

Manfred Thaller

Vom verschwindenden Unterschied zwischen Datenbanken und Texten: Konsequenzen neuerer www-Technologie am Beispiel von museumsna- hen Datenbanken

Beobachtungen

Datenbanken sind besonders gut geeignet, um Informationen so aufzubereiten, dass sie für die systematische Suche und die Ausgabe in vielen verschiedenen Formen von Listen zugänglich werden. Damit sie dies können, muss die Information, die in ihnen enthalten ist, sorgfältig aufbereitet und vereinheitlicht werden. Dies bedeutet zweierlei: einerseits, dass die vorhandenen Informationen in Tabellenform gespeichert werden, wobei exakt definiert werden muss, welche Informationen aufgenommen werden sollen und welche Art von Information welchen Feldern zugewiesen wird. Ein Beispiel dafür bietet die unten stehende Tabelle, ein (leicht vereinfachter) Auszug aus der Datenbank, welche die theaterwissenschaftliche Sammlung der Universität zu Köln verwendet, um einen Bestand von Theaterphotos zu verwalten.

Beispiel 1

Die Felder sind genau vorgegeben und entsprechen folgenden Modell:

1. [I]Signatur
2. Personen
3. Objekt-Urheber
4. Titel
5. Werk-Urheber
6. Bühne
7. Ort
8. Regie
9. Ausstattung
10. Zeit

Diese Art der Aufteilung von Informationen auf klar definierte Abschnitte muss begleitet werden von einer sehr präzisen Festlegung, wie genau die Angaben in den einzelnen Feldern zu formulieren sind. Einerseits strukturell: „Familiename, gefolgt vom Vornamen des oder der Dargestellten. Dahinter steht die auf der Abbildung dargestellte Rolle in einem Paar runder Klammern, wobei der Text ‚Rolle:‘ vor der Bezeichnung derselben steht.“ Andererseits inhaltlich: durch die exakte Festlegung dessen, welcher Wortschatz in den einzelnen Feldern verwendet werden darf. Diese Voraussetzungen müssen erfüllt werden, wenn Informationen vom Rechner verwaltet werden müssen; werden sie berücksichtigt, so kann die Information neuerdings nicht nur lokal, sondern auch über das Internet angesprochen werden. Dementsprechend ist es einfach Daten wie die obigen als Teil eines für die Lehre entwickelten Kleinservers im Internet bereitzustellen, wo er unter der URL <http://www.hki.uni-koeln.de/demo/hybrid/server1/index.html> angesprochen werden kann. (Einige weitere Erläuterungen zum Hintergrund dieses Servers sind dort zu finden unter <http://www.hki.uni-koeln.de/demo/hybrid/index.html>).

Unsere bisherigen Darlegungen entsprechen einer Meinung, die mittlerweile so verbreitet ist, dass ihre Wiederholung völlig überflüssig erscheint. Aber: Ist sie richtig? Wenn dies der Fall ist, so ist folgendes Dokument ganz offensichtlich ungeeignet, um im Internet datenbankmäßig verwaltet zu werden:

Bild 12 © Günther Ventur

Gotische Türme in abendlicher Festbeleuchtung – das politische und das religiöse Wahrzeichen des alten Köln: der historische Rathausturm (1414) und die Türme des Doms (1880), damals mit 157,38 m für ein Jahrzehnt das höchste Bauwerk der Welt.

Gothic towers in night-time gala illumination – the political and religious landmarks of ancient Cologne: the historic town hall tower (1414) and the spires of the Cathedral (1880), at that time, the world's highest building for a decade, at a height of 157.38 m.

Beispiel 2

Unter <http://www.hki.uni-koeln.de/demo/hybrid/server2/index.html> finden wir im Internet aber einen anderen, sehr kleinen, Lehrserver, der exakt dieses Material in genau der gleichen Zugriffsweise anbietet, wie dasjenige, das wir eben in Tabellenform kennen gelernt haben. Nicht sehr detailliert aufgeschlüsselt – nach den drei Kategorien „Copyrightbesitzer“, „deutscher Volltext“ und „englischer Volltext“ –, aber eben doch in einer Form, die unverkennbar mit einem Datenbanksystem in Verbindung steht.

Und unter <http://www.hki.uni-koeln.de/demo/hybrid/server3/index.html> finden wir dieselbe Art von Material in einem Zugriffssystem, das geringfügig weitergeht und auch den Zugriff über Schlagworte ermöglicht, so dass die in den Volltextmenüs angebotenen, für die meisten Benutzer weniger interessanten Suchworte, wie „alten“ oder „political“ wegfallen. Aber auch hier liegt keineswegs eine Datenbank zu Grunde. Den Hintergrund bilden Dokumente wie das folgende:

Bild 12 © Günther Ventur

Gotische Türme in abendlicher Festbeleuchtung – das politische und das religiöse Wahrzeichen des alten Köln: der historische Rathausturm (1414) und die Türme des Doms (1880), damals mit 157,38 m für ein Jahrzehnt das höchste Bauwerk der Welt.

Gothic towers in night-time gala illumination - the political and religious landmarks of ancient Cologne: the historic town hall tower (1414) and the spires of the Cathedral (1880), at that time, the world's highest building for a decade, at a height of 157.38 m.

Beispiel 3

Dass diese Beschreibungen – mit offensichtlicher Ähnlichkeit zu denen herkömmlicher Kataloge – offensichtlich gültige „Records“ oder „Dokumente“ einer Datenbank darstellen, ist verblüffend, wenn wir uns an das eingangs Gesagte erinnern. Hier liegt das Missverständnis aber in einem Detail, das mit einer etwas unpräzisen Wiedergabe vermeintlich informationswissenschaftlicher Begrifflichkeiten zusammenhängt. Als Hintergrund des in den einleitenden Absätzen Gesagten wird gerne eine Aussage, wie etwa die folgende zitiert: „Die Anwendung einer Datenbank setzt voraus, dass die zu verwaltenden Daten in Form von Tabellen vorliegen, die bestimmten strukturellen Einschränkungen unterliegen.“ Richtiger wäre eine Formulierung der folgenden Form: „Die Anwendung einer Datenbank setzt voraus, dass die Daten in einer Form vorliegen, die sie in eine klar erkennbare Struktur bringt.“ Die Umsetzung in Tabellen ist eine Möglichkeit dies zu erreichen, und aus der Sicht einiger Softwareprodukte auch die vorgeschriebene. Unter „Strukturen“ in Daten verstehen wir in der Rechnertechnologie aber ein wesentlich weiteres Spektrum als das

von Tabellen. Genau wie der Mensch, übrigens. Wenn wir das Beispiel 2 betrachten, verstehen wir implizit, dass diese Textstücke etwa in folgender Weise aufbereitet sind: „Eine neue Beschreibung beginnt mit der Nummer der dokumentierten Fotografie, gefolgt vom Copyrightinhaber in großem Fettdruck. Daran schließt sich eine deutsche Beschreibung in normaler Schriftgröße und Art, eine englische in normaler Schriftgröße, aber im Kursivdruck.“

Genau diese Anweisungen – Schriftgröße, Schriftart – verwenden wir, wenn wir den Text für das Internet aufbereiten wollten, wie im folgenden Beispiel 4:

Bild 12 © Günther Ventur

Gotische Türme in abendlicher Festbeleuchtung – das politische und das religiöse Wahrzeichen des alten Köln: der historische Rathausurm (1414) und die Türme des Doms (1880), damals mit 157,38 m für ein Jahrzehnt das höchste Bauwerk der Welt.

Gothic towers in night-time gala illumination - the political and religious landmarks of ancient Cologne: the historic town hall tower (1414) and the spires of the Cathedral (1880), at that time, the world's highest building for a decade, at a height of 157.38 m.i

Beispiel 4

Als Menschen interpretieren wir dabei ein visuelles Attribut intellektuell, das heißt, wir verhalten uns in etwa so, als ob Folgendes hier stünde:

```
<<Bild ref="012" copyright="Günther Ventur">>
```

```
<<deutsch>>
```

Gotische06 Türme in abendlicher Festbeleuchtung – das politische und religiöse Wahrzeichen des alten Köln: der historische Rathausurm (1414) und die Türme des Doms (1880), damals mit 157,38 m für ein Jahrzehnt das höchste Bauwerk der Welt.06

```
<</deutsch>>
```

```
<<englisch>>
```

Gothic towers in night-time gala illumination - the political and religious landmarks of ancient Cologne: the historic town hall tower (1414) and the spires of the Cathedral (1880), at that time, the world's highest building for a decade, at a height of 157.38 m.

```
<</englisch>>
```

```
<<Bild>>
```

Beispiel 5

Diese Art, den Text zu betrachten, nennen wir in der Informationstechnologie „konzeptuelle oder logische Auszeichnung“ (conceptual or logical markup). Sie beginnt durch die verstärkte Aufmerksamkeit für XML, das gerne als Nachfolger von HTML gehandelt wird, auch für die Erzeugung von Texten fürs Internet eine immer größere praktische Bedeutung zu erlangen. Hier ist das entscheidende Argument für eine derartige „Auszeichnung“ eines Textes, im Gegensatz zum unmittelbaren Einfügen von Anweisungen, mit welchen Attributen er dargestellt werden sollte, etwa Folgendes: „Die Vorstellungen davon, was ein gefälliges visuelles Äußeres eines Dokuments im Internet ausmacht, wandeln sich ziemlich rasch. Deshalb ist es zweckmäßiger, festzulegen, aus welchen Abschnitten, die potentiell unterschiedlich dargestellt werden sollen, ein Text besteht und bei jedem Wandel der Mode die Regeln, wie diese Abschnitte visuell umzusetzen sind, zu ändern, als sich der Mode von heute anzupassen und dann schon im nächsten Jahr den Text nach dem letzten Schrei umformatieren zu müssen.“

Ein weniger beachteter Effekt besteht jedoch darin, dass wir beim Umsetzen von Beispiel 4 zu Beispiel 5 nebenbei eine Struktur erzeugt haben, die visuelle Attribute aufnehmen kann, aber auch zu anderen Dingen verwendet werden kann. Denn, die Strukturierung der Daten als

```
<<Bild ref="xxx" copyright="yyy"
  <<deutsch>>
  <</deutsch>>
  <<englisch>>
  <</englisch>>
<<Bild>>
```

hat nebenbei auch eine „Struktur“ erzeugt, wie wir sie als Verallgemeinerung der Vorstellung, dass Datenbanken Tabellen enthalten müssten, als echte Voraussetzung für den Einsatz von Datenbanken gefordert haben.

Freilich, auch hier ist eine gewisse Disziplin erforderlich: Ist die obige Struktur festgelegt, dann wird eine Datenbank einen Beschreibungsteil der Form von Beispiel 6 zunächst nicht verstehen.

```
<<englisch>>
```

Gothic towers in night-time gala illumination – the political and religious landmarks of ancient <<deutsch>> Köln<</deutsch>>: the historic <<deutsch>>Rathausturm<</deutsch>> (1414) and the spires of the Cathedral (1880), at that time, the world's highest building for a decade, at a height of 157.38 m.

```
<<englisch>>
```

Beispiel 6

Dies liegt aber keineswegs daran, dass derartige Strukturen notwendigerweise extrem einfach bleiben müssten. In der Tat sind sie potenziell „mächtiger“ als die Tabellen klassischer Datenbanken. Dass wir, um Beispiel 3 verarbeiten zu können, eine Struktur des Typs

```
<<Bild ref="xxx" copyright="yyy">>
  <<deutsch>>
    <<term>>
    <</term>>
  <</deutsch>>
  <<englisch>>
    <<term>>
    <</term>>
  <</englisch>>
```

```
<<Bild>>
```

benötigen, dürfte offensichtlich sein. Der recht mächtige Server, mit dem das Institut für Mittelalterliche Realienskunde in Krems an der Donau seine in Theuern schon mehrfach vorgestellte Datenbank mittlerweile im Internet anbietet (<http://dione.imareal.oeaw.ac.at/server/index.html>) beruht auf einer Struktur, die im Folgenden extrem vereinfacht wiedergegeben ist:

```
<<Nummer>>
  <<Gegenstand>>
  <</Gegenstand>>
  <<Dokumentation>>
  <</Dokumentation>>
  <<Kommentar>>
```

```

<</Kommentar>>
<<Person>>
  <<Körperteil>><<R>          <</Körperteil>>
  <<Attribut>>
    <<Teil>>
      <<Unterteilung>>
      <</Unterteilung>>
    <</Teil>>
  <</Attribut>>
  <<Kleidungsstück>>
    <<Teil>>
      <<Unterteilung>>
      <</Unterteilung>>
    <</Teil>>
  <</Kleidungsstück>>
<</Person>>
<<Objekt>>
  >>Teil>>
    <<Unterteilung>>
    <</Unterteilung>>
  <</Teil>>
<</Objekt>>
<</Bild>>
<<Nummer>>

```

Tatsächlich verarbeitet der Server in diesem Fall eine Struktur, bei der jedes der hier eingeführten Symbole fünf bis sechs weitere enthalten kann und an den meisten Stellen noch eine zusätzliche dreistufige Hierarchie „abzweigen“ kann.

Und Server, wie der unter <http://www.ceec.uni-koeln.de> zugängliche für die Handschriften der Kölner Dom- und Diözesanbibliothek, verwenden Daten, die den Inhalt der klassischen Handschriftenbeschreibungen und Ausstellungskataloge enthalten und nach den gleichen Prinzipien erstellt wurden, aber auf den so genannten „TEI Spezifikationen“ beruhen, die zwei kleingedruckte Bände im Telefonbuchformat füllen.

Bedeutung: Eine Zwischenbilanz

Nach dem eben Bemerkten verwischt sich also die Grenze zwischen Datenbanken und Texten der Art, wie sie insbesondere für das Internet angelegt werden, fast völlig. Als Nebeneffekt bedeutet dies, dass auch die Unterschiede zwischen unterschiedlichen Datenbanksystemen weniger wichtig werden: Da alle davon gezwungen sind Möglichkeiten anzubieten, wie sie ihre Inhalte im Internet zugreifbar machen, wird es zunehmend einfacher Software zu generieren, die eine allgemeine Struktur der hier verwendeten Art voraussetzt und sich mit einzelnen Datenbanken darüber „unterhält“, welche Informationen, die dieser allgemeinen Struktur in etwa entsprechen, sie enthalten.

Dementsprechend fördert das Bundesministerium für Bildung und Forschung in den nächsten Jahren mit nicht unerheblichen Beträgen ein Projekt, das es sich zum Ziel setzt unterschiedliche Bilddatenbanken, wie sie in kunsthistorischen Instituten üblich sind, unter einer gemeinsamen Oberfläche abfragbar zu machen. (Der Projektbeginn ist April 2001; unter <http://www.prome->

theus-bildarchiv.de werden also erst ab etwa Sommer/Herbst substantiellere Informationen vorliegen.)

Beide Effekte zusammen können in ihrer Bedeutung kaum überschätzt werden. Die Verwendung von Datenbanken im kulturellen Bereich hatte de facto in den letzten Jahren einen zentralisierenden Effekt. Nachdem Datenbanken vermeintlich nur mit sehr einheitlichen, tabellenartigen Datenstrukturen und streng vereinheitlichtem Vokabular umgehen konnten, schien ein technischer Sachzwang zu möglichst großer Reglementierung vorzuliegen — mit dem Idealbild eines für möglichst große Regionen einheitlichen konzeptuellen Systems, hinter dem sich die Utopie einer bundesweiten – oder europa-, ja weltweiten – „Museumseinheitssoftware“ nur mühsam verbarg. In dem Maße, in dem „die Datenbank“ durch „den Webserver“ gewissermaßen als „Leitfossil“ der Anwendung der Informationstechnologie im Bereich des Kulturerbes ersetzt wird, wird diese Tendenz, über deren intellektuelle Sinnhaftigkeit unterschiedliche Meinungen möglich sind, jedenfalls technisch nachgerade widersinnig. Heterogenität, Dezentralisierung, verteiltes Rechnen, hybride Server – so heißen die Schlagworte der neuen Generation von IT-Anwendungen. Und definitiv nicht Zentralisierung und Vereinheitlichung im alten Sinne.

Technischer Hintergrund I: XML

Wir haben uns bei den vorangehenden Beispielen des Werkzeugs der „Extensible Markup Language“ bedient, die auf dieser Tagung auch in anderen Beiträgen angesprochen wird. Sie ist an sich keine Voraussetzung der hier behandelten Entwicklungen und vorhergesagten Trends, da alle durch sie bereit gestellten Werkzeuge für den Bereich, der uns hier interessiert – das Verschwimmen der Grenzen zwischen maschinenlesbaren Texten und dem Inhalt von Datenbanken – an sich schon sehr viel länger existieren. Schon in den 80er Jahren gab es im Bereich der historischen Forschung an der Universität Cambridge ein System, das sich zum Ziel gesetzt hatte historische Dokumente unverändert zu transkribieren und durch eingesetzte Steuersequenzen so aufzubereiten, dass die Texte gleichzeitig auch als Inhalte von Datenbanken verwendet werden konnten (King, Tim J.: *The Use of Computers for Storing Records in Historical Research*. In: *Historical Methods* Bd. 14, 1981, S. 59– 64.). Und wenig später fanden im Bereich der Arbeiten des Verfassers Entwicklungen statt, die ähnliche verallgemeinerte Möglichkeiten im Bereich der PC-Anwendungen boten (Martin Gierl et al.: *StanFEP: Ein Arbeitsbuch*, St. Katharinen, 1990 =Halbgraue Reihe zur Historischen Fachinformatik, Bd. A6) .

Die Bedeutung von XML liegt also nicht so sehr darin, dass dadurch grundsätzlich Neues geschaffen würde: Sie liegt vielmehr darin, dass Überlegungen, die bisher als exotische Randbereiche der Forschung im Grenzbereich von Geisteswissenschaften und Informatik erschienen, plötzlich überraschend nahe an einer der derzeitigen Hauptentwicklungsrichtungen der Informationstechnologie liegen.

XML ist nicht, wie HTML, eine „Sprache“ um bestimmte Formen der Auszeichnung von Dokumenten zu ermöglichen, sondern vielmehr eine „Sprache“ um Auszeichnungssysteme zu definieren. In der Tat gibt es in den für die weitere Entwicklung des Internet zuständigen Gremien mittlerweile eine ganz bestimmte in XML spezifizierte Definition eines Auszeichnungssystems zur Präsentation von Inhalten in WWW-Seiten, welche die Regeln von HTML als einem ganz speziellen Sonderfall von XML wiedergibt. Ein derartiges Regelsystem – eine so genannte „Document Type Definition“ (DTD) kann verwendet werden um das visuelle Aussehen von Dokumenten zu beschreiben. Es kann aber auch völlig anderen Zwecken dienen. Beispielsweise verwendet

die unter [http://www. mpier.uni-frankfurt.de/dlib/](http://www.mpier.uni-frankfurt.de/dlib/) erreichbare Bibliothek privatrechtlicher Literatur – mit in absehbarer Zeit 1,2 Millionen Seiten die derzeit größte ihrer Art in der Bundesrepublik – eine DTD um Regeln zu formulieren, wie Inhaltsverzeichnisse von Zeitschriften so umgesetzt werden können, dass sie als Suchbehelfe in digitalen Bibliotheken verwendet werden können. Für viele Gebiete, auch für viele des kulturellen Bereichs, gibt es in der Zwischenzeit solche DTDs, die Regeln vorgeben, wie bestimmte Inhalte in XML ausgezeichnet werden sollen: so etwa die Ebind DTD (<http://sunsite.berkeley.edu/Ebind/>), die der erwähnten Zeitschriftenbibliothek zu Grunde liegt; die „Encoded Archival Description“ (EAD: <http://www.loc.gov/ead/>), die versucht universelle Regeln für die Vorbereitung von archivarischen Findbüchern durch die Rechner-technologie bereitzustellen; die „Text Encoding Initiative“ (TEI: <http://www.tei-c.org/>), die den bescheidenen Anspruch erhebt ein Modell für die Vorbereitung beliebiger geisteswissenschaftlicher Texte für beliebige geisteswissenschaftlich relevante Bearbeitungen durch den Rechner bereitzustellen (ein Beispiel für eine Datenbank, die auf Dokumenten aufsetzt, die nach diesen Regeln kodiert sind, findet sich unter [http://www.ceec.uni-koeln.de.](http://www.ceec.uni-koeln.de/)). Und auch das Marburger Midas System bietet in seiner 4. Auflage eine Abbildung der bisher verwendeten Datenstrukturen in XML an: nochmals eine nachdrückliche Bestätigung, dass nicht nur Inhalte, die im Internet dargestellt werden sollen, sondern nahezu beliebige Informationsstrukturen in XML ausgedrückt werden können.

Lautet die Antwort auf alle Fragen der Verarbeitung museumsrelevanter Informationen also „Hast Du Daten, kodiere Sie in XML und alle Probleme sind gelöst“? Wenn man die Diskussionen der letzten Zeit in manchen Bereichen der Informationstechnologie verfolgt, hat man in der Tat manchmal den Eindruck, dass dies die derzeitige gängige Meinung sei.

Darum eine bewusst scharf formulierte Warnung: Die Hoffnung, wenn man Daten nur in XML kodiere – und sei es in Form einer der zitierten DTDs –, dann werde es schon Software geben, die sie verarbeiten könne, erinnert ein wenig an die Voodoo-Idee, wenn man eine Puppe durch eine Nadel durchbohrt, dann werde sich schon ein Geist finden, der den dadurch ausgedrückten frommen Wunsch in die Wirklichkeit umsetzt. Gerade weil sich XML eignet, um Informationsstrukturen stark unterschiedlicher Charakteristika auszudrücken, kann keineswegs garantiert werden, dass eine beliebige in XML ausgedrückte Struktur von beliebiger Software auch tatsächlich verstanden wird. Dies ist leider ein Punkt, der in der heutigen Debatte auch von IT-Spezialisten gerne übersehen wird; wenn eine Softwareumgebung vorliegt, die in der Lage ist Strukturen zu bearbeiten, die durch die Markupsprache ausgezeichnet werden, ist XML aber ein vorzügliches Mittel um Zukunftssicherheit zu gewährleisten, da die Daten dann eben von allen Softwareprodukten verstanden werden, die diese Struktur bearbeiten (und XML Daten verarbeiten) können.

Da dieser Punkt „nicht jedes Softwareprodukt versteht jede Struktur; wenn eine Struktur verarbeitet werden kann, dann aber durch jede Software, die die Struktur versteht“ erfahrungsgemäß ein äußerst schwer verständliches, aber sehr wichtiges Prinzip ist, soll es an einem Beispiel aus der Textverarbeitung erläutert werden. Auch heute noch versteht nicht jedes Textverarbeitungsprogramm, was eine Fußnote ist. Versteht es das nicht, fallen all die angenehmen Funktionen von Programmen, die dies tun, flach: Man kann zwar einen Textblock in einer anderen Schriftart an das Ende einer Seite setzen, der dann wie eine Fußnote aussieht. Für das Textverarbeitungsprogramm ist das aber keine Fußnote, das bedeutet beispielsweise, dass später davor eingegebener Text dafür sorgt, dass, was wie eine Fußnote aussieht, an den Anfang der Folgeseite geschoben wird. Versteht ein Textverarbeitungsprogramm das Konzept „Fußnote“ nicht, ist es völlig belanglos, auf welche Weise man versucht den visuellen Effekt herbeizuführen – der gewünschte Verarbeitungseffekt tritt nie ein. Analog dazu: versteht eine Datenbank das Konzept „hierarchische

Beziehung“ nicht, so ist es völlig egal, ob man versucht die Tatsache, dass ein Beschreibungselement eine Untergliederung der Beschreibung des Gesamtobjektes ist, in XML auszudrücken oder durch inbrünstiges Gebet. Die Software verarbeitet keine Struktur, die sie nicht auf abstrakter Ebene „versteht“, technischer: die dem ihr zu Grunde liegenden Datenmodell nicht entspricht.

Andererseits: viele von uns waren irgendwann einmal gezwungen von einer Generation von Textverarbeitungsprogramm zu einer anderen überzugehen. Vor allem in früheren Jahren war oft die einzige Möglichkeit den Text aus dem älteren der beiden Systeme in eine ASCII- Datei auszugeben und in das neuere der beiden dann in dieser Form einzulesen – und wenn wir heute versuchen dasselbe zu tun, sind wir oft immer noch in derselben Situation, es sei denn, das neuere System versucht das ältere aus dem Markt zu drängen und bietet daher (fast immer unvollständige) Optionen dessen spezifische oder „proprietäre“ Dateien zu lesen. Tun wir dies, sind wir genau in der Ausgangslage: Der Text der Fußnoten ist zwar noch da, sie verhalten sich aber nicht als Fußnoten, sondern als etwas anders aussehende Textblöcke.

Das Versprechen von XML liegt nun einfach darin, die „Eigenschaft“ eines Textteils einer Fußnote darzustellen, sozusagen zu vererben – auch über einen neutralen, von keinem spezifischen Softwaresystem vorgeschriebenen Zustand, wie einer ASCII-Datei, hinweg. Und diese Eigenschaft kann dann von jeder Software wieder verstanden werden, die das Konzept Fußnote „versteht“, genau wie jede Software, die in der Lage ist ASCII-Dateien zu lesen, ASCII-Dateien von allen Programmsystemen lesen kann, die sie erzeugen können.

Technischer Hintergrund II: Das Document Object Model

Ob Datenbanken auf schwierigeren oder einfacheren Strukturen aufbauen als Texte mit mehrschichtigen Verweissystemen, ist ein Frage, deren Beantwortung meist mehr davon abhängt, womit der Antwortende besser vertraut ist, als mit objektiven Kriterien. Tatsächlich ist es aber so, dass die Vorherrschaft des relationalen Datenmodells in den letzten Jahren dazu geführt hat, dass zumindest Datenbanken, die hierarchische Strukturen enthalten, als wesentlich komplexer erscheinen als Texte.

Dieses Dilemma – es ist schön eine Struktur in XML auszeichnen zu können, aber keine Garantie, dass Software existiert, die sie verarbeiten kann – trifft uns daher mit voller Härte, wenn wir versuchen die Verschmelzung der Grenzen zwischen Volltexten und Datenbanken auszunutzen, die wir in den einleitenden Beispielen beschrieben haben. Nun existiert offensichtlich Software, die diese Probleme löst – anders wäre die Existenz der angeführten URLs nicht zu erklären. Doch diese Software – vom Verfasser als Teil seiner Forschungstätigkeit entwickelt – interessiert uns hier nicht in erster Linie. Denn: so lange es eben nur diese Software wäre, die die diskutierten Leistungen böte, ginge der wesentliche Vorteil, den die XML Welt bietet – problemloser Überstieg von einer Softwaregeneration auf eine andere – ja wieder verloren.

Die zweite entscheidende Entwicklung, die den Verfasser zu seinen abschließenden Vermutungen inspiriert, ist daher wesentlich allgemeiner: Im so genannten „W3C“, also jenem Komitee, das für die Koordination oder zumindest Dokumentation der Weiterentwicklung des World Wide Web zuständig ist, wird sei einiger Zeit intensiv an dem so genannten Document „Object Model“ oder DOM gearbeitet. (Der Vollständigkeit halber: <http://www.w3.org/DOM/> – für die Majorität der Leser dieses Textes aber mit Sicherheit zu technisch.) Dahinter verbirgt sich vordergründig ein Standard, der beschreibt, wie man XML- aus HTML-Seiten heraus optimal nutzt, mit starker Anlehnung an die Java-Programmierung. Entscheidend ist jedoch, dass dieser Standard mit einer

nur ganz geringfügigen konzeptuellen Erweiterung so etwas wie einen allgemeinen Bauplan für die Erstellung von Programmen darstellt, welche die eingangs als Beispiel beschriebene Verschmelzung zwischen stark formalisierten Datenbanken und weitgehend offenen Texten ermöglicht.

Bedeutung: Einige abschließende Thesen

In vollem Bewusstsein, wie gefährlich dies in der Informationstechnologie ist, lassen Sie mich abschließend thesenhaft einige Vorhersagen wagen:

- Durch die Entwicklung des Internet hin zur Aufbereitung immer größerer Textmengen einerseits, immer größerer Datenmengen andererseits verschieben sich die Grenzen zwischen der Textverarbeitung und der Datenbanktechnologie.
- Durch die voneinander unabhängige, aber aufeinander bezogene Entwicklung von Auszeichnungssprachen wie XML und von Konzepten für ihre von spezifischen Softwaresystemen unabhängige Nutzung, wie dem DOM, wird es in naher Zukunft zunehmend zur Entstehung von Softwareprodukten kommen, die typische Datenbankdienstleistungen für Sammlungen von stark textorientierten Dokumenten anbieten.
- Da gleichzeitig ein erhebliches Interesse daran besteht Dokumente aus höchst unterschiedlichen Systemen, die wenig oder gar keine terminologische Standardisierung aufweisen, gemeinsam in integrierten Systemen zu verwalten, wird die Zahl von dafür bereitgestellten technischen Instrumenten stark zunehmen.
- Für große, nur zum kleinen Teil dokumentierte Objektsammlungen wäre es daher von großem Vorteil bewusst „flache“ Beschreibungsmodi zu entwickeln, also solche, welche die rasche Anfertigung von Minimalbeschreibungen erlauben, die später ergänzt und/oder mit Softwareunterstützung miteinander verbunden werden können.