T-XPath: Ein zeitliches Modell für XML-Datenbanken

Markus Kalb[1], Kerstin Schneider[2], Günther Specht[1]

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

[2]European Media Laboratory GmbH (EML), Heidelberg
Kerstin.Schneider@eml.villa-bosch.de

Abstract: Bisherige XML-Datenbanken erlauben lediglich eine eingeschränkte Unterstützung von zeitlichen Daten und Anfragen. Das hier vorgestellte zeitliche Modell T-XPath erweitert das bisherige Datenmodell und die Anfragesprache XPath um eine flexible und effiziente Modellierung, Verwaltung und Abfragemöglichkeit von zeitlichen Informationen. Als Grundlage dienen abstrakte zeitliche Datentypen (ADT), die die gesamte zeitliche Entwicklungsgeschichte eines Wertes kapseln und zusätzlich auch unscharfe und ungenaue Zeitangaben berücksichtigen können. Die Anfragesprache von T-XPath ist voll abwärtskompatibel zu XPath und stellt für zeitliche Anfragen eine Reihe neuer Operationen und Funktionen zur Verfügung.

1 Einleitung

In vielen Anwendungsgebieten von XML-Datenbanken kommt zunehmend die Anforderung hinzu, auch komplexe zeitliche Daten und Anfragen zu unterstützen. Die bisherigen Möglichkeiten von XML, insbesondere XPath und XSchema, bieten hierfür jedoch lediglich eine sehr eingeschränkte Unterstützung.

Die Verwaltung und Verarbeitung von komplexer zeitlicher Information, welche zum Teil sogar unscharf und ungenau ist, wird beispielweise im Projekt GEIST[1] erforderlich [Kr01], das am European Media Laboratory GmbH (EML) in Kooperation mit dem Fraunhofer Institut für Graphische Datenverarbeitung sowie dem Zentrum für Graphische Datenverarbeitung in Darmstadt durchgeführt wird. Es hat die Entwicklung eines mobilen Augmented-Reality-Systems zum Ziel, welches es erlaubt, Geschichte an Ort und Stelle zu erleben und zu erfahren. Als Beispielszenario wurde der 30jährige Krieg (in Heidelberg) gewählt. Neben dem Erleben von interaktiven Erzählungen auf der Grundlage historischer Information, soll es den Benutzern möglich sein, jederzeit weitergehende historische Information zu erfragen.

Für XML-Datenbanken wurde vom W3C die Anfragesprachen XPath und (darauf aufbauend) XQuery vorgeschlagen [W3C02]. Im folgenden wird die Anfragesprache T-XPath vorgestellt, die XPath um eine flexible und effiziente zeitliche Verwaltung erweitert, die unscharfe und ungenaue zeitliche Information berücksichtigen kann.

Das Papier ist wie folgt gegliedert: In Abschnitt 2 werden die bisherigen Möglichkeiten von XPath zur Unterstützung von zeitlichen Informationen dargestellt. In Abschnitt 3 wer-

1. Das GEIST Projekt wird gefördert vom BMBF (01 IRA 12A, 01 IRA 12B, 01 IRA 12C) und von der Klaus Tschira Stiftung (KTS).

den neue, flexible zeitliche Datentypen vorgestellt, die zusammen mit dem in Abschnitt 4 definierten temporalen Datenmodell die Grundlage von T-XPath bilden. In Abschnitt 5 wird die Anfragesprache von T-XPath vorgestellt. Abschließend werden in Abschnitt 6 verwandte Arbeiten betrachtet.

2 Unterstützung von zeitlichen Information in 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. 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. Neue zeitliche Datentypen sowie ein Konzept ihrer Interaktionen mit konkreten Werten werden benötigt um komplexe zeitliche Informationen effizient zu verwalten.

3 Neue zeitliche Datentypen für Zeitangaben in T-XPath

Die in der letzten Dekade im Bereich temporale Datenbanken eingeführten Konzepte bilden die Grundlage für die nachfolgenden Definitionen der zeitlichen Datentypen in T-XPath. Dabei besitzen alle Typen die folgenden grundlegenden Eigenschaften:
Einen zeitlichen Wertebereich: Die Zeit wird als eine eindimensionale, lineare und geordnete Linie betrachtet. Diese ist in diskrete Einheiten, Chronons, unterteilt und ist isomorph zu den ganzen Zahlen $\mathbb{Z}$.
Kalenderunabhängigkeit: Allen Zeitprimitiven liegen Chronons zugrunde. Sie sind somit unabhängig von einem spezifischen Kalender. Umrechnungen in konkrete Kalendarien erfolgen über Abbildungsfunktionen.
Unschärfen: Zeitliche Information ist charakterisiert durch Ungenauigkeit und Unbestimmtheit, da (1.) die Messung und Speicherung der Zeit in Kalendern mit unterschiedlichen Basisgranularitäten zu Ungenauigkeiten führt und (2.) die Information über zeitliche Angaben oft nicht exakt vorhanden ist. Beispielsweise sind geschichtliche Ereignisse häufig ungenau überliefert: "Der Schlossbau begann zwischen 1500 und 1502". Beide Formen der Unschärfe werden durch die neuen Datentypen unterstützt.
Spezielle Variablen ermöglichen die besondere Behandlung bestimmter Zeitangaben.

Beispielsweise bezeichnet die Variable *now* die aktuelle Zeit. Die Variablen *until change* und *since beginning* beschreiben den Anfang und das Ende der Zeitlinie.

3.1 Atomare zeitliche Datentypen

Als Grundlage führen wir folgende drei zeitliche Datentypen in T-XPath ein: Zeitdauer (*interval*), Zeitpunkt (*instant*) und Zeitperiode (*period*).

Zeitdauer (*interval*)

Der Typ *interval* charakterisiert die Dauer eines Ereignisses, z.B. "30 Tage". Eine ungenaue Zeitdauer entspricht z.B. der Aussage "das Ereignis dauerte zwischen 5 und 7 Stunden". Es werden das Minimum d_{min} bzw. Maximum d_{max} von möglichen Werten definiert. Die Zeitangabe ist genau dann exakt, wenn $d_{min} = d_{max}$.

$$interval := \{(d_{min}, d_{max}) \mid d_{min}, d_{max} \in \mathbb{Z}, 0 \leq d_{min} \leq d_{max}\} \quad \textit{(Definition 1)}$$

Zeitpunkt (instant)

Ein Zeitpunkt ist eine Zeitangabe, die der Granularität genau eines Chronons entspricht, z.B. der "01.10.2000" wenn die Granularität des Kalenders einem Tag entspricht. Ungenaue Zeitangaben sind z.B. "Ein Tag in der ersten Woche im Oktober 2000". Ein ungenauer Zeitpunkt wird definiert durch die Angabe des frühest möglichen Zeitpunkts *c* des Ereignisses sowie die Anzahl der möglichen nachfolgenden Zeitpunkte d_Δ.

$$instant := \{(c, d_\Delta) \mid c, d_\Delta \in \mathbb{Z}, d_\Delta \geq 0\} \quad \textit{(Definition 2)}$$

Zeitperiode (period)

Eine Zeitperiode entspricht einer Zeitangabe, die einen Bereich beschreibt während dem ein Ereignis stattgefunden hat, z.B. "Im gesamten März 2001" oder "01.03.01 - 15.03.01". Eine Zeitperiode wird durch die Angabe ihres Anfangs- und Endzeitpunktes definiert. Ungenaue Zeitperioden besitzen einen ungenauen Anfangs und/oder Endzeitpunkt. Zusätzlich gilt die Bedingung, dass der Endpunkt nicht vor dem Anfangszeitpunkt liegen darf. Zeitangaben wie z.B. "Es begann zwischen 1500-1502 und endete zwischen 1600 und 1603" sind damit beschreibbar.

$$period := \{(i_1, i_2) \mid i_1, i_2 \in instant, (before(i_1, i_2) \vee overlaps(i_1, i_2) \vee equal(i_1, i_2))\}^{1} \quad \textit{(Definition 3)}$$

Diese drei neuen atomaren Datentypen ersetzten die bisherigen zeitlichen Datentypen von XSchema respektive XPath.

3.2 Operationen über den atomaren zeitlichen Datentypen

In der Literatur wird eine Vielzahl von zeitlichen Operationen auf zeitlichen Datentypen beschrieben [Al84, Je00, Sn00]. Die Einbeziehung unscharfer Zeitangaben erfordert jedoch eine Erweiterung einiger Operationen. Insbesondere boolesche Operationen müssen um einen dreiwertigen booleschen Datentyp (*3-bool* [CP01]) ergänzt werden. In Tabelle 3.1 wird am Beispiel von zwei Operationen deren Signatur und Semantik defi-

1. Die verwendeten Operationen entsprechen den Operation von J.F. Allen und werden später noch näher erklärt.

niert. Dies sind die Vergleichsoperation (<) für Zeitintervalle und die Operation *before()* aus Allen's Operationen für Zeitperioden [Al84].

Operation	Signatur	Semantik
<	$interval \times interval \rightarrow 3\text{-}bool$	$I_1 < I_2 \equiv \begin{cases} true : I_1.d_{max} < I_2.d_{min} \\ false : I_1.d_{max} > I_2.d_{min} \\ maybe : else \end{cases}$
before	$period \times period \rightarrow 3\text{-}bool$	$before(P_1,P_2) \equiv \begin{cases} true : ((P_1.I_2.c + P_1.I_2.d_\Delta) < P_2.I_1.c) = true \\ false : (P_1.I_2.c \geq (P_2.I_1.c + P_2.I_1.d_\Delta)) = true \\ maybe : else \end{cases}$

Tabelle 3.1: Beispiele für zeitliche Operationen

In T-XPath stehen folgende zeitliche Operationen zur Verfügung:
- arithmetische Operationen auf *interval*, *instant*, *period*
- Vergleichsoperationen auf *interval* (<, =, >,...)
- Allen's Operationen auf *period*, *instant* (*before*, *meets*, *after*,...)
- Cast Operationen (*duration_of_period*)

4 Das T-XPath Datenmodell

4.1 Klassifikation der Objekte des bisherigen XPath-Modells

XPath operiert auf der abstrakten, logischen Struktur eines XML-Dokuments und repräsentiert diese in seinem Modell als eine geordnete Baumstruktur. Innerhalb des Baumes werden Knoten, atomare Werte und Sequenzen als Grundelemente unterschieden. Für die Definition der Semantik der zeitlichen Verwaltung ist eine genauere Klassifikation dieser Objekte erforderlich.
Es gibt sieben Knotenarten: Dokument, Element, Attribut, Text, Namensraum, Prozess-Instruktion und Kommentar. Die Dokument- und Element-Knoten können als einzige weitere Knoten, so genannte Kindknoten, beinhalten und repräsentieren die hierarchische Struktur eines XML-Dokuments. Konkrete Informationen können lediglich in den anderen fünf Knotenarten gespeichert werden. Somit lassen sich Knoten in die zwei Klassen der "strukturierenden" Objekte und der "quantifizierenden" Objekte disjunkt einteilen.

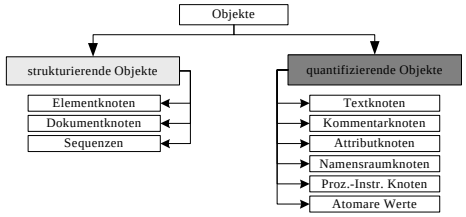

Abbildung 4.1: Vollständige Klassifizierung der XPath-Grundelemente

Die atomaren Werte des Datenmodells von XPath entsprechen atomaren Datentypen und korrespondieren mit den Datentypen aus XSchema. Atomare Werte gehören zu den "quan-

tifizierenden" Objekten. Eine Sequenz entspricht einer geordneten Menge von Knoten oder atomaren Werten, die jeweils auch mehrfach enthalten sein können. Sequenzen sind ‚flach', d.h. sie dürfen keine Sequenzen enthalten, und sind der Klasse der "strukturierenden" Objekte zuzuordnen. Ein vollständiger Überblick über die von uns vorgestellte Klassifikation der Grundelemente findet sich in der Abbildung 4.1.

4.2 Semantik der zeitlichen Verwaltung

Anhand der Klassifikation können zwei unterschiedliche Semantiken der zeitlichen Verwaltung abgeleitet werden:

a) In den strukturierenden Objekten von XPath werden Strukturen zeitlich verwaltet, d.h. es wird die Geschichte von Elementen und somit der Entwicklungsverlauf der Struktur eines XML-Dokuments repräsentiert (Schemaversionisierung).

b) Innerhalb der quantifizierenden Objekte werden konkrete Informationen zeitlich verwaltet. Dies entspricht dem geschichtlichen Verlauf eines konkreten Wertes (Objektversionisierung)

4.3 Die Integration der zeitlichen Verwaltung

Bei der zeitlichen Verwaltung von Information bilden die *Gültigkeitszeit* (*valid time*), wann war ein Wert in der realen Welt gültig, und die *Aufzeichnungszeit* (*transaction time*), wann wurde ein Wert gespeichert, die grundlegenden Zeitarten. Stehen beide zur Verfügung spricht man von *Bitemporaler Zeit*.
Orthogonal dazu werden Tupel- und Attributzeitstempelverfahren zur Verknüpfung der Information mit Zeitangaben unterschieden [Sk97, JE00]. Neuere Ansätze erweitern das Attributzeitstempelverfahren dahingehend, dass ein zeitabhängiger Wert als ein abstrakter zeitlicher Datentyp (ADT) modelliert wird [Er99, Gü00, CR01]. Die zeitliche Verwaltung findet ausschließlich innerhalb des ADT statt und muss nicht, wie bei den bisherigen Verfahren, explizit realisiert werden. 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]. Daher bilden ADTs die Grundlage für das T-XPath Modell.

4.4 Die abstrakten zeitlichen Datentypen

Die zeitliche Verwaltung mit ADTs erweitert lediglich das Typsystem von T-XPath. D.h. es werden keine neuen zeitlichen Elemente eingeführt, sondern neue Datentypen und Operationen, die in den quantifizierenden Objekten verwendet werden können. Somit wird die hierarchische Baumstruktur des zugrundeliegenden XPath-Datenmodells durch die zeitliche Verwaltung nicht beeinflusst.

In T-XPath werden für alle bisherigen Datentypen (z.B. *string*, *integer*, etc.) drei korrespondierende abstrakte zeitliche Datentypen zur Verfügung gestellt, die deren Gültigkeitszeit-, Aufzeichnungszeit- oder bitemporale Entwicklungsgeschichte repräsentieren. Die jeweiligen Eigenschaften der Zeitarten, beispielsweise keine Lücken in der Aufzeichnungszeit zu erlauben, sind für jeden der Datentypen formal definiert. Im folgenden sind am Beispiel des Datentyps *string*, dessen korrespondierende zeitliche Datentypen definiert. Die zusätzlichen Bedingungen in den Definitionen von *t_string* und *vt_string* stellen sicher, dass keine Löcher in der Aufzeichnungszeitgeschichte existieren.

validtime_string *(v_string)* *(Definition 4)*

$$v\_string := \left\{ B = \{(v_1, vt_1), \ldots, (v_m, vt_m)\} \middle| \begin{array}{l} v_i \in string, vt_i \in interval \cup instant \cup period \\ 1 \le i \le m, m \in \mathbb{N} \end{array} \right\}$$

transactiontime_string *(t_string)* *(Definition 5)*

$$t\_string := \left\{ B = \{(v_1, tt_1), \ldots, (v_m, tt_m)\} \middle| \begin{array}{l} v_i \in string, tt_i \in period, 1 \le i \le m, m \in \mathbb{N}, \\ 1 \le k \le m-1, meets(tt_k, tt_{k+1}) = true \end{array} \right\}$$

bitemporal_string *(vt_string)* *(Definition 6)*

$$vt\_string := \left\{ B = \{(v_1, vt_1, tt_1), \ldots, (v_m, vt_m, tt_m)\} \middle| \begin{array}{l} v_i \in string, tt_i \in period, vt_i \in interval \cup instant \cup period \\ 1 \le i \le m, m \in \mathbb{N}, 1 \le k \le m-1 \\ meets(tt_k, tt_{k+1}) = true \end{array} \right\}$$

4.5 Zeitliche Eigenschaftsattribute der ADTs

In den Definitionen der abstrakten zeitlichen Datentypen sind bereits Bedingungen enthalten, die eine konsistente zeitliche Verwaltung sicherstellen, beispielsweise die lückenlose Geschichte der Aufzeichnungszeit. Diese reichen jedoch nicht aus. So können bei der Gültigkeitszeit in einer Anwendung Überlappungen der zeitlichen Angaben erlaubt bzw. erwünscht sein und in einer anderen nicht. Diese Eigenschaft wird bei der Modellierung der Anwendung über ein zeitliches Eigenschaftsattribut des ADT explizit festgelegt und bei der Instanziierung des ADTs überprüft. Zusätzlich können Eigenschaften, wie beispielsweise ein automatisches Coalescing [Je00], eine temporale Aufwärtskompatibilität [Sn00], oder Einschränkungen des Wertebereichs, explizit über die Eigenschaftsattribute definiert werden.

4.6 Operationen der abstrakten zeitlichen Datentypen

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. Durch die Mengenwertigkeit der ADTs kann das Ergebnis mehrere Elemente enthalten, die zusammen wiederum einen instantiierten ADT bilden.

Für den Umgang der Mengenwertigkeit stehen Operationen zur Verfügung, die z.B. die Anzahl der Elemente (z.B. *count*()) oder deren zeitliche Ordnung (z.B. *last*(), *first*(), *next*()) innerhalb eines ADTs ermitteln. Zusätzlich existieren für die Änderungsrate eines Wertes (*rate_of_change*()) und dessen zeitlichen Durchschnitt (*temporal_average*()) weitere Operationen. Die Operationen *time*() und *value*() sind Projektionsoperationen die für einen ADT mit *count*()=1 die Zeitangabe oder den Wert zurückliefern. Für die Unterscheidung der Zeitarten in den bitemporalen ADTs stehen die Operationen *valid_time*() und *transaction_time*() zur Verfügung, welche für nachfolgende Operationen die zu verwendende Zeitart festlegen.
Beim Vergleich zweier abstrakter zeitlicher Datentypen *A* und *B* ruft deren mögliche Mengenwertigkeit Probleme hervor. Für den einfacheren Fall, dass entweder *A* oder *B* ein einzelnes Element *x* beinhaltet (d.h. *count*()=1), ist das Ergebnis wiederum ein ADT mit genau den Elementen, die bei dem Vergleich mit *x* das gewünschte Prädikat besitzen. Der schwierigere Fall liegt vor, wenn sowohl *A* als auch *B* jeweils mehr als ein Element besitzen. Dadurch ist es möglich, dass die einzelnen Elemente aus *A* mit jeweils unterschiedlichen Teilmengen aus *B* korrelieren. Im Ergebnis müsste somit für jedes Element aus *A* die jeweilige Ergebnismenge aus *B* dargestellt werden. Ein derartiger Vergleich liefert eine Multimenge als Ergebnis. Dieses ist mit den bisher vorgestellten Datentypen nicht darstellbar. Es kann aber, wie später noch gezeigt wird, mit Hilfe einer neuen Funktion in der Anfragesprache von T-XPath realisiert werden.

4.7 Repräsentation der abstrakten Datentypen in XML und XSchema

Die abstrakten zeitlichen Datentypen sind 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[1] ist diese Struktur beispielhaft anhand einer ComplexTyp-Definition in XSchema dargestellt (links oben in der Abbildung). Die explizite (nicht vollständige) Definition des ADTs würde bei der Integration in XSchema wegfallen.

```
<!-- Complex Typ-Definition eines v_string -->
<xs:complexType name="v_string">
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="value" type="xs:string"/>
    <xs:element name="valid_time" type="Zeitwert"/>
  </xs:sequence>
</xs:complexType>
...
<!-- Anwendungsbeispiel eines v_string -->
<xs:complexType name="Gebäude">
    <xs:sequence maxOccurs="unbounded">
      <xs:element name="Verwendung" type="v_string"/>
    </xs:sequence>
</xs:complexType>
```

```
...
<Gebäude>
  <Verwendung>
    <value>Stallungen</value>
    <valid_time>1500-1730</valid_time>
     <value>Hospital</value>
     <valid_time>1730-1735</valid_time>
    <value>Lagerhaus</value>
    <valid_time>1735-uc</valid_time>
  </Verwendung>
...
</Gebäude>
...
```

Abbildung 4.2: Beispiel für die Repräsentation des abstrakten Datentyp v_string in XSchema (links) und XML (rechts)

Die Abbildung 4.2 zeigt gleichzeitig die Anwendung des ADT *v_string* in einer Beispielmodellierung (linker, unterer Bereich der Abbildung), in welcher der Verwendungszweck eines Gebäudes einer zeitlichen Entwicklung unterliegt.

1. Der Type "Zeitwert" für das Element "valid_time", repräsentiert die Menge der vorgestellten atomaren zeitlichen Datentypen

Die Repräsentation des Beispiels in XML ist auf der rechten Seite der Abbildung dargestellt. Alle im Laufe der Geschichte aufgetretenen Verwendungen eines Gebäudes und deren jeweilige Gültigkeitszeiten sind vollständig in dem XML-Dokument enthalten. 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.

5 Die Anfragesprache von T-XPath

T-XPath repräsentiert zeitliche Daten durch neue atomare und abstrakte zeitliche Datentypen. 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 4.6) 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.

5.1 Beispiel für Anfragen in T-XPath

Die nachfolgenden Anfragen beziehen sich auf das Beispiel von Abschnitt 4.7. Die folgende Anfrage liefert alle Gebäude deren Verwendungszweck im Jahre 1750 exakt bekannt war.

```
//Gebäude[Verwendung valid[1] '1750'= true]
```

Für den unsicheren Fall müsste die Anfrage folgendermaßen umformuliert werden:

```
//Gebäude[Verwendung valid '1750'= maybe]
```

Im Ergebnis sind ausschlieslich die Gebäude enthalten, bei denen aufgrund unscharfer Zeitangaben für das Jahr 1750 zumindest die Möglichkeit bestand, dass sie zu dieser Zeit dem entsprechenden Verwendungszweck dienten.
Alle Gebäude, die ab 1800 (bis heute oder bis zu ihrem Abriss) ihren Verwendungszweck nicht mehr geändert haben, werden mit Hilfe der nachfolgenden Anfrage ermittelt.

```
//Gebäude[Last(Verwendung) valid '1800'= true]
```

Die Funktion *Last*() findet zunächst den letzten Verwendungszweck eines Gebäudes und vergleicht anschließend, ob dieser bereits im Jahre 1800 gültig war.
Die bisher vorgestellten Anfragen wurden ausschließlich auf den zeitlichen Werten durchgeführt. Eine Anfrage, die zusätzlich konkrete Werte berücksichtigt, könnte wie folgt aussehen.

```
//Gebäude[Verwendung valid '1600' and Verwendung = 'Stallungen']
```

In der Anfrage wurde für das zeitliche Prädikat kein Wert angegeben. In diesem Fall wird *true* und *maybe* angenommen. Das Ergebnis bilden diejenigen Gebäude, die im Jahre 1600 (eventuell) als Stallungen verwendet wurden.

1. Die Operation `valid()`setzt sich aus mehreren von Allen's Operationen zusammen und vergleicht Zeitangaben ob diese mindestens ein gemeinsames Chronon besitzen.

5.2 Operationen zwischen zwei abstrakten zeitlichen Datentypen

Wie in Abschnitt 4.6 festgestellt, kann der Vergleich zweier ADTs ohne zusätzliche Unterstützung durch die Anfragesprache nicht realisiert werden. Als problematisch erweisst sich die Darstellung der Ergebnismenge, in der zu jedem Elemente aus A die entsprechende Teilmenge aus B zugeordnet ist.
Für die Realisierung führen wir zunächst die neue Funktion *expand*() ein, die als Übergabeparameter einen (instantiierten) ADT erhält. Die Operation konvertiert jedes einzelne Element des übergebenen ADTs in einen ADT mit genau einem Wert, d.h. *count*()=1, und liefert eine Sequenz derartiger ADTs als Ergebnis. Diese können analog zu Abschnitt 4.6 mit dem zweiten mengenwertigen ADT operieren.
Für die Iteration über eine oder mehrere Sequenz(en) wird in XPath 2.0 ein neuer Ausdruck, die For-Expression, eingeführt, der auch in T-XPath zur Verfügung steht. Mit Hilfe der For-Expression und der *expand*()-Funktion wird der Vergleich von zwei mengenwertigen ADTs realisiert. Abschließend ist dies anhand eines Beispiels verdeutlicht.

```
for $i in expand(//Gebäude/Verwendung)
return($i, for $j in //Gebäude[Verwendung valid valid_time($i)]
           return($j/Verwendung))
```

Die Anfrage liefert eine Sequenz, in der jedem Verwendungszweck eines Gebäudes, die zur gleichen Zeit gültigen Verwendungen der anderen Gebäude zugeordnet ist.

6 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 im vorderen Teil dieser Arbeit gezeigt, sind die zeitlichen Kanten jedoch eher für die zeitliche Verwaltung der Struktur des XML-Dokumentes geeignet. 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 Abfragefunktionaltität durch eine eigene SQL-artige Anfragesprache TS-QL realisiert ist.

7 Zusammenfassung

Wir haben gezeigt, wie zeitliche Informationen in XML verwaltet werden können. Die Grundlage bildet das XPath 2.0 Datenmodell. Dieses wurde um temporale ADTs erweitert, die neue zeitliche atomare Datentypen und Operationen kapseln und eine flexible, kalenderunabhängige Unterstützung sowohl von exakten als auch von unscharfen zeitlichen Angaben ermöglichen. Anhand einer Klassifikation von XPath-Elementen wurden die unterschiedlichen Semantiken der zeitlichen Verwaltung von Daten in T-XPath aufgezeigt. Sowohl Schemaversionisierung als auch Objektversionisierung sind möglich. Zusätzlich werden Gültigkeitszeiten und Aufzeichnungszeiten unterstützt. Die Anfragesprache von T-XPath stellt für zeitliche Anfragen eine Reihe neuer Operationen und Funktionen zur Verfügung. Dabei musste insbesondere das Problem der Multimengenwertigkeit von Ergebnissen gelöst werden.
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. In der historischen Anwendung des GEIST Projektes wird T-XPath praktisch evaluiert.

8 Literaturverzeichnis

[Al84] Allen, J.F. Towards a General Theory of Action and Time. AI, 1984, pp. 123-154.

[AMU00] Amagasa, T., Masatoshi, Y., Uemura, S. A Data Model for Temporal XML Documents. Proc. DEXA 2000, 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 Syst. 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-46,1999.

[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.

[Kr01] Kretschmer, U., Coors, et.al. Meeting the Spirit of History. In Proc. of the Int. Symp. on Virtual Reality, Archaeology and Cultural Heritage, VAST 2001, Greece, 2001.

[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, 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/