

# Temporale Datenmodelle und Metainformationssysteme für Geoinformationssysteme

Dipl. Inf. Jan Ramsch

Dr. Dieter Sosna

ramsch@informatik.uni-leipzig.de

dieter@informatik.uni-leipzig.de

Universität Leipzig

Institut für Informatik

Augustusplatz 10/11

Leipzig, den 20. August 1997

**Schlüsselworte:** ATKIS, GIS, temporale Datenmodelle, temporale  
Metainformationssysteme

## Zusammenfassung

Das Amtliche Topographisch-Kartographische Informationssystem ATKIS ist das Vorhaben der Landesvermessungsverwaltungen zum Aufbau eines einheitlichen digitalen Datenbestand für das Gebiet der Bundesrepublik Deutschland. Für die im Rahmen des Verfahrens zur Datenabgabe an die Nutzer bereitzustellende inkrementelle Fortführungsinformation ist die ATKIS-Datenbank um Informationen über neu erzeugte, veränderte oder gelöschte Datensätze zu erweitern. Neben dieser existieren von Seiten des IfAG<sup>1</sup> weitere Anforderungen, die so nicht in [ATKIS] festgehalten sind, wie zum Beispiel die Rekonstruktion des ATKIS-Datenbank-Zustandes zu einem früheren Zeitpunkt und die Unterscheidung von korrekten und erfaßten Daten bezüglich eines bestimmten Zeitpunktes. Diese Anforderungen sind beim Entwurf der Datenbank zu berücksichtigen. Nach einer Diskussion prinzipieller Lösungswege werden in den folgenden Abschnitten verschiedene Ansätze, Modelle und Lösungsvorschläge für temporale Datenmodelle und Versionenverwaltungen vorgestellt und besonders im Hinblick auf eine Implementation in einem relationalen DBMS und den speziellen Anforderungen von ATKIS diskutiert.

---

<sup>1</sup> Institut für Angewandte Geodäsie

# Inhaltsverzeichnis

<b>1</b>	<b>Temporale Datenmodelle</b>	<b>2</b>
1.1	Notwendigkeit eines temporalen GIS für ATKIS . . . . .	2
1.2	Geschichten in ATKIS . . . . .	5
1.2.1	Geschichten geometrischer Daten . . . . .	5
1.2.2	Geschichten nicht-geometrischer Daten . . . . .	6
1.3	Versionen . . . . .	7
1.4	Versionen im relationalen Modell . . . . .	10
1.4.1	Unterscheidung bitemporal erweiterter Relationenmodelle . . . . .	11
1.5	Temporale Operationen für ein relationales Datenmodell . . . . .	13
1.5.1	Änderungsoperationen für aktuelle Tupel . . . . .	14
1.5.2	Änderungsoperationen für historische Tupel . . . . .	14
<b>2</b>	<b>Versionen von Metadaten</b>	<b>15</b>
2.1	Allgemeine Begriffe . . . . .	15
2.2	Metadaten von zeitveränderlichen Daten . . . . .	16
2.3	Zusammenfassung . . . . .	17
	<b>Glossar</b>	<b>19</b>
	<b>Literaturverzeichnis</b>	<b>23</b>

## 1 Temporale Datenmodelle

### 1.1 Notwendigkeit eines temporalen GIS für ATKIS

Das 1985 durch die Landesvermessungsverwaltungen der Bundesrepublik Deutschland ins Leben gerufene Vorhaben ATKIS (Amtliches Topographisch-Kartographisches Informationssystem) zum Aufbau Digitaler Landschaftsmodelle und Digitaler Kartographischer Modelle stellt topographische Daten für Endnutzer nach dem in Abbildung 1 dargestellten Schema bereit. Die Abgabe erfolgt über die *Einheitliche Datenbankschnittstelle* (EDBS) durch das Verfahren *Bezieher Sekundärnachweis* (BZSN). Laut [AGH96] ist die angestrebte Verfahrensweise die einmalige Abgabe des gesamten Primärdatenbestandes oder benutzerspezi-

fischer Auszüge (Erstausrüstung). Danach werden nur noch die durch Fortführung veränderten, neu erzeugten oder gelöschten Datensätze abgegeben (*Inkrementelle Fortführungsinformation*). ATKIS-Nutzer müssen durch dieses Verfahren ihre Fachdaten nicht jedesmal neu mit den ATKIS-Daten verknüpfen, es ist lediglich eine Bearbeitung der neu hinzugekommenen und der gelöschten Datensätze notwendig.

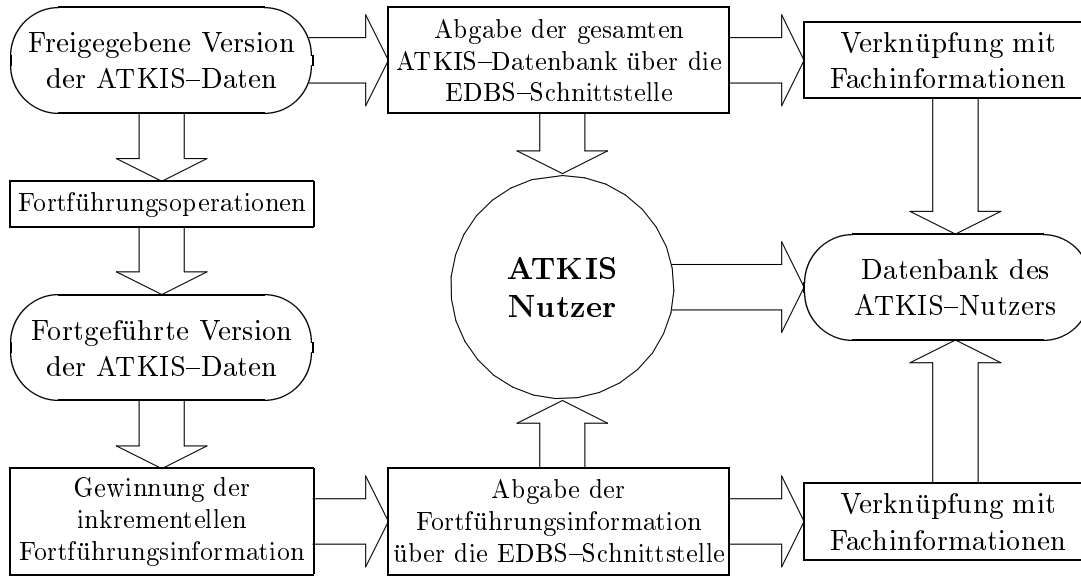


Abbildung 1: Abgabe und Fortführung der ATKIS-Daten

Die Gewinnung der inkrementellen Fortführungsinformation setzt die Verwaltung aller Manipulationen an ATKIS-Objekten und -Objektteilen und eine benutzerspezifische Kontrolle der im Rahmen des BZSN abgegebenen Daten voraus. Die ATKIS-Datenbank ist also um einen Mechanismus zur Verwaltung aller Änderungsoperationen oder adäquater Informationen zu erweitern. Hierfür sind eine Reihe von Verfahren denkbar. Nach [AGH96] werden zur Zeit die Fortführungsdaten für den BZSN gewonnen, indem alle Veränderungen am Primärdatenbestand beziehungsweise gespeichert werden. In vorliegendem Abschnitt werden neben dieser verschiedene andere Möglichkeiten diskutiert. Dabei wird nachgewiesen, daß das dort beschriebene *temporale* ATKIS-Datenmodell den Anforderungen am besten gerecht wird.

Eine einfach zu realisierende Lösung für die beschriebene Anforderung (Bereitstellung aller Einfüge-, Lösch- und Änderungsoperationen für die Erzeugung der Inkrementellen Fortführungsinformation) ist die Aufzeichnung aller Veränderungsoperationen am ATKIS-Datenbestand (Logging-Verfahren). Das versetzt den Anwender in die Lage, seine Daten

durch die Einarbeitung dieser Fortführungsinformationen immer auf dem aktuellen Stand zu halten. Er ist aber verpflichtet, die Änderungen *jeder* Fortführung in der korrekten Reihenfolge einzuarbeiten. Die spätere Rekonstruktion der einmal an einen Nutzer abgegebenen Daten wird durch ein solches Verfahren wenn nicht unmöglich gemacht, so doch extrem kompliziert.

Der Forderung nach Rekonstruierbarkeit kann man dadurch gerecht werden, daß nach jeder Fortführung ein Schnappschuß der ATKIS-Daten aufgehoben wird (Schnappschußverfahren). Dies ist einfach zu realisieren und erlaubt die Recherche alter Daten. Die Gewinnung der inkrementellen Fortführungsinformation wird jedoch schwierig. Außerdem ist dieses Verfahren mit einem Speicheraufwand von mehreren 100 Mbyte pro Sicherung selbst angesichts rapide sinkender Preise für Speichermedien keine empfehlenswerte Vorgehensweise.

Eine bessere Lösung des Problem es ist die Integration der Zeit in das ATKIS-Datenmodell. Ein solches *temporales* Datenmodell ermöglicht die Abbildung der Geschichte eines Objektes, wobei jeder einmal aktuelle Objektzustand auch nach Änderungs- oder Löschope-rationen in der Datenbank verbleibt. Diese Modellierung eröffnet eine Reihe von Vorteilen:

- genaue Kontrolle über Versionen von Datenbeständen,
- leichte Erzeugung der inkrementellen Fortführungsinformation,
- Verfügbarkeit jedes einmal aktuellen Datenbestandes,
- Möglichkeit der Verfolgung von zeitlichen Entwicklungen bei Objekten,
- Möglichkeit der Korrektur älterer Objektzustände ohne Zerstörung des fehlerhaften, aber schon an Dritte abgegebenen Zustandes.
- geringer Speicherplatzbedarf

Voraussetzung dafür ist ein temporales Datenmodell, das Einträge und Abfragen bezüglich zweier Zeitebenen unterstützt, der Zeit, zu der ein Datensatz erfaßt bzw. aktualisiert wurde (Aufzeichnungszeit) und der Zeitspanne, in der seine Informationen als gültig angenommen werden (Gültigkeitszeit). Als Hauptnachteile eröffnen sich der im Vergleich zu den beiden erstgenannten Verfahren höhere Aufwand und der höhere Bedarf an aktivem Speicherplatz. Der letztgenannte Punkt läßt sich jedoch durch die Auslagerung „historischer“ Daten relativieren.

Ein anderes Problem bei der Fortführung der ATKIS-Daten eröffnet sich durch die anhaltende Entwicklung von ATKIS. Auch in Zukunft ist mit Veränderungen der OK zu rechnen. Wir sind also auch bei der ATKIS-Datenstruktur mit temporalen Veränderungen konfrontiert. Die Aspekte einer zeitveränderlichen Datenstruktur und der Integration der Zeit in das Metadatenystem DST werden in Abschnitt 2 besprochen.

Zunächst sollen jedoch temporale Datenmodelle vorgestellt, diskutiert und für ATKIS angewendet werden.

## 1.2 Geschichten in ATKIS

### 1.2.1 Geschichten geometrischer Daten

Aus der Theorie der temporalen Datenmodelle sind mehrere Methoden zur Beschreibung der Geschichten zeitveränderlicher Objekte bekannt. Die wichtigsten sind zustandsverändernde, zustandserhaltende, ereignisorientierte und ableitbare Geschichten (siehe Glossar oder vgl. [Tan93], [Käf92] oder [Ram97/1]). Mit welcher dieser Geschichtsarten läßt sich nun die Entwicklung der ATKIS-Objekte und -Objektteile am besten beschreiben? Sehen wir uns zunächst die Veränderungen auf der geometrischen Ebene an. In der Natur treten sowohl kontinuierliche Veränderungen auf, zum Beispiel das Vorrücken einer Steilküste ins Landesinnere, als auch abrupte, wie die Erweiterung der Grundfläche eines Gebäudes durch Anbau.

Der erste Fall eröffnet sofort gewisse Schwierigkeiten. Aufgrund des enormen technischen Aufwandes ist es in der heutigen Praxis nicht möglich, mehr als Schnappschüsse von der Oberflächenstruktur der Erde zu erhalten. Zudem sind die funktionalen Zusammenhänge kontinuierlicher Lageänderungen im allgemeinen nicht bekannt. Man kann also davon ausgehen, daß solche Veränderungen nur durch eine zustandsverändernde Geschichte beschrieben werden können.

Bei dieser Art von Geschichte wird davon ausgegangen, daß Zustände zwischen den gesicherten Zeitpunkten mit vertretbarer Genauigkeit interpoliert werden können. Zur Zeit sind die Abstände zwischen den Fortführungsphasen topographischer Datenbestände noch sehr groß, eine Simulation der Entwicklung der Geometrie eines topographischen Objektes muß aber nicht zwingend zu ungenau werden, da solche Veränderungen im allgemeinen eher langsam vonstatten gehen. Nehmen wir zum Beispiel den Verlauf eines Flusses. Seine Ände-

rungen werden im Laufe eines Jahres nur wenige Meter betragen. So erhält man selbst bei Fortführungszeiträumen von einigen Jahren ein hinreichend genaues Bild der Flußveränderungen.

Im zweiten genannten Fall handelt es sich um eine zustandserhaltende Geschichte, wobei der Zeitpunkt der tatsächlichen Änderung im allgemeinen nicht bekannt ist. Entgegen den Anforderungen an eine zustandserhaltende Geschichte ist es also hier nicht möglich, zu jedem Zeitpunkt eine Aussage zur Geometrie des Objektes zu treffen. Ein korrekter Objektzustand ist nur für den Zeitpunkt der Erfassung bzw. Fortführung vorauszusetzen.

Leider ist es nicht möglich, die Art der Geschichte aus der Art des abgebildeten Objektes zu folgern. So kann sich die Lage des als Beispiel genannten Flusses auch plötzlich durch Eingriffe von Menschenhand oder Naturkatastrophen verändern.

Eine mögliche Lösung ist das Verzichten auf zustandsverändernde Geschichten und die Abbildung aller temporalen Vorgänge mittels einer zustandserhaltenden Geschichte. Eine ereignisorientierte Geschichte, wie sie sich scheinbar anbietet, ist nicht ausreichend, da zwischen den gesicherten Zeitpunkten sehr wohl ein Objektzustand existiert. Man muß sich allerdings bewußt sein, daß dieser, obwohl bei zustandserhaltender Geschichte immer definiert, nur mit einer gewissen Wahrscheinlichkeit korrekt ist. Wie bereits besprochen, wird er aber in den meisten Fällen eine gute Annäherung an den tatsächlichen Zustand sein.

### **1.2.2 Geschichten nicht–geometrischer Daten**

Bei der Modellierung temporaler Entwicklungen der attributiven ATKIS–Daten sind wieder verschiedene Arten von Geschichten anzutreffen. Viele ATKIS–Attribute beschreiben topographisch relevante Sachverhalte, indem die hervorstechendsten Eigenschaften eines Objektes in möglichst treffende Kategorien eingeordnet werden. Erst diese Abstraktion von ansonsten nahezu unbegrenzt vielfältigen Charakteristika macht eine sinnvolle Verarbeitung und vergleichende Auswertung topographischer Objekte möglich. Sie verhindert aber auf der anderen Seite die Dokumentation stetig verlaufender Veränderungen in der Landschaft. So läßt sich die allmähliche Verwandlung eines Laubwaldes in einen Mischwald nur durch die Änderung des Attributes Waldtyp festhalten, wenn ein bestimmter Nadelbaumanteil überschritten wurde. Der zeitliche Verlauf der „Vernadelung“ kann nicht nachvollzogen werden. Wir beobachten hier also wiederum eine zustandserhaltende Geschichtsmodellierung, die dem Wesen nach eigentlich eine zustandsverändernde ist.

Etwas anders liegt der Fall bei Attributen, die nicht kodierte Eigenschaften von topographischen Objekten, sondern echte Maßangaben beinhalten. Dabei handelt es sich um Breiten- und Höheninformationen, die nicht direkt aus der Geometrie ablesbar sind, die Tragfähigkeiten von Bauwerken, die Spannung eines Umspannwerkes und vieles andere. Einige dieser Attribute sind von Natur aus diskret, wie zum Beispiel die Anzahl der Startbahnen eines Flughafens oder die Anzahl der Geschosse eines Gebäudes. Andere, wie die mittlere Pegelhöhe eines Gewässers, haben einen stetigen Charakter.

Die Situation ähnelt der der geometrischen ATKIS-Daten. Für Teilaspekte wäre eine zustandsverändernde Geschichte wünschenswert. Die Voraussetzungen für eine relativ genaue Interpolation sind auch hier trotz der großen Abstände gesicherter Zeitpunkte positiv zu bewerten, da sich die betroffenen Attributwerte meist nur in längeren Zeiträumen wesentlich verändern. Der größere Teil der Attribute unterliegt aber einer zustandserhaltenden Geschichte.

Es ergeben sich somit zwei Möglichkeiten, die Erschaffung einer komplexen Geschichtsverwaltung mit mehreren Geschichtsarten oder die vereinfachte Modellierung nur mit zustandserhaltenden Geschichten und einem wesentlich geringerem technischen Aufwand.

Eine die wahre Natur vieler topographischer Objekte besser beschreibende Lösung mit verschiedenen Arten von Geschichten ist ohne Zweifel im Sinne einiger Anwendergruppen wie Ökologen, Historiker und Umweltschützer, die vor allen Dingen an Entwicklungen der Landschaft interessiert sind. Das Gros der ATKIS-Benutzer ist aber eher im kartographischen, topographischen und raum- und landschaftsplanerischen Umfeld angesiedelt. Für solche Anwender ist in erster Linie eine Abbildung der Landschaft zu einem konkreten Zeitpunkt wichtig, weniger aber die temporale Entwicklung einzelner Objekte. Es gilt also, die Recherchierbarkeit des Zustandes des DLM zu einem beliebigen Zeitpunkt sicherzustellen, wobei für den aktuellen Stand der einfachste Zugriff gewährleistet sein sollte.

Zusammenfassend ist festzustellen, daß für alle in ATKIS verwalteten Informationen, die temporalen Änderungen unterworfen sind, die Modellierung mit Hilfe einer zustandserhaltenden Geschichte akzeptabel ist.

### **1.3 Versionen**

Im folgenden soll der in Zusammenhang mit den zustandserhaltenden Geschichten benutzte Versionenbegriff näher betrachtet werden.

Aufzeichnungszeit

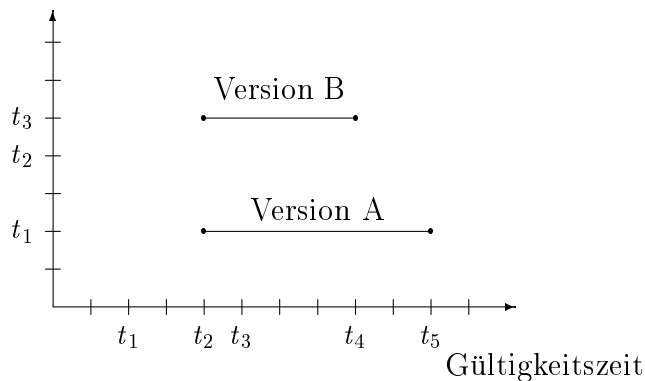


Abbildung 2: Beispiel für eine bitemporale Objektgeschichte

Objektgeschichten in ATKIS sind *bitemporalen Objektgeschichten*. Betrachten wir den in Abbildung 2 dargestellten Fall. Zum Zeitpunkt  $t_1$  wird die Version A eines Objektes gespeichert. Sie soll einen Gültigkeitszeitraum von  $t_2$  bis  $t_5$  haben. Zu einem Zeitpunkt  $t_3 > t_1$  stellt man jedoch fest, dass die Version A für den Zeitraum  $t_2$  bis  $t_4$  fehlerhaft ist. Nun wurde die Version A aber bereits an andere Nutzer weitergegeben. Eine Änderung der Version A in die korrekte Version B würde zwar die Linearität der Objektgeschichte erhalten, die Recherchierbarkeit der wahren Objektgeschichte aber unmöglich machen. Also wird eine neue Version B erzeugt.

Also ist sowohl der Aufzeichnungsszeitpunkt  $t_a$  als auch das Gültigkeitszeitintervall  $t_g$  eines Objektes festzuhalten. Eine Version eines Objektes existiert dann immer bezüglich des Tupels  $(t_a, t_g)$ . Bezüglich eines der beiden Zeitstrahlen können also mehrere Versionen eines Objektes existieren.

Aus diesen Überlegungen heraus wird in der moderneren Literatur (siehe z.B. [Kim90], [Sno95/2] und [Cat91]) die Version eines Objektes etwa wie folgt eingeführt:

**Definition 1.1 (Version)** *Eine Version ist ein Zustand eines Objektes. Sie besitzt einen Gültigkeitszeitraum und ist je nach Art der Version unveränderlich oder veränderlich. Nach der Erzeugung einer ersten Version werden alle weiteren Versionen eines Objektes von dieser rekursiv abgeleitet und bilden eine Versionenhierarchie.*



Dadurch ergibt sich ein Versionenschema wie in Abbildung 3 dargestellt. Aus objekt-orientierter Sicht sind alle Versionen Instanzen eines Objektes. In einem relationalen Modell sind das Objekt und seine Versionen durch eine 1 :  $n$ -Beziehung verbunden.

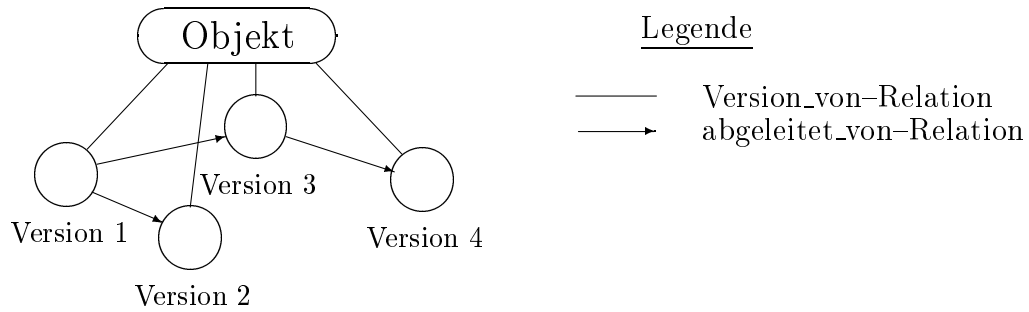


Abbildung 3: Beispiel einer Versionenhierarchie [Kim90]

Wie in der Definition bereits angedeutet, werden mehrere Arten von Versionen unterschieden. Diese Notwendigkeit ergibt sich vor allem aus der praktischen Arbeit mit temporalen Daten. Eine der Grundeigenschaften temporaler Daten ist ihre unbegrenzte Dauerhaftigkeit, das heißt sie werden, einmal in eine Datenbank eingebracht, nie wieder verändert oder gar gelöscht. Soll ein Objekt verändert werden, so ist eine neue Version des Objektes zu erzeugen. Nun befinden sich aber gerade im GIS-Bereich Daten während eines Bearbeitungszeitraumes oft in einem unfertigen Zustand, der noch nicht dem erwünschten Endzustand entspricht. Für die praktische Handhabung werden also Versionenzustände benötigt, die noch Änderungen oder gar Löschungen zulassen, ohne das eine neue Version erzeugt und die alte in der Datenbank belassen werden muß.

Won Kim unterscheidet in [Kim90] drei Arten von Versionen, die sich durch den Grad ihrer *Robustheit* unterscheiden. Unter Robustheit ist hier ein Maß für die Unveränderlichkeit eines Datensatzes zu verstehen.

**Definition 1.2 (Flüchtige Version (transient version))** *Eine flüchtige Version eines Objektes ist eine temporäre Version, die geändert und gelöscht werden kann. Für das Objekt geforderte Integritätsbedingungen können bei flüchtigen Versionen für ungültig erklärt werden.*

**Definition 1.3 (Arbeitsversion (working version))** *Eine Arbeitsversion eines Objektes kann gelöscht, aber nicht geändert werden.*

**Definition 1.4 (Freigegebene Version (released version))** *Eine freigegebene Version eines Objektes kann weder geändert noch gelöscht werden.*

Eine neu erzeugte Version eines Objektes ist immer eine flüchtige Version. Eine flüchtige Version wird durch *Beförderung* (promotion) zu einer Arbeitsversion und durch eine weitere Beförderung zu einer freigegebenen Version. Eine Arbeitsversion kann durch *Degradierung* (demote) wieder zur flüchtigen Version werden. Eine freigegebene Version kann jedoch nicht degradiert werden.

Diese Unterscheidung von Versionen ist auch für ATKIS-Daten sinnvoll. Ein ATKIS-Objekt oder -Objektteil durchläuft in seiner Entwicklung mehrere Stadien. Während seiner Erfassung oder Erzeugung durch automatische Verfahren kann es sich in einem inkonsistenten Zustand befinden, bestimmte Komponenten werden evtl. erst zu einem späteren Zeitpunkt vervollständigt oder auf ihre Richtigkeit überprüft<sup>2</sup>. Diese flüchtige Version geht mit der Beendigung der Erfassung durch Beförderung in eine Arbeitsversion über. Alle Integritätsbedingungen müssen nun erfüllt sein. An die eigentliche Erfassung schließt sich ein Prozeß der inhaltlichen Überprüfung der ATKIS-Daten an. Hier können Objekte und Objektteile noch geändert oder gelöscht werden. Für eine Änderung muß die Arbeitsversion zuerst zu einer flüchtigen Version degradiert werden, eine Löschung kann direkt erfolgen. Nach diesem letzten Arbeitsschritt werden die Daten freigegeben und können an Dritte abgegeben werden. Sie dürfen nun nicht mehr verändert oder gelöscht werden; der Erhalt von an Dritte abgegebenen Daten muß immer garantiert sein.

## 1.4 Versionen im relationalen Modell

Für die eingeführten Arten von Versionen müssen nun Repräsentationen im relationalen Modell<sup>3</sup> einschließlich der Operationen Erzeugen, Löschen, Ändern und Abfrage definiert werden. Hierzu gibt es eine Reihe von Ansätzen. Eine große Anzahl von Arbeiten zu diesem Thema (z.B. [Cli83], [Tan86]) sieht nur eine Abbildung der Gültigkeitszeit vor, während andere Modelle nur die Aufzeichnungszeit berücksichtigen (z.B. [Jen92], [Sto87]). Wie bereits begründet, benötigt ATKIS die Abbildung von bitemporalen Geschichten.

---

<sup>2</sup> Ein typisches Beispiel hierfür ist die automatische Erzeugung von Objektteilen, die erst im Laufe einer späteren manuellen Bearbeitung Objekten zugeordnet werden.

<sup>3</sup> Der Natur von temporalen Daten entsprechend bietet sich für die Implementation eines temporalen Datenmodells ein objektorientiertes DBMS an. Aus technologischen Gründen kommen zur Zeit für die Haltung der ATKIS-Daten jedoch nur relationale Datenbanksysteme in Frage. Aus diesem Grunde soll hier nur dieser Ansatz verfolgt werden.

Das erste bitemporale Modell wurde bereits 1982 von Ben-Zvi unter dem Namen Time Relational Model in [Ben82] beschrieben. Die Komplexität der Materie brachte noch eine Reihe von sehr unterschiedlichen Ansätzen hervor. Für ATKIS müssen wir uns für eine Variante entscheiden, die die folgenden Bedingungen erfüllt:

- Erweiterung des Relationenmodells unter Wahrung der Einfachheit (bitemporal erweitertes Relationenmodell).
- Anordnung der Zeit auf Tupelebene. Die Unterstützung temporaler Attribute ist für ATKIS nicht sinnvoll, da die Anzahl der Attribute pro Tupel i. allg. gering ist, der technologische Aufwand zur Unterstützung temporaler Attribute jedoch sehr hoch.
- Bevorzugung der jüngsten Version eines Objektes (bezüglich Gültigkeits- und Aufzeichnungszeit). Getrennte Speicherung der jüngsten und älteren Versionen. Der Zugriff auf aktuelle Daten muß gegenüber dem wesentlich selteneren Zugriff auf ältere Daten bevorzugt werden.
- Unterstützung von Versionen von Metadaten (Strukturversionen) für nicht-aktuelle Daten. Die aktuellen Daten besitzen immer eine einheitlichen Metastruktur.
- Unterstützung von Anfragen bezüglich eines bitemporalen Zeitpunktes (Snapshot- oder AS OF-Anfrage) und bezüglich eines Intervalls der Gültigkeits- oder Aufzeichnungszeit (WALK THRU TIME-Anfrage).
- Unterstützung von temporalen Joins (Selektion von Geschichten aufgrund gegenseitiger zeitlicher Abhängigkeiten) unter besonderer Berücksichtigung eines temporalen Equi Join (Verbund zweier Geschichten, wenn die Verbundbedingung zur gleichen Zeit in beiden Geschichten gültig ist). Temporale Joins müssen bezüglich einer oder beider Zeitebenen definiert werden können.

#### **1.4.1 Unterscheidung bitemporal erweiterter Relationenmodelle**

Die Geschichte temporaler Datenmodelle brachte bereits eine Reihe von bitemporalen Ansätzen auf Basis des Relationenmodells hervor. Diese unterscheiden sich in einigen wesentlichen Punkten, welche im folgenden kurz erläutert und im Hinblick auf ihre Relevanz für ATKIS ausgewertet werden.

Der erste wichtige Unterschied ist die Art und Weise, wie Zeitstempel dargestellt werden. Ein Zeitstempel ist dabei die Kombination aus Zeitpunkten und -intervallen, die für jede Version eines Objektes gespeichert werden. Diese Zeitstempel können als normale Attribute, deren Semantik dem RDBMS nicht bekannt ist, oder als neue Datentypen einschließlich der entsprechenden Operationen abgebildet werden. Die letzte Variante verlangt tiefere Eingriffe in das RDBMS und wurde deshalb hier nicht weiter verfolgt.

Ein weiterer Punkt ist die Frage, ob sich die temporalen Relationen in erster Normalform befinden. Wenn ja, dann muß eine Objektgeschichte durch eine Menge von Tupeln dargestellt werden, wobei der Zeitstempel zum Primärschlüssel der Relation hinzugenommen werden muß. Dieser Ansatz ist konzeptionell sehr nahe am relationalen Datenmodell, bringt aber eine gesteigerte Auswertungskomplexität mit sich, da Tupel nicht geordnet sind. Im anderen Fall kann die gesamte Objektgeschichte in einem Tupel abgelegt werden. Dadurch ergeben sich Relationen folgender Struktur:

$$REL(\underline{A_1, \dots, A_n}, A_{n+1}, \dots, A_k, H(TS, A_{k+1}, \dots, A_l))$$

$A_1, \dots, A_n$ : Primärschlüssel der Relation

$A_{n+1}, \dots, A_k$ : nicht-temporale Attribute der Relation

$H(TS, A_{k+1}, \dots, A_l)$ : Wiederholungsgruppe aller zeitveränderlichen Aussagen der Geschichte der Relation mit dem Zeitstempel TS

Dies vereinfacht entscheidend Anfragen nach Objektgeschichten (Walk Thru Time Anfrage), insbesondere Anfragen über mehrere temporale Relationen (temporal join).

Eng verbunden damit ist die Frage, ob die Zeit auf Relationen-, Tupel- oder Attributebene angeordnet werden sollte. Eine Anordnung auf Relationenebene ist offensichtlich ungünstig, da bei jeder Änderung eines Tupels die gesamte Relation dupliziert werden müßte. Bei Anordnung auf Tupelebene ergibt sich in Abhängigkeit von der Normalform eines der folgenden Schemen:

$$NF^2 : REL(\underline{A_1, \dots, A_n}, A_{n+1}, \dots, A_k, H(TS, A_{k+1}, \dots, A_l))$$

$$1NF : REL(\underline{TS, A_1, \dots, A_n}, A_{n+1}, \dots, A_k, A_{k+1}, \dots, A_l)$$

Für ATKIS sollte auf eine dieser beiden Varianten zurückgegriffen werden. Der dritte Weg verlangt die Führung eines Zeitstempels für jedes temporale Attribut der Relation:

$$REL(\underline{A_1, \dots, A_n}, A_{n+1}, \dots, A_k, H_1(TS, A_{k+1}), \dots, H_m(TS, A_l))$$

Diese Möglichkeit ist für ATKIS aufgrund des hohen Aufwandes nicht diskutabel.

Ein weiterer bedeutender Unterschied zwischen den verschiedenen temporal erweiterten

Relationenmodellen ist die Frage der Bevorzugung aktueller vor historischen Tupeln und eine getrennte Speicherung dieser. Eine solche Vorgehensweise ist von großem Vorteil bei Systemen wie ATKIS, die bevorzugt mit aktuellen Daten arbeiten. Der Zugriff auf diese wird beschleunigt und es eröffnet sich die Möglichkeit, ältere Daten zu komprimieren oder auf andere Speichermedien auszulagern. Das Führen von Versionen von Metadaten (Strukturversionen) wird ermöglicht. Allerdings werden Anfragen nach Geschichten von Objekten erschwert.

Die solcherart unterschiedenen Tupel können unterschiedlich repräsentiert werden. Ist bei aktuellen Tupeln die Darstellung der Zeit als normales Attribut ausreichend, sollte bei historischen Tupeln die Zeit dem Primärschlüssel hinzugefügt werden.

Das letzte Unterscheidungsmerkmal ist die Art der Darstellung von Zeitstempeln. Für die Gültigkeitszeit sind einzelne Zeitpunkte, offene, halboffene oder geschlossene Zeitintervalle oder ein Gültigkeitszeitelement (Menge von Zeitpunkten oder -intervallen) denkbar, die Aufzeichnungszeit kann als einzelner Zeitpunkt (*now*), als Zeitintervall (*now, until changed*) oder als drei Zeitpunkte (Aufzeichnungszeitpunkte der Gültigkeitszeitpunkte und der *Todeszeit*) abgebildet werden. Mit Todeszeit wird der Zeitpunkt bezeichnet, ab dem keine Version eines Objektes mehr gültig ist.

Für ATKIS ist die Verwendung unterschiedlicher Zeitstempel für aktuelle und historische Daten günstig. Aktuelle Tupel sind durch einen Aufzeichnungszeitpunkt und einen Gültigkeitszeitpunkt (Beginn des Gültigkeitszeitraumes) zeitlich ausreichend eingeordnet. Für historische Daten ist zusätzlich die Aufzeichnung der Löschezit und des Endzeitpunktes des Gültigkeitszeitraumes notwendig.

## 1.5 Temporale Operationen für ein relationales Datenmodell

Für ein temporales ATKIS-Datenmodell sind einerseits Anfrage- und Änderungsoperationen, andererseits Operationen über aktuellen und historischen Tupeln zu unterscheiden. Anfrageoperationen können sich natürlich auch auf aktuelle und historische Tupel beziehen. Sie sollen jedoch kein Diskussionspunkt dieser Arbeit sein, da sie sehr spezifisch an eine Implementation anzupassen sind, die bis zum jetzigen Zeitpunkt noch nicht verwirklicht ist.

Für Änderungsoperationen, an dieser Stelle ein Oberbegriff für alle Operationen, die Manipulationen am Datenbestand vornehmen, lassen sich unabhängig von einer speziellen Implementation gewisse Grundsätze festhalten.

### 1.5.1 Änderungsoperationen für aktuelle Tupel

Als Voraussetzung für andere Operationen muß zunächst die Beförderung und Degradierung von aktuellen Tupeln entsprechend der aufgestellten Regeln bereitgestellt werden. Des weiteren müssen neue Objekte erzeugt werden können. Dabei werden Gültigkeitszeit und Aufzeichnungszeit auf *now*<sup>4</sup> gesetzt. Die neuen Tupel sind flüchtige Versionen. Für bereits existierende Objekte müssen neue temporale Instanzen erschaffen werden können. Die alte Version wird zur freigegebenen Version befördert, wird historisches Tupel und bekommt als Gültigkeitsendzeit die (Gültigkeitszeit des neuen Tupels) -1 (bezüglich der Granularität). Die neu erzeugte Version ist eine flüchtige Version. Die letzte notwendige Operation ist das logische Löschen eines Objektes. Das aktuelle Tupel wird historisches Tupel und bekommt als Gültigkeitsendzeit (*now*) -1. Eine aktuelle Version existiert dann nicht mehr.

### 1.5.2 Änderungsoperationen für historische Tupel

Für historische Tupel sind nur zwei Operationen notwendig. Die erste ist das Erzeugen eines Tupels in Geschichte. Das entspricht der Änderung von historischen Tupeln. Das Tupel wird sofort freigegebene Version. Das Gültigkeitszeitintervall ist frei wählbar, darf sich aber nicht mit dem des aktuellen Tupels überschneiden und muß innerhalb der Lebenszeit des Objektes liegen. Die Aufzeichnungszeit ist *now*. Das Tupel überlagert alle Tupel mit älterer Aufzeichnungszeit. Dies ist wichtig für Anfrageoperationen.

Die zweite ist das Löschen eines Tupels in Geschichte. Die Löschezit wird *now* gesetzt. Das Löschen eines Tupels kann zu Lücken in der Objektgeschichte führen. Das ist bei Anfrageoperationen zu berücksichtigen.

---

<sup>4</sup> Mit *now* wird der aktuelle Zeitpunkt bezeichnet.

## 2 Versionen von Metadaten

Die Geschichte der Geoinformationssysteme und ATKIS zeigt, daß Modelle für Geodaten selten über längere Zeiträume unveränderlich sind. Die beschreibende Struktur verändert sich durch die rasante Entwicklung der Anforderungen an Geodaten. Auch die komplizierte Modellierung und die damit oft notwendig werdenden Korrekturen an Strukturen tragen zu diesem Prozess bei. Auf der anderen Seite besteht aber nicht nur ein Bedarf an aktuellen Daten, es werden auch ältere Versionen von Geodaten benötigt. Darüber hinaus ist es notwendig, nachträgliche Korrekturen in ältere Datenbestände einzubringen, dabei aber die originalen Daten zu erhalten. All dies erfordert eine komplexe Zeitverwaltung auf Daten- und Metadatenebene. Um die Voraussetzungen dafür zu schaffen, sollen im folgenden einige Konzepte zur Verwaltung von Versionen von Metadaten (Strukturversionen) entwickelt und näher untersucht werden.

### 2.1 Allgemeine Begriffe

Zunächst sollen einige grundlegende Begriffe eingeführt werden.

**Definition 2.1 (Metadaten)** *Metadaten beinhalten Informationen über Struktur und Eigenschaften von Daten.*

Metadaten erlauben den automatischen Zugriff auf komplexe, sich evt. dynamisch ändernde Datenstrukturen effizient zu gestalten.

**Definition 2.2 (Version von Metadaten)** *Eine Version ist ein Zustand der Metadaten. Dabei kann eine Version über einen längeren Zeitraum gültig sein. Eine neue Version entsteht, wenn Metadaten in einer Weise verändert werden, so daß sich die Struktur oder die Eigenschaften der beschriebenen Daten ändern.*

**Definition 2.3 (Tabellentyp)** *Ein Tabellentyp ist eine Klasse von gleichartigen Tabellen, welche in gleicher Weise definierte und nach einheitlichen Namenskonventionen bestimmte Schlüssel- und Fremdschlüsselattribute enthalten, in gleicher Weise mit Tabellen anderer Typen verbunden sind (über Primär-Fremdschlüsselbeziehungen und weiterführende Verknüpfungen) und gleiche oder artgleiche technologische Attribute beinhalten (z.B. für Zeitverwaltung). Eine Tabelle eines Tabellentypes heißt Instanz dieses Tabellentypes. Informa-*

tionen über Tabellentypen sind üblicherweise nicht in Metainformationssystemen verfügbar und sind somit zusätzliche Voraussetzung für eine Interpretation der Metadaten.

**Definition 2.4 (Tabellencluster)** Als Tabellencluster bezeichnet man eine Menge von Tabellentypen, die sich aufgrund von Primär–Fremdschlüsselbeziehungen oder anderer Verknüpfungen gegenseitig bedingen und erfordern. Eine Instanz eines Tabellenclusters ist eine Menge real existierender Tabellen, wobei von jedem Tabellentyp des Tabellenclusters jeweils eine Instanz existiert.

## 2.2 Metadaten von zeitveränderlichen Daten

Für die nachfolgenden Überlegungen nehmen wir an, daß nur die Metadaten veränderlich sind, die Struktur der Metadatenbank aber fest ist. Metadaten verändern sich nur bezüglich einer linearen Zeitachse, bitemporale Operationen wie Veränderungen in der Vergangenheit werden nicht unterstützt. Des weiteren unterscheiden wir zwei Arten von Veränderungen von Metadaten, die, welche die Existenz von Tabellen beginnen oder beenden, und die, die Änderungen von existierenden Tabellen nach sich ziehen. Das können Veränderungen der Spalten und der Eigenschaften von Tabellen sein.

Es wird nur der spezielle Fall betrachtet, daß sowohl eine Versionierung der Metadaten als auch der eigentlichen Daten erfolgt. Daraus ergeben sich bereits einige Einschränkungen.

Die Granularität der Zeit bezüglich der Metadaten muß in der Granularität der Zeit der beschriebenen Daten enthalten sein. Das heißt zu jedem technischen Zeitpunkt bezüglich der beschriebenen Daten darf nur eine Version der Metadaten gültig sein.

Des weiteren dürfen Tabellen zwar neu erzeugt, später jedoch nur wieder gelöscht werden, wenn sie leer sind, da jede nichtleere Tabelle freigegebene Versionen von Objekten beinhalten kann, die nie wieder gelöscht werden dürfen. Es ist nur eine Markierung der Tabelle als *inaktiv* möglich.

Wird eine neue Instanz eines Tabellentypes erzeugt, der Element eines Tabellenclusters ist, so müssen auch Instanzen aller anderen Tabellentypen dieses Clusters erzeugt werden.

Die folgenden Komponenten einer Tabelle dürfen sich nicht mehr verändern, sobald Daten in die Tabelle eingebracht sind:

- der Name der Tabelle,



- der Tabellentyp, also alle Primär und Fremdschlüssel sowie allgemeine Attribute für die Zeitverwaltung und evtl. stets vorhandene Attribute dieses Tabellentyps inklusive aller diese Spalten betreffenden Integritätsbedingungen,
- Spalten, in denen sich Daten befinden, deren Informationsgehalt nicht vollständig in anderen Spalten enthalten ist, die also nicht redundant abgelegt sind.

Existierende Tabellen können durch die folgenden Operationen verändert werden:

- Anfügen einer Spalte,
- Erweitern, Einschränken oder neu Definieren einer Integritätsbedingung, wenn dadurch keine in dieser oder einer anderen Tabelle enthaltenen Tupel undefiniert werden,
- Löschen einer redundanten Spalte, sofern die Recherchierbarkeit historischer Daten dadurch nicht eingeschränkt wird, und
- logisches Löschen einer Spalte, in die in Zukunft keine Daten mehr eingebracht werden sollen. Jedem neuen Tupel der Tabelle wird für die logisch gelöschte Spalte ein definierter Wert mit der Bedeutung „Attribut nicht mehr belegt“ zugewiesen.

Jede dieser Operationen darf nur ausgeführt werden, wenn sich dadurch der Tabellentyp nicht ändert. Die Metadatenbank muß so konzipiert und kontrolliert werden, daß nur Veränderungen an Metadaten erlaubt sind, die den vorgenannten Operationen entsprechen. Ausgenommen hiervon ist die Schaffung neuer Tabellentypen.

Diese weitreichenden Einschränkungen sind notwendig, um die effiziente Verwaltung temporaler Daten in zeitveränderlichen Strukturen zu ermöglichen, ohne daß jede Strukturänderung die Schaffung neuer Tabellen nach sich zieht. Eine solche Vorgehensweise wäre flexibler, aber auch ungleich schwerer zu beherrschen. Die ohnehin sehr hohe Anzahl von Tabellen der ATKIS-Datenbank würde sich weiter vergrößern, Recherchen zu Objektgeschichten würden erschwert.

## 2.3 Zusammenfassung

In den vorangegangenen Abschnitten wurde eine kurze Einführung in die äußerst komplexe Materie der temporalen Datenmodelle gegeben und verschiedene Anwendungsmöglichkeiten

für den Aufbau des ATKIS–Datenmodells diskutiert. Dabei wurde die prinzipielle Eignung bestimmter temporaler Modelle für ATKIS nachgewiesen und Wege für die Verwirklichung aufgezeigt. Dies stellt aber lediglich einen Einstieg in eine temporale ATKIS–Modellierung dar. Verschiedene Fragen der Modellierung und einer konkreten Implementation, wie z.B. die Gestaltung einer temporalen Anfragesprache, sind noch offengelassen worden und müssen noch gelöst werden, bevor die hier dargelegten Ansätze verwirklicht werden können. Dieses muß aber Anliegen weitergehender Untersuchungen sein.

Die in diesem Report aufgestellten Grundsätze für temporale Metainformationssysteme führten bislang noch nicht zur Schaffung eines temporalen Metainformationssystemes. Sie fanden aber Beachtung bei Konzeption und Implementation des in [Ram97/2] vorgestellten Metainformationssystemes DST, das auf einfache Art und Weise um temporale Attribute erweitert werden kann. Diese Aufgabe und die Umsetzung der gegebenen Anregungen müssen Zielsetzung weiterführender Bemühungen sein.

# Glossar

**Ableitbare Geschichte:** Geschichte von Objektzuständen, die sich sprunghaft und zwischen diesen Zeitpunkten kontinuierlich verändern. Die kontinuierliche Änderung folgt einem bekannten funktionalen Zusammenhang.

**Arbeitsversion (working version):** Eine Arbeitsversion eines Objektes kann gelöscht, aber nicht geändert werden. Sie besitzt eine mittlere Robustheit. (vgl. *flüchtige Version* und *freigegebene Version*)

**Attribut:** Ein Attribut bezeichnet quantitative und qualitative Eigenschaften, die ein *Objekt* oder ein *Objektteil* näher beschreiben. [ATKIS] Attribute können konkrete Angaben wie Namen und Maße, aber auch durch vierstellige Zahlen kodierte Angaben enthalten. Sie werden als Unterscheidungskriterium von *Objekten* und *Objektteilen* herangezogen.

**Aufzeichnungszeit  $t_a$ :** Zeitpunkt, zu dem eine Aussage in ein Datenbanksystem eingebracht wurde.

**Aufzeichnungszeitmodell:** Temporales Datenmodell auf der Basis der Aufzeichnungszeit. Es wird Aufzeichnungszeit gleich Gültigkeitszeit bzw. Beginn des Gültigkeitszeitintervalls angenommen.

**Beförderung (promotion) einer Version:** Vorgang der Überführung einer *flüchtigen Version* in eine *Arbeitsversion* und von einer *Arbeitsversion* in eine *freigegebene Version*.

**Beobachtungszeit  $t_b$ :** Zeitpunkt, zu dem die Richtigkeit einer Aussage erkannt wurde.

**Bitemporales Modell:** Temporales Datenmodell auf der Basis der von einander unabhängigen Aufzeichnungs- und Gültigkeitszeit. Die Gültigkeitszeit kann ein Zeitpunkt oder ein Zeitintervall sein.

**Chronon:** In Bezug auf das Maßsystem kleinste sinnvoll zu unterscheidende Zeiteinheit.

**Degradierung (demote) einer Version:** Vorgang der Überführung einer *Arbeitsversion* in eine *flüchtige Version*. Eine *freigegebene Version* kann nicht degradiert werden.

**Ereignisorientierte Geschichte:** Geschichte von Objektzuständen, die nur zu gewissen Zeitpunkten definiert sind. Es kann Zeitpunkte ohne und Zeitpunkte mit mehreren Objektzuständen geben.

**Flüchtige Version (transient version):** Eine flüchtige Version eines Objektes ist eine temporäre Version, die geändert und gelöscht werden kann. Für das Objekt geforderte Integritätsbedingungen können bei flüchtigen Versionen für ungültig erklärt werden. Sie besitzt die geringste Robustheit. (vgl. *Arbeitsversion* und *freigegebene Version*)

**Freigegebene Version (released version) :** Eine freigegebene Version eines Objektes kann weder geändert noch gelöscht werden. Sie besitzt die höchste Robustheit. (vgl. *flüchtige Version* und *Arbeitsversion*)

**Geschichte:** Eine Geschichte ist eine zeitlich geordnete Folge von Zuständen oder Ereignissen von Aussagen, die zu einem Objekt oder einer Beziehung gehören. [Här84]

**Granularität, konstante:** Eine Granularität ist eine Einteilung einer linearen Zeitachse oder eines Ausschnittes einer linearen Zeitachse und wird bestimmt durch eine Länge und einen Ausgangspunkt. Die Länge wie auch der Ausgangspunkt werden in Chronons der zugrundeliegenden Uhr angegeben.

**Granularität, variable:** Eine Granularität, deren Länge im Gegensatz zur *konstanten Granularität* variabel sein kann.

**Granulat:** Bezugsmaß für die durch die Granularität definierten Zeitabschnitte. Ein Granulat besitzt eine konstante Größe bei konstanter Granularität der Zeit.

**Gültigkeitszeit  $t_g$ :** Zeitpunkt oder Zeitintervall, in dem eine Aussage wahr ist bzw. als wahr angesehen wird. Die Gültigkeitszeit kann in eine oder beide Richtungen unbeschränkt sein.

**Gültigkeitszeitmodell:** Temporales Datenmodell auf der Basis der Gültigkeitszeit. Aufzeichnungszeit und Beobachtungszeit werden nicht beachtet.

**Modellobjekt:** Die Abbildung eines topographischen Objektes in der DLM-Datenbank. Auch als DLM-Objekt bezeichnet.

**Objekt, topographisches:** Konkreter, einer *Objektart* zugeordneter, geometrisch begrenzter, durch *Attribute* beschriebener und mit Namen versehbarer Gegenstand der Landschaft. Es besteht aus einer nach fachlichen Gesichtspunkten gebildeten Menge von *Objektteilen*. Topographische Objekte können unter Berücksichtigung des unterschiedlichen Strukturierungsgrades punktförmig, linienförmig oder flächenförmig sein. Auf der Erdoberfläche selbst sind sie stets flächenförmig. [ATKIS]

**Objektart:** Zusammenfassende Bezeichnung für gleichartige *topographische Objekte* zum Zwecke der Katalogisierung. [ATKIS]

**Objektattribut:** Attribut, das der Objektebene zugeordnet ist.

**Objektteil:** Ein Objektteil ist ein konkreter, geometrisch begrenzter und durch einheitliche *Attribute* und *Relationen* gekennzeichnete(r) Gegenstand der Landschaft als Teil eines *Objekts*. Beim Wechsel eines *Attributs* entsteht ein neuer Objektteil. [ATKIS]

**Objektteilattribut:** Attribut, das der Objektteilebene zugeordnet ist.

**Todeszeit:** Zeitpunkt, ab dem ein Objekt keine gültige Version mehr besitzt.

**Version:** Eine Version ist ein Zustand eines Objektes. Sie besitzt einen Gültigkeitszeitraum und ist je nach Art der Version unveränderlich oder veränderlich. Nach der Erzeugung einer ersten Version werden alle weiteren Versionen eines Objektes von dieser rekursiv abgeleitet und bilden eine Versionenhierarchie.

**Zeitdauer:** Die Zeitdauer eines Zeitintervalls im technischen Sinne ist die Differenz von Anfangs- und Endpunkt des Intervalls in Chronons. Dabei wird für den Anfangspunkt der Beginn und für den Endpunkt das Ende des zugeordneten Zeitabschnittes auf dem Zeitstrahl gerechnet. Bei konstanter Granularität kann die Zeitdauer auch in Granulaten gerechnet werden.

**Zeitintervall:** Abbildung einer *natürlichen* Zeitspanne auf die betrachtete Granularisierung der Zeit. Dabei wird der Anfangs- und Endpunkt der Zeitspanne auf die entsprechenden *technischen* Zeitpunkte abgebildet.

**Zeitpunkt, technischer:** Im Unterschied zum eigentlichen Sinne des Wortes beschreibt ein technischer Zeitpunkt nicht einen konkreten Punkt, sondern ein Abschnitt der

Zeitachse entsprechend der betrachteten Granularität. Dieser *technische* Zeitpunkt wird aus einem *natürlichen* Zeitpunkt durch Abbildung auf den entsprechenden Abschnitt der betrachteten Granularität gewonnen.

**Zeitstempel:** Die Kombination von Zeitpunkten und –intervallen, die für jede Version eines Objektes gespeichert werden.

**Zustandserhaltende Geschichte:** Geschichte von Aussagen, die jeweils einen neuen Zustand eines Objektes oder einer Beziehung beschreibt. Ein Zustand bleibt bis zum Inkrafttreten eines neuen Zustandes invariant.

**Zustandsverändernde Geschichte:** Geschichte von kontinuierlich veränderlichen Objektzuständen. Der funktionale Zusammenhang von Zeit und Zustand ist stetig, aber unbekannt. Die Erfassung kann exemplarisch durch Zustände hinreichender zeitlicher Dichte erfolgen.

# Literatur

- [AGH96] AG Hannover im Vorhaben ALK/ATKIS: *Bezieher Sekundärnachweis*. Niedersächsisches Landesverwaltungsamt, Landesvermessung, Okt. 1996
- [ATKIS] AdV: *Amtliches Topographisch-Kartographisches Informationssystem ATKIS, Gesamtdokumentation*. AdV, Hannover, 1995
- [Ben79] J. Ben-Zvi et al.: *PGS-1, Payroll Generator System-1*. In Proceedings of the I.P.A. Conference, S.309-332, Tel-Aviv, Israel, 1979
- [Ben82] J. Ben-Zvi: *The Time Relational Model*. PhD. Diss., Computer Science Department, UCLA, 1982
- [Ber] Jochen van den Bercken, Bernhard Seeger: *Query Processing Techniques for Multiversion Access Methods*. Universität Marburg, technischer Report
- [Cat91] Roderic Geoffrey Galton Cattell: *Object Data Management*. Addison-Wesley Publishing Company, 1991
- [Cli83] James Clifford, D. S. Warren: *Formal Semantics for Time in Databases*. ACM TODS, Vol. 8, No. 2, S. 214 - 254, Juni 1983
- [Cli86] J. Clifford, G. Ariav: *Temporal Data Management: Models and Systems*. Ablex Publishing Corporation, Norwood, New Jersey, 1986
- [Gri95/1] Bernd Griefahn: *Zugriffsmodelle für temporale Daten*. Rostocker Informatik-Berichte 17, 1995
- [Gri95/2] Bernd Griefahn: *Eine Erweiterung der relationalen Algebra für temporale Attribute mit unterschiedlichen Zeitfolgetypen*. Universität Rostock, technischer Report, 1995
- [Gri96] Bernd Griefahn: *Implementierungstechniken für temporale Daten*. Universität Rostock, technischer Report, 1996
- [Här84] Th. Härder: *Überlegungen zur Modellierung und Integration der Zeit in temporalen Datenbankensystemen*. Bericht Nr. 19/84, Universität Kaiserslautern, Okt. 1984
- [Jen92] Christian S. Jensen, J. Clifford, S. K. Gadia, A. Segev, R. T. Snodgrass: *A glossary of temporal database concepts*. SIGMOD RECORD, 21(3):35-43, Sep. 1992

- [Jen93/2] Christian S. Jensen, Richard T. Snodgrass: *The TEMPIS Project. Proposal for a Data Model for the Temporal Structured Query Language*. TEMPIS Technical Report No. 37, University of Arizona, Tucson, AZ 85721, 1993
- [Käf92] Wolfgang Käfer: *Geschichts- und Versionenmodellierung*. Dissertation, Universität Karlsruhe, 1992
- [Kim90] Won Kim: *Introduction to Object-Oriented Databases*. The MIT Press, Massachusetts Institute of Technology, Cambridge, 1990
- [Kim95] Won Kim: *Modern Database Systems*. ACM Press, 1995
- [Org93] Mehmet A. Orgun, Hausi A. Müller: *A Temporal Algebra Based on an Abstract*. In Proceedings of the ADC'93, 1993
- [Pis94] Niki Pissinou u.a.: *Towards an Infrastructure for Temporal Databases: Report of an Invitational ARPA/NSF Workshop*. University of Arizona, Technical Report 94-01, Januar 1994
- [Ram97/1] Jan Ramsch: *Implementation des ATKIS-Datenmodells in GIS mit relationaler Attributverwaltung unter dem Aspekt der Fortführung*. Diplomarbeit, Universität Leipzig, April 1997
- [Ram97/2] Jan Ramsch, Dieter Sosna: *Das Metainformationssystem DST für ATKIS*. Technischer Report 11(97), Universität Leipzig, August 1997
- [Sno90] R. Snodgrass: *Temporal databases: Status and research directions*. ACM SIGMOD Record, 19(4):83-89, Dez. 1990
- [Sno93] Richard T. Snodgrass: *Overview of the Special Section on Temporal Database Infrastructure*. University of Arizona, 1993
- [Sno95/1] Richard T. Snodgrass, The TSQL2 Language Design Committee: *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, Boston, Dordrecht, London, 1995
- [Sno95/2] Richard T. Snodgrass: *Temporal Object-Oriented Databases: A Critical Comparison*. Modern Database Systems, S. 386 - 408, ACM Press, 1995
- [Sto87] M. Stonebraker: *The Design of the POSTGRES Storage System*. In Proceedings of the Conference on Very Large Databases, Brighton, England, S.289-300, Sep. 1987



- [Tan86] Abdullah Uz Tansel: *Adding Time Dimension to Relational Model and Extending Relational Algebra Information Systems*. Vol. 11, Nr.4, S. 343-355, 1986
- [Tan93] Abdullah Uz Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, R. T. Snodgrass: *Temporal databases: theory, design, and implementation*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, 1993
- [Wed] Hartmut Wedekind: *Are the Terms „Version“ and „Variant“ Orthogonal to One Another*. Universität Erlangen–Nürnberg