

## Zeitliche Verwaltung XML-basierter Metadaten in digitalen Bibliotheken

Markus Kalb, Günther Specht

Universität Ulm  
Abt. Datenbanken und Informationssysteme  
{kalb, specht}@informatik.uni-ulm.de

**Zusammenfassung:** In digitalen Bibliotheken hat sich die Verwendung von XML als Austauschformat für Metadaten etabliert. Zunehmend kommt die Anforderung hinzu, auch komplexe zeitliche Daten und Anfragen innerhalb von XML effizient zu unterstützen. Allerdings sind die heutigen Möglichkeiten der XML-Technologien für eine zeitliche Verwaltung von Daten nicht ausreichend. Das im folgenden vorgestellte Modell T-XPath erweitert das Datenmodell und die Anfragesprache XPath um eine flexible und effiziente zeitliche Verwaltung, die auch unscharfe und ungenaue zeitliche Information berücksichtigt. Zusätzlich werden verschiedene Implementierungsvarianten von T-XPath vorgestellt und ihre jeweiligen Vor- und Nachteile hinsichtlich Performanz, Effizienz und Realisierbarkeit diskutiert. Damit wird eine korrekte zeitliche Annotation von Metadaten sowohl innerhalb digitaler Bibliotheken als auch in ihren Austauschformaten erreicht.

**Abstract (Extended):** The use of XML as an interchange format for meta data in digital libraries is commonly established. The requirement to manage complex temporal data and queries effective inside XML is increasingly important. Temporal meta data arises especially from historical data or series of measurements in digital libraries and E-Journals as well as in their user management. The possibilities of XML especially XPath and XSchema, are insufficient for a robust and flexible representation and handling of dates or rather nonexistent for a temporal management of information. In the following we present a temporal data model T-XPath that extends the data model and query language XPath, by a flexible and robust temporal management. The concept based on abstract temporal data types (ADT), which encapsulate the entire history of a value. Additionally supported are indeterminacy and fuzzy dates. The query language is downward compatible to XPath and supports temporal queries by new temporal operations and functions. Thus a correct temporal annotation of meta data is guaranteed both inside digital libraries and in its interchange formats. Consequently, it also saves the quality and history of data. In addition to the fundamentals of T-XPath, different variants for its implementation and its respective advantages and disadvantages are discussed. Thereby, the points performance, efficiency and short-term realisation are also captured.

### 1 Einleitung

XML hat sich als Austauschformat für Metadaten in digitalen Bibliotheken etabliert. Zunehmend kommt die Anforderung hinzu, auch komplexe zeitliche Daten und Anfragen innerhalb von XML effizient zu unterstützen. Zeitliche Metadaten fallen insbesondere an bei zeitbasierten (z.B. historischen) Daten oder Messreihen in digitalen Bibliotheken und E-

Journals sowie bei der Benutzerverwaltung in digitalen Bibliotheken.

Für XML-Dokumente wurde vom W3C die Anfragesprachen XPath und (darauf aufbauend) XQuery vorgeschlagen [W3C02]. Die bisherigen Möglichkeiten von XML, insbesondere XPath und XSchema, sind für eine robuste und flexible Behandlung von Zeitangaben nicht ausreichend bzw. für eine zeitliche Verwaltung von Information nicht vorhanden.

Das im folgenden vorgestellte zeitliche Modell T-XPath erweitert das Datenmodell und die Anfragesprache XPath um eine flexible und effiziente Verwaltung und Abfragemöglichkeit von zeitlichen Informationen.

Das Papier ist wie folgt gegliedert: In Abschnitt 2 werden die bisherigen Möglichkeiten von XML zur Unterstützung von zeitlichen Informationen dargestellt. In Abschnitt 3 wird nach Betrachtung der grundlegenden Aspekte einer zeitlichen Verwaltung, das temporale Datenmodell und die Anfragesprache, T-XPath, vorgestellt. In Abschnitt 4 werden verschiedene Implementierungsansätze für T-XPath vorgestellt und diskutiert.

## 2 Unterstützung von zeitlichen Informationen in XML

Bei der Betrachtung der bisherigen Möglichkeiten zur Unterstützung von zeitlichen Informationen in XML liegt der Fokus auf den Zusatztechnologien zu XML, XSchema und XPath. XPath ist eine Anfragesprache zur Navigation innerhalb der hierarchischen Struktur eines XML-Dokumentes, zur Selektion von Teilen des Dokumentes sowie zur Manipulation der selektierten Daten [W3C02]. Im Folgenden wird die derzeit noch in Arbeit befindliche Version 2.0 zugrunde gelegt.

In XPath werden Zeitangaben durch die Verwendung der einfachen zeitlichen Datentypen aus XSchema definiert: *duration*, *dateTime*, *time*, *date*, *gYearMonth*, *gYear*, *gMonthDay*, *gMonth*, *gDay*. Diese bieten lediglich eingeschränkte Möglichkeiten zur Behandlung von zeitlichen Angaben. So sind die einzelnen Datentypen entsprechend dem gregorianischen Kalender definiert. Jeder Datentyp entspricht einer anderen Granularität des Kalenders. Das hat vielfältige Nachteile: Vergleiche zwischen den unterschiedlichen Datentypen werden dadurch erschwert, Änderungen des Datenformates oder des Kalenders gestalten sich aufwendig, beliebige zeitliche Perioden, wie beispielweise "03.10.01 - 06.10.01", müssen explizit modelliert werden. Des Weiteren werden Aggregationen (Coalescing) von zeitlichen Perioden nicht direkt unterstützt. Ebenfalls problematisch erweist sich die Abhängigkeit der Operationen von dem Datumsformat der Zeitangaben (z.B. *add-year-MonthDuration-to-dateTime(A,B)*). Schließlich sind ungenaue und unscharfe Zeitangaben nicht definierbar.

Die bisherigen Möglichkeiten von XPath und XSchema sind für eine robuste und flexible Behandlung von Zeitangaben nicht ausreichend bzw. für eine zeitliche Verwaltung von Information nicht vorhanden. Das im folgenden vorgestellte Modell T-XPath, erweitert die XML-Technologie um eine effiziente Verwaltung von komplexen zeitlichen Informationen, durch Einführung neuer zeitlicher Datentypen sowie ein Konzept ihrer Interaktionen mit konkreten Werten.

## 3 Zeitliche Verwaltung mit T-XPath

### 3.1 Neue zeitliche Datentypen für Zeitangaben in T-XPath

Die drei zeitlichen Datentypen, *Zeitdauer (interval)* zur Charakterisierung der Dauer eines Ereignisses, *Zeitpunkt (instant)* für ein Ereignis mit der Dauer eines Chronons<sup>1</sup> und *Zeitraum (period)* zur Angabe eines Zeitbereiches während dem ein Ereignis stattgefunden hat, bilden die Grundlage von T-XPath.

Alle zeitliche Datentypen legen Chronons zugrunde. Sie sind somit unabhängig von einem spezifischen Kalender und die Umrechnung in konkrete Kalendarien erfolgt über Abbildungsfunktionen. Zusätzlich unterstützen alle Datentypen verschiedene Formen von Unschärfen wie sie durch die Messung der Zeit oder durch unzureichende zeitliche Informationen entstehen können. Beispielsweise, entspricht eine ungenaue *Zeitdauer* der Aussage "das Ereignis dauerte zwischen 5 und 7 Stunden". Dies wird im Datentyp *interval* durch Minimum und Maximum von möglichen Werten angegeben. Die Einbeziehung unscharfer *Zeitangaben* erfordert jedoch eine Erweiterung einiger Operationen. Insbesondere boolesche Operationen müssen um einen dreiwertigen booleschen Datentyp (*3-bool* [CP01]) ergänzt werden, da aufgrund der Unschärfe nicht immer nach *True* oder *False* ausgewertet werden kann, sondern auch nach *Vielleicht (Maybe)*.

Die drei Datentypen *interval*, *instant* und *period* sowie ihre zugehörigen Operationen (z.B. arithmetische, Vergleichs- und Cast-Operationen) ersetzen die bisherigen neun zeitlichen Datentypen von XSchema respektive XPath. Eine ausführlichere Betrachtung aller Komponenten von T-XPath findet sich in [KSS03].

### 3.2 Die abstrakten zeitlichen Datentypen von T-XPath

Die Grundlage der zeitlichen Verwaltung in T-XPath bilden abstrakte Datentypen (ADT). Diese Form des Attributzeitstempelverfahren [Sk97, JE00], kapselt die gesamte Geschichte eines Wertes innerhalb eines ADTs [Er99, Gü00, CR01]. Ein ADT erweitert einen nicht-zeitabhängigen Datentyp um eine zeitliche Verwaltung, wobei dessen ursprüngliche Eigenschaften bestehen bleiben und lediglich um neue zeitliche Eigenschaften ergänzt werden. Die gekoppelte Speicherung von konkreten Werten und Zeitangaben in einem Datentyp ermöglicht effiziente Algorithmen, insbesondere für Operationen, die gleichzeitig auf zeitlichen und konkreten Werten operieren (z.B. Änderungsrate eines Wertes). Diese Operationen waren mit den bisherigen Verfahren nur schwer oder nicht effizient lösbar [Er99].

Die zeitliche Verwaltung mit ADTs erweitert lediglich das Typsystem von XPath bzw. XSchema. D.h. es werden neue zeitliche Datentypen und Operationen eingeführt, die in XSchema verwendet werden können. Die Baumstruktur des zugrundeliegenden XPath-Datenmodells wird durch die zeitliche Verwaltung nicht beeinflusst. In T-XPath werden für alle bisherigen Datentypen (z.B. *string*, *integer*, etc.) drei korrespondierende abstrakte zeitliche Datentypen eingeführt, die deren Gültigkeitszeit- (*valid time*), wann war ein Wert in der realen Welt gültig, Aufzeichnungszeit- (*transaction time*), wann wurde der Wert ge-

<sup>1</sup> Ein Chronon entspricht der kleinsten nicht weiter teilbaren, diskreten Einheit einer eindimensionalen, linearen Zeitlinie.

speichert oder bitemporale Entwicklungsgeschichte, Gültigkeit- und Aufzeichnungszeit werden zusammen betrachtet, repräsentieren, z.B. für den Datentyp *string*: *v\_string* (Gültigkeitszeit), *t\_string* (Aufzeichnungszeit) und *vt\_string* (bitemporale Zeit).

Die jeweiligen Eigenschaften der Zeitarten, beispielsweise keine Lücken in der Aufzeichnungszeit, sind für jeden der Datentypen formal definiert und gewährleisten eine konsistente zeitliche Verwaltung. Weiterführende Eigenschaften der zeitlichen Verwaltung werden bei der Modellierung der Anwendung über zusätzliche zeitliche Eigenschaftsattribute des ADT explizit festgelegt und bei der Instanziierung von dem ADT überprüft, z.B. ob Überlappungen der zeitlichen Angaben zulässig oder nicht gewünscht sind. Auf einem ADT sind alle Operationen seines ursprünglichen Datentyps sowie alle Operationen der zeitlichen Datentypen definiert. Der Unterschied zu den originalen Operationen liegt lediglich in dem neuen Ergebnistyp. Das Ergebnis kann durch die Mengenwertigkeit der ADTs mehrere Elemente enthalten, die zusammen wiederum einen instantiierten ADT bilden. Hierfür stehen ebenfalls Operationen zur Verfügung (z.B. *count()*, *last()*, *next()*) die den Umgang mit der Mengenwertigkeit erleichtern.

### 3.3 Repräsentation der abstrakten Datentypen in XML und XSchema

Für T-XPath sind die abstrakten zeitlichen Datentypen in XSchema integriert und können durch den Anwender analog zu den einfachen Datentypen bei einer Modellierung verwendet werden. Im Unterschied zu diesen ist ihre Struktur komplexer. In Abbildung 4.2 ist beispielhaft die Anwendung eines ADTs, d.h. eines Strings mit Gültigkeitszeit (*v\_string*), aufgezeigt. In dem Beispiel unterliegt die Auflage eines Buches einer zeitlichen Entwicklung.

<pre> ... &lt;!-- Anwendungsbeispiel eines v_string --&gt; &lt;xs:complexType name="Buch"&gt;   &lt;xs:sequence maxOccurs="unbounded"&gt;     &lt;xs:element name="Titel" type="string"/&gt;     &lt;xs:element name="Auflagen" type="v_string"       value="Auflage" time="valid_time"       format="MM.YYYY"&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; ... </pre>	<pre> ... &lt;Buch&gt;   &lt;Titel&gt; Physik I &lt;/Titel&gt;   &lt;Auflagen&gt;     &lt;Auflage&gt;1. Auflage&lt;/Auflage&gt;     &lt;valid_time&gt;02.1970-09.1975&lt;/valid_time&gt;     &lt;Auflage&gt;2. Auflage&lt;/Auflage&gt;     &lt;valid_time&gt;10.1975-05.1976&lt;/valid_time&gt;     &lt;Auflage&gt;3. Auflage&lt;/Auflage&gt;     &lt;valid_time&gt;06.1976-uc&lt;/valid_time&gt;   &lt;/Auflagen&gt; ... &lt;/Buch&gt;... </pre>
--	---

Abbildung 3.1: Beispiel für die Repräsentation des abstrakten Datentyp *v\_string* in XSchema (links) und XML (rechts)

Auf der linken Seite in der oberen Abbildung ist die Struktur des XML-Dokumentes definiert. Dabei repräsentiert z.B. der Eintrag `<xs:element name="Auflagen" type="v_string" . . . >` die Auflage eines Buches und somit den Wert der zeitlich verwaltet wird. Die Attribute `time="valid_time"` und `value="Auflage"` bestimmen, dass die Bezeichnung einer Buchauflage im XML-Dokument durch das Tag `<Auflage>` und die Gültigkeitszeit des Verwendungszwecks durch das Tag `<valid_time>` gekennzeichnet ist. Das Datumsformat für Zeitangaben ist hier durch den Eintrag `format="MM.YYYY"`

definiert. Die Repräsentation des Beispiels in einem XML Dokument ist auf der rechten Seite der Abbildung dargestellt. Alle im Laufe der Geschichte aufgetretenen Auflagen eines Buches und deren jeweilige Gültigkeitszeiten sind vollständig in dem XML-Dokument enthalten. Damit sind auch Drucke und Auflagen historischer Bücher modellierbar zu denen es keine exakte Auflagenzählung (z.B. Abdruck in „Augsburg“) und kein exaktes Auflagedatum gibt (z.B. nur bekannt, daß es im 16 Jhr. war).

Die XML Repräsentation der Daten ist abwärtskompatibel zum XPath-Datenmodell. XPath könnte die zeitlichen Daten repräsentieren, allerdings mit der Einschränkung, dass keine zeitlichen Anfragen und Konsistenzbedingungen möglich bzw. überprüfbar wären.

### 3.4 Die Anfragesprache von T-XPath

Die Anfragesprache von T-XPath ist für nicht-zeitliche Anfragen identisch mit der Anfragesprache X-Path. Für die Verarbeitung zeitlicher Anfragen ist sie um eine Reihe neuer zeitlicher Operationen und Funktionen (siehe Abschnitt 3.2 und Abschnitt 3.6) erweitert. Die zeitliche Unterstützung von T-XPath konzentriert sich auf die Prädikate, die mittels zeitlicher Ausdrücke eine Knotenmenge weiter verfeinern. Anhand einiger Beispiele wird im folgenden die Sprache von T-XPath näher vorgestellt

Die nachfolgenden Anfragen beziehen sich auf das Beispiel von Abschnitt 3.3. Die folgende Anfrage liefert alle Bücher die im Jahre 1978 eine Auflage hatten und diese exakt bekannt war.

```
//Buch[Auflagen valid1 '1978' = true]
```

Für den unsicheren Fall, z.B. bei historischen Büchern bei denen das Auflagedatum nur ungenau bekannt ist, müsste die Anfrage folgendermaßen umformuliert werden:

```
//Buch[Auflagen valid '1750' = maybe]
```

Im Ergebnis sind ausschließlich die Bücher enthalten, bei denen aufgrund unscharfer Zeitangaben für das Jahr 1750 zumindest die Möglichkeit bestand, dass sie zu dieser Zeit eine entsprechende Auflage besaßen.

Die bisher vorgestellten Anfragen wurden ausschließlich auf den zeitlichen Werten durchgeführt. Eine Anfrage, die zusätzlich konkrete Werte berücksichtigt, sieht wie folgt aus.

```
//Buch[Auflagen valid '1978' and Auflagen = '3.Auflage']
```

In der Anfrage wurde für das zeitliche Prädikat kein Wert angegeben. In diesem Fall wird *true* und *maybe* angenommen. Das Ergebnis bilden diejenigen Bücher, die 1978 bereits die 3. Auflage hatten. Durch die Auswertung nach *maybe* sind auch jene Bücher enthalten von denen es nicht genau bekannt ist.

---

<sup>1</sup>. Die Operation `valid()` setzt sich aus mehreren Vergleichsoperationen zusammen und vergleicht Zeitangaben ob diese mindestens ein gemeinsames Chronon besitzen.

## 4 Implementationsaspekte von T-XPath

Für die Realisierung von T-XPath ergeben sich zwei verschiedene Varianten der Architektur und Implementierung. In der ersten Variante erfolgt eine vollständige Integration in die XML-Speicher- und Auswertssysteme bzw. Programmbibliotheken. In der zweiten Variante erfolgen die temporalen Erweiterungen nicht in den bestehenden Systemen selbst, sondern werden über eine zusätzliche Schicht bereitgestellt, welche die temporale Funktionalität auf bestehende XML-Standards abbildet. In Anlehnung an die Entwurfsprinzipien für temporale Datenbanken [Je00] wird diese Variante auch als „On Top“ Architektur bezeichnet. Beide vorgestellte Varianten haben ihre spezifischen Vor- und Nachteile die im folgenden näher betrachtet werden.

### 4.1 Integrierte Architektur

Bei der integrierten Architektur erfolgt eine vollständige Integration der temporalen Funktionalität in die XML-Speicher- und Auswertungs-systeme. Auf Grund der unterschiedlichen Komponenten von T-XPath sind die Implementierungen der verschiedenen XML-Standards (XPath, XQuery, XSchema, XSLT etc.) von der Erweiterung betroffen. Die neuen Datentypen für Zeitangaben (z.B. *period*) und die Entwicklungsgeschichte von Werten (z.B. *t\_string*) stehen dann als Basisdatentypen in der Erweiterung von XSchema zur Verfügung. Die neuen zeitlichen Ausdrücke und Operationen von T-XPath werden durch eine Erweiterung der bisherigen Implementierungen der Anfragesprache XPath erreicht.

#### 4.1.1 Vor- und Nachteile der integrierten Architektur

Der Vorteil dieser Architektur liegt in der direkten Unterstützung und Integration von zeitlichen Daten in den einzelnen XML-Komponenten. Hierdurch können konkrete Implementierungen eine effiziente Repräsentation von zeitlichen Datentypen und eine optimierte Ausführung von T-XPath Anfragen und Operationen realisieren. Nachteilig ist, dass fast alle bisherigen Komponenten und bestehenden Systeme reimplementiert werden müssen. Da solche Systeme meist nicht in der Hand des Entwicklers liegen, stellt sich eine Erweiterung dieser als sehr aufwendig oder nicht möglich heraus. Dann könnten aufgrund der Abwärtskompatibilität von T-XPath zwar T-XPath XML-Dokumente in einer Datenbank gespeichert, aber keine zeitlichen Anfragen oder Operationen an die gespeicherten Dokumente gestellt werden.

### 4.2 „On Top“ Architektur

Die „On-Top“ Architektur stellt die temporale Funktionalität mit Hilfe einer zusätzlichen Zwischenschicht, welche die zeitliche Aspekte auf bestehende Techniken abbildet, bereit. Die Zwischenschicht verfügt über eine Ein- und Ausgabeschnittstelle, welche die Kommunikation zu den einzelnen Anwendungen realisiert sowie die Verbindung zu den zugrundeliegenden Techniken bildet. Die Schicht ist jedoch nicht als reiner Konvertierer zu verstehen, sondern realisiert die vollständige zeitliche Unterstützung, unter Abstützung

auf ein darunterliegendes XPath-Modell.

In Abbildung 4.1 wird die Architekturform schematisch verdeutlicht. Die Idee der Zwischenschicht ist es auf Grundlage existierender Implementierungen von XML-Standards, eine zeitliche Unterstützung für XML-Dokumenten zu realisieren. Alle zeitlichen Anfragen werden an die Zwischenschicht gestellt und durch die Schicht in X-Path konforme Ausdrücke übersetzt und an das darunter liegende XML-Dokument geleitet. Das Ergebnis wird von der Zwischenschicht entsprechend aufbereitet und als Ergebnismenge der Anfrage übergeben.

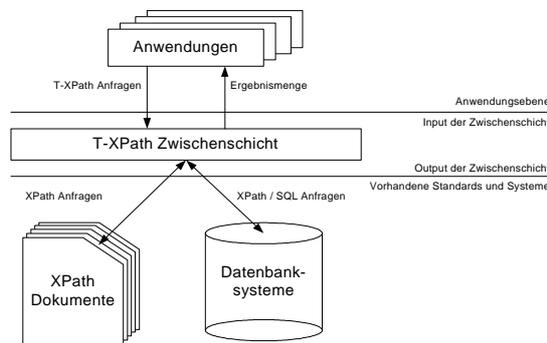


Abbildung 4.1: T-XPath als „On-Top“ Architektur

Die Implementierung der Zwischenschicht kann auf unterschiedliche Art und Weise erfolgen, abhängig von der jeweiligen Anwendungsumgebung:

- Die Zwischenschicht befindet sich direkt zwischen den Anwendungen und der XML-Dokumentenschicht. Somit handelt es sich um eine eigenständige zusätzliche Schicht, welche z.B. sowohl innerhalb von digitalen Bibliotheken, als auch beim Datenaustausch zwischen diesen verwendet werden kann. Die Architekturform bietet sich insbesondere für den direkten Umgang mit XML-Dokumentensysteme bzw. XML-basierten Datenbanken an.
- Die Zwischenschicht wird in eine bereits vorhandene Middleware zwischen der Anwendung und den XML-Dokumenten integriert. Eine solche Middleware existiert z.B. bei der persistenten Speicherung von XML-Dokumenten in relationalen Datenbanksystemen und realisiert dabei eine Mappingschicht, welche ein XML-Dokument in relationale Tabellen abbildet. In diese Mappingschicht wird die zeitliche Funktionalität integriert, wobei es hierbei wiederum zwei Integrationsvarianten existieren. In Variante (1) werden die T-XPath Anfragen zunächst in XPath Anfragen umgewandelt und anschließend durch die Mappingschicht z.B. in SQL-Anfragen konvertiert. In der (2) Variante erfolgt die Umwandlung unmittelbar von T-XPath Anfragen in SQL-Anfragen. Durch die Erweiterung der Middleware können dann auch Datenbanken bzw. Dokumentenformate angesprochen werden, die ursprünglich kein XML unterstützen bzw. nicht XML-konform sind.

#### 4.2.1 Vor- und Nachteile der „On-Top“ Architektur

Aus dem „On-Top“ Ansatz ergibt sich der Vorteil, dass bereits vorhandenen XML-Standardimplementierungen ohne Probleme weiterverwendet werden können und eine temporale Erweiterung ohne Modifikation der bereits bestehenden Umgebung möglich ist. Im Gegensatz zur integrierten Architektur sind mit dieser Architektur auch die persistente Speicherung von XML-Dokumenten mit zeitlichen Anfragen über diesen realisierbar. Je nach Implementierungsvariante für die Zwischenschicht eignet sich die „On-Top“-Architektur sowohl für XML-basierte Austauschformat als auch für die Speicherung von zeitabhängigen XML-Daten. Allerdings führt diese Form der Architektur zu einer schlechteren Performanz gegenüber der integrierten Architektur. Dadurch, dass die verwendeten Techniken nicht modifiziert wurden, befindet sich die zeitliche Unterstützung ausschließlich in der Zwischenschicht. Die tieferliegenden Systeme besitzen keine zeitliche Unterstützung und können somit auch zeitliche Anfragen und Operationen nicht optimiert unterstützen.

### 5 Verwandte Arbeiten

Unsere Behandlung von Unschärfen ist eine Integration und Weiterentwicklung der Ansätze wie sie in der Literatur zu zeitlichen Primitiven zu finden sind [DS98, CP01, PT01]. Auf dem Gebiet der Repräsentation und Abfrage von zeitlichen Daten innerhalb von XML wurde bisher nur wenig Forschung betrieben. So präsentierten Grandi und Mandroli ein Modell mit expliziten Tupelzeitstempeln zur Beschreibung von Gültigkeitszeiten innerhalb von XML-Dokumenten [GM99]. In dem Modell wird kein Datentypkonzept verwendet, sondern die gesamte Funktionalität von zeitlichen Anfragen beruht auf parametrisierten XSL-Stylesheets. Alle zeitlichen Operation werden als Stylesheets definiert, die das ursprüngliche XML-Dokument, unter Berücksichtigung der zeitlichen Bedingung, in ein (Ergebnis-) XML-Dokument transformieren. Es handelt sich bei diesem Vorschlag weniger um eine Integration zeitlicher Funktionalität in XML, sondern mehr um eine Modellierung einer solchen mit Hilfe von XML. In der Arbeit von Amagasa et.al. wird eine temporale Erweiterung des XPath-Modells vorgeschlagen, in der lediglich den Kanten zwischen den Elementen ein Zeitstempel zugeordnet wird und nicht den Elementen selbst [AMU00]. Wie in der Arbeit von Kalb, Schneider und Specht gezeigt, eignet sich dies jedoch besser für die zeitliche Verwaltung der Struktur eines XML-Dokuments [KSS03]. Von Dyreson werden zusätzlich zu den Kanten die Knoten mit Zeitinformation behaftet und formal in einer Erweiterung des XPath-Modells vorgestellt [Dy01]. Der Fokus liegt dabei mehr auf der Transaktionszeit und ihrer impliziten Gewinnung unter Verwendung der Änderungszeit einer XML-Datei. Ein vergleichbarer Ansatz wird von Oliboni et. al. gewählt [OQT01], wobei die Abfragefunktionalität durch eine eigene SQL-artige Anfragesprache TS-QL realisiert ist. Die vorgestellten Architekturformen wurden in ähnliche Form in anderem Kontext von C. Jensen für die Realisierung von Temporalen Datenbanken vorgestellt [Je00].

## 6 Zusammenfassung

Wir haben gezeigt, wie XML-basierte Metadaten in digitalen Bibliotheken, zeitlich verwaltet werden können. Den Ausgangspunkt der Betrachtung bildet die Untersuchung der bisherigen Unterstützung von zeitlichen Information durch den heutigen Stand der XML-Technologie. Allerdings zeigte sich, dass sowohl die Datentypen für Zeitangaben als auch die Interaktion von Zeitangaben mit konkreten Werten den Ansprüchen einer robusten und flexiblen zeitlichen Verwaltung nicht gerecht werden konnten.

Eine effizientere und flexiblere Lösung wurde durch Datenmodell und Anfragesprache T-XPath vorgestellt, dass eine kalenderunabhängige Unterstützung sowohl von exakten als auch von unscharfen zeitlichen Angaben ermöglicht. Die Anfragesprache und die ADTs von T-XPath ermöglichen eine robuste und flexible Verwaltung von zeitlicher Information, wie sie für XML-Daten bisher nicht zur Verfügung stand. Dies dient somit auch der Sicherstellung der Qualität und Historisierbarkeit von Metadaten, sowohl innerhalb digitaler Bibliotheken als auch in ihren Austauschformaten.

Bei den Implementierungsvarianten von T-XPath gibt es mehrere Alternativen. Da der integrierte Ansatz Neuimplementationen aller beteiligten Systeme und Komponenten erfordern würde und damit eine kurzfristige Bereitstellung von zeitlicher Funktionalität in XML nicht möglich wäre, bildet der „On-Top“ Ansatz die bessere Alternative. Dabei wird eine neue T-XPath Zwischenschicht eingezogen, während darunterliegende Systeme und Komponenten gleich bleiben. Dieser Ansatz ermöglicht auch kurzfristig die Realisierung einer zeitlichen Unterstützung in XML.

## 7 Literaturverzeichnis

- [Al84] Allen, J.F. Towards a General Theory of Action and Time. *Artificial Intelligence*, 1984, pp. 123-154.
- [AMU00] Amagasa, T., Masatoshi, Y., Uemura, S. A Data Model for Temporal XML Documents. In *Proc. Int. Conf. on Database and Expert Systems Applications - DEXA 2000*, Springer Verlag LNCS 1873, London, 2000, pp. 334-344.
- [CP01] Combi, C., Pozzi, G. HMAP - A temporal data model managing intervals with different granularities and indeterminacy from natural sentences. *VLDB Journal* 9, 2001, pp. 294-311.
- [CR01] Chomicki, J., Revesz, P.Z. Parametric Spatiotemporal Objects. *Bulletin IA\*AI (Italian Association for Artificial Intelligence)*, Vol. 14, No.1, 2001
- [DS98] Dyreson, C.E., Snodgrass, R.T. Supporting Valid-Time Indeterminacy. *ACM Trans. Database Systems* 23(1), 1998, pp. 1-57.
- [Dy01] Dyreson, C.E. Observing Transaction-time Semantics with TT-XPath. *Proc. 2nd Int. Conf. on Web Information Systems Engineering (WISE2001)*, Kyoto, Japan, 2001, pp. 193-202.
- [Er99] Erwig, M., Güting R.H., Schneider M., Vazirgiannis M. An Approach to Modeling and Querying Moving Objects in Databases. In *GeoInformatica Vol.3*, 1999
- [GM99] Grandi, F., Mandreoli F. The Valid Web: it's Time to Go. *TimeCenter TR-46* ([www.cs.auc.dk/TimeCenter/](http://www.cs.auc.dk/TimeCenter/)), 1999.
- [Gü00] Güting, R.H., Böhlen, M.H., et.al. A Foundation for Representing and Querying Moving Objects. In *ACM Trans. on Database Systems Vol. 25 No. 1*, 2000, pp. 1-42.
- [Je00] Jensen, C.S. Temporal Database Management. PhD thesis, Univ. of Arizona, 2000.
- [KSS03] Kalb, M., Schneider K., Specht, G. T-XPath: Ein zeitliches Modell für XML-Datenbanken. In *Proc. Datenbanksysteme für Business, Technologie und Web, BTW-2003*, Gesellschaft für Informatik, Lecture Notes in Informatics Vol. P-26, 2003, pp. 157-166

- [OQT01] Oliboni, B., Quintarelli, E. and Tanca, L. Temporal aspects of semi-structured data. Proc. 8th Int. Symp. on Temporal Representation and Reasoning (TIME-01), 2001.
- [PT01] Pfoser, D., Tryfona, N. Capturing Fuzziness and Uncertainty of Spatiotemporal Objects. TimeCenter Technical Report, TR-59 ([www.cs.auc.dk/TimeCenter/](http://www.cs.auc.dk/TimeCenter/)), 2001
- [Sk97] Skjellaug, B. Temporal Data: Time and Object Databases. Technical report, University Oslo, April 1997.
- [Sn00] Snodgrass, R. T. Developing time-oriented database applications in SQL. Morgan Kaufmann Publishers, 2000.
- [W3C02] W3C. XQuery 1.0 and XPath 2.0 Data Model (Working Draft 16.8.2002)  
<http://www.w3.org/TR/query-datamodel/>