenten

Prof. Dr.-Ing. Frank-Dieter Dorloff / Dipl.-Wirt.-Inf. Jörg Leukel /
Dipl.-Inf. Volker Schmitz, Essen

Wegen der verschiedenen Datenformate und Dokumenttypen ist das Verarbeiten und Konvertieren von Daten bei elektronischen Marktplätzen schwierig. Mit dem zwischenbetrieblichen Austausch XML-basierter Geschäftsdokumente hat die Transformation von Datenformaten neue Bedeutung erlangt.

1. Datentransformationen

Darstellung ändern, Informationsgehalt erhalten

Unter Datentransformation wird die Umwandlung von Daten, die in einem **Quelldatenformat** vorliegen, in ein **Zieldatenformat** verstanden. Eine solche Umwandlung sollte den Informationsgehalt der Daten nicht verändern oder verringern, sondern nur die Darstellung der Informationen dem jeweiligen Zieldatenformat anpassen. Dies entspricht der Begriffsauffassung der Wirtschaftsinformatik, dass Daten eine für die maschinelle Verarbeitung notwendige Darstellung von Informationen sind. Quell- und Zieldatenformat beschreiben die Art und Weise dieser Darstellung. Sie definieren dabei sowohl die Struktur der Daten — beispielsweise die Anzahl und Anordnung von Datenelementen — als auch die Kodierung der Datenelementwerte anhand von Datentypen und Zeichensätzen.

Das Konzept der Datentransformation hat seinen Ursprung in der Integration unterschiedlicher relationaler Schemata und wird in diesem Zusammenhang auch als **Schema-Integration** oder **Schema-Matching** bezeichnet. Integration heißt hier, dass die unterschiedlichen Schemata (Quelldatenformate) in ein gemeinsames Schema (Zieldatenformat) überführt werden.

Die Einsatzgebiete von Datentransformationen gehen jedoch in Folge der umfassenden Verbreitung von Datenverwaltungssystemen und insbesondere relationaler Datenbanksysteme weit darüber hinaus. Zu nennen sind die Migration von Datenbeständen aus Legacy-Systemen, die Verbesserung der Performance von Datenbanken, der Aufbau von Data Warehouses sowie allgemein die unternehmensweite und -übergreifende Integration von Informationssystemen. Damit sind Datentransformationen zu einem zentralen Bestandteil des Datenmanagements geworden, deren Bedeutung durch den zunehmenden zwischenbetrieblichen Datenaustausch und angesichts der **Parallelität von relationalem, objektorientiertem und XML-Datenmodell** verstärkt wird (vgl. IEEE Computer Society 1999).

Typ- vs. Instanzebene von Daten

Datentransformationen können in eine Entwicklungs- und eine Ausführungsphase unterteilt werden. In der Entwicklungsphase werden die Voraussetzungen geschaffen, damit Datenbestände konvertiert und Transformationen maschinell ausgeführt werden können. Während sich die Ausführung auf konkrete Datenbestände und damit auf die Instanzebene der Daten bezieht, ist für die Entwicklung in der Regel eine Betrachtung der Typebene notwendig, d.h., es sind die Schemata der Quell- und Zieldatenformate zu untersuchen. Dazu werden Beziehungen zwischen inhaltlich korrespondierenden Datenelementen aufgedeckt und die Art der Beziehung definiert.

Dieser Vorgang wird allgemein als Mapping bezeichnet, das im Ergebnis zu einzelnen **Mapping-Definitionen** oder Transformationsanweisungen führt. Eine Mapping-Definition beschreibt in einer strukturierten Form den Zusammenhang zwischen einem oder mehreren Datenelementen des Quell- und Zielformats.

2. Verarbeitung XML-basierter Geschäftsdokumente

XML-Geschäftsdokumente erstellen, übertragen und importieren

Ein **Einsatzgebiet für Datentransformationen** ist die Verarbeitung von XML-Dokumenten, die beispielsweise zwischen Unternehmen ausgetauscht werden — insbesondere Geschäftsdokumente wie z.B. Auftrag, Auftragsbestätigung, Auftragsänderung, Lieferavis und Rechnung. Dieser Datenaustausch vollzieht sich in der Regel in drei Schritten. Das sendende Unternehmen erstellt das zu übermittelnde XML-Dokument auf Basis von betrieblichen Daten, die in einem Anwendungssystem bzw. der zugehörigen Datenbank enthalten sind. Dazu werden Transformationsanweisungen benötigt, die die relevanten Bestandteile des Datenbankschemas auf das jeweilige XML-Austauschformat abbilden. Im zweiten Schritt wird das XML-Dokument zu dem Empfänger unter Nutzung eines Protokolls übertragen (z.B. FTP, HTTP oder Offline-Übertragung). Der Empfänger importiert das XML-Dokument in ein Anwendungssystem gemäß der Mapping-Definitionen, die das XML-Austauschformat auf das Datenbankschema des Anwendungssystems abbilden. Zwar steht im Folgenden die Verarbeitung XML-basierter Geschäftsdokumente im Vordergrund, die Datentransformationskonzepte sind jedoch von allgemeiner Gültigkeit für XML-Dokumente und beziehen sich damit auf jeglichen XML-Datenaustausch.

In der beschriebenen Situation unterscheiden sich die Quell- und Zielformate nicht nur in ihren Schemata. Auch die Unterschiede zwischen relationalem Datenmodell auf der einen und XML-Datenmodell auf der anderen Seite sind zu berücksichtigen. Dies ist nur dann nicht der Fall, wenn sowohl das Quell- als auch das Zielformat bereits auf XML basieren und zwischen beiden eine direkte Konvertierung erfolgen soll. Die Entwicklung von Mapping-Definitionen ist hierbei einfacher, da Formatunterschiede, die sich allein aufgrund divergierender Modellsprachen ergeben, nicht auftreten. Damit können die Entwicklungsarbeiten auf **syntaktische und semantische Unterschiede der Schemata** konzentriert werden. Zur Erfassung und Klassifizierung der auftretenden Schemaunterschiede bietet es sich an, von den zu Grunde liegenden Modellsprachen zu abstrahieren, d.h., die Klassifizierung sollte unabhängig von den spezifischen Eigenschaften des relationalen, objektorientierten bzw. XML-Datenmodells sein. Weiterhin sollten auch Mapping-Definitionen bezüglich sonstiger Dokumentformate des zwischenbetrieblichen Datenaustausches, insbesondere sog. Trennzeichenformate (CSV — Comma Separated Values) ermöglicht werden.

Werkzeuggestützte Entwicklung

Häufig überführen Softwarewerkzeuge für Datentransformationen die in unterschiedlichen Modellsprachen vorliegenden Schemata in eine einheitliche, interne Repräsentation. Hierbei handelt es sich um gerichtete Graphenmodelle, hierarchische Datenmodelle oder hybride Formen, die relationale oder objektorientierte Datenmodelle erweitern. Viele Softwarewerkzeuge verwenden bereits **XML als gemeinsame Modellsprache** für alle Quell- und Zieldatenformate und überführen daher relationale Schemata und CSV-Formate zunächst in eine XML-basierte Repräsentation. Somit lässt sich die Mapping-Definition auf die direkte Konvertierung zwischen zwei XML-Formaten reduzieren. Kennzeichnend für das XML-Datenmodell sind die explizite Auszeichnung aller Datenwerte durch sog. Tags, die Unterscheidung zwischen XML-Tags und XML-Attributen sowie die Möglichkeit zur Hierarchisierung durch geschachtelte Tags. Übergeordnete Tags können so als Container für die darin enthaltenen Tags dienen.

XML-Geschäftsdokumente hoher Komplexität

Einen wichtigen Anwendungsbereich des zwischenbetrieblichen XML-Datenaustausches bilden elektronische Produktkataloge, die von Lieferanten bereitgestellt werden und in B2B-Marktplätzen und einkaufsseitigen Beschaffungssystemen Verwendung finden (vgl. Leukel/Schmitz/Dorloff 2002). Anhand der **Domäne Katalogdaten** lassen sich die grundsätzlichen Probleme beschreiben, die bei der Verarbeitung von und der Konvertierung zwischen XML-Dokumenten auftreten. Dazu wird nachfolgend auf exemplarische Fälle aus den XML-Katalogstandards BMEcat 1.2, cXML 1.2.009 und xCBL 4.0 zurückgegriffen (vgl. Schmitz/Kelkar/Pastoors 2001; Ariba 2003; CommerceOne 2003). Einerseits weisen Katalogdaten und die genannten Standards eine im Vergleich zu anderen Geschäftsdaten hohe Komplexität auf, die sich am Umfang der Formatspezifikationen, der Anzahl der Datenelemente und inhaltlichen Abhängigkeiten der Datenelemente untereinander zeigt. Andererseits sind XML-Formate für den Katalogdatenaustausch bereits von sehr hoher Bedeutung, sodass die Beschränkung auf XML als alleiniges Datenmodell kein Nachteil ist.

3. Mapping-Typen

Vier Grundtypen

In Abhängigkeit von der Anzahl der Datenelemente, die im Quell- und Zielformat an einer Mapping-Definition beteiligt sind, lassen sich die vier Grundtypen **1:1-, 1:N-, N:1- und N:M-Mapping** bilden. Die Typisierung basiert also auf der Kardinalität des Mapping.

Eine Mapping-Definition wird generell so vorgenommen, dass die Datenelemente des Zielformats vollständig erfasst werden, d.h., es ist zu fragen, welche Schritte zur Erfüllung der Anforderungen des Zielformats notwendig sind. Die Entwicklungsreihenfolge für diese Definitionen kann sich jedoch auch an anderen, insbesondere inhaltlichen Kriterien orientieren. Im Ergebnis handelt es sich bei den Mapping-Definitionen um eine deklarative Beschreibung, was bei der Konvertierung zu transformieren ist, nicht jedoch, wie die Transformationen auszuführen sind (prozedurale Beschreibung).

3.1. Das 1:1-Mapping

Einfachster Fall

Beim 1:1-Mapping besteht eine Beziehung zwischen genau einem Datenelement des Quellformats und genau einem Datenelement des Zielformats. Die zugehörige Datentransformation bildet den Informationsgehalt des Quell- auf das Zieldatenelement ab. Dieser Fall tritt dann auf, wenn ansonsten übereinstimmende Datenelemente unterschiedlich benannt sind oder sich in der Wertkodierung unterscheiden. Bezogen auf XML-Dokumente ist außerdem die **Einordnung der Datenelemente in der Dokumentstruktur** von Bedeutung, die es erlaubt, Datenelemente auf mehreren Hierarchieebenen anzuordnen.

Vor allem standardisierte XML-basierte Austauschformate nutzen das Instrument der Hierarchisierung, um logisch zusammenhängende Datenelemente eines Datenobjektes zu gruppieren. In der Folge dient ein erheblicher Anteil der Tags nicht zur Aufnahme von Datenwerten, sondern zur Gruppierung von weiteren Tags, die wiederum Tags oder tatsächlich Datenwerte enthalten. Ferner ist zu beachten, dass auf allen Hierarchiestufen **Tags mit XML-Attributen** angereichert werden können. Damit können die enthaltenen Tags bzw. Datenwerte typisiert und in ihrer Bedeutung näher beschrieben werden. Unabhängig von den Entscheidungskriterien für die Verwendung von Tags und Attributen ist bei der Konvertierung die Möglichkeit zu berücksichtigen, dass der im Quellformat über ein Attribut wiedergegebene Sachverhalt im Zielformat als Tag beschrieben wird (und umgekehrt).

Ein Beispiel demonstriert anhand von drei Tags eines Produktdatensatzes, der im Quellformat xCBL 4.0 vorliegt, die Konvertierung in das Zielformat BMEcat 1.2:

Quellformat xCBL 4.0:

```
<Product>
  <Action value="Add"></Action>
  <ProductID Type="Supplier">2002471</ProductID>
  <ProductName> Herkules 500 S</ProductName>
  ...
</Product>
```

Zielformat BMEcat 1.2:

```
<ARTICLE mode="new">
  <SUPPLIER_AID>2002471</SUPPLIER_AID>
  <ARTICLE_DETAILS>
    <DESCRIPTION_SHORT>Herkules 500 S</DESCRIPTION_SHORT>
    ...
  </ARTICLE_DETAILS>
  ...
</ARTICLE>
```

Abb. 1: Beispiel XML-Daten für das 1:1-Mapping

Ausprägungen des 1:1-Mapping

In diesem Beispiel können folgende Ausprägungen des 1:1-Mapping festgestellt werden:

- Dem Tag Product als Container für die Produktdatensätze entspricht in BMEcat das Tag ARTICLE.
- Das Tag Action besitzt anstatt eines Datenwertes nur ein Attribut value mit der Belegung Add, die sich im Zielformat im Attribut mode des Tags ARTICLE wieder findet und dort mit dem Wert new kodiert ist. Das **Mapping betrifft XML-Attribute**, die sich hinsichtlich ihrer Bezeichnung und ihres zulässigen Wertebereiches unterscheiden und zudem auf verschiedenen Hierarchiestufen angeordnet sind.
- Das Tag ProductID besitzt einen Datenwert, der direkt auf das Ziel-Tag SUPPLIER_AID abgebildet werden kann. Dagegen existiert für das Attribut Type kein korrespondierendes Attribut im Zielformat, da dessen inhaltliche Bedeutung bereits durch das Ziel-Tag wiedergegeben wird.
- Das Tag ProductName besitzt ebenfalls einen Datenwert, der direkt auf ein Ziel-Tag (DESCRIPTION_SHORT) abgebildet werden kann. Allerdings befindet sich dieses auf einer tieferen Hierarchiestufe innerhalb des Containers ARTICLE_DETAILS.

Differenzierung des Mapping über die Kardinalität hinaus

Das 1:1-Mapping für XML-Daten lässt sich weiter systematisieren, indem die beeinflussenden Parameter und ihre Ausprägungen angeführt werden:

1. Umfang des Datenelementes im Quellformat (Tag oder Tag + Attribut)
2. Umfang des Datenelementes im Zielformat (Tag oder Tag + Attribut)
3. Benennung der Datenelemente (identisch oder verschieden)
4. Kodierung der Datenwerte (identisch oder verschieden)
5. Hierarchiestufe der Datenelemente (identisch oder verschieden)

Damit ergeben sich bereits für diesen einfachen Fall des 1:1-Mapping eine Reihe von Varianten, die in ähnlicher Form auch bei einem Mapping höherer Kardinalität anzutreffen sind.

Frage 1: Wie ist das 1:1-Mapping des Produktidentifikators (Tag ProductID) aus Abb. 1 anhand der fünf Parameter zu klassifizieren?

3.2. Das 1:N-Mapping

Beim 1:N-Mapping besteht eine Beziehung zwischen genau einem Datenelement des Quellformats zu mehreren Datenelementen des Zielformats. Die zugehörige Datentransformation bildet den Informationsgehalt des Quelldatenelementes durch **Verteilung und Hinzufügung** auf die Zieldatenelemente ab. Dieser Fall tritt vor allem dann auf, wenn das Zielformat einen höheren Detaillierungsgrad aufweist.

Die **Verteilung des Informationsgehaltes auf mehrere Zieldatenelemente** bedeutet, dass die Werte des Quelldatenelementes gemäß einer festgelegten Regel zerlegt und anschließend den Zieldatenelementen zugewiesen werden. Anwendungsbeispiele sind unter anderem die Trennung von Anschrift in Straßenname und Hausnummer, von Telefonnummer in Vorwahl und Anschluss sowie von Personenname in Vor- und Nachname. Die entsprechenden Zerlegungsvorschriften lassen sich auf Basis der Formatvorgabe des Quelldatenelementes formulieren. Fehlen diese oder sind unterschiedliche Formate zugelassen, so ist während der Datentransformation die Zerlegungsstelle durch eine Analyse der Datenwerte zu ermitteln, beispielsweise durch eine Suche nach üblichen Trennzeichen.

Frage 2: Woran könnte eine Verteilungsoperation scheitern?

Höhere Detaillierung im Zielformat

Im nächsten Beispiel wird die als BMEcat-Format in dem Tag Phone abgelegte Telefonnummer auf vier Tags des cXML-Formats verteilt, das separate Tags für die Landesvorwahl, die Gebietsvorwahl, die Anschlussnummer und die Durchwahl vorsieht, die durch einen zusätzlichen Container TelephoneNumber zusammengefasst werden (Abb. 2).

Quellformat BMEcat 1.2:

```
<PHONE>+49 201 183 4020</PHONE>
```

Zielformat cXML 1.2.009:

```
<Phone>
  <TelephoneNumber>
    <CountryCode isoCountryCode="DE">049,CountryCode>
    <AreaOrCityCode>201,AreaOrCityCode>
    <Number>183</Number>
    <Extension>4020,Extension>
  </TelephoneNumber>
</Phone>
```

Abb. 2: Beispiel XML-Daten für das 1:N-Mapping

Die Verteilung ist im Beispiel deshalb möglich, weil im BMEcat-Standard ein Inhaltsformat für das Phone-Tag definiert wird, das als Trennzeichen jeweils Leerstellen verwendet. Bei der Transformation der Ländervorwahl nach cXML ist die Kodierung so zu verändern, dass das führende Pluszeichen in eine führende Null umgewandelt wird. Weiterhin ist ein Attributwert zu erzeugen, der das mit der Landesvorwahl referenzierte Land gemäß der ISO-Norm 3166-1 benennt (Hinzufügung).

Adressdatenkonvertierung ist ein allgemeines Problem

Die zuvor genannten Anwendungsfälle sind nicht besonders charakteristisch für Katalogdaten. Zwar werden Adressdaten auch in elektronischen Katalogen verwendet und sind folglich zu konvertieren (vgl. Omelayenko/Fensel 2001), jedoch ist der Bereich der produktbezogenen Daten von größerer Bedeutung. Als Beispiele für Schemaunterschiede, die zu einem 1:N-Mapping führen, können hier genannt werden:

Beispiele für Schemaunterschiede

- Bei Produktmerkmalen werden im Quellformat **Wert und Einheit** zusammengefasst (Beispiel: „15 cm"), während diese im Zielformat getrennt sind.
- Bei Produktmerkmalen werden im Quellformat **Merkmalsbezeichnung und Wert** zusammengefasst (Beispiel: „Farbe: schwarz"), während diese im Zielformat getrennt sind.
- Bei Produktpreisen werden im Quellformat **Preis und Währung** zusammengefasst (Beispiel: „25.50 EUR"), während diese im Zielformat getrennt sind.

Ein 1:N-Mapping liegt auch dann vor, wenn im Quellformat der Informationsgehalt eines Tags durch einen oder mehrere Attributwerte ergänzt wird und im Zielformat eine Verteilung auf zwei oder mehr Tags notwendig ist, da keine entsprechenden Attribute vorgesehen sind. Dieser Fall ist auf die oben angeführten Schemaunterschiede übertragbar, wenn beispielsweise der Merkmalswert in einem Tag und die Einheit in einem XML-Attribut enthalten sind. Analog dazu zeigt das nächste Beispiel die Transformation eines Produktpreises von cXML nach BMEcat (Abb. 3), indem das currency-Attribut dem CURRENCY-Tag, das Money-Tag dem PRICE_AMOUNT-Tag und der UnitPrice-Container dem ARTICLE_PRICE-Container zugeordnet werden. Letzterer besitzt außerdem ein XML-Attribut für den Preistyp, das immer zu belegen ist (Hinzufügung).

Quellformat cXML 1.2.009:

```
<UnitPrice>
  <Money currency="EUR">88.95</Money>
</UnitPrice>
```

Zielformat BMEcat 1.2:

```
<ARTICLE_PRICE price_type="net_list">
  <PRICE_AMOUNT>88.95,PRICE_AMOUNT>
  <CURRENCY>EUR</CURRENCY>
</ARTICLE_PRICE>
```

Abb. 3: Beispiel XML-Daten für das 1:N-Mapping

3.3. Das N:1-Mapping

Geringere Detaillierung im Zieldatenformat

Beim N:1-Mapping besteht eine Beziehung zwischen mehreren Datenelementen des Quellformats zu genau einem Datenelement des Zielformats. Die zugehörige Datentransformation muss den Informationsgehalt der Quelldatenelemente durch **Konkatenation, Aggregation und Selektion** auf das Zieldatenelement abbilden. Komplementär zum 1:N-Mapping tritt dieser Fall vor allem auf, wenn das Quellformat einen höheren Detaillierungsgrad als das Zielformat besitzt.

Die Anwendungsfälle des N:1-Mapping bei der Konvertierung zwischen Katalogformaten ergeben sich bereits über die Umkehrung der oben beschriebenen 1:N-Mappings. Damit werden jedoch nur die Konkatenations- und die Selektionsoperation abgedeckt. Konkatenation heißt, dass alle Elemente der N-Seite auf das Zielelement abgebildet werden können, indem die Werte in einer bestimmten Reihenfolge miteinander verknüpft werden. Selbstverständlich können diese Werte vor der Konkatenation einer Änderung der Kodierung unterliegen. Von einer Selektion kann gesprochen werden, wenn für das Mapping nicht alle Elemente der N-Seite herangezogen werden, da es im Zielformat keine Entsprechung gibt. Im Beispiel aus Abb. 3 trifft dies auf das XML-Attribut price_type zu.

Frage 3: Welchen Effekt zieht die Operation Selektion nach sich?

Die Aggregationsoperation überführt den Informationsgehalt mehrerer Quelldatenelemente unter Anwendung einer Aggregationsvorschrift in ein Zieldatenelement. Sie wird für die **Neuberechnung numerischer Werte** — bei Katalogdaten insbesondere für Produktpreise, Lieferzeiten und Logistikmerkmale — genutzt. Besitzt das Quellformat beispielsweise ein stärker detailliertes Preismodell, so ist wie im nachfolgenden Beispiel der Produktpreis über die Komponenten des Preismodells zu berechnen: Während im BMEcat-Format zu den preisbestimmenden Faktoren unter anderem der Preistyp, der Umsatzsteuersatz und ein durchgerechneter Rabattfaktor zählen, sind diese Komponenten im cXML-Preismodell nicht vorhanden. Stattdessen handelt es sich bei cXML-Preisen um Bruttoendpreise. Deshalb ist der in der Abb. 4 kodierte Produktpreis über die Formel

$$\text{Money} = \text{ROUND}((\text{PRICE\_AMOUNT} * \text{PRICE\_FACTOR}) / (1 + \text{TAX}), 2)$$

neu zu berechnen.

Quellformat BMEcat 1.2:

```
<ARTICLE_PRICE price_type="gros_list">
  <PRICE_AMOUNT>80,PRICE_AMOUNT>
  <CURRENCY>EUR</CURRENCY>
  <TAX>0.16</TAX>
  <PRICE_FACTOR>0.9</PRICE_FACTOR>
</ARTICLE_PRICE>
```

Zielformat cXML 1.2.009:

```
<UnitPrice>
  <Money currency="EUR">62.07</Money>
</UnitPrice>
```

Abb. 4: Beispiel XML-Daten für das N:1-Mapping

3.4. Das N:M-Mapping

Zerlegung in 1:1-, 1:N- und N:1-Mapping nicht ausreichend

Beim N:M-Mapping besteht eine Beziehung zwischen mehreren Datenelementen des Quellformats zu mehreren Datenelementen des Zielformats. Die zugehörige Datentransformation bildet den Informationsgehalt der Quelldatenelemente durch eine **komplexe Operation**, die sich aus Teiloperationen vom Typ Verteilung, Hinzufügung, Konkatenation, Selektion und Aggregation zusammensetzt, auf die Zieldatenelemente ab. Dies ist der Fall, wenn die beiden Formate (in Teilbereichen) eine stark abweichende Dokumentstruktur aufweisen.

Die zuvor beschriebenen Datentransformationen haben sich auf gleiche Objekttypen bezogen, d.h., es wurden Mappings zwischen Preisen, Merkmalen, Adressdaten, Produktdaten usw. definiert, ohne dass die zwischen diesen Datenbereichen bestehenden Beziehungen zu berücksichtigen gewesen wären. Diese inhaltlichen Beziehungen spiegeln sich gerade in den Dokumentstrukturen der XML-Formate wieder, die anders als etwa CSV-basierte Formate in der Lage sind, mehrere **unterschiedlich strukturierte Datenbereiche in einer gemeinsamen XML-Dokumentstruktur** zu integrieren. Ein N:M-Mapping zeichnet sich nun dadurch aus, dass auf mindestens einer der beiden Seiten auf Datenelemente unterschiedlicher Objekttypen Bezug genommen wird.

Anforderungen eines N:M-Mapping

Im nächsten Beispiel unterscheidet sich die Zuordnung von Produkten zu Produktgruppen dadurch, dass im Quellformat xCBL die Zuordnung Bestandteil der Produktdaten ist, sie jedoch im Zielformat BMEcat getrennt von den Produktdaten erfolgt (Abb. 5). Wie der XML-Code zeigt, besteht eine inhaltliche Beziehung zwischen den beiden Bereichen Produktdaten (ARTICLE) und Gruppenzuordnung (ARTICLE_TO_CATALOGGROUP_MAP), die über den Produktidentifikator (SUPPLIER_AID bzw. ART_ID) hergestellt wird. Damit sind die Anforderungen eines N:M-Mapping erfüllt: Die M-Seite weist zwei unterschiedliche Objekttypen auf, und für die Datentransformation auf der N-Seite werden mehrere Datenelemente angesprochen.

Quellformat xCBL 4.0:

```
<Product>
  <ProductID>88021</ProductID>
  <SchemaCategoryRefList>
    <CategoryIDRef>50047463,CategoryIDRef>
  </SchemaCategoryRefList>
</Product>
```

Zielformat BMEcat 1.2:

```
<ARTICLE mode="new">
  <SUPPLIER_AID>88021</SUPPLIER_AID>
  ...
</ARTICLE>
<ARTICLE_TO_CATALOGGROUP_MAP>
  <ART_ID>88021</ART_ID>
  <CATALOG_GROUP_ID>50047463</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
```

Abb. 5: Beispiel XML-Daten für das N:M-Mapping

Die verwendete Operation ist vom Typ Verteilung, da die Datenelemente des einen Objekttyps im Zielformat auf mehrere Objekttypen verteilt werden. Sie kann ergänzt werden um:

- **Hinzufügung**, wenn zusätzliche Zieldatenelemente befüllt werden
- **Konkatenation**, wenn mehrere Quelldatenelemente auf ein Zieldatenelement abgebildet werden
- **Aggregation**, wenn eine Berechnungsvorschrift für die Zieldatenelementwerte existiert
- **Selektion**, wenn nicht jedes Quelldatenelement eine Entsprechung im Zielformat besitzt

4. Entwicklung von Datentransformationen

Formate analysieren und verstehen

Die Erarbeitung von Mapping-Definitionen basiert auf der **Analyse der formalen und deskriptiven Formatspezifikationen**, soweit diese verfügbar und hinreichend sind. Die formale Spezifikation beschreibt den Aufbau gültiger Dokumente vergleichbar mit dem Schema einer relationalen Datenbank. Dazu zählen die Definition von Datentypen und Datenelement, die Verwendung dieser Elemente zur Beschreibung der hierarchischen Dokumentstruktur, die Hinzufügung von Integritätsbedingungen und je nach eingesetzter Schemasprache weitere Eigenschaften des Dokumenttyps. Die wichtigsten Schemasprachen für XML-Dokumente sind **DTD (Document Type Definition) und XSDL (Extensible Schema Definition Language)**. Letztere ist nicht zuletzt aufgrund ihrer Mächtigkeit mittlerweile zur Standardsprache für die Spezifikation XML-basierter Geschäftsdokumente geworden (vgl. Schmitz/Leukel/Dorloff 2003).

Zunächst ist für jedes Datenelement zu prüfen, welche Informationen es enthält und ob diese im Zieldokument relevant sind. Die semantische Zuordnung erfordert daher eine genaue Kenntnis der Dokumenttypen. Zu vielen Formaten kann die zugehörige Spezifikation zwar Hilfe bei der Klärung von Elementinhalten bieten. Die Bedeutung ist jedoch oft nur mit fachlichem Wissen zweifelsfrei feststellbar, d.h., es ist **Wissen über die jeweilige Domäne** notwendig (z.B. E-Procurement, Logistik).

Entwicklung planen und dokumentieren

Die Mapping-Definitionen sind bei umfangreichen Dokumenttypen zu dokumentieren und nach den oben beschriebenen Kriterien zu klassifizieren. Dazu bieten sich der Aufbau und die schrittweise Vervollständigung von **Mapping-Tabellen** an, welche die Dokumentstruktur von Quell- und Zielformat gegenüberstellen. Zu den Inhalten einer solchen Beschreibung gehören hinsichtlich der Formate die Elementbezeichner entsprechend ihrem Auftreten in der hierarchischen Dokumentstruktur, die Elementkardinalitäten und die Datentypen. Die Gegenüberstellung von Quell- und Zielformat erfolgt durch die Zuordnung korrespondierender Datenelemente und die Angabe der Mapping-Kardinalität. Je nach Ansatz und Bedeutung dieser geplanten Entwicklung von Datentransformationen kann so eine fundierte und vollständige Beschreibung entstehen, die anschließend auch von Nicht-Domänenexperten in **Datentransformationswerkzeuge** (z.B. Microsoft BizTalk, Seeburger Business Integration Server) überführt oder in Skripte umgesetzt werden kann.

Die Ausführung von Datentransformationen für XML-Dokumente geschieht in der Regel auf der Grundlage von Skripten, die in der standardisierten Sprache **XSLT (Extensible Stylesheet Language Transformations)** erstellt sind (vgl. Wüstner/Hotzel/Buxmann 2002). Gerade bei aufwändigen Transformationen, komplexen Dokumentstrukturen und einer Vielzahl von Formaten, die zudem in unterschiedlichen Versionen vorliegen, bietet es sich an, den Entwicklungsprozess an **Prinzipien der Softwareentwicklung** auszurichten. Im Speziellen ist die direkte Umsetzung in XSLT ohne eine ausreichende inhaltliche Dokumentation zu vermeiden. Üblicherweise wird dazu die konzeptionelle Entwicklung wie skizziert von der Codierung getrennt.

Ist Konvertierung überhaupt möglich und sinnvoll?

Die Entwicklung von Datentransformationen wird in vielen Fällen einen unterschiedlichen Informationsgehalt von Quell- und Zielformat aufdecken. Dies bedeutet, dass sich für bestimmte Datenelemente des Quellformates keine Äquivalente im Zielformat feststellen lassen, sodass es bei der Konvertierung zum einem **Informationsverlust** kommt. Im umgekehrten Fall erfordert das Zielformat über definierte Pflichtdatenelemente bestimmte Informationen, die im Quellformat nicht abgelegt sind. Hier liegt ein **Informationsdefizit** vor, das die Konvertierung zunächst verhindert. Dieses Problem kann gelöst werden, indem die fehlenden Informationen nicht aus dem Quelldokument abgeleitet, sondern vor der Ausführung der Datentransformation manuell hinzugefügt werden (vgl. Beul/Bittscheidt/Leukel/Spies 2003).

Frage 4: Wie kann über akzeptable Informationsverluste entschieden werden?

Literaturempfehlungen:

Ariba, Inc.: cXML 1.2.009. Online: http://xml.cxml.org (Stand: 10.11.2003).

Beul, M./Bittscheidt, C./Leukel, J./Spies, T: Behandlung von Informationsdefiziten und -verlusten bei der Transformation von XML-Geschäftsdaten. In: Proceedings der 5. Paderborner Frühjahrstagung „Innovationen im E-Business". Paderborn 2003, S. 159 - 168.

CommerceOne, Inc.: XML Common Business Library (xCBL) 4.0, 2003. Online: http://www.xcbl.org, (Stand: 10.11.2003).

IEEE Computer Society (1999): Bulletin of the Technical Committee on Data Engineering. Special Issue on Data Transformations, Vol. 22 (1999), No. 1.

Leukel, J./Schmitz, V./Dorloff, F.-D.: Coordination and Exchange of XML Catalog Data in B2B. In: Proceedings of the 5th International Conference on Electronic Commerce Research (ICECR-5). Montreal 2002.

Omelayenko, B./Fensel, D.: An Analysis of Integration Problems of XML-Based Catalogs for B2B Electronic Commerce. In: Proceedings of the 9th IFIP 2.6 Working Conference on Database Semantics (DS-9). Hong-Kong 2001, S. 232 - 246.

Schmitz, V./Kelkar, O./Pastoors, T.: Spezifikation BMEcat Version 1.2, 2001. Online: http://www.bmecat.org (Anmeldung erforderlich, Stand: 10.11.2003).

Schmitz, V./Leukel, J./Dorloff, F.-D.: Does B2B Data Exchange Tap the Full Potential of XML Schema Languages. In: Proceedings of the 16th Bled Electronic Commerce Conference. Bled (Slowenien) 2003, S. 172 - 182.

Wüstner, E./Hotzel, T./Buxmann, P.: Converting Business Documents: A Classification of Problems and Solutions using XML/XSLT. In: Proceedings of the 4th IEEE International Workshop on Advanced Issues of E-Commerce and Web-based Information Systems (WECWIS 2002). Newport Beach 2002, S. 61 - 68.

Die Fragen werden im WISU-Repetitorium beantwortet.

BMEcat-Telefonnummern ist die Verteilung nur mit Hilfe der vorgegebenen Leerzeichen möglich. Werden diese nicht eingefügt, so ist die Verteilung nicht möglich oder führt zu falschen Ergebnissen. Sind im Quelldatenelement verschiedene Informationen abgelegt (z.B. „Hr. Dr. rer. nat. Hans Dieter Dressler"), die zudem in jeder Instanz vorkommen (z.B. Nachname), mehrfach auftreten (z.B. Vornamen) oder optional sind (z.B. akademischer Grad), dann muss die Verteilungsoperation jede Zeichenkette genau analysieren, mit bekannten Mustern vergleichen und Inhalte möglicherweise kontextabhängig separieren. Dies ist nicht in allen Fällen zweifelsfrei möglich.

Frage 3: Welchen Effekt zieht die Operation Selektion nach sich?

Da die Selektion nur einen Teil des Informationsgehaltes der Quelldatenelemente auf das Zieldatenelement abbildet, ist zu überprüfen, ob der nicht berücksichtigte Informationsgehalt bei der Konvertierung insgesamt verloren geht und damit ein Informationsverlust entsteht. Es kann jedoch auch sein, dass im Zuge einer anderen Mapping-Definition sichergestellt wird, dass keine Informationen des Quellformats verloren gehen — sie werden nur an einer anderen Stelle im Zieldokument platziert.

Frage 4: Wie kann über akzeptable Informationsverluste entschieden werden?

Ob ein Informationsverlust akzeptabel ist oder die Konvertierung aus inhaltlicher Sicht verhindert wird, ist stark von den jeweiligen Einsatzanforderungen an die Geschäftsdokumente abhängig. Diese Bewertung kann beispielsweise bilateral von den Geschäftspartnern vorgenommen werden und zu anderen Ergebnissen führen als bei einer allgemeinen, unternehmensunabhängigen Betrachtung. Informationsverluste können zum Beispiel toleriert werden, wenn die betreffenden Datenelemente in der konkreten Geschäftsbeziehung nicht benötigt werden.

Wirtschaftsinformatik/Hauptstudium

Fragen und Antworten 1 - 4 zu „Autorisierung und Zugriffskontrolle bei Wissensportalen" von Prof. Dr. G. Pernul/Dipl.-Wirt.-Inf. E. Masovic/Dipl.-Wirt.-Inf. T. Priebe. WISU 1/04, S. 94 - 100.

Frage 1: Welche Probleme ergeben sich bei der Zugriffskontrolle in Wissensportalen?

Wissensportale können sensible Informationen enthalten, deren Vertraulichkeit gewährleistet werden muss. Um diese Sicherheitsanforderung erfüllen zu können, muss eine Zugriffskontrolle erfolgen. Innerhalb von Wissensportalen ist die Anzahl der Benutzer i.d.R. sehr hoch. Gleiches gilt für die zu schützenden Objekte, bspw. Dokumente. Zudem sind meist individuell unterschiedliche Zugriffsrechte erforderlich. Somit ergeben sich in diesem Zusammenhang Probleme bei der Administration der Berechtigungen. Weiterhin erfolgt in Portalsystemen eine Zugriffskontrolle auf verschiedenen Ebenen (sowohl Struktur als auch Inhalt).

Frage 2: Was versteht man unter Autorisierung und wie wird sie von der Zugriffskontrolle abgegrenzt?

Durch eine Zugriffskontrolle sollen Vertraulichkeit und Integrität von Informationen gesichert werden. Die Festlegung bzw. Verwaltung der Zugriffsrechte (welche Subjekte dürfen auf welche Objekte, z.B. Dokumente, wie zugreifen?) wird Autorisierung genannt. Die Zugriffskontrolle hingegen umfasst die Überprüfung der Zugriffsrechte zur Laufzeit. Dabei werden die während der Autorisierung festgelegen Rechte geprüft und der Zugriff entsprechend gewährt oder verweigert.

Frage 3: Welche klassischen Zugriffskontrollmodelle kennen Sie und wodurch sind sie gekennzeichnet?

— Benutzerbestimmbare Zugriffskontrolle (Discretionary Access Control — DAC): Die benutzerbestimmbare Zugriffskontrolle basiert auf dem Eigentümerprinzip. Das heißt, dass die Eigentümer

Wirtschaftsinformatik/Grundstudium

Fragen und Antworten 1 - 4 zu „Datentransformation bei XML-basierten Geschäftsdokumenten" von Prof. Dr.-Ing. F.-D. Dorloff/Dipl.-Wirt.-Inf. J. Leukel/Dipl.-Inf. V. Schmitz. WISU 1/04, S. 87 - 94.

Frage 1: Wie ist das Mapping der Produktidentifikatoren (Tag ProductID) aus Abb. 1 anhand der fünf Parameter zu klassifizieren?

Das Datenelement ProductID des Quellformats wird dem Datenelement SUPPLIER_AID des Zielformats zugeordnet. Das Quelldatenelement umfasst ein Tag und ein XML-Attribut (1. Parameter), während das Zieldatenelement einen geringeren Umfang besitzt (nur Tag, 2. Parameter). Die Benennung der Datenelemente ist verschieden (3. Parameter), die Kodierung der Datenwerte dagegen identisch (4. Parameter), sodass keine Wertumwandlung erforderlich ist. Auch ohne Kenntnis der genauen Formatspezifikation ist anzunehmen, dass in beiden Formaten als Datentyp Zeichenketten begrenzter Länge definiert sind. Die Datenelemente sind auf der gleichen Hierarchiestufe angeordnet (5. Parameter), und zwar innerhalb eines Containers zur Aufnahme produktbezogener Daten.

Frage 2: Woran könnte eine Verteilungsoperation scheitern?

Jede Verteilungsoperation basiert auf Annahmen über das Format des Quelldatenelementes. Im Fall der Zeichenketten zeigt sich dies an der Funktion von Trennzeichen, die signalisieren, an welcher Stelle sich Informationsgehalte trennen und anschließend verteilen lassen. Die Verteilung scheitert, wenn solche Trennzeichen fehlen oder nicht eindeutig identifiziert werden können. Im Beispiel der