



Fakultät Umweltwissenschaften  
Fachrichtung Geowissenschaften

Professur für Geoinformationssysteme  
**Bachelorarbeit**  
im Studiengang Geodäsie und Geoinformation

Realisierung der Zeitkomponente einer  
Geodatenbank durch einen ISO19108  
konformen Datentyp

*Eingereicht von:* Peter Broßeit

*Betreuer:* Prof. Dr. Lars Bernard, TU Dresden  
Dipl.-Geogr. Matthias Müller, TU Dresden

*Bearbeitungszeitraum:* 10.07 – 17.09.2012

## **Kurzfassung**

Anwender aus verschiedensten Fachgebieten stellen weitreichende Anforderungen an die Modellierung der Zeitdimension in den Geoinformationssystemen. Von Interesse ist dabei oft speziell eine Analyse der Dynamik der betrachteten Phänomene. Das Erfordernis, geeignete Methoden zur Erfassung und Verarbeitung von Zeitinformationen bereitzustellen, stand in den letzten Jahrzehnten im Fokus diverser Untersuchungen und Publikationen. Im Kontext dieser Entwicklung ist im Jahr 2002 auch eine ISO-Norm (ISO19108:2002, Zeitliche Schema) zu dieser Thematik erschienen. Die Möglichkeiten, die hinsichtlich dessen von aktuellen Geoinformationssystemen angeboten werden, bleiben im Allgemeinen hinter der genannten Norm zurück.

In dieser Arbeit wird untersucht, inwiefern das Konzept des Zeitlichen Schemas der ISO durch einen Abstrakten Datentyp in einer Geodatenbank umgesetzt werden kann. Es erfolgt die Konzeption hierfür notwendiger Objekte und Funktionen. Ziel ist dabei die integrierte Erfassung, Verwaltung und Analyse von Zeitpunkten, -perioden sowie einfacher zeitlicher Komplexe. Weiter wird für den Informationsaustausch zwischen Nutzer und Datenbanksystem, aufbauend auf bestehenden Standards, eine textuelle Repräsentation der Objekte konzipiert.

Die Umsetzbarkeit des Konzepts wurde durch eine prototypische Implementierung, in einer PostgreSQL-Datenbank mit PostGIS-Erweiterung, untersucht. Die prinzipielle Machbarkeit konnte nachgewiesen werden, Details werden in der Arbeit behandelt.

## **Abstract**

Users from various fields of expertise place a wide range of demands on the subject of time modelling in geographic information systems. Of particular interest in this area is often the analysis of the dynamics of spatial features. The demand of providing suitable methods for the collection and processing of time data has been the focus of diverse research projects and publications over the last decades. In the context of the previously mentioned development, an ISO standard (ISO19108:2002, Temporal Schema) regarding this issue was published in 2002. In general, the capabilities of today's geographic information systems are far behind the theoretical potential provided by this ISO standard.

This bachelor thesis examines the feasibility of how the Temporal Schema of the ISO standard can be implemented in a Geodatabase with an abstract data type. Therefore necessary objects and functions are conceptualized. The aim is the integrated capture, management and analysis of instants, periods of time and simple temporal complexes. Furthermore, according to existing standards, a textual representation of objects is designed for the exchange of information between users and database systems.

The practicability of the concept is examined with an implementation prototype in a PostgreSQL database, using the PostGIS extension. In summary, it can be said that the theoretical feasibility of this approach is proven, the details are discussed in this thesis.

# Inhaltsverzeichnis

1.	Einführung und Zielsetzung .....	1
1.1	Zielsetzung .....	2
2.	Grundlagen .....	3
2.1	Kalender- und Uhrzeitsysteme.....	3
2.1.1	Kalendersysteme.....	4
2.1.2	Uhrzeitsysteme .....	5
2.2	Bedeutung der Zeit in GIS und Geodatenbanken .....	7
2.3	Begriffe und Konzepte zur Dynamik.....	8
2.3.1	Raum-zeitliche Modelle .....	8
2.3.2	Zeitbezugssysteme für GIS .....	9
2.3.3	Temporale Datenbanken .....	10
2.4	Zeitliches Schema der ISO .....	11
2.4.1	Temporale Objekte .....	11
2.4.2	Temporale Referenzsysteme und Zeitpositionen .....	13
2.4.3	Temporale Objekte und Geoobjekte.....	16
2.5	Zeitbehandlung in aktueller GIS-Software .....	18
2.5.1	Desktop-GIS.....	18
2.5.2	Geodatenserver .....	19
2.5.3	AAA-Modell .....	20
2.6	Zusammenfassung .....	21
3.	Konzeption .....	22
3.1	Konzept einer universellen Behandlung der valid time .....	23
3.2	Die time-Objekte .....	24
3.2.1	Klassifikation der Objekte .....	24
3.2.2	Bedeutung der zeitlichen Bezugssysteme .....	26
3.3	Textuelle Repräsentation der Objekte und Bezugssysteme .....	28
3.3.1	Kalendersysteme – ISO8601-Format .....	29
3.3.2	Bezugssysteme.....	30
3.3.3	Beschreibung der Geometrie in anderen als Kalendersystemen.....	31
3.4	Operationen .....	33
3.4.1	Basisfunktionen.....	33
3.4.2	Möglichkeiten raum-zeitlicher Analysen .....	38
4.	Beschreibung der Implementierung .....	41
4.1	Abstrakte Datentypen in PostgreSQL .....	41
4.1.1	Physischer Entwurf .....	42
4.2	Umgesetzte Referenzsysteme und Typen .....	43
4.2.1	Referenzsysteme.....	43
4.2.2	<i>Time</i> -Typen.....	45

4.3	Umgesetzte Funktionen .....	46
4.3.1	Basisfunktionen .....	46
4.3.2	Aufbauende PostgreSQL-Funktionen .....	49
5.	Fazit und Ausblick .....	52
	Quellenverzeichnis .....	55
	Selbstständigkeitserklärung .....	58
	Anhang .....	59

## Abbildungsverzeichnis

2-1:	Die 13 möglichen zeitliche Beziehungen zwischen zwei Zeitspannen nach der ISO19108 .....	12
3-1:	UML - Die Klassen TM_Instant und TM_Period .....	25
3-2:	UML - Die Klassen MultiInstant und MultiPeriod .....	25
3-3:	UML - Die Klassen RegularMultiInstant und RegularMultiPeriod .....	26
3-4:	UML - Übernommene TM_TemporalPosition-Subtypen .....	27
3-5:	UML - Konzept der internen und externen TM_TemporalPosition-Objekte .....	27
3-6:	Ablauf bei der Ermittlung der ISO-Fortführungsbeziehungen .....	39
4-1:	Einbindung Abstrakter Datentypen in PostgreSQL .....	42
4-2:	Die interne Struktur .....	43
4-3:	Eine simple Funktion als externe C-Funktion in PostgreSQL .....	46
4-4:	PostgreSQL-Anweisung – Deklaration des ADT geotime .....	47
4-5:	PgAdmin3 – Die relativePosition-Funktionen .....	48
4-6:	PgAdmin3 – Die distance-Funktionen .....	48
4-7:	Geometrie der verwendeten Testdaten .....	50
4-8:	PgAdmin3 – Vorgänger und Nachfolger verschiedener Staaten – Ausschnitt .....	50
4-9:	PgAdmin3 – Am häufigsten von Tornados betroffene Bundesstaaten der USA .....	51
4-10:	PgAdmin3 – Histogramm aus Instant-Objekten - Ausschnitt .....	52

## Tabellenverzeichnis

2-1:	Funktionen der TM_Primitive Objekte .....	13
3-1:	Auszug aus der Simple Feature Access Spezifikation – WKT Beispiele .....	23
3-2:	Beispiele für die sechs Time-Objekte des Datentyps .....	25
3-3:	Verwendete Terminale .....	29
3-4:	Beispiele für die textuelle Darstellung .....	32
3-5:	Der Aufzählungstyp relativePosition .....	33
3-6:	Konstruktoren und Casts .....	34
3-7:	Logische Operationen .....	34

3-8: Simple Zugriffe.....	35
3-9: Simple Datenmanipulation.....	36
3-10: ISO19108-Funktionen.....	36
3-11: Erweiterte ISO19108-Funktionen.....	37
3-12: Raum-zeitliche Pseudofunktionen.....	39
4-1: Implementierte Referenzsysteme.....	44
4-2: Implementierte Typen.....	46
4-3: Umgesetzte Funktionen – Ein- und Ausgabe des ADT geotime.....	47
4-4: ISO19108-Funktionen – Relative Position zweier Primitive.....	47
4-5: ISO19108-Funktionen – Zeitlicher Abstand zweier Primitive.....	48
4-6: Überladene Funktion – Temporale Bounding Box.....	48
4-7: Charakterisierung der verwendeten Testdaten.....	49
4-8: Raum-zeitliche Funktionen – Fortführungsbeziehungen.....	50
4-9: Raum-zeitliche Funktionen – Bezugsobjekt mit den häufigsten Ereignissen ...	51

## Abkürzungsverzeichnis

ADT	Abstrakter Datentyp
AdV	Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland
ALFIS	Amtliches Festpunktinformationssystem
ALKIS	Amtliches Liegenschaftskatasterinformationssystem
ATKIS	Amtliches Topographisch-Kartographisches Informationssystem
BIPM	<i>International Bureau of Weights and Measures</i>
BNF	Backus-Naur-Form
DBMS	Datenbankmanagementsystem
GIS	Geographisches Informationssystem
GiST	<i>Generalized Search Tree</i>
IERS	<i>International Earth Rotation and Reference System Service</i>
ISO	<i>International Organization for Standardization</i>
OGC	<i>Open Geospatial Consortium</i>
OSGeo	<i>Open Source Geospatial Foundation</i>
SI	<i>International System of Units</i>
SQL	<i>Structured Query Language</i>
SRID	<i>Spatial Reference System Identifier</i>
TAI	Internationale Atomzeit
UML	<i>Unified Modeling Language</i>
UT	<i>Universal Time</i>
UTC	<i>Coordinated Universal Time</i>
WKB	<i>Well-known Binary Representation for Geometry</i>
WKT	<i>Well-known Text Representation for Geometry</i>

# 1. Einführung und Zielsetzung

Die Auseinandersetzung mit der Zeit ist in der Geoinformatik, dort wo es um die Abstraktion der Wirklichkeit geht, eine sich unmittelbar aus dem Vorhaben ergebende Notwendigkeit. Einer Ortsangabe liegt immer eine Position in den vier Dimensionen der Raumzeit zugrunde (GIANCOLI 2006 S.1223, PEUQUET 2002 S.20). Wie weit die erforderliche Abstraktion hier gehen darf, richtet sich nach der zu lösenden Fragestellung. Theoretisch reicht das Spektrum von einer statischen Betrachtungsweise bis zu der Modellierung des Raum-Zeit-Kontinuums. In der Praxis ist häufig die zeitliche Variation bestimmter Eigenschaften der betrachteten Phänomene von Interesse.

Dies gilt auch für die Geowissenschaften. So schlägt sich das besondere Interesse an den zeitlichen Veränderungen auch in der Strategieschrift der Senatskommission für Geowissenschaftliche Gemeinschaftsforschung der Deutschen Forschungsgemeinschaft nieder, die den Begriff der Dynamik schon im Titel<sup>1</sup> trägt (WEFER 2010). Aber auch für viele weitere Fachgebiete, die durch die Geoinformationstechnologien unterstützt werden, ist eine adäquate Behandlung des Zeitbezugs der Geoinformationen zwingende Voraussetzung für den zielführenden Einsatz der Technologien (BARTELME 2005 S.221).

Das Erfordernis, Methoden zur Erfassung und Verarbeitung von Zeitinformationen bereitzustellen, stand in den letzten Jahrzehnten immer wieder im Fokus diverser Untersuchungen und Publikationen. Im Laufe dieses Prozesses haben sich verschiedene Ansätze zur Modellierung der Zeit entwickelt. Diese haben zum Teil auch ihren Platz in den GIS-Standardwerken gefunden. So sei hier beispielhaft auf BARTELME (2005), BILL (2010) sowie WORBOYS UND DUCKHAM (2004) verwiesen, die diesem Aspekt jeweils einen Abschnitt widmen. Im Kontext dieser Entwicklung ist im Jahr 2002 auch die ISO19108 - Zeitliches Schema - von dem Technischen Komitee ISO/TC211<sup>2</sup> publiziert worden (Iso 2002a). Dieses Schema dient als Grundlage für die Zeitbehandlung in weiteren Normen des ISO/TC211 und fließt so auch in die Arbeiten des OGC<sup>3</sup> ein, welches einige Standards in Kooperation mit der ISO erarbeitet und veröffentlicht. So bildet es z.B. auch die Basis für die temporalen Aspekte der *Geography Markup Language* (Ogc 2007 S.148). Es kann hier also eine weitreichende Relevanz diese Schemas festgestellt werden, das im Jahr 2005 auch auf europäischer und deutscher Ebene als Norm übernommen wurde.

Infolge der weitreichenden Anforderungen der Nutzer, aber auch durch die konzeptuellen Fortschritte hinsichtlich der Zeit, gilt die zwingende Berücksichtigung dieser Dimension in GIS heute als Konsens in Theorie und Praxis.

---

<sup>1</sup> Dynamische Erde – Zukunftsaufgaben der Geowissenschaften – Strategieschrift.

<sup>2</sup> Die ISO/TC211 ist das technische Komitee der ISO für den Bereich Geoinformation/Geomatik.

<sup>3</sup> Das *Open Geospatial Consortium* ist eine internationale Organisation, die sich der Entwicklung öffentlich zugänglicher Standards für Geodaten und -dienste in einem offenen Konsensprozess verschrieben hat.

Klassische Systeme, die die Vorteile der computergestützten Datenverarbeitungen für Geodaten erschlossen haben, werden dem im Allgemeinen nicht gerecht. Als Grund hierfür wird die statische Welt der analogen Karten gesehen, zu der ein wichtiger Bezug bestand (BILL 2010 S.193, PEUQUET 2001 S.13). Davon abgesehen mussten zunächst aber auch die konzeptuellen und technischen Voraussetzungen geschaffen werden.

Die Ergebnisse der ersten Arbeiten, die sich bereits Ende der achtziger Jahre mit der Erweiterung einer Geodatenbank um zeitliche Funktionalitäten auseinandersetzten, wurden als pragmatische Lösungen ohne universellen Charakter eingestuft (PEUQUET 2001 S.14f). Auch in der Informatik existiert bis heute keine allgemein anwendbare Methode zur Zeitmodellierung, so kommen FURIA et al. (2010) zu dem Schluss:

*“despite the common understanding that time is a basic, unique conceptual entity, there are ‘many notions of time’ in our reasoning; this is reflected in the adoption of different formal models when specifying and analyzing any type of system where timing behavior is of any concern.”* (FURIA et al. 2010 S.6:53).

Eine besondere Schwierigkeit im Zusammenhang mit GIS besteht hier in der Integration der Zeit- und Raumdimensionen. Mittlerweile werden zwar von diversen Arbeitsplatz-GIS und Geowebdiensten Zeitfunktionalitäten angeboten, jedoch ist die Kombination der Zeit mit den Raumdimensionen für viele Anwendungen nicht weitreichend genug. So ist die Einbindung der Zeitdimension in den Geoinformationstechnologien als laufender Prozess zu bewerten.

## 1.1 Zielsetzung

In dieser Arbeit soll untersucht werden, inwiefern die Modellierung des Zeitbezugs durch die Erweiterung einer bestehenden Geodatenbank um einen geeigneten Datentyp realisiert werden kann. Die zu entwickelnde Datenstruktur soll dabei verschiedenartige Zeitinformationen repräsentieren können. Dabei sind Konzepte für Zeitpunkte, -perioden und einfacher temporaler Komplexe zu integrieren. Die Konzeption wird sich dabei an dem Zeitlichen Schema der ISO orientieren, insofern dies möglich ist, um bestehende Konventionen aufzugreifen und deren Potential zu nutzen.

In der Praxis ist es üblich, stattdessen auf bestehende Datentypen, insbesondere Datums-typen, zurückzugreifen<sup>4</sup>. Daraus ergibt sich, dass Zeitperioden nur implizit, durch getrennt verwaltete Start- und Endzeitpunkte, abgebildet werden. Zeitkomplexe könnten nur durch eine große Anzahl von Zeitattributen oder durch ansonsten kongruente Datensätze verwaltet werden. Wird hier, wie teilweise praktiziert, auch ein Intervalltyp mit

---

<sup>4</sup> Um hier einige Beispiele zu nennen sei auf ArcGIS (ESRI 2012a), QuantumGIS (GRASER 2011) und GeoServer (GEOSOLUTIONS 2011a) verwiesen.

eingebunden, bleibt die Problematik der Erfassung komplexer Typen, ebenso wie ein einheitlicher Zugriff auf die Daten, ungelöst.

Die Vorteile, die eine alternative Herangehensweise hier schaffen könnte, sind zum einen die explizite Modellierung von Zeitperioden und Komplexen. Zum anderen könnten hierauf dann Funktionen aufbauen, die homogene Abfragen für alle Objekte mit Zeitbezug ermöglichen würden.

Dabei wird keinesfalls der Anspruch erhoben, die bestehenden Konzepte weiterzuentwickeln, sie sollen nur in spezieller Weise umgesetzt werden.

Im nachfolgenden Teil dieser Arbeit werden zunächst die Grundlagen für ein solches Schema geschaffen, indem die Eigenschaften von Zeitbezugssystemen und die relevanten Konzepte zur Zeitbehandlung in der Geoinformatik herausgearbeitet werden. Hierauf wird im dritten Kapitel die Konzeption der Datenstruktur aufbauen. Im vierten Kapitel wird die entsprechende Implementierung erläutert. Eine Einschätzung der Ergebnisse ist Gegenstand des fünften Kapitels.

## 2. Grundlagen

In diesem Abschnitt werden die notwendigen Grundlagen für die beabsichtigte Konzeption dargestellt. Zunächst wird dazu auf generelle Aspekte von Kalender- und Uhrzeitsystemen eingegangen, welche die Basis für die meisten Zeitangaben bilden. Dem wird ein Einstieg in die Thematik der Zeit im Zusammenhang mit der Geoinformatik folgen. Zunächst wird es dabei um die Bedeutung der Zeit in GIS gehen, dann um die wesentlichen Definitionen. Anschließend wird mit dem zeitlichen Schema der ISO ein geschlossenes Konzept zur Modellierung der Zeit vorgestellt. Zum Schluss dieses Abschnitts wird beispielhaft auf aktuelle Umsetzungen eingegangen und es erfolgt eine kurze Zusammenfassung.

### 2.1 Kalender- und Uhrzeitsysteme

Die Gegenwart in einen Kontext mit Zukunft und Vergangenheit einordnen zu können, ist seit Anbeginn der Menschheitsgeschichte erforderlich. So war es schon für die Menschen der Frühgeschichte notwendig, sich auf die verschiedenen Bedingungen der Jahreszeiten einzustellen. Durch die wachsende Komplexität menschlicher Gesellschaften stiegen auch die Anforderungen an eine Systematisierung der Zeit als solide Planungs- und Organisationsgrundlage in allen Lebensbereichen. Solche Systeme basieren in der Regel auf Himmelsbeobachtungen, aber auch solchen von wiederkehrenden Naturereignissen. Verschiedene Zivilisationen entwickelten ihre eigenen Systeme und bauten dabei zum Teil auf bestehenden auf (RICHARDS 1998 S.5-7). Im Folgenden sollen die wesentlichen Gemeinsamkeiten dieser Systeme so wie das heute Gebräuchlichste vorgestellt werden.



### 2.1.1 Kalendersysteme

Die elementare Einheit eines jeden Kalenders ist der Tag, der früher durch den beobachteten Sonnenaufgang bzw. -untergang definiert war und an dem sich das Leben der Menschen und vieler Tiere ausrichtet. Heute wird unter einem Tag im Allgemeinen der Bürgerliche Tag verstanden, die Zeitdauer, die vergeht bis die mittlere Sonne erneut über demselben Meridian steht<sup>5</sup>. Die Tage werden meist in Monaten und Jahren gruppiert, die wiederum selbst auf astronomischen Perioden basieren. Ein Monat ist immer in einer bestimmten Weise mit dem Umlauf des Mondes um die Erde verknüpft, wobei diese Kopplung zwischen Erdtrabant und Kalendermonat in verschiedenen Systemen unterschiedlich stark ausgeprägt ist. Ebenso uneinheitlich sind die Definitionen des Zeitpunktes an dem ein Monat beginnt oder endet. Unter einem Jahr wurde historisch meist die Periode der Jahreszeiten verstanden, die entweder über Beobachtungen des Sonnenstandes oder bestimmter Umweltphänomene definiert wurde. Heute entspricht dies weitgehend dem Tropischen Jahr der Astronomie, dessen Länge aufgrund verschiedener Einflüsse variiert (RICHARDS 1998 S.24-36).

Ein wesentliches Problem von Kalendersystemen besteht darin, dass sich weder die Dauer eines Monats noch die eines Jahres in Tagen ganzzahlig exakt angeben lässt<sup>6</sup>. In frühen Kalendern wurden Monats- und Jahresanfang häufig empirisch durch Beobachtungen bestimmt, so dass dies vernachlässigbar war. Ein solcher indeterministischer Kalender ist für Zeitangaben in der Zukunft jedoch kaum geeignet. Daher wurden Systeme entwickelt, die durch Regeln die Anzahl der Tage und Monate sowie deren Abfolge festlegen. Infolge dessen können die wahren Zeitspannen der zugrundeliegenden Phänomene nur approximiert werden. Die ersten Kalender dieser Art waren darauf ausgerichtet, dass die Monate sich bestmöglich dem Mondzyklus annähern. Diese werden als Lunarkalender bezeichnet. In Gesellschaften mit einer ausgeprägten Agrarkultur ist es jedoch wichtiger, dass das Kalenderjahr synchron zum tropischen Jahr verläuft. Daher wurden Lunisolarkalender entwickelt, die versuchen dem Mond- und Sonnenzyklus gerecht zu werden, was die Komplexität wesentlich steigert. Aufgrund dessen wurde bei Solarkalendern auf eine Kopplung an den Mond gänzlich verzichtet. Durch die Kalenderregeln muss dann nur noch eine gute Näherung des Kalenderjahres an das tropische Jahr gewährleistet sein (RICHARDS 1998 S.92-100). In den verschiedenen Kalendersystemen ergeben sich durch die

---

<sup>5</sup> Durch die Synergie der Bewegung der Erde um die Sonne und der Erdrotation ist diese Zeitspanne etwa vier Minuten länger als ein Siderischer Tag, der einer vollen Erdumdrehung relativ zu einem Fixpunkt entspricht (RICHARDS 1998 S.24-26).

<sup>6</sup> Ein Monat hat eine Länge zwischen 29 und 30 Tagen, ein Jahr zwischen 365 und 366 Tagen. Demnach hat ein Jahr circa 12,4 Monate. Die Angaben beziehen sich dabei auf Mittelwerte, da die tatsächlichen Himmelskörperbewegungen in ihrer Dauer variieren. Im Jahr 2000 betrug die Länge eines tropischen Jahres 365,24219Tage, dieser Mittelwert verringert sich um ca. eine halbe Sekunde/Jahrhundert (RICHARDS 1998 S.34; 91f).

notwendigen Kalenderregeln Monate und Jahre unterschiedlicher Länge, bei Lunisolar-kalendern häufig sogar eine variierende Anzahl an Kalendermonaten in einem Jahr. Um dennoch eine dem Tag übergeordnete Einheit von konstanter Dauer zur Verfügung zu haben, wird eine Woche verwendet, die genau dem Vielfachen eines Tages entspricht. In zahlreichen Kalendersystemen und Kulturkreisen hat eine Woche eine Dauer von sieben Tagen, dies ist jedoch nicht notwendigerweise so (RICHARDS 1998 S.265f).

Heute wird in vielen Ländern der Gregorianische Kalender (ein Solarkalender) verwendet, der auch den Standard im internationalen Rahmen darstellt. Er wurde im Jahr 1582 in den ersten Ländern in Europa eingeführt. In diesen galt bis dahin der Julianische Kalender, ebenfalls ein Solarkalender, der mit einer durchschnittlichen Jahreslänge von 365,25 Tagen entscheidend länger war als das tropische Jahr. Dadurch drohten die Monate nicht mehr den mit ihnen verbundenen Jahreszeiten zu entsprechen. Im Gregorianischen Kalender hat ein gewöhnliches Jahr 365 Tage, die in bestimmter Weise auf 12 Monate aufgeteilt sind. Alle vier Jahre erfolgt dabei ein Schaltjahr, in dem das Jahr einen zusätzlichen Tag am Ende des zweiten Monats zugewiesen bekommt. Ist das Jahr auch durch 100 teilbar, aber nicht durch 400, erfolgt eine Ausnahme von dieser Schaltjahresregel. Daraus ergibt sich eine durchschnittliche Jahreslänge von 365,2425 Tagen, wodurch eine bessere Näherung des tropischen Jahres erreicht wird als im Julianischen Kalender (RICHARDS 1998 S.247-252).

### 2.1.2 Uhrzeitsysteme

Wie gezeigt bildet der Tag die elementare Einheit der Kalendersysteme. Doch vielfach ist es erforderlich, die Zeit weiter zu unterteilen, um genauere Zeitangaben zu ermöglichen.

In der Frühzeit wurde die ‚Tageszeit‘ direkt durch den Sonnenstand abgeschätzt, was durch immer bessere Hilfsmittel perfektioniert wurde. Diese basieren auf dem Schattenwurf eines Stabes, dessen Position direkt von der Phase der Erdrotation abhängt. Dies ermöglicht eine Einteilung der Zeitspanne von Sonnenaufgang bis -untergang, wobei die Länge dieser Zeitspanne über das Jahr variiert. Während der Nacht muss auf andere Beobachtungen zurückgegriffen werden, wie z.B. der von Sternen. In Ägypten waren spätestens seit 1500 v.Chr. Wasseruhren bekannt, die eine Zeitmessung unabhängig von den Himmelskörpern ermöglichten. Hier wurde auch die Stunde als der vierundzwanzigste Teil der Summe aus Tag und Nacht definiert. Im antiken babylonischen Reich wurde eine Stunde weiter in 60 Minuten unterteilt. Davon abgesehen bildeten sich in den verschiedenen Zivilisationen diverse Einteilungen für die Tageszeit (RICHARDS 1998 S.44).

Das abstrakte Prinzip der frühen Wasseruhren war es, eine Schwingung mit konstanter Frequenz zu realisieren und durch die Anzahl der vergangenen Schwingungen die Zeit zu messen. Auch wenn die Genauigkeit dieser Geräte relativ gering war, liegt diese Methode allen Uhren zugrunde. Mit der einsetzenden Neuzeit machte die Entwicklung mechanischer Uhren große Fortschritte, gefolgt von Quarzuhren und den heutigen hochgenauen Atomuhren. Dies machte es möglich, auch die Sekunde als den sechzigsten Teil einer Minute zu messen. Durch die hochgenauen Beobachtungen mit Atomuhren stellte sich dann jedoch

heraus, dass die Tageslänge nicht konstant ist, sondern Variationen unterliegt (RICHARDS 1998 S.62f). Dies hatte sehr weitreichende Folgen, da die gesamte Zeitmessung in Sekunden, Minuten und Stunden erfolgte. Also in Größen, die sich aus der Tageslänge ableiten und entsprechend mit dieser variieren. Um dieses Dilemma zu lösen, wurde die SI-Definition der Sekunde 1967 geändert<sup>7</sup> und ist seitdem unabhängig von der Planetenbewegung der Erde. Auch die Längen aller weiteren Uhrzeit- und Kalendereinheiten sind nachfolgend durch eine konstante Anzahl von SI-Sekunden definiert worden (Iso 2004).

Heute sind verschiedene etablierte Uhrzeitsysteme zu unterscheiden. Die streng gleichförmige Internationale Atomzeit (TAI) wird vom BIPM realisiert und beruht auf SI-Sekundenmessungen durch verschiedene Atomuhren. Im Gegensatz dazu wird die *Universal Time* (UT) nach wie vor auf Basis der Erdrotation, genauer auf dem Bürgerlichen Tag in Greenwich, bestimmt<sup>8</sup>. Die Koordinierte Weltzeit (UTC) unterliegt wie die TAI dem BIPM und ist sowohl gleichförmig wie auch an die Erdrotation gebunden. Sie unterscheidet sich durch eine ganze Anzahl von SI-Sekunden von der TAI, den Schaltsekunden. Diese werden eingefügt, wenn sich die Differenz zwischen UTC und UT 0,9 Sekunden annähert, um die UTC synchron zur Erdrotation zu halten. Die Entscheidung Schaltsekunden anzubringen obliegt dem IERS, wobei sie im Juni oder Dezember an den letzten Tag des Monats angefügt oder weggelassen werden, wenn nötig auch Ende März oder September. Seit der Einführung von UTC 1972 gab es nur positive Schaltsekunden die Ende Juni oder Dezember angehängt wurden. UTC liegt aktuell (Juli 2012) 35 Sekunden hinter der TAI zurück. Da das Auftreten der zukünftigen Schaltsekunden auf Beobachtungen basiert, lässt es sich nicht durch deterministische Algorithmen beschreiben (Usno 2012).

UTC ist zusammen mit dem Gregorianischen Kalender das Zeitbezugssystem der ISO-Norm für die Darstellung von Datum und Uhrzeit. Die gesetzlichen Uhrzeitsysteme leiten sich in den meisten Ländern direkt aus der UTC ab. Die lokalen Zeiten weichen dabei um eine bestimmte Dauer von der reinen UTC Zeit ab. Diese Zeitverschiebungen ergeben sich aus den Zeitzonen, in die die Erde durch internationale Konventionen eingeteilt ist. Generell orientieren sich diese Zonen an den Längengraden, um der Verschiebung der lokalen Sonnenscheindauer gegenüber ebendieser am Greenwich-Meridian gerecht zu werden. Die Zonengrenzen verlaufen über der Landoberfläche jedoch nicht streng entlang der Meridiane, sondern orientieren sich hier untergeordnet an den Staatsgrenzen, um die Anzahl der Zeitzonen innerhalb eines Staates zu reduzieren. Die Differenz zur Greenwich-Zone beträgt maximal 12 Stunden, dies ergibt sich aus der Dauer einer halben Erdumdrehung. Entlang des

---

<sup>7</sup> „Eine Sekunde ist die Zeitdauer von 9.192.631.770 Perioden der Strahlung, die dem Übergang zwischen den beiden Hyperfein-Niveaus des Grundzustandes des Cäsium-133-Atoms entspricht.“ (BIPM 2006 S.112).

<sup>8</sup> Sie wird vom IERS bestimmt und als Differenz zur UTC und TAI angegeben. Effekte der Polbewegung werden dabei korrigiert.

180. Längengrades verläuft die Datumsgrenze, auf den beiden Seiten gilt bei gleicher Ortszeit ein anderes Kalenderdatum (ISO 2004 S.12, 16; RICHARDS 1998 S.61f).

## 2.2 Bedeutung der Zeit in GIS und Geodatenbanken

In den verwendeten Quellen sind diverse Fragestellungen zu finden, die die Ergänzung einer Geodatenbank um Methoden der Zeitbehandlung erforderlich machen. Diese reichen von auf die temporale Dimension beschränkten Abfragen (Wann) bis zu wesentlich komplexeren, die eine Kombination der zeitlichen Daten mit der Geometrie, Topologie und Thematik erforderlich machen (Was-Wann-Wo-Wie-Warum)<sup>9</sup>. Analysen die eine Zeitkomponente aufweisen bzw. erfordern, lassen sich dabei im Querschnitt der Anwendungsgebiete finden (BILL 2010 S.193).

Grob lassen sich diese Anwendungsfälle in zwei abstrakte Kategorien einteilen. Dies ist zum einen das Interesse an den bereits eingangs dargelegten zeitlichen Veränderungen von Objekten (z.B. Landnutzung), also ihrer Dynamik.

Die zweite Kategorie bilden die bewegten Objekte (z.B. Fahrzeuge), die ihre Position im Raum mit der Zeit verändern und in ihren sonstigen Charakteristika, insbesondere Form und Größe, statisch sind. Beide Fälle werden häufig gesondert betrachtet, da diese grundlegenden Unterschiede auch eine Differenzierung auf der konzeptuellen Ebene zweckmäßig machen (PEUQUET 2001 S.17, WORBOYS UND DUCKHAM 2004 S.371).

Ursächlich für die Veränderungen, welcher Geometrie und Attribute der modellierten Geoobjekte dabei ausgesetzt sind, ist in erster Linie anthropogenes Handeln, aber auch die Dynamik des Systems Erde. Relativistische Effekte kommen nur bei extraterrestrischen Betrachtungen zum tragen.<sup>10</sup>

Neben der Zeit zu der ein Objekt in der Wirklichkeit gültig war (*valid time*), ist oft auch die Zeit zu der das Objekt in die Datenbank eingetragen oder geändert wurde zu modellieren (*transaction time*). Darüber hinaus sind in bestimmten Anwendungsfällen noch weitere Zeitinformationen von Interesse, z.B. die Zeit zu der ein Objekt beobachtet wurde (*user defined time*). Eine Analyse der Veränderungen der zugrunde liegenden Phänomene basiert auf den Angaben der ersten Art. Die Datenbankzeit ist wichtig, um es zu ermöglichen eine vorherige Version des Datenbestandes wiederherzustellen. Ein bitemporales System ist in der Lage beide Ebenen zu verwalten (BILL 2010 S.197, WORBOYS UND DUCKHAM 2004 S.367).

Die Anforderungen an eine Geodatenbank hinsichtlich der Zeit sind also gleichermaßen vielseitig wie auch weitreichend.

---

<sup>9</sup> WORBOYS UND DUCKHAM (2004) S.360, BILL (2010) S.198, PEUQUET (2001) S.17. Das auch weiterhin Abfragen ohne Einbeziehung der Zeit möglich sein sollten, bleibt hiervon unberührt.

<sup>10</sup> Aufgrund großer Relativgeschwindigkeiten und Unterschiede im Gravitationspotential könnte die Zeit nicht mehr vereinfacht als absolut angenommen werden. Hier wäre bei der Modellierung immer zu berücksichtigen, für welchen Beobachter die Angabe gültig ist bzw. sein soll.

## 2.3 Begriffe und Konzepte zur Dynamik

In diesem Abschnitt werden die wesentlichen Begriffe erläutert, die sich im Zusammenhang mit der Modellierung der Dynamik als Konsens herausgebildet haben. Abschließend wird auf die Konzepte der temporalen Datenbanken eingegangen, die seit langem Einfluss auf die Zeitbehandlung in GIS nehmen und daher nicht unerwähnt bleiben können.

### 2.3.1 Raum-zeitliche Modelle

Die in den meisten GIS implementierten Modelle sind in erster Linie für eine adäquate Behandlung von geographischen Informationen konzipiert. Es wurden verschiedene Ansätze entwickelt, um die temporale Dimension in diese Modelle zu integrieren und eine Erfassung der Dynamik möglich zu machen. Die Zeit wird dabei gewöhnlich separat von den Raumdimensionen konzipiert und nur semantisch mit ihnen in Beziehung gesetzt. Es erfolgt keine echte 4D-Modellierung.

Einen ersten Ansatz, um die Überwindung der statischen Modelle zu erreichen, stellen die Momentaufnahmen (*Snapshots*) dar. Diese sind mit Landkarten unterschiedlicher Aktualität zu vergleichen. Hierbei wird ein Gebiet zu verschiedenen *valid time* Momenten in gleicher Weise abstrahiert und gespeichert. Die Dynamik wird dabei nur implizit und insofern erfasst, dass sie aus einem Vergleich der Datensätze extrahierbar sein muss. So würde beispielsweise durch jährlich erfolgende Momentaufnahmen die Dynamik aufgrund der Jahreszeiten nicht erfasst werden. Ein weiterer Nachteil ist die notwendige komplette Erfassung eines Gebiets für jede Momentaufnahme und die daraus folgende hohe Redundanz sowie unnötiger Speicherplatzverbrauch.

Eine diese Probleme lösende explizite Beschreibung der Dynamik bietet sich insbesondere in der objektorientierten Welt an. Dabei können die zeitlichen Variationen von Geometrie und Thematik durch verschiedene Versionen eines Geoobjekts modelliert werden. Durch Objektbeziehungen lassen sich auch Vorgänge wie Objektvereinigungen oder -teilungen abbilden (*Object lifelines*). Dabei werden die Ursachen der Variationen allerdings nur indirekt als Veränderung der Geoobjekte erfasst. Dies ist für Anwendungen, bei denen eine Analyse dieser Ursachen im Fokus steht, nicht weitreichend genug. Ein Beispiel könnte hier die Analyse von Wetterereignissen sein. Solche Ereignisse, die keine unmittelbaren Auswirkungen auf die Attribute der Geoobjekte haben, bleiben von vornherein unberücksichtigt.

Dies macht es erforderlich die Ereignisse selbst als Objekte aufzufassen und sie mit den bestehenden Geoobjekten in Beziehung zu setzen. WORBOYS UND DUCKHAM (2004) unterscheiden dabei drei Formen von Ereignissen. Sie alle können punktuell sein oder eine Dauer haben. Dies sind zum einen die *events*, welche diskrete, abgrenzbare Entitäten darstellen und zum anderen Prozesse (*processes*), welche nicht zählbar sind. Aktionen (*actions*) sind solche Ereignisse, die von einem anderen Objekt ausgelöst wurden. Die Zeitabschnitte oder Zeitpunkte in denen ein Objekt in all seinen Charakteristika statisch ist, werden als Zustände (*state*) bezeichnet. Sind die Ereignisse als Objekte modelliert, können

auch Beziehungen zwischen ihnen, zu beeinflussten Geobjekten oder zu ihren Ursachen beschrieben werden. Hier kann demnach eine sehr umfassende Erfassung der Dynamik erfolgen (KEMP 2008 S.442-444, WORBOYS UND DUCKHAM 2004 S.361-367).

### 2.3.2 Zeitbezugssysteme für GIS

Ähnlich wie für die Raumdimensionen sehr unterschiedliche Konzepte existieren, sind auch für die Modellierung der Zeitdimension unterschiedliche Herangehensweisen entwickelt worden. Im Folgenden sollen die prägnanten Merkmale und Differenzen solcher Modelle aufgezeigt werden.

Die Zeit wird auf konzeptueller Ebene als zusätzliche Dimension mit Unterschieden, aber auch vielen Analogien, zu den räumlichen Dimensionen aufgefasst. Ein wesentlicher Unterschied ist, dass die Zeit eine feste Ordnung hat. So lässt sich für jeden Zeitpunkt eine Vergangenheit (Davor) und Zukunft (Danach) bestimmen, eine Bewegung erfolgt dabei immer gleichgerichtet und auch gleichmäßig<sup>11</sup> (WORBOYS UND DUCKHAM 2004 S.372). Wie für den Raum werden auch für die Zeit die Begriffe der Geometrie und Topologie definiert. Dabei wird unter der Geometrie die Position von Zeitpunkten oder Zeitabschnitten in einem geeigneten Zeitbezugssystem verstanden. Die Zeitpunkte unterscheiden sich von den Zeitabschnitten dadurch, dass man ihnen keine Ausdehnung in der Zeit zuschreibt. Die Topologie beschreibt die Verbindungen oder Abfolge verschiedener Phänomene in der temporalen Dimension (Iso 2002a).

Einen Unterschied der Modelle macht die Verwendung von dichter oder diskreter Zeit aus, wobei zum Verständnis die erstgenannte der Menge der reellen oder rationalen<sup>12</sup> Zahlen und die zweite der Menge der natürlichen Zahlen gleichgesetzt werden kann. Demnach liegt bei der ersten Variante zwischen zwei Zeitpunkten immer ein weiterer, bei der diskreten Zeit ist dies nicht zutreffend. Mit der Granularität eines diskreten Zeitmodells wird die temporale Auflösung, also die Dauer zwischen zwei Zeitpunkten bezeichnet. Unter einem Chronon oder *tick* wird ein Zeitpunkt in diesem System verstanden. Dies kann analog zu einem Pixel bei einer diskreten Einteilung des Raumes gesehen werden. Daher kann es sein, dass zwei eigentlich nicht gleichzeitig stattfindende Ereignisse aufgrund einer groben Granularität in das gleiche Chronon abgebildet werden. Im Modell würden sie dann als gleichzeitige Ereignisse gelten. Da die zeitliche Auflösung immer begrenzt ist, sei es durch die implementierungsbedingte Diskretisierung der Zeitpunkte oder die Datengrundlage, wird auch bei dichten Zeitmodellen von Granularität und Chronon gesprochen (ELMASRI UND NAVATHE 2002 S.799f; FURIA et al. 2010 S.6:6-8; WORBOYS UND DUCKHAM 2004 S.372).

---

<sup>11</sup> Das dies physikalisch nicht exakt ist, ist für die betrachteten Phänomene nicht relevant.

<sup>12</sup> Aufgrund der implementierungsbedingt begrenzten Genauigkeit der Gleitpunktzahlen wird von einer Differenzierung zwischen kontinuierlicher ( $\mathbb{R}$ ) und dichter ( $\mathbb{Q}$ ) Zeit abgesehen (FURIA et al. 2010 S.6:6, GRUMM UND SOMMER 2011 S.29).

Bei den bisherigen Betrachtungen wurde immer von einer expliziten Modellierung der Geometrie der Zeit ausgegangen, auf einer Ordinalskala ist jedoch auch eine bloße Beschreibung der Topologie möglich. FURIA et al. verwenden hier die Begriffe der Quantität und Qualität, bei einem quantitativen System ist zwischen jedem Paar von Zeitpunkten eine Distanz definiert (FURIA et al. 2010 S.6:8f). Eine rein qualitative Modellierung kann dann sinnvoll sein, wenn nur die Topologie, aber nicht die exakten Zeitpunkte verschiedener Ereignisse bekannt sind, wie z.B. in der Geologie. Aber auch in Anwendungen die sich auf die topologischen Aspekte konzentrieren, kann deren explizite Modellierung zielführend sein.

Neben dem bisherigen streng linearen Verständnis der Zeit sind auch sogenannte verzweigte oder zyklische Ansätze denkbar. Die Zyklischen heben die strenge Ordnung der Zeit auf, indem sie diese ringförmig modellieren. Dementsprechend sind sie besonders für periodische Analysen geeignet, z.B. zur Ermittlung der Auslastung des Nahverkehrs in einem zyklischen Zeitsystem der Wochentage. In einem verzweigten Modell sind mehrere zukünftige oder vergangene Zeitverläufe möglich, was sich für Simulationen anbietet (FURIA et al. 2010 S.6:9f, WORBOYS UND DUCKHAM 2004 S.372f).

### 2.3.3 Temporale Datenbanken

In Abschnitt 2.2 wurde das Erfordernis der Zeitbehandlung in GIS-Anwendungen dargelegt. Darüber hinaus existiert eine solche Anforderung auch in vielen Datenbankanwendungen die den Raumbezug nicht zwingend berücksichtigen, wie z.B. im Gesundheits- oder Versicherungswesen. Auch hier sind zahlreiche Abfragen mit einem Zeitaspekt denkbar, die eine besondere Zeitsemantik erfordern. Um dem gerecht zu werden, wurden temporale Datenbankkonzepte entwickelt (ELMASRI UND NAVATHE 2002 S.799, VOSSEN 2008 S.751).

Da eine (Geo-)Datenbank heute im Allgemeinen den Kern eines GIS bildet<sup>13</sup>, sind die Entwicklungen in DBMS und GIS eng verzahnt und dementsprechend lassen sich die Konzepte der temporalen Datenbanken in der GIS-Welt wiederfinden. Nach PEUQUET besteht hier seit gut 15 Jahren ein intensiver Austausch (PEUQUET 2001 S.15). So ist die Differenzierung zwischen *valid*, *transaction* und *user defined time* auch ein wesentliches Merkmal von temporalen Konzepten in DBMS. Es wird zwischen zwei Ansätzen unterschieden, wie diese Zeitdimensionen in temporalen Datenbanken abgebildet werden.

Die Tupel-Versionierung (*Tuple timestamping*) dient der Zeiterfassung in relationalen Datenbanken. Hierzu wird eine Relation um Attribute erweitert, welche Start- und Endzeitpunkt für die entsprechende Zeitdimension mit einem geeigneten Datentyp verwalten. Muss ein Datensatz geändert werden, um eine Änderung in der Wirklichkeit (*valid*) zu erfassen oder einen Fehler im Datensatz zu korrigieren (*transaction*), wird der Datensatz nicht ersetzt, sondern mit entsprechend modifizierten Werten neu angelegt. Als Attribute können auch

---

<sup>13</sup> WORBOYS UND DUCKHAM (2004) S.2f.

Intervalltypen verwendet werden, die Start- und Endzeitpunkt in einem Datentyp einschließen. Die Tupel-Versionierung entspricht dem *Snapshot*-Modell (Abschnitt 2.3.1) und hat denselben Nachteil, der zum Teil hohen Redundanz der Datensätze.

In Objektdatenbanken oder objektrelationalen Systemen kann hingegen eine Attributversionierung erfolgen. Dabei werden für jedes Attribut die erforderlichen Zeitinformationen gesondert verwaltet. Die verschiedenen Versionen eines Attributes könnten dann z.B. durch eine Liste mit dem Objekt verknüpft werden. Es würden dann nur noch die Daten erfasst werden, die neue Informationen enthalten (ELMASRI UND NAVATHE 2002 S.799-809, WORBOYS UND DUCKHAM 2004 S.369f).

## 2.4 Zeitliches Schema der ISO

Das zeitliche Schema der ISO<sup>14</sup> ist ein konzeptuelles Schema zur Beschreibung der zeitlichen Gültigkeit (entsprechend der *valid time*) von Geoinformationen. Dies umfasst Objektattribute, ihre Methoden und Beziehungen sowie Metadatenelemente (Iso 2005a S.6). Dabei wird dem ISO Referenzmodell folgend eine differenzierte Modellierung von Raum- und Zeitbezug vorgesehen:

*“Reference systems are subtyped into spatial reference systems and temporal reference systems.”* (Iso 2005b S.25).

Die Norm definiert hierfür zwei zu implementierende Objektpakete. Dies sind zum einen die temporalen Objekte<sup>15</sup> zur Beschreibung der zeitlichen Geometrie und Topologie. Weiter werden konzeptuelle Modelle für geeignete Bezugssysteme<sup>16</sup> vorgestellt, die der Festlegung der zeitlichen Position der temporalen Objekte dienen. Darüber hinaus wird beschrieben, wie das Schema in Verbindung mit Geoinformationen verwendet werden sollte<sup>17</sup> (Iso 2005a S.12). Diese Inhalte sollen hier kurz vorgestellt werden.

### 2.4.1 Temporale Objekte

Die temporalen Objekte gliedern sich grob in primitive (*TM\_Primitive*) und komplexe (*TM\_Complex*) Typen. Die Primitive umfassen dabei geometrische und topologische Primitive, die Komplexe hingegen nur topologische Komplexe.

#### Geometrische Primitive

Die Geometrie (*TM\_GeometricPrimitive*) wird durch zwei verschiedene Klassen von geometrischen Primitiven modelliert. Zum einen sind dies Zeitpunkte (*TM\_Instant*), zum anderen Zeitperioden (*TM\_Period*). Zeitpunkte haben dabei nur ein Attribut, welches die

---

<sup>14</sup> Die 2006 veröffentlichte - das zeitliche Schema betreffende - Berichtigung wurde vom Autor zur Kenntnis genommen (Iso 2006).

<sup>15</sup> *Temporal Objects* - Iso 2005a S.12-21

<sup>16</sup> *Temporal Reference System* – a.a.O. S.21-29

<sup>17</sup> *Temporal aspects of geographic information components* – a.a.O. S.29-35



Position in der Zeit durch ein *TM\_Position*-Objekt festlegt. Diese Klasse wird in Abschnitt 2.4.2 behandelt werden. Eine Periode ist aus zwei Zeitpunkten zu bilden, die Start- und Endpunkt der Periode festlegen. Bedingung ist, dass der Beginn der Periode zeitlich vor dem Ende liegt.

Die geometrischen Primitive implementieren die Schnittstellen *TM\_Order* und *TM\_Seperation* und müssen daher Methoden anbieten, die es ermöglichen die relative Position (*relativePosition*) oder Distanz (*distance*) zwischen zwei geometrischen Primitiven abzufragen. Eine weitere Objektmethode dient der Bestimmung der Länge (*length*) des Primitives.

Die relative Position beschreibt die Lage eines geometrischen Primitives rein qualitativ zu einem anderen. Für zwei Zeitpunkte ergeben sich somit drei mögliche Konstellationen (Davor - Gleichzeitig - Danach). Im komplexesten Fall, dem Vergleich zweier Zeitspannen, sind jedoch 13 zeitliche Beziehungen zu berücksichtigen (Siehe Abbildung 2-1). Die Rückgabewerte der Funktion sind durch den Aufzählungstyp *TM\_RelativePosition* definiert.

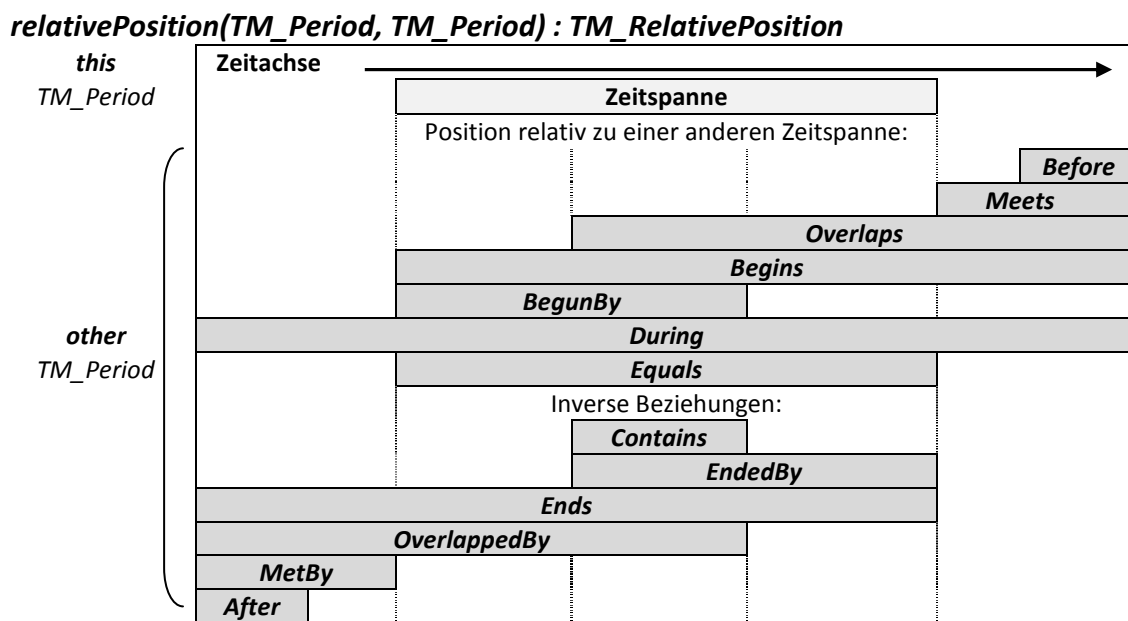


Abbildung 2-1: Die 13 möglichen zeitliche Beziehungen zwischen zwei Zeitspannen nach der ISO19108.

Mit den Funktionen *length* und *distance* lassen sich quantitative Informationen über die modellierten Primitive abfragen. Die Länge gibt die Distanz zwischen den Start- und Endzeitpunkten einer Zeitspanne auf der Zeitachse an und beträgt für *TM\_Instant*-Objekte per Definition Null. Mit *distance* werden analog die Distanzen zwischen zwei Zeitprimitiven ermittelt, wobei der geringste Abstand zwischen ihnen anzugeben ist. Der Rückgabewert

dieser Funktionen ist vom Typ *TM\_Duration*, mit den abgeleiteten Klassen *TM\_PeriodDuration* und *TM\_IntervallLength*, die als Datentypen dienen. Die erstgenannte Struktur gibt die Dauer in Kalender- und Uhrzeiteinheiten an, wie es in der ISO8601<sup>18</sup> festgelegt ist. Der zweite Typ basiert auf dem Intervalltyp der *General-Purpose Datatypes* der ISO (Iso 2007) und dient der Beschreibung einer Dauer als dem Mehrfachen einer einzigen Zeiteinheit (Iso 2005a S.13-18).

### Topologische Primitive und Komplexe

Durch die topologischen Primitive (*TM\_TopologicalPrimitive*) wird die zeitliche Verbindung der Objekte modelliert. Die Basis hierfür ist der nulldimensionale Knoten (*TM\_Node*), aus dem sich die eindimensionalen Kanten (*TM\_Edge*) bilden. Dies erfolgt analog zu der Modellierung der Zeitperioden aus Zeitpunkten in der Geometrie, durch Angabe von einem Start- und Endknoten. Eine Knoteninstanz kann dabei für verschiedene Kanten Start- oder Endknoten sein, wodurch es möglich wird auch nicht lineare Zeitverläufe (Siehe 2.3.2) topologisch zu beschreiben. Sowohl Knoten als auch Kanten können mit den ihnen entsprechenden Geometrieinstanzen über eine Objektbeziehung verbunden werden. Beide Primitive sind immer in mindestens einem topologischen Komplex (*TM\_TopologicalComplex*) enthalten, der alle zeitlich untereinander verknüpften Objekte enthält.

Auch für die Topologie ist durch das *TM\_Order*-Interface eine Implementierung der Funktion *relativePosition* vorgegeben. Der Rückgabewert richtet sich dabei nach der Ordnung, welche die Elemente in ihrem gemeinsamen topologischen Komplex einnehmen. Sollten sie keinem gemeinsamen Komplex angehören, gibt die Funktion eine Exception zurück (a.a.O. S.18-21). Tabelle 2-1 enthält die Funktionen der geometrischen Primitive.

Tabelle 2-1: Funktionen der *TM\_Primitive* Objekte.

Funktion	Eingabeparameter	Rückgabewert
<i>relativePosition</i>	( <i>TM_GeometricPrimitive</i> , <i>TM_GeometricPrimitive</i> )	: <i>TM_RelativePosition</i>
<i>relativePosition</i>	( <i>TM_TopologicalPrimitive</i> , <i>TM_TopologicalPrimitive</i> )	: <i>TM_RelativePosition</i>
<i>distance</i>	( <i>TM_GeometricPrimitive</i> , <i>TM_GeometricPrimitive</i> )	: <i>TM_Duration</i>
<i>length</i>	( <i>TM_GeometricPrimitive</i> )	: <i>TM_Duration</i>

### 2.4.2 Temporale Referenzsysteme und Zeitpositionen

Die Zuweisung einer absoluten Position in der Zeit zu einem geometrischen Primitiv erfolgt durch ein *TM\_Position*-Objekt, welches sich auf eines der Bezugssysteme bezieht, die durch die Objekte der *TM\_ReferenceSystem*-Klasse beschrieben werden.

<sup>18</sup> *Representation of Dates and Times* (Iso 2004).

## Referenzsysteme

Als zeitliches Bezugssystem für geographische Informationen sollte vorzugsweise der Gregorianische Kalender mit der UTC verwendet werden. Wie Zeitangaben in diesem System erfolgen wird in der ISO8601 festgelegt. Die ISO19108 definiert Konzepte für weitere mögliche Referenzsysteme mit der Klasse *TM\_ReferenceSystem*. Ein solches wäre dann durch einen Verweis oder direkt in den Metadaten hinreichend zu beschreiben. Wird in einer Anwendung mehr als ein Referenzsystem verwendet, ist zu jeder Zeitinformation das entsprechende Bezugssystem anzugeben. Dabei werden drei mögliche Typen von Referenzsystemen unterschieden. Dies sind die Kalendersysteme (*TM\_Calendar*), zeitliche Koordinatensysteme (*TM\_CoordinateSystem*) und Ordinalskalen (*TM\_OrdinalReferenceSystem*) (a.a.O. S.21f).

Eine *TM\_Calendar*-Instanz ist mit mindestens einem *TM\_CalendarEra*-Objekt verknüpft. In dieser Klasse werden Informationen über das verwendete Kalendersystem gespeichert, wie der Name des Systems, die Zeitspanne zu der es gültig war und der Referenzzeitpunkt auf den sich die Daten beziehen. Um Transformationen zu ermöglichen, ist der Referenzzeitpunkt dabei sowohl durch ein Kalenderdatum innerhalb des Kalenders selbst, als auch durch den Julianischen Tag<sup>19</sup> anzugeben. Das eine Kalenderinstanz mehr als ein *TM\_CalendarEra*-Objekt haben kann, ist zum Beispiel erforderlich, wenn ein System modelliert werden soll, indem Herrschaftsperioden verschiedener Dynastien mit jeweiligen Referenzzeitpunkten unterschieden werden. Sollen mit einem *TM\_Calendar* Zeitangaben mit einer Auflösung von mehr als einem Tag erfolgen, kann ihm hierzu ein *TM\_Clock*-Objekt zugeordnet werden. Dies gibt den Referenzzeitpunkt der Zeitzählung an, z.B. der Sonnenaufgang. Die Angabe erfolgt dabei sowohl im jeweiligen Uhrzeitsystem wie auch in der Koordinierten Weltzeit (UTC), um eine Basis für Uhrzeittransformationen zu schaffen. Ein *TM\_Calendar* Objekt bietet immer die Methoden *dateTrans* und *julTrans* an, welche für eine gegebene Position in dem Kalendersystem dieselbe Position als Julianischen Tag zurückgeben oder die inverse Transformation durchführen. Demnach kann ein Kalendersystem über das Julianische Datum in jedes andere überführt werden. Durch die Verwendung eines *TM\_Clock*-Objekts stehen auch die Methoden *clkTrans* und *utcTrans* zur Verfügung, die eine angegebene Uhrzeit in UTC überführt oder vice versa (a.a.O. S.22-24).

Ein *TM\_CoordinateSystem* zeichnet sich dadurch aus, dass Positionen auf einer kontinuierlichen Zeitskala durch das Vielfache einer Basiseinheit angegeben werden. Da ein Zeitpunkt dann vollständig durch eine Zahl beschrieben werden kann, sind Berechnungen von Zeitdistanzen und weitere Operationen hier wesentlich primitiver als in den Kalender-

---

<sup>19</sup> Bei Verwendung des Julianischen Tages (auch Julianisches Datum) wird eine Zeitposition durch die Zahl der vollen Tage die seit dem Beginn des Jahres 4713v.Chr. (im Julianischen Kalender) vergangen sind, angegeben. Die Maßeinheit der Werte ist demnach ein Tag. Auflösungen im Sub-Tages Bereich können durch die Angabe von Nachkommastellen erreicht werden, der Tageswechsel erfolgt dabei zur Mittagszeit (MEES 1991 S.59).

systemen. Ein solches Koordinatensystem hat einen Ursprung, der in dem Gregorianischem Kalender und UTC angegeben wird, sowie den Namen der verwendeten Einheit. Die Klassenmethoden dienen auch hier der Transformation, in dem Fall zwischen dem Gregorianischen Kalender mit UTC und dem temporalen Koordinatensystem. Das Julianische Datum ist ein zeitliches Koordinatensystem, welchem aufgrund der expliziten Einbindung in die *TM\_Calendar*-Transformationen eine besondere Bedeutung in der ISO19108 zukommt (a.a.O. S.24f).

Durch die Klasse *TM\_OrdinalReferenceSystem* lassen sich Zeitsysteme beschreiben, bei denen die Topologie der Zeit im Vordergrund steht. Sie sind eine Komposition aus *TM\_OrdinalEra*-Objekten. Dies sind Zeitspannen, die die Elemente der Ordinalskala bilden. Zu ihren Attributen zählt ein Name sowie der Anfangs- und Endzeitpunkt, angegeben im Gregorianischem Kalender und UTC. Die *TM\_OrdinalEra*-Objekte können durch weitere Kompositionsbeziehungen Hierarchien bilden. So könnte die griechische Antike eine Instanz sein, der das archaische, das klassische und das hellenistische Zeitalter untergeordnet sind (a.a.O. S.25f).

### Zeitpositionen

Durch die *TM\_Position*-Objekte erfolgt die Benennung eines eindeutigen Zeitpunktes innerhalb eines Referenzsystems. Die Zeitzuweisung innerhalb des Gregorianischen Kalenders mit UTC erfolgt dabei durch eine Zeichenkette gemäß der ISO8601. Für jedes andere Bezugssystem wird die Klasse *TM\_TemporalPosition* verwendet. Da die Art und Weise der Zeitpunktbezeichnung abhängig vom verwendeten Bezugssystem ist, existieren für die oben vorgestellten *TM\_ReferenceSystem*-Objekte entsprechende *TM\_TemporalPosition*-Unterklassen. Diese sind über eine Objektbeziehung mit ihrem jeweiligen Bezugssystem verknüpft.

Das einzige Attribut der Klasse *TM\_TemporalPosition* ist vom Typ *TM\_IndeterminateValue*. Wird aus der Klasse selbst eine Instanz gebildet, kann hiermit ausgedrückt werden, dass der Zeitpunkt nicht bekannt ist. Wird eine der Unterklassen verwendet, kann angegeben werden, dass der Zeitpunkt nicht genau bekannt ist, aber vor oder nach dem angegebenen liegt. Eine dritte Variante ermöglicht es einen Zeitpunkt als Gegenwart zu definieren, so dass er bei jedem Zugriff dem aktuellen Zeitpunkt entspricht, angegeben im jeweiligen System.

Bei der Verwendung eines *TM\_IndeterminateValue* ist zu beachten, dass ein Funktionsaufruf von *relativePosition*, *length* oder *distance* dann eine Exception auslöst.

Die beiden *TM\_TemporalPosition*-Unterklassen *TM\_CalDate* und *TM\_ClockTime* dienen der Bezeichnung eines Zeitpunktes innerhalb eines Kalender- oder Uhrzeitsystems. Für eine Kalenderangabe wird auf den Namen der zugrunde liegenden *TM\_CalendarEra* verwiesen und durch eine Sequenz von Zahlen die Position festgelegt. Die Werte beziehen sich dabei auf die Jahre, Monate und Tage des Kalenders. Das Format sollte sich dabei an dem der ISO8601 orientieren und dieses auf andere Kalendersysteme übertragen. Ein Uhrzeitsystem wird gewöhnlich nur in Verbindung mit einem Kalender verwendet. Es wäre jedoch auch

möglich, die Zeitposition eines täglich wiederkehrenden Ereignisses nur durch eine Uhrzeit anzugeben. Das einzige Attribut von *TM\_ClockTime* ist analog zu der Zahlen-Sequenz von *TM\_CalDate* und gibt Stunden, Minuten und Sekunden an. Vorzugsweise auch hier im ISO8601-Format. Die Klasse *TM\_DateAndTime* vereinigt *TM\_CalDate* und *TM\_ClockTime* um Zeitangaben innerhalb eines Kalender- und Uhrzeitsystems mit einer Instanz zu ermöglichen. Durch nur einen Zahlenwert kann die Position relativ zu einem *TM\_CoordinateSystem* durch eine *TM\_Coordinate*-Instanz vollständig beschrieben werden. Das Julianische Datum wird als *TM\_Coordinate*-Kindklasse aufgeführt.

Durch die Klasse *TM\_OrdinalPosition* erfolgt die Positionsbeschreibung innerhalb eines *TM\_OrdinalReferenceSystem*, indem sie in eine Beziehung zu einem *TM\_OrdinalEra*-Objekt gesetzt wird. Das *TM\_OrdinalEra*-Objekt muss dabei auf der untersten Ebene der Ordinalskala angesiedelt sein (a.a.O. S.26-29).

### 2.4.3 Temporale Objekte und Geoobjekte

Der letzte Abschnitt der ISO19108 definiert, wie die eingeführten Elemente in einem Anwendungsschema verwendet werden sollten, um die zeitlichen Eigenschaften von Geoobjekten zu beschreiben.

#### Temporale Objektattribute

Um die zeitlichen Charakteristika als Objektattribut abzubilden, wird der *GF\_TemporalAttributeType* verwendet. Dieser ist eine Unterklasse des *GF\_AttributeType* der ISO19109<sup>20</sup>. Dabei existieren eigene Kindklassen für *events* und *states* (Siehe Abschnitt 2.3.1), wobei unter *events* in dieser Norm nur punktuelle Ereignisse fallen und *states* immer eine Dauer haben müssen. Der Wert einer *TM\_EventType*-Instanz ist entweder ein *TM\_Instant*, *TM\_Node* oder direkt ein *TM\_TemporalPosition*-Objekt. Entsprechend kann eine *TM\_StateType*-Instanz durch ein *TM\_Period*- oder *TM\_Edge*-Objekt realisiert werden. Sollen mehr Informationen über die *states* gespeichert werden als auf diesem Wege möglich, kann *TM\_TemporalPosition* auch als Unterklasse von *TM\_Period* oder *TM\_Edge* gebildet und um die nötigen Attribute erweitert werden. Über das von der *GF\_AttributeType*-Klasse geerbte Attribut *cardinality* kann festgelegt werden, wie viele *TM\_TemporalPosition*-Instanzen einem einzigen Geoobjekt maximal zugewiesen werden dürfen. Durch eine Kardinalität größer eins können demnach entweder verschiedene *states* oder *events* einem Geoobjekt zugeordnet werden. Die ISO19108 bietet auch eine Möglichkeit zur Modellierung von regelmäßig wiederkehrenden Ereignissen bzw. Zuständen. Ein hierfür geeignetes *TM\_TemporalAttributeType*-Objekt wird dazu als gesonderte Klasse gebildet und enthält mindestens zwei Attribute. Das erste dient dabei der Angabe eines spezifizierten Zeitpunktes bzw. einer Zeitperiode der Abfolge und repräsentiert ein Ereignis

---

<sup>20</sup> *Rules for application schema* (Iso 2005).

oder einen Zustand. Das zweite Attribut ist vom Typ *TM\_Duration* und gibt die Zeitspanne zwischen den *events* bzw. *states* an (a.a.O. S.29-31).

### Temporale Objektoperationen

Temporale Operationen von Geoobjekten sind Funktionen, denen ein Zeitparameter übergeben wird und deren Rückgabewert von diesem abhängt. Dadurch lässt sich beschreiben, wie sich die Charakteristika des Objekts mit der Zeit verändern (a.a.O. S.31).

### Temporale Objektbeziehungen

Zeitliche Objektbeziehungen werden in der ISO19108 in zwei grundlegend verschiedene Arten eingeteilt. Dies sind zum einen solche, die von der *valid time* der Objekte existenzabhängig sind und zum anderen solche, bei denen die Zeit nur eine untergeordnete Rolle spielt. Der Klasse *GF\_TemporalRelationship* sind nur die der ersten Art angehörig.

Der zweiten Variante könnte zum Beispiel eine Angabe über die zeitliche Gültigkeit einer Objektbeziehung entsprechen, die sich aus der thematischen Dimension ableitet. In diesen Fällen wird der Zeitaspekt der Beziehung durch ein Attribut vom Typ *TM\_StateType* oder *TM\_EventType* berücksichtigt.

Die Klasse *GF\_TemporalRelationship* ist in der ISO19109 als Kindklasse der generellen Objektbeziehungen (*GF\_AssociationType*) festgelegt. Die möglichen zeitlichen Objektbeziehungen beruhen auf den Rückgabewerten der *relativePosition*-Funktion und sollten vorzugsweise direkt aus dieser abgeleitet werden. Neben den einfachen Beziehungen (*TM\_SimpleRelationship*), die sich allein aus der temporalen Topologie der Objekte ergeben, schließen die *TM\_FeatureSuccessions* auch die räumliche Dimension mit ein. Solche Beziehungen bestehen zwischen zwei Mengen von Geoobjekten, bei denen die eine durch die andere hinsichtlich des Raumbezuges ersetzt wurde. Hier werden drei Fälle unterschieden, aus denen sich auch komplexere Fortführungsbeziehungen bilden lassen. Dies ist zum einen die *TM\_FeatureSubstitution*, die zwischen zwei Objektinstanzen besteht, weil eine die andere vollständig ersetzt hat. Geoobjekte stehen in einer *TM\_FeatureDivision* Beziehung, wenn ein Objekt durch zwei oder mehrere andere ersetzt wurde. Der inverse Fall ist die *TM\_FeatureFusion*, bei dem mehrere Objekte ein neues Objekt gebildet haben. Durch eine geeignete Verknüpfung der Funktion *relativePosition* und den räumlichen Operationen der ISO19107 lassen sich die *TM\_FeatureSuccessions*-Beziehungen ableiten (a.a.O. S.32-34).

### Temporale Metadaten

Temporale Metadaten können als zusätzliche Metadatenelemente im Sinne der ISO19115 definiert werden. Die Zuweisung erfolgt dabei analog zu den temporalen Objektattributen, mit dem Unterschied, dass hier Information über die Daten und nicht über die Objekte beschrieben werden. Eine solche Information kann auf ein einzelnes Objektattribut abzielen oder auch ganze Datensätze betreffen (a.a.O. S.34f).

## 2.5 Zeitbehandlung in aktueller GIS-Software

Um einen Eindruck der Zeitbehandlung in aktuellen Systemen der Geoinformatik zu vermitteln, werden hier temporale Funktionen ausgewählter Anwendungen vorgestellt. Angesichts der Vielzahl der verschiedenen Produkte kann eine solche Übersicht in dieser Arbeit nicht umfassend erfolgen. Trotz der notwendigen Beschränkung auf einige Beispiele, soll durch deren Auswahl der Stand der Dinge allgemeiner skizziert werden. Im Fokus stehen dabei populäre Desktop-GIS und Geodienste, die eine breite Anwendung finden. Weiter wird auf die Zeitbehandlung im AAA-Modell des amtlichen Vermessungswesens eingegangen werden.

### 2.5.1 Desktop-GIS

ArcGIS Desktop ist ein in verschiedenen Versionen verfügbares, kommerzielles Desktop-GIS der GIS-Produktfamilie des Unternehmens ESRI, welches international verwendet wird (BILL 2010 S.148). Aufgrund der führenden Marktposition wird hier auf dieses Produkt eingegangen. In der aktuellen Version 10.1 wird für eine Vielzahl von Datenformaten eine Modellierung der Dynamik angeboten.

Dazu wird die *valid time* der Objekte durch Attribute analog zu der Tupel-Versionierung temporaler Datenbanken erfasst (Abschnitt 2.3.3). Dieses Prinzip wird dabei auch auf die nicht unmittelbar in Datenbanken gehaltenen Daten wie NetCDF übertragen (ESRI 2012a). Für die entsprechenden Attribute können die Datentypen *Date*, *Numeric* oder *String* verwendet werden. Eine Speicherung in einem Intervalltyp ist nicht dokumentiert. Stattdessen wird die Gültigkeit durch Start- und Endzeitpunkt angegeben oder sie umfasst nur einen einzigen Zeitpunkt (ESRI 2012b). Durch eine entsprechende Einstellung kann für einen solchen Layer bekannt gegeben werden, dass er Zeitinformaten enthält. Dadurch wird er für das Visualisierungswerkzeug *Time Slider* verwendbar. Dieses ist dafür konzipiert, die Geodaten in Abhängigkeit von ihrer zeitlichen Gültigkeit darzustellen, so dass die Darstellung dem aktuellen Zustand für einen gegebenen Zeitpunkt entspricht. Dem Nutzer wird ermöglicht, mithilfe eines *Sliders* durch die Zeit zu navigieren, das Verhalten durch zahlreiche Optionen zu steuern und die einzelnen Bilder als Film abzuspielen. Die Bildsequenzen lassen sich als solche auch exportieren (ESRI 2012c).

Durch die Programmiererweiterung *Tracking Analyst* stehen noch weitere Zeitfunktionen zur Verfügung. Diese sind in erster Linie für eine adäquate Behandlung von sich bewegenden Objekten ausgerichtet und basieren auf einem speziellen Datenformat, dem *tracking layer*. Hierin werden einzelne Positionen eines Objektes erfasst, aus denen dann ein Pfad approximiert werden kann. Eine zeitabhängige Visualisierung kann auch hier mit dem *Tracking Analyst* erfolgen. Mit dem *Playback Manager* ist aber auch ein ähnliches, speziell auf bewegte Objekte zugeschnittenes Werkzeug realisiert. Darüber hinaus ist es auch möglich, spezielle Diagramme zu berechnen welche die Zeitdimension mit der Thematik der Objekte verknüpfen. Diese sollen den Nutzer dabei unterstützen, Muster bzw. Periodizitäten in den Zeitreihen zu erkennen (ESRI 2012d).

Mit der QuantumGIS-Erweiterung *Time Manager* sind die *Time Slider* Funktionen auch für Nutzer dieses Open Source GIS weitgehend verfügbar. Anders als beim proprietären Vorbild können dabei auch Intervalltypen verwendet werden (GRASER 2011).

Es lässt sich feststellen, dass in einem modernen GIS Methoden integriert sind, die eine Analyse der Dynamik ermöglichen. Primär dienen sie einer geeigneten Visualisierung, um den Anwender bei einer solchen Untersuchung zu unterstützen. Die Semantik der *valid time* wird dabei innerhalb eines Systems durch verschiedene Datenschemata repräsentiert. Infolgedessen muss vom Anwender für jeden Layer festgelegt werden, ob es sich um Zeitpunkte oder Zeitspannen handelt. Komplexere raum-zeitliche Abfragen sind nicht realisiert. Sollten sie durch eine Kombination der bestehenden Abfragen erfolgen, müssten sie vom Nutzer immer auf das jeweilige Schema abgestimmt werden.

## 2.5.2 Geodatenserver

Als Beispiel für die Zeitbehandlung durch GIS-Serverdienste wird hier auf die beiden Open Source Projekte *Geoserver* und *Mapserver* eingegangen. Zahlreiche Geodienste bauen auf diesen Plattformen auf. Beide Projekte unterstützen neben weiteren Standards des OGC auch deren WMS-Standard<sup>21</sup> (GEOSERVER 2012, MAPSERVER 2012). Dieser bietet die Möglichkeit die Zeit als zusätzliche Dimension der Geoinformationen mit einzubeziehen. Die entsprechenden Funktionen sind in Mapserver implementiert (MAPSERVER 2006) und werden zurzeit in das Geoserver-Projekt aufgenommen (GEOSOLUTIONS 2011a). Um die Zeitdimension in dem Dienst verfügbar zu machen, ist die Angabe des Namens der Dimension, der verwendeten Einheiten und der von den Werten abgedeckte Zeitbereich erforderlich. Daneben können optional noch weitere Attribute angegeben werden. Zum besseren Verständnis ist ein Beispiel aus der WMS (Ogc 2006 S.53) übernommen:

```
<Dimension name="time" units="ISO8601" default="2003-10-17">
  1996-01-01/2003-10-17/P1D </Dimension>.
```

Ist ein Layer entsprechend konfiguriert, wird in der Rückgabedatei einer WMS-*GetCapabilities*-Anfrage die Zeit als Dimension für diesen Layer wie im Beispiel angegeben. Die Informationsübermittlung über die *valid time* sollte im ISO8601-Format und einigen hieran anknüpfenden Erweiterungen<sup>22</sup> erfolgen. In eine WMS-*GetMap*-Anfrage kann dann die Zeit als Parameter aufgenommen werden und somit die Zustände für bestimmte Zeitpunkte oder Zeitspannen abgefragt werden (Ogc 2006 S.51-58). Für den Geoserver steht auch eine Funktion bereit, die es ermöglicht die zeitliche Abfolge der Zustände durch eine Bildsequenz im *Graphics Interchange Format* wiedergeben zu lassen (GEOSOLUTIONS 2011b).

---

<sup>21</sup> *OpenGIS® Web Map Service Implementation Specification* (Ogc 2006). Die aktuelle Version ist 1.3, *Mapserver* und *Geoserver* unterstützen neben dieser aber auch noch frühere WMS-Versionen.

<sup>22</sup> Diese sind in Ogc (2006) S.57f definiert. *Geo-* und *Mapserver* unterstützen jedoch nicht alle der spezifizierten Formate.



Diese Beispiele zeigen, dass die Zeit sowohl in den maßgebenden Standards als auch durch die darauf aufbauenden Umsetzungen in Geodiensten, berücksichtigt wird. Wie die Daten im *Back end* gehalten werden, wird dabei den Entwicklern überlassen.

### 2.5.3 AAA-Modell

Das AAA-Modell der Adv dient einer einheitlichen Verwaltung der Geodaten des amtlichen Vermessungswesens. Um dies zu erreichen, wird ein gemeinsames Anwendungsschema für die drei Informationssysteme ALKIS, ATKIS und AFIS festgelegt. Die Umsetzung der Vorgaben ist ein zurzeit laufender Prozess, der den einzelnen Bundesländern obliegt (Adv 2012).

Im zugrunde liegenden konzeptuellen Modell, welches in der GeoInfoDok<sup>23</sup> der Adv dokumentiert ist, wird die zeitliche Gültigkeit der Geodaten durch ein Versionskonzept berücksichtigt. Dabei wird immer dann, wenn eine Änderung eines oder mehrerer Objektattribute auftritt, eine zusätzliche, aktuelle Version des Objektes angelegt. Es wird auf die Konzepte der Objektlebenslinien und der Tupel-Versionierung zurückgegriffen (Vergleiche Abschnitt 2.3.1 und 2.3.3). Die resultierende Redundanz wird dabei zugunsten der Zugriffsgeschwindigkeit ausdrücklich akzeptiert. Maßgebend für das Versionskonzept ist die *transaction time*. So richtet sich die Gültigkeit eines Objektes hier nach den Zeitpunkten der Erfassung der Zustände in der Datenbank. Jedes Objekt führt ein Entstehungs- und Untergangsdatum. Wird es erstmalig angelegt, geschieht dies mit dem Zeitpunkt der Eintragung als Entstehungsdatum. Bei einer Aktualisierung wird bei dieser Objektversion das Untergangsdatum gesetzt, welches äquivalent zu dem Entstehungsdatum der neuen Version ist. Endet die Existenz eines Objektes, wird zur Erfassung lediglich das Untergangsdatum der bis dahin aktuellen Version gesetzt. So werden alle Objektänderungen dokumentiert und die historischen Datenzustände sind immer rekonstruierbar (Adv 2009 S.57-63). Die Angabe der Zeitinformationen erfolgt im ISO8601-Format, ohne Zeitonenverschiebung gegenüber UTC (Adv 2009 S.21).

Im Zusammenhang mit der Dynamik sind auch die Prozesse zu erwähnen, die in der GeoInfoDok festgelegt werden. Durch einen Prozess wird hier ein Datenbestand in einen anderen überführt. Es wird zwischen Erhebungs-, Qualifizierungs-, Führungs-, Benutzungs- und Transferprozess unterschieden. Die Prozesse untergliedern sich in Vorgänge und weiter in Aktivitäten, deren Modellierung folgendes Ziel hat:

„Für eine vollständige Anwendungsbeschreibung sind Vorgänge und Aktivitäten zu definieren, die Daten in funktionelle Abhängigkeiten setzen und das dynamische Verhalten der Anwendung definieren.“ (Adv 2009 S.69).

Diese lassen sich mit den in Abschnitt 2.3.1 beschriebenen Ereignisobjekten vergleichen, wenn auch nicht gleichsetzen. So wird hier durch die Prozesse, abgesehen vom

---

<sup>23</sup> Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens (Adv 2009).

Erhebungsprozess, lediglich das Systemverhalten durch die Abhängigkeiten der Ausgabe von den Eingabeparametern beschrieben. Jedoch erfolgt eine explizite Beschreibung eines Vorgangs, dessen Art über eine reine Datenmanipulation hinausgeht. So kann z.B. der Führungsprozess, bei dem amtliche Daten ihren Zustand ändern, als justizielles Ereignis aufgefasst werden. Die Notwendigkeit einer solchen Interpretation besteht indes nicht, da sich hier die relevanten Eigenschaften der Ereignisse immer ausreichend in der Versionierung widerspiegeln (Adv 2009 S.68-73).

Die Modellierung der Dynamik ist demnach mit der Versionierung ein wesentlicher Bestandteil des AAA-Modells. Dieses Konzept wurde speziell für die amtlichen Geodaten entwickelt. Diese unterliegen einer relativ langsamen zeitlichen Variation (Liegenschaftskataster und Festpunkte) und ihr aktueller Zustand wird nur in größeren Zeitabständen durch Messverfahren erfasst (Topographie). Aus diesem Grund lässt sich die hier zweckmäßige Zeitbehandlung auch nicht ohne weiteres auf andere Anwendungen übertragen. Hier könnte die hohe Redundanz oder die nicht mögliche explizite Modellierung von Ereignissen einer Verwendung im Wege stehen.

## 2.6 Zusammenfassung

In diesem Abschnitt wurde die Komplexität der Zeitmodellierung verdeutlicht. Wie gezeigt, wird der Aufbau von Kalender- und Uhrzeitsystemen durch verschiedenste Regeln definiert. Da diese Systeme die Basis der meisten Zeitangaben bilden, müssen sie auch in den Umsetzungen berücksichtigt werden. Weiter wurde beschrieben, wie vielfältig die konzeptuellen Modelle der Zeitdimension als solcher sind. Dies konnte ebenso für die Konzepte zur Verknüpfung der Zeit- mit den Raumdimensionen festgestellt werden. Davon abgesehen wurden für die Modellierung der Dynamik, in dem zeitlichen Schema der ISO, konkretere Festsetzungen getroffen. Die selektive Betrachtung einiger Umsetzungen zeigte, dass die Konzepte auch in der Praxis angewandt werden. So stellen verschiedenste Anwendungen ihren Nutzern Zeitfunktionalitäten zur Verfügung. Die angebotenen Möglichkeiten sind dabei meist speziell auf einen Anwendungsfall ausgerichtet und können die Anforderungen nicht umfassend erfüllen. In ihrem Umfang bleiben sie hinter der ISO19108 zurück. Die oben gezeigten Funktionen ermöglichen es Zustände für diskrete Zeitpunkte oder Zeitspannen abzufragen und zu visualisieren. Raum-zeitliche Abfragen, wie z.B. die zeitlichen Objektbeziehungen der ISO19108, werden nicht direkt angeboten. Auch in den Standards der OGC und ISO wird die temporale Dimension meist in einem konkreten Kontext betrachtet, so zum Beispiel unter dem Gesichtspunkt der Dynamik oder den bewegten Objekten.

Aus diesen Gründen lässt sich die in der Einführung gemachte Behauptung, die Zeitmodellierung in GIS sei ein laufender Prozess, bestätigen. Symptomatisch hierfür ist auch die *Abstract Specification zu Features* des OGC. So wurde hier 2009 noch geschrieben, dass Abstraktionen gewöhnlich statisch sind, aber auch, dass bereits eine steigende Anzahl von Anwendungen dynamische Abstraktionen anwendet (Ogc 2009 S. 7). Es wird auch darauf

hingewiesen, dass sich eine spätere Version detaillierter mit der zeitlichen Charakteristik von Geoinformationen auseinandersetzen wird (OGC 2009 S. 3).

Zu der Tatsache, dass bisher kein geschlossenes, universelles Raum-Zeit-Modell für GIS existiert, sollen zwei Dinge bemerkt werden. Zum einen wäre ein solches Modell wohl so komplex, dass es für viele Anwendungen viel zu weitreichend ist und sie unnötig verkomplizieren würde. Demnach ist es nicht möglich ein Konzept zu finden, das für alle Anwendungsfälle adäquat ist. Eine gewisse Heterogenität der Modelle ist also erforderlich, ähnlich wie es auch für den reinen Raumbezug verschiedene Ansätze gibt. So stellt PEUQUET (2002) dazu fest:

*"[...] in extending the power of representational models, the cost will always be an increase in complexity. This has been the case with nonspatial database models, spatial database models, and knowledge representations – thus the famous adage: There is no such thing as the perfect model, except the thing you are modeling itself."*  
(PEUQUET 2002 S.320).

Davon abgesehen lässt sich vermuten, dass eine Weiterentwicklung in Theorie und Praxis hinsichtlich der Zeit, so wie es vor etwa zehn Jahren von Experten eingeschätzt wurde, nach wie vor erforderlich ist. Auch komplexere Konzepte sollten, wenn sie benötigt werden, zur Verfügung stehen (PEUQUET 2001 S.27, WORBOYS UND DUCKHAM 2004 S.361).

### 3. Konzeption

Aufbauend auf den oben dargelegten Grundlagen, insbesondere auf dem zeitlichen Schema des ISO/TC211, erfolgt hier die Konzeption einer Datenstruktur<sup>24</sup>. Diese soll geeignet sein, verschiedene zeitliche Objekte zu repräsentieren. Wie in der Zielstellung formuliert, soll dies einheitlich für die verschiedenen zeitlichen Geometrieformen erfolgen. Die Struktur soll als Datentyp in einer Geodatenbank dienen. Der Datentyp trägt hier den Titel *Time*.

Zu Beginn wird kurz auf den zugrundeliegenden Ansatz eingegangen. Danach werden die unterschiedlichen Objekte klassifiziert, die in die Datenstruktur eingebunden werden sollen. Dies schließt auch eine Auseinandersetzung mit den unterschiedlichen temporalen Bezugssystemen ein. Dem folgt die Konzeption einer eindeutigen alphanumerischen Darstellung der unterschiedlichen Objekte. Diese soll der Konstruktion sowie der Repräsentation der *Time*-Instanzen in der Datenbank dienen. Anschließend wird auf die Operationen eingegangen, die unmittelbar auf der Datenstruktur möglich sein sollten.

---

<sup>24</sup> Der Literatur folgend werden die Begriffe Datenstruktur und -typ synonym verwendet und meinen: „Eine Menge gleichartiger Daten, auf denen eine Sammlung von Operationen definiert ist. Eine Operation ist dabei eine Verknüpfung, die einer festen Anzahl von Eingabedaten ein Ergebnis zuordnet.“ (GRUMM UND SOMMER 2011 S.102).

Abschließend werden mögliche Kombinationen der konzipierten Operationen mit räumlichen Analysefunktionen aufgezeigt.

### 3.1 Konzept einer universellen Behandlung der *valid time*

Der Ansatz, die *valid time*-Informationen ungeachtet der konkreten temporalen Geometrie einheitlich behandeln zu wollen, kann in einen Kontext zu entsprechenden Konzepten für die räumliche Geometrie gestellt werden. So wird von OGC und ISO in der Implementierungsbeschreibung<sup>25</sup> des räumlichen Schemas der ISO19107<sup>26</sup> die *Well-known Text Representation for Geometry* als einheitliche Repräsentation für die unterschiedlichen Geometrieobjekte definiert. Diese kann für die Erzeugung der Objekte ebenso wie für die alphanumerische Darstellung genutzt werden (Ogc 2011 S.51). Ähnlich wird mit der *Well-known Binary Representation for Geometry* die Darstellungsweise der Objekte in Bytes festgelegt (Ogc 2011 S.62). Die *Well-known Text Representation of Spatial Reference Systems* standardisiert die textuelle Repräsentation eines Raumbezugssystems (Ogc 2011 S.73). Im zweiten Teil der *Simple Feature Access* Spezifikation wird unter anderem beschrieben, wie diese Konzepte in SQL-Schema integriert werden können. Hierzu werden benutzerdefinierte Typen und Funktionen auf diesen Typen definiert (Ogc 2010 S.2). Eine populäre Umsetzung ist PostGIS, eine Erweiterung für das DBMS PostgreSQL. Wesentlicher Bestandteil der Erweiterung ist der Datentyp *geometry*, der es erlaubt Geometrieobjekte aus ihrer WKT- oder WKB-Repräsentation in der Datenbank anzulegen (POSTGIS 2012a). Aufgrund der einheitlichen Datenhaltung im *geometry*-Typ erfolgen die Funktionsaufrufe - beispielsweise für räumliche Analysen oder das Abfragen geometrischer Parameter - auf der Datenbankebene unabhängig vom Geometriotyp (POSTGIS 2012b). Die Tabelle 3-1 enthält einige Beispiele für die WKT-Darstellung:

Tabelle 3-1: Auszug aus der *Simple Feature Access* Spezifikation – WKT Beispiele (Ogc 2011 S.61).

Typ	Literal der WKT	OGC Kommentar
<i>Point</i>	Point (10 10)	<i>a Point</i>
<i>LineString</i>	LineString (10 10, 20 20, 30 40)	<i>a LineString with 3 points</i>
<i>Polygon</i>	Polygon(10 10, 10 20, 20 20, 20 15, 10 10)	<i>a Polygon with 1 exteriorRing and 0 interiorRings</i>
<i>Multipoint</i>	MultiPoint((10 10), (20 20))	<i>a MultiPoint with 2 points</i>
...		

Im Folgenden soll versucht werden, Ansätze eines ähnlichen Konzepts für die *valid time* zu entwickeln.

<sup>25</sup> Der Titel des OGC Standard ist *OpenGIS® Implementation Standard for Geographic information - Simple feature access* (Ogc 2011 und Ogc 2010), das entsprechende ISO-Dokument ist die ISO19125. Die Dokumente sind jeweils in einen Allgemeinen und einen SQL-Teil untergliedert.

<sup>26</sup> *Geographic information – Spatial schema*.

## 3.2 Die time-Objekte

In diesem Abschnitt werden die temporalen Objekte klassifiziert und die für die Informationsrepräsentation notwendigen Daten herausgestellt. Die Objekte werden dabei aus der ISO19108 übernommen, wobei das Modell im Detail von dieser Norm abweichen wird. Die Modifikationen werden entsprechend erläutert und sind in dem Standard vorgesehen (Iso 2002a S.11):

„[...] to conform to this International Standard, an implementation shall provide the capabilities described by these elements of the abstract model, but it need not implement them in the same way.“

Insbesondere soll von der Möglichkeit Gebrauch gemacht werden nur die temporale Geometrie explizit zu beschreiben und die Topologie nur durch Operationen auf den geometrischen Objekten zu berücksichtigen. Dies scheint für den überwiegenden Anteil der möglichen Anwendungen adäquat und soll einer Überfrachtung der Datenstruktur vorbeugen (Iso 2002a S.12). Die ISO19108-Funktionen *length*, *distance* und *relativePosition* lassen sich auf diesem Wege realisieren.

### 3.2.1 Klassifikation der Objekte

Für die beabsichtigte Verwaltung der Zeitinformationen in einem elementaren Datentyp ist besonders der in Abschnitt 2.4.3 beschriebene *GF\_TemporalAttributeType* relevant. Wie oben bereits erläutert, dient er einer Beschreibung der *valid time* von Geoobjekten durch geeignete Attribute. Dabei wurde grundsätzlich zwischen punktuellen Ereignissen und ausgedehnten Zuständen unterschieden. Von einer solchen weiterführenden Semantikuweisung zu den beiden primitiven Zeitgeometrien soll hier abgesehen werden. Stattdessen wird angenommen, dass auch Zeitpunkte Zustände repräsentieren und Ereignisse eine Dauer haben können. Ein Beispiel wäre eine tägliche Wasserstandsmessung eines Fließgewässers. Hier wären die zeitlich punktuellen Messwerte Zustände und die Zeitspannen dazwischen Ereignisse (die Änderung des Wasserstandes). Aus diesem Grund werden direkt die Klassen *TM\_Instant* und *TM\_Period* in die Datenstruktur integriert. Ob es sich um Zustände oder Ereignisse handelt ist dann durch das Anwendungsschema auszudrücken.

Neben diesen beiden primitiven Typen werden noch vier weitere Klassen gebildet. Zwei sollen jeweils der Modellierung multipler *TM\_Instant*- oder *TM\_Period*-Instanzen dienen. Sie werden mit *MultiInstant* und *MultiPeriod* bezeichnet. Durch diese können kongruente Zustände, in denen sich ein Objekt mit zeitlichen Unterbrechungen befand, mit einer Instanz temporal referenziert werden. Ähnliches leisten die beiden weiteren Objekttypen für regelmäßig wiederkehrende Zustände oder Ereignisse. Dies sind *TM\_Instant*- oder *TM\_Period*-Objekte, die nach einer konkreten Dauer erneut gültig sind. Sie werden hier als *RegularMultiInstant* und *RegularMultiPeriod* eingeführt.

Diese Klassifikation beruht auf den Ausführungen der ISO19108, speziell die den *GF\_TemporalAttributeType* betreffenden (Iso 2002a S.30f). Die Details dieser sechs Klassen

werden nachfolgend behandelt. Über eine bloße Modellierung von Zeitgeometrien gehen diese Klassen bewusst nicht hinaus. Fragen wie:

- ob ein Phänomen als Zeitpunkt oder Zeitspanne modelliert werden soll
- ob als einzelnes *TM\_Instant*- oder *MultiInstant*-Objekt
- ob es als Zustand oder Ereignis (Vergleiche Abschnitt 2.3.1) zu sehen ist

müssen für den jeweiligen Anwendungsfall beantwortet werden. Je nach Vorhaben kann dasselbe Phänomen dabei auf unterschiedliche Weise modelliert werden. Beispielphänomene sind in Tabelle 3-2 enthalten, durch die oben genannten Gründe kann eine Zuordnung dabei nicht eindeutig sein.

Tabelle 3-2: Beispiele für die sechs Time-Objekte des Datentyps.

Typ	Zustand	Ereignis
<i>TM_Instant</i>	Blitzeinschlag	Änderung eines Grundstückeigentümers
<i>TM_Period</i>	Dauer in der ein Grundstück den gleichen Eigentümer hat	Bau eines Gebäudes
<i>MultiInstant</i>	Radarfalle wurde ausgelöst	Cholerafall wird in Gebäude diagnostiziert
<i>MultiPeriod</i>	Phasen in denen auf einem Acker die gleiche Feldfrucht gepflanzt wurde	Waldbrand in einem bestimmten Waldgebiet
<i>RegularMultiInstant</i>	Aufnahmezeitpunkte einer automatischen Webcam	Fällig werden einer Grundstückspacht
<i>RegularMultiPeriod</i>	Wochenmarkt	Eine Brücke ist Nachts geschlossen

*TM\_Instant* und *TM\_Period* sind ausführlich in der ISO definiert und wurden dementsprechend in Abschnitt 2.4.1 behandelt. Das einzige Attribut der Klasse *TM\_Instant* nutzt ein *TM\_TemporalPosition*-Objekt als Datentyp, um einen Zeitpunkt zu referenzieren (Siehe 2.4.2). Die *TM\_Period*-Objekte bilden sich aus zwei *TM\_Instant*-Instanzen. Dabei liegt das *begin*- zeitlich vor dem *end-TM\_Instant*-Objekt (Siehe Abbildung 3-1).

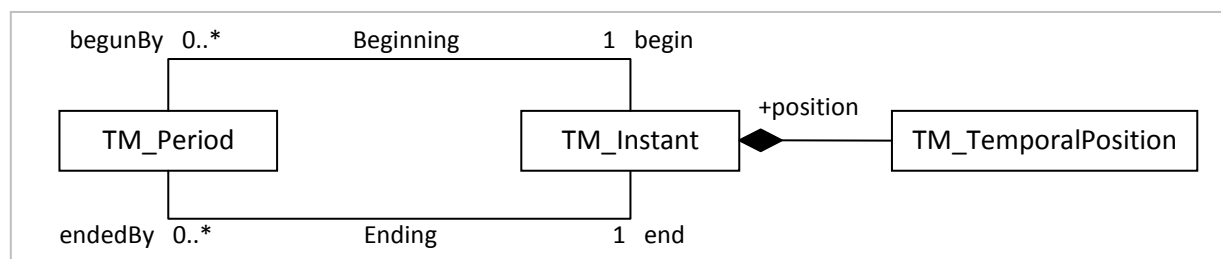


Abbildung 3-1: UML – Die Klassen *TM\_Instant* und *TM\_Period* (nach Iso 2002a S.14).

Die Klassen *MultiInstant* und *MultiPeriod* ergeben sich jeweils aus Kompositionen temporaler Primitive. Konzeptuell kann ein Multi-Objekt auch aus einem einzigen temporalen Primitiv gebildet werden, dann ist es diesem gleich (Siehe Abbildung 3-2).

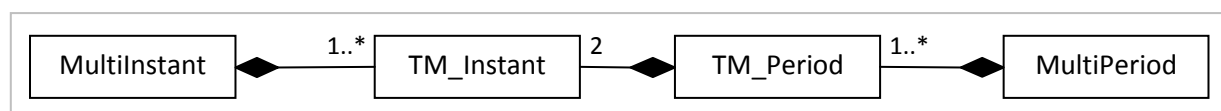


Abbildung 3-2: UML – Die Klassen *MultiInstant* und *MultiPeriod*.

Für die Beschreibung eines regelmäßig wiederkehrenden Phänomens werden in der ISO19108 zwei minimal erforderliche Informationen festgesetzt. Diese sind das Temporale Primitiv des ersten Auftretens sowie die Dauer, die zwischen den gültigen Zeitspannen liegt. Demnach wäre die Periodizität nicht terminiert. Dieser Umstand soll hier so ausgelegt werden, dass das Phänomen bis zum gegenwärtigen Zeitpunkt gültig ist. Daneben soll optional die Kardinalität festgelegt werden können (Siehe Abbildung 3-3).

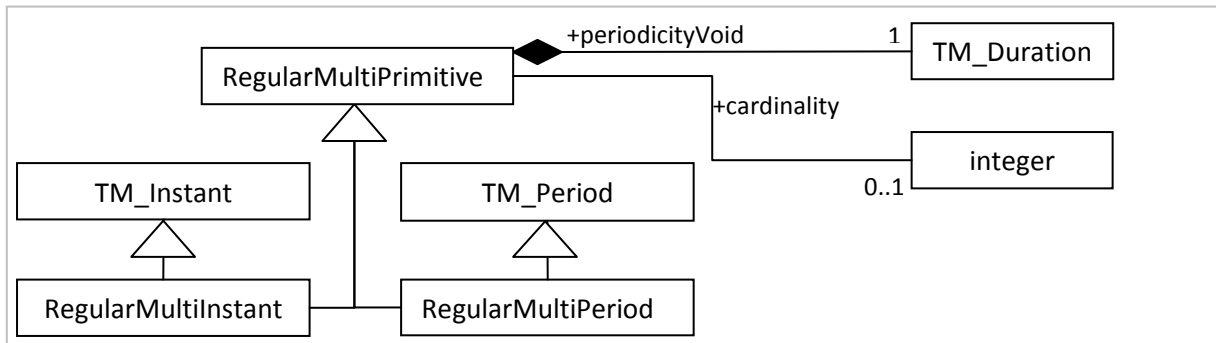


Abbildung 3-3: UML - Die Klassen RegularMultiInstant und RegularMultiPeriod.

Die hier einbezogene Dauer ist eine definierte Ausdehnung in der Zeit, ohne absolute Position. Eine Dauer sollte in den Einheiten eines Kalendersystems oder zeitlichen Koordinatensystems angegeben werden können. Bei Kalendersystemen ergibt sich die exakte Dauer unter Umständen erst durch den Kontext. So ist ein Jahr beispielsweise 365 oder 366 Tage lang. Ähnliches gilt für Monate und bei Verwendung von UTC sogar für Tage. Daher müssen, für eine kalenderbezogene Dauer, diese initialen Informationen bereitgehalten werden (Iso 2004 S.30f). Eine Gleichsetzung mit einem gewöhnlichen *Period*-Objekt mit fester Zeitgeometrie, gebildet durch Start- und Endzeitpunkt, ist nicht möglich. Auch wenn theoretisch auch innerhalb von Ordinalsystemen regelmäßig wiederkehrende Phänomene gebildet werden könnten, soll dies in dieser Konzeption ausgeschlossen werden. Eine Dauer könnte hier nur qualitativ angegeben werden und ein Ereignis dann zum Beispiel in jeder dritten *TM\_OrdinalEra* auftreten. Um das Konzept der wiederkehrenden Phänomene nicht derartig auszuweiten, sollen sie auf Kalender- und zeitliche Koordinatensysteme beschränkt sein.

### 3.2.2 Bedeutung der zeitlichen Bezugssysteme

Die Zuweisung der Zeitpunkte zu den oben definierten Objekten erfolgt wie angegeben durch die in Abschnitt 2.4.2 behandelte Klasse *TM\_TemporalPosition*. Es wurde herausgestellt, dass sich die Zeitpositionen auf den Gregorianischen Kalender und UTC oder eines der *TM\_ReferenceSystem*-Objekte beziehen können. Der überwiegende Anteil der Anwendungen wird den Gregorianischen Kalender und UTC nutzen. In die hier erfolgende Konzeption soll dennoch die Möglichkeiten eingebunden werden, ein anderes System zu verwenden. Die *TM\_TemporalPosition*-Instanzen, die durch Kompositionsbeziehungen eines

der sechs oben beschriebenen Objekte bilden, sollen dasselbe Bezugssystem verwenden. Jedes *time*-Objekt steht demnach in einer Beziehung zu einem *TM\_ReferenceSystem*.

Die von der ISO19108 vorgegebenen *TM\_TemporalPosition*-Kindklassen werden nicht in der gleichen Weise übernommen. Auf die Klassen *TM\_CalDate* und *TM\_ClockTime* soll verzichtet werden und stattdessen nur die Vereinigung durch die Klasse *TM\_DateAndTime* realisiert werden. Die *TM\_ClockTime* Objekte dienen der Beschreibung täglich wiederkehrender Ereignisse, dies kann stattdessen analog durch die Verwendung einer *RegularMultiPrimitive*-Instanz erfolgen. Im Folgenden ist mit einem Kalender immer die Kombination aus Kalender- und Uhrzeitsystem gemeint. Der Gregorianische Kalender und die UTC werden zu einem gewöhnlichen *TM\_DateAndTime*-Objekt zusammengefasst. Daher ist eine Trennung zwischen *TM\_Position* und *TM\_TemporalPosition* nicht weiter erforderlich. Zeitpositionen werden einheitlich durch *TM\_TemporalPosition*-Objekte beschrieben. Neben den Kalendersystemen beinhaltet dies auch die Subklassen *TM\_OrdinalPosition* und *TM\_Coordinate* (Abschnitt 2.4.2). Die Abbildung 3-4 enthält eine UML Darstellung der aus der ISO19108 übernommenen Bezugssysteme.

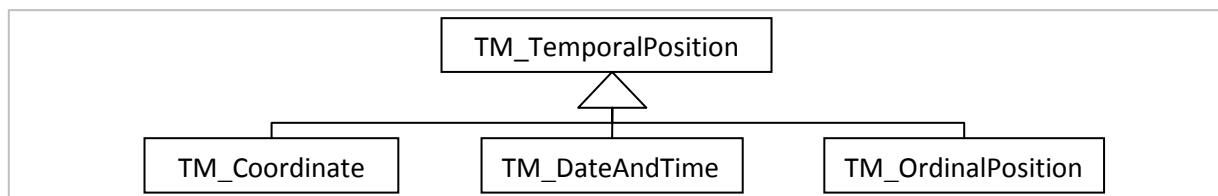


Abbildung 3-4: UML – Übernommene *TM\_TemporalPosition*-Subtypen.

Konzeptuell soll vorgegeben werden, dass sich die aus dem jeweiligen *TM\_ReferenceSystem* ergebenden *TM\_TemporalPosition*-Objekte intern in das Julianische Datum überführt werden (Siehe Abbildung 3-5). Dieses Zeitsystem sowie die besondere Bedeutung in der ISO19108 wurden bereits in Abschnitt 2.4.2 herausgestellt. Das Julianische Datum eines Zeitpunktes entsprach der Anzahl der vergangenen Tage seit dem Jahr 4713v.Chr. (im Julianischen Kalender). Durch die Verwendung von Dezimalstellen kann die Auflösung theoretisch beliebig gesteigert werden. Für zahlreiche Kalendersysteme existieren Algorithmen für Transformationen von und in das Julianische Datum. Die wesentlichen Vorteile der Verwendung liegen in den performanten Berechnungen sowie den einfachen Transformationen zwischen verschiedenen Kalendersystemen durch die jeweiligen Algorithmen. Darüber hinaus kann auch von einer Datenreduktion gesprochen werden.

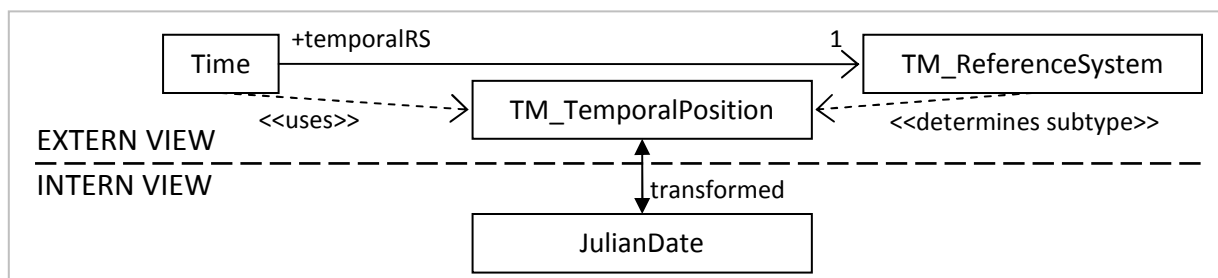


Abbildung 3-5: UML – Konzept der internen und externen *TM\_TemporalPosition*-Objekte.



Die Einbindung der Klasse *TM\_IndeterminanteValue* ist problematisch zu sehen. Dies sind Zeitpositionen, deren Lage nur qualitativ oder gänzlich unbekannt ist. Durch die ISO wird vorgegeben, dass ein Funktionsaufruf der diese Objekte betrifft, eine Exception verursacht (Abschnitt 2.4.2). Dies könnte umgangen werden, indem bei einem Zugriff die Objekte durch absolute Pseudozeitpunkte ersetzt werden. Funktionen, die auf quantitative Informationen abzielen, würden dann allerdings auch nur unbrauchbare Pseudoergebnisse liefern. Ausgenommen hiervon ist die Instanz *now*, die immer der aktuellen Gegenwart entspricht. Diese kann in den Wertebereich für Zeitpunkte aufgenommen werden, wobei durch besondere Bedingungen zu gewährleisten ist, dass erzeugte Objekte auch für alle späteren Zeitpunkte gültig sind. Das Beispiel einer *Period*-Instanz, welche mit *now* als Startzeitpunkt und einem später liegenden Endzeitpunkt A definiert ist, verdeutlicht das Problem. Nach dem Zeitpunkt A ist diese *Period* nicht länger ein gültiges Objekt und kann zu unerwartetem Verhalten führen. Die unterminierten Multiobjekte, durch die bereits implizit *now* eingeführt wurde, sind eine mögliche Anwendung. Die weiteren *TM\_IndeterminanteValue*-Objekte werden aus der Konzeption ausgeschlossen.

Typischerweise werden die temporalen Charakteristika eines Phänomens immer innerhalb desselben Bezugssystems beschrieben werden. Daher scheint es sinnvoll, die Daten über das verwendete System nicht mit jeder Instanz des Datentyps zu speichern. Zweckmäßiger wäre die Haltung entsprechender Informationen für die jeweiligen Attribute der Entitätstypen oder sogar an einer Stelle für die gesamte Datenbank. Der bereits in Abschnitt 3.1 erwähnte SQL-Teil der *Simple Feature Access* Spezifikation gibt vor, wie entsprechend Informationen über räumliche Systeme abgelegt werden sollten. Dort ist ein konkretes Schema vorgegeben, wie durch die Einführung von Tabellen und Beziehungen das Referenzsystem losgelöst von den einzelnen Geometrieinstanzen definiert wird (Iso 2010). Trotz der Kenntnis dessen, wird der Datentyp hier so konzeptualisiert, dass jede Instanz Informationen über das verwendete Bezugssystem enthält. Grund ist das Fehlen entsprechender Vorgaben für zeitliche Bezugssysteme, ein zu großes Maß an eigenmächtigen Festsetzungen soll an der Stelle vermieden werden.

### 3.3 Textuelle Repräsentation der Objekte und Bezugssysteme

In diesem Abschnitt wird die textuelle Repräsentation der in Abschnitt 3.2 definierten Objekte festgelegt. Diese soll der Erzeugung der Objekte und deren alphanummerischen Darstellung dienen. Die Beschreibung der erlaubten Syntax basiert auf der Backus-Naur-Form. Es werden die in Tabelle 3-3 angegebenen Terminale verwendet. Darüber hinaus dienen Buchstaben und Ziffern der Produktion. Aus der ebenfalls in BNF gehaltenen WKT-Definition sind einfache Elemente übernommen worden (OGC 2011 S.52-54).

Tabelle 3-3: Verwendete Terminale.

Symbol	Semantik
<>	Die eingeschlossenen Terminale bilden einen Nonterminal.
::=	Produktion des Linksstehenden Nonterminals durch die Ausdrücke der rechten Seite
	Wahl zwischen zwei Nonterminalen, steht zwischen diesen.
*	Zeigt die Möglichkeit an, mehrere Instanzen des Nonterminals zu verwenden, bezieht sich auf den vorstehenden Nonterminal.
()	Gruppierung mehrerer Nonterminale in einen einzigen.
{ }	Eingeschlossener Nonterminal ist optional.

### 3.3.1 Kalendersysteme – ISO8601-Format

Da hier nur eine praktische Darstellung der vorgesehenen textuellen Repräsentation erfolgen soll, sind einige Nonterminale nachfolgend direkt durch ISO8601-Formate beschrieben, auch wenn dies formell nicht exakt ist. Sie sollen der Beschreibung von Zeitpositionen, bezogen auf einen Kalender, dienen.

Datum und Uhrzeit werden im erweiterten Format angegeben, welches gegenüber dem einfachen Format eine bessere Lesbarkeit bietet. Die Uhrzeitangabe soll sich vorzugsweise auf die Greenwich-Zeitzone beziehen. Eine Zeitonenverschiebung kann jedoch angegeben werden. Das Format soll, wie in der ISO19108 festgelegt, für die Beschreibung eines Zeitpunktes bezogen auf ein beliebiges Kalender-Uhrzeitsystem verwendet werden. Der Wertebereich der einzelnen Elemente ergibt sich aus dem Bezugssystem. Zeitpunkte vor dem Referenzzeitpunkt des Kalendersystems können durch ein vorgestelltes Minuszeichen beschrieben werden. Die Genauigkeit der Angabe kann durch die Verwendung von Dezimalzahlen für die Sekunden verfeinert werden. Die erfolgten Festsetzungen basieren vollständig auf den Inhalten der ISO19108 sowie ISO8601 (Iso 2002a S.28f; Iso 2004 S.25-29).

`<8601 datetime> ::= YYYY-MM-DDThh:mm:ssZ | YYYY-MM-DDThh:mm:ss±hh:mm27`

Die Angabe eines Zeitintervalls durch Start- und Endzeitpunkt (bezogen auf ein Kalender-Uhrzeitsystem) soll in dem entsprechenden erweiterten Format der ISO8601 erfolgen.

`<slash> ::= /`

`<8601 interval> ::= <8601 datetime><slash><8601 datetime>`

Die Angabe einer Dauer, einer Zeitspanne ohne absolute Position, erfolgt wie durch den *TM\_PeriodDuration*-Typ der ISO19108 definiert. Dieser nutzt die ISO8601-Darstellung eines nur durch die Dauer bestimmten Zeitintervalls und überträgt es auf andere Bezugssysteme.

`<8601 duration> ::= PnnYnnMnnDTnnHnnMnnS28`

<sup>27</sup> Es sind Jahr, Monat und Tag durch Bindestriche getrennt anzugeben. Der Buchstabe T kennzeichnet die nachfolgende Uhrzeit. Diese wird in Stunden, Minuten und Sekunden angegeben, die Werte werden durch Doppelpunkte getrennt. Folgt der Buchstabe Z, beziehen sich die Angabe auf UTC, dies wird auch angenommen wenn die Zeichenkette mit dem Sekundenwert endet. Optional kann eine Zeitverschiebung gegenüber UTC durch ein Vorzeichen gefolgt von Stunden, Doppelpunkt und Minuten am Ende der Zeichenkette angegeben werden (Iso 2002a S.28f).

Zeichenketten, welche regelmäßig wiederkehrende Zeitintervalle repräsentieren, werden in der ISO8601 durch ein vorstehendes R in Verbindung mit einer optionalen Ganzzahl gekennzeichnet. Die Zahl folgt unmittelbar dem R und gibt die Anzahl der Wiederholungen an. Fehlt diese Zahl, terminiert die Periodizität nicht. Dieses Element soll hier übernommen werden.

```
<8601 repetitions number> ::= Rn
```

Die wiederkehrenden Zeitintervalle der ISO8601 beschreiben Zeitspannen, welche nach ihrem Auftreten unmittelbar an sich selbst anschließen. Der Endpunkt einer Periode fällt also immer mit dem Startpunkt der Folgenden zusammen. Hier soll dieses Format so geändert werden, dass auch ein wiederkehrendes Phänomen beschrieben werden kann, bei dem zwischen dem erneuten Auftreten der Gültigkeit jeweils eine konstante Zeitspanne liegt. Dies soll durch das nachfolgende Format geschehen, bei dem der Zeitpunkt in Verbindung mit der nachfolgenden Dauer die erste gültige Periode beschreibt und die abschließende Dauer die Zeit bis zum nächsten auftreten definiert.

```
<8601 regular period> ::= {(<8601 repetitions number> <slash>)}
                           <8601 datetime> <slash> <8601 duration>29
                           <slash> <8601 duration>
```

Die Repräsentation wiederkehrender Zeitintervalle durch Startpunkt und Dauer der ISO8601 soll hier zur Beschreibung wiederkehrender Zeitpunkte verwendet werden.

```
<8601 regular instant> ::= {(<8601 repetitions number> <slash>)}
                           <8601 datetime> <slash> <8601 duration>
```

So lassen sich die sechs definierten Klassen der zeitlichen Geometrie, bezogen auf einen Kalender, textuell wie folgt ausdrücken:

```
<comma> ::= ,
<calendar position> ::= (<8601 datetime> {(<comma> <8601 datetime>)}*)
                        | (<8601 interval> {(<comma> <8601 interval>)}*)
                        | <8601 regular instant> | <8601 regular period>
```

Dabei wird ein Komma zur Trennung verschiedener Primitive verwendet. Dies ist aus der WMS übernommen worden und erscheint darüber hinaus als intuitiv (Ogc 2006 S.58, Iso 2004).

### 3.3.2 Bezugssysteme

Auch die Informationen über das verwendete Bezugssystem müssen durch eine eindeutige Darstellung beschreibbar sein. Dabei soll bloß ein Verweis auf das entsprechende System

<sup>28</sup> Der Buchstabe P kennzeichnet die nachfolgende Zeichenkette als Dauer. Sie kann über Jahre, Monate, Tage, Stunden, Minuten oder Sekunden definiert werden. Dabei folgt dem entsprechenden Wert der Anfangsbuchstabe der entsprechenden Einheit in englischer Sprache, als Großbuchstabe. Bei Verwendung mehrerer Einheiten ordnen sich die Werte nach der Granularität, von der größten zur feinsten. Der Buchstabe T trennt Kalender- und Uhrzeitangaben (Iso 2004 S.30f).

<sup>29</sup> In dieser Zeile wird eine Periode abweichend von den sonstigen Festsetzungen und auch der ISO19108 durch Startzeitpunkt und Dauer definiert. Grund ist das Problem der kontextbezogenen Dauer in Kalendersystemen.

erfolgen und keine vollständige Beschreibung dessen Charakteristika. Dies ist ähnlich der Verwendung eindeutiger Codes für räumliche Bezugssysteme, wie z.B. durch SRID. Da in den verwendeten Quellen kein Hinweis auf die Existenz von hierfür geeigneten Vereinbarungen für temporale Systeme gefunden werden konnte, erfolgen eigenmächtige Festsetzungen.

Die Codierung setzt sich aus zwei Segmenten zusammen. Das erste beschreibt die Klasse des Systems, also ob es sich um ein Kalender-, Ordinal- oder temporales Koordinatensystem handelt. Hierfür dient eine entsprechende Signatur aus drei Großbuchstaben. Die Instanz der Klasse wird durch das zweite Segment festgelegt, einen Block aus drei Ziffern, Großbuchstaben oder Kleinbuchstaben.

```

<simple Latin upper case letter> ::=
    A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<simple Latin lower case letter> ::=
    a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
<digit> ::= 0|1|2|3|4|5|6|7|8|9
<system code> ::= (<digit> <digit> <digit>) |
    (<simple Latin upper case letter> <simple Latin upper case letter>
    <simple Latin upper case letter>) | (<simple Latin lower case letter>
    <simple Latin lower case letter> <simple Latin lower case letter>)
<calendar system flag> ::= CAL
<ordinal system flag> ::= ORD
<coordinate system flag> ::= TCS
<calendar system> ::= <calendar system flag> <system code>
<ordinal system> ::= <ordinal system flag> <system code>
<temporal coordinate system> ::= <coordinate system flag> <system code>

```

### 3.3.3 Beschreibung der Geometrie in anderen als Kalendersystemen

Hier erfolgen die Festsetzungen wie ein time-Objekt bezogen auf ein Ordinal- oder zeitliches Koordinatensystem beschrieben werden kann. Für die zeitlichen Koordinatensysteme wird das Kalenderformat übertragen, mit dem Unterschied, dass einzelne Zeitpunkte durch einen Gleitpunktwert angegeben werden. Analog wird eine Dauer in der Maßeinheit des Systems angegeben.

```

<unsigned integer> ::= (<digit>)*
<plus sign> ::= +
<minus sign> ::= -
<sign> ::= <plus sign> | <minus sign>
<signed integer> ::= {<sign>} <unsigned integer>
<decimal point> ::= .
<signed numeric literal> ::= <signed integer> {(<decimal point>
    <unsigned integer>)}
<unsigned numeric literal> ::= <unsigned integer> {(<decimal point>
    <unsigned integer>)}
<tcs instant> ::= <signed numeric literal>
<tcs period> ::= <tcs instant> <slash> <tcs instant>
<tcs regular instant> ::= {(<8601 repetitions number> <slash>)}
    <tcs instant> <slash> <unsigned numeric literal>
<tcs regular period> ::= {(<8601 repetitions number> <slash>)}
    <tcs period> <slash> <unsigned numeric literal>

```

Der folgende Ausdruck fasst die Darstellung aller sechs Geometrieobjekte zusammen:

```
<tcs position> ::= (<tcs instant> {(<comma> <tcs instant>)}*)
                | (<tcs period> {(<comma> <tcs period>)}*)
                | <tcs regular instant> | <tcs regular period>
```

Eine Ordinalposition wird durch einen Verweis auf ein *TM\_OrdinalEra*-Objekt angegeben. Dies soll durch einen eindeutigen Schlüssel geschehen. Welche solcher Zeichenketten gültig sind, richtet sich nach dem verwendeten Ordinalsystem. Hier wird die willkürliche Beschränkung getroffen, dass als Schlüssel ein oder mehrere Buchstaben verwendet werden.

```
<simple Latin letter> ::= <simple Latin lower case letter> |
                        <simple Latin upper case letter>
<ordinal instant> ::= (<simple Latin letter>)*
<ordinal period> ::= <ordinal instant> <slash> <ordinal instant>
```

Die für Ordinalsysteme vier möglichen time-Klassen lassen sich wie folgt bilden:

```
<ordinal position> ::= (<ordinal instant> {(<comma> <ordinal
instant>)}*) | (<ordinal period> {(<comma> <ordinal period>)}*)
```

### Vollständige Beschreibung einer time-Instanz

Durch die Angabe eines konkreten Bezugssystems und eines zeitlichen Geometrieobjekts innerhalb dieses Systems ist ein time-Objekt vollständig beschrieben. So lassen sich Zeitpunkte, -perioden und komplexere Typen eindeutig erzeugen.

```
<time> ::= ({<calendar system>} <calendar position>)
          | (<temporal coordinate system> <tcs position>)
          | (<ordinal system> <ordinal position>)
```

Fehlt die Angabe eines Bezugssystems, wird die temporale Geometrie dem Gregorianischen Kalender mit UTC zugeordnet. Beispiele sind in Tabelle 3-4 enthalten.

Tabelle 3-4: Beispiele für die textuelle Darstellung.

Typ	Beispiel
	Mögliche Tokendarstellung
TM_Instant	2012-8-15T18:6:34.22 <8601 instant>
TM_Instant	CAL0032012-8-15T18:6:34.22 <calendar system> <calendar position>
TM_Instant	ORD001Triassic <ordinal system> <ordinal position>
TM_Instant	TCS0012456157.07553 <temporal coordinate system> <tcs position>
TM_Period	TCS0012456157.07553/2456158.07553 <temporal coordinate system> <tcs position>
TM_Period	CAL0012012-8-15T18:06/2012-8-15T20:20:20 <calendar system> <8601 period>
MultiPeriod	CALGRE2012-8-15T18:06/2012-8-15T20:20:20, 2012-9-15T18:06/2012-9-15T20:20:20 <calendar system> <8601 period> <comma> <8601 period>
RegularMultiPeriod	CAL003R8/2012-8-15T10:00/P1DT12H/P1Y2M15DT12H <calendar system> <8601 repetitions number> <slash> <8601 interval> <slash> <8601 duration>

Ergänzend wird in Tabelle 3-5 der verwendete Aufzählungstyp für die Rückgabewerte der ISO19108-Funktion *relativePosition* auf einfache Weise festgelegt (Siehe Abbildung 2-1).

Tabelle 3-5: Der Aufzählungstyp *relativePosition*

0 = <i>Before</i>	3 = <i>Begins</i>	6 = <i>Contains</i>	9 = <i>Ends</i>	12 = <i>After</i>
1 = <i>Meets</i>	4 = <i>BegunBy</i>	7 = <i>Equals</i>	10 = <i>EndedBy</i>	
2 = <i>Overlaps</i>	5 = <i>During</i>	8 = <i>OverlappedBy</i>	11 = <i>MetBy</i>	

Im Folgenden ist mit `<relative position text>` der Wertebereich aus den in Tabelle 3-5 definierten konstanten Zeichenketten gemeint. Analog bezieht sich `<relative position integer>` auf die Zahlenwerte.

### 3.4 Operationen

In diesem Teil werden die Operationen definiert, die auf der Datenstruktur möglich sein sollten. Die Operationen werden durch Signaturen und eine Beschreibung der Abhängigkeit der Eingabe- von den Ausgabedaten eingeführt. Um den Abschnitt übersichtlich zu gestalten, werden die Funktionen in Kategorien eingeteilt. Die nachfolgende Liste enthält die fundamentalsten Operationen. Die Optionen sind jedoch so umfassend, dass über die hier definierten noch zahlreiche weitere Funktionen möglich sind.<sup>30</sup> Am Ende des Abschnitts wird auf die Möglichkeiten raum-zeitlicher Abfragen eingegangen.

#### 3.4.1 Basisfunktionen

Aufgrund der einheitlichen internen Behandlung der Zeitpunkte im Julianischen Tag könnten die Funktionen unabhängig vom jeweiligen Referenzsystem operieren. Ausgenommen davon sind die Ordinalsysteme, aufgrund der grundsätzlich anderen Semantik. Dennoch soll hier festgesetzt werden, dass Operationen, die mehr als eine *Time*-Instanz einschließen, nur ausgeführt werden, wenn sich die Objekte ein Bezugssystem teilen. Andernfalls wird eine *Exception* ausgelöst. Dieser Umstand ist im Folgenden nicht mehr explizit erwähnt. Wird ein *Time*-Objekt zurückgegeben, erfolgt dies im Bezugssystem der Eingabedaten.

#### Konstruktoren und Casts

Die Operationen, die zur Erzeugung der *Time*-Instanzen angeboten werden, sollten sehr umfassend sein, um eine optimale Integration in die Datenbank zu erreichen. Elementar ist die Konstruktion aus der textuellen Darstellung und umgekehrt, sowie die Transformation von und in weitere Datentypen. Insbesondere sollten solche *Casts* zu allen in der Datenbank existierenden Datumstypen umgesetzt werden (Siehe Tabelle 3-6).

---

<sup>30</sup> Der Stil dieses Abschnitts ist an den des entsprechenden Teils in der PostGIS Dokumentation angelehnt (PostGIS 2012b).

Tabelle 3-6: Konstruktoren und Casts.

<i>Time</i> → other Type	other Type → <i>Time</i>
<b>time (Text<sup>31</sup>&lt;time&gt;) : Time</b>	
Beschreibung: Erzeugt aus der Textform ein <i>Time</i> -Objekt. Beispiel: time ( Text<time> ) -> physische <i>Time</i> -Instanz erzeugt Exceptions: 'Invalid string syntax' wenn der Text syntaktisch kein Objekt repräsentiert 'Unsupported reference system key' der Bezugssystem-Schlüssel ist nicht unterstützt. 'Invalid date value' dieser Tag existiert in dem entsprechendem Kalendersystem nicht. 'Invalid temporal reference value' Zahl außerhalb des Wertebereichs des TCS 'Invalid ordinal instant' Schlüssel entspricht keiner Instanz des Ordinalsystems.	
<b>text (Time) : Text&lt;time&gt;</b>	
Beschreibung: Erzeugt die Textform aus den gespeicherten Daten. Beispiel: text (physische <i>Time</i> -Instanz) -> Text<time> Exceptions: -	
<b>date (Time) : Date</b>	<b>time (Date) : Time</b>
Beschreibung: Beispiel eines Casts, wandelt einen Datumstyp in ein <i>Time</i> -Objekt um oder vice versa. Steht beispielhaft für alle weiteren Casts. Die Implementierung richtet sich nach den in der Datenbank angebotenen Typen. Beispiel: date(CAL0012012-09-15T12:30:50) : 2012-09-15 Exceptions: 'Not supported for ordinal systems' im Falle eines Ordinalsystems 'Not supported for this type of <i>Time</i> ' wenn z.B. eine <i>MultiInstant-Time</i> in ein <i>Date</i> -Objekt gecastet werden soll.	

### Logische Operationen

Die Logischen Operationen geben einen Wahrheitswert zurück. Mit ihnen lässt sich abfragen, ob für eine oder mehrere Instanzen ein bestimmter Zustand gilt oder nicht (Siehe Tabelle 3-7).

Tabelle 3-7: Logische Operationen.

<i>Time</i> → Boolean	<i>Time</i> × <i>Time</i> → Boolean
<b>isPrimitive (Time) : Boolean</b>	
Beschreibung: Ist Wahr, wenn die Instanz vom Typ <i>Instant</i> oder <i>Period</i> ist. Analog lässt sich eine Funktion <i>isComplex</i> definieren. Für die Implementierung ist die Negation von <i>isPrimitive</i> ausreichend. Beispiel: isPrimitive (CAL0012012-09-15T12:30:50,2012-08-15T2:00:00) : FALSE Exceptions: -	
<b>isInstant (Time) : Boolean</b>	
Beschreibung: Wie <i>isPrimitive</i> , nur spezieller. Wahr wenn die Instanz ein <i>Instant</i> -Objekt ist. Auch diese Funktion steht für eine ganze Klasse von Operationen: <i>isPeriod</i> , <i>isMultiInstant</i> etc. Beispiel: isInstant (CAL0012012-09-15T12:30:50) : TRUE Exceptions: -	

<sup>31</sup> Die getroffenen Typbezeichnungen sind generisch zu sehen und meinen die entsprechenden Umsetzungen in einer Datenbank.

<b>simultaneous (Time, Time) : Boolean</b>	
Beschreibung:	Ist Wahr, wenn die Objekte einen gemeinsamen Zeitpunkt haben. Ist kommutativ und assoziativ, sollte auch für <i>Time</i> -Mengen (z.B. in Arrays) angeboten werden.
Beispiel:	simultaneous (CAL0012012-9-15T12:30:50, CAL0012012-08-15T2:00:00/2014-05-20T22:32:00) : <i>TRUE</i>
Exceptions:	-
<b>equal (Time, Time) : Boolean</b>	
Beschreibung:	Ist Wahr, wenn die Objekte identisch sind. Hier ist auch <i>equalValues</i> denkbar, eine Operation die Wahr ist, wenn die Objekte den gleichen Typ und die gleichen Zeitpunkte haben, ungeachtet in welchem Referenzsystem diese Werte angegeben sind. Ist kommutativ und assoziativ.
Beispiel:	equal (CAL0012012-09-15T12:30:50, CAL0012012-09-15T12:30:50) : <i>TRUE</i>
Exceptions:	-
<b>overlaps (Time, Time) : Boolean</b>	
Beschreibung:	Ist Wahr, wenn eine Zeitspanne existiert, zu der beide Objekte gültig sind. Ähnlich, aber strenger als <i>simultaneous</i> . Kommutative und assoziative Operation, sollte für Mengen implementiert werden.
Beispiel:	overlaps (CAL0012012-09-15T12:00:00/2012-09-16T12:00:00, CAL0012012-09-16T12:00:00/2012-09-20T12:00:00) : <i>FALSE</i>
Exceptions:	-

## Simple Zugriffe

Die simplen Zugriffe liefern Teilinformationen über die Objekte (Siehe Tabelle 3-8).

Tabelle 3-8: Simple Zugriffe.

	<i>Time</i> → <i>Time</i>	<i>Time</i> → Integer
<b>temporalBoundingBox (Time) : Time</b>		
Beschreibung:	Gibt die temporale <i>Bounding Box</i> einer oder mehrerer <i>Time</i> -Objekte als Zeitspanne zurück. Im Unterschied zu gewöhnlichen <i>Period</i> -Instanzen kann der Rückgabewert den gleichen Start- und Endzeitpunkt haben, was der temporalen <i>Bounding Box</i> eines Zeitpunktes entspricht. Der Rückgabewert wird von anderen Operationen sonst wie eine <i>Period</i> -Instanz behandelt. Diese Funktion ist sowohl kommutativ wie auch assoziativ und sollte auch für Mengen implementiert werden.	
Beispiel:	temporalBoundingBox(CAL0012012-08-15T2:00:00) : (CAL0012012-08-15T02:00:00 /2012-08-15T02:00:00)	
Exceptions:	„Reference System is ordinal“	
<b>startInstant (Time) : Time</b>		
Beschreibung:	Liefert den Startzeitpunkt einer Periode als <i>Instant</i> -Objekt. Analog sollte <i>endInstant</i> angeboten werden. Darüber hinaus ist eine Operation <i>multiStart(End)Instants</i> möglich, die alle relevanten Zeitpunkte einer <i>MultiPeriod</i> oder auch <i>RegularMultiPeriod</i> liefert. Dies könnte in einer geeigneten Form wie z.B. Arrays oder Sets erfolgen.	
Beispiel:	startInstant (CAL0012012-09-15T12:30:50/2012-09-20T11:30:50) : CAL0012012-09-15T12:30:50	
Exceptions:	‘Time-Object has to be a period‘	
<b>countPrimitives (Time) : Integer</b>		
Beschreibung:	Liefert die Anzahl der enthaltenen Primitive. Eine Operation <i>countInstants</i> gibt ähnlich die Anzahl der Zeitpunkte wieder, welche die Zeitgeometrie bilden.	
Beispiel:	countPrimitives (CAL0012012-09-15T12:30:50/2012-09-20T11:30:50) : 1	
Exceptions:	-	



## Simple Datenmanipulation

Datenmanipulationen dienen der Veränderung bestehender Instanzen. Teilweise überschneiden sie sich mit den Konstruktoren, werden aber dennoch separiert betrachtet, da sie *Time*-Instanzen in andere *Time*-Instanzen überführen (Siehe Tabelle 3-9).

Tabelle 3-9: Simple Datenmanipulation.

<i>Time</i> × <i>Time</i> → <i>Time</i>	<i>Time</i> × Text → <i>Time</i>
<b>period (<i>Time, Time</i>) : <i>Time</i></b>	
Beschreibung: Erzeugt aus zwei <i>Instant</i> - ein <i>Period</i> -Objekt. Weiter sind Funktionen möglich um MultiInstant oder MultiPeriod Objekte aus entsprechenden Primitiven zu bilden.	
Beispiel: period (CAL0012012-09-15T12:30:50, CAL0012012-09-15T15:00:00) : CAL0012012-09-15T12:30:50/2012-09-15T15:00:00	
Exceptions: 'Can only create period from two instants'	
<b>transformRefSys (<i>Time, Text</i>) : <i>Time</i></b>	
Beschreibung: Transformiert ein <i>Time</i> -Objekt in ein anderes Bezugssystem. Das Bezugssystem wird wie in Abschnitt 3.3 beschrieben angegeben.	
Beispiel: transformRefSys (CAL0012012-09-15T12:00:00,TCS001) : TCS0012456186,0	
Exceptions: 'Can not transform from an ordinal reference system' 'Can not transform to an ordinal reference system'	
<b>addDuration (<i>Time, Text</i>&lt;8601 duration&gt;) : <i>Time</i></b>	
Beschreibung: Addiert eine Dauer zu einem Zeitpunkt und liefert den resultierenden Zeitpunkt. Die Dauer sollte alternativ auch als Dezimaltag ausgedrückt werden können.	
Beispiel: addDuration (CAL0012012-09-15T12:00:00,P1DT2H) : CAL0012012-9-16T14:00:00	
Exceptions: 'Can not add duration to an ordinal reference system'	
<b>regularMultiToMulti (<i>Time</i>) : <i>Time</i></b>	
Beschreibung: Erzeugt aus einem gegebenen RegularMulti-Objekt ein einfaches Multi-Objekt mit derselben Zeitgeometrie.	
Beispiel: regularMultiToMulti (CAL001R3/2012-09-05T12:00:00/P1D) : 2012-09-05T12:00:00, 2012-09-06T12:00:00, 2012-09-07T12:00:00	
Exceptions: 'Can only create MultiPeriod/Instant with RegularMultiPeriod/Instant'	

## ISO19108-Funktionen

Die hier genannten Funktionen entsprechen den bereits vorgestellten aus dem zeitlichen Schema der ISO. Um konform zu dem Standard zu bleiben, werden diese Operationen nur auf den primitiven Typen *Instant* und *Period* definiert. Die Komplexen sollen durch spezielle Rückgabewerte berücksichtigt werden und keine *Exceptions* auslösen um einen flüssigen Zugriff auf große Datensätze zu gewähren (Siehe Tabelle 3-10).

Tabelle 3-10: ISO19108-Funktionen

<i>Time</i> × <i>Time</i> → Text	<i>Time</i> × <i>Time</i> → Integer	<i>Time</i> × <i>Time</i> → Double	<i>Time</i> → Text	<i>Time</i> → Double
<b>relativePosition (<i>Time, Time</i>) : Text&lt;relative position text&gt;</b>				
Beschreibung: Implementierung der ISO19108-Funktion <i>relativePosition</i> . Gibt die temporale Position des ersten <i>Time</i> -Objekts relativ zum Zweiten als Zeichenkette zurück. Die Ausdrücke entsprechen dem Text des <i>relativePosition</i> -Aufzählungstyps. Ist eines der <i>Time</i> -Objekt komplex, wird ‚NAP‘ zurückgegeben.				
Beispiel: relativePosition (CAL0032012-08-15T18:6:34.22,CAL0032012-09-15T12:10:0) : Before				
Exceptions: -				

<b>relativePositionInteger (Time, Time) : Integer&lt;relative position integer&gt;</b>	
Beschreibung:	Implementierung der ISO19108-Funktion <i>relativePosition</i> . Ähnlich wie <i>relativePosition</i> , mit dem Unterschied, dass der Rückgabewert durch einen Integerwert codiert ist. Komplexe lösen die Zahl -1 aus.
Beispiel:	<code>relativePositionInteger (CAL0032012-8-15T18:6:34.22,CAL0032012-9-15T18:6:34.22) : 1</code>
Exceptions:	-
<b>distance (Time, Time) : Text&lt;8601 duration&gt;</b>	
Beschreibung:	Implementierung der ISO19108-Funktion <i>distance</i> . Gibt die Zeitdauer zwischen den beiden <i>Time</i> -Objekten im erweiterten ISO8601-Format wieder. ‚NAP‘ bei Komplexen. Ist Kommutativ.
Beispiel:	<code>distance(CAL0012012-08-15,CAL0012012-09-15T12:00:00) : P1MT12H</code>
Exceptions:	‘Not supported for ordinal systems‘ im Falle eines Ordinalsystems
<b>distanceJulDay (Time, Time) : Double</b>	
Beschreibung:	Implementierung der ISO19108-Funktion <i>distance</i> . Ähnlich <i>distance</i> . Gibt die minimale Zeitdauer zwischen den Zeitgeometrien in Dezimaltagen als Double zurück. -1 bei Komplexen.
Beispiel:	<code>distanceJulDay(CAL0012012-08-15, CAL0012012-09-15T12:00:00): 31.5 [Tage]</code>
Exceptions:	‘Not supported for ordinal systems‘ im Falle eines Ordinalsystems
<b>duration (Time) : Text&lt;8601 duration&gt;</b>	
Beschreibung:	Implementierung der ISO19108-Funktion <i>duration</i> . Gibt die Dauer des <i>Time</i> -Objekts im erweiterten ISO8601-Format zurück. Bei Komplexen ‚NAP‘. Auch diese Funktion ist kommutativ.
Beispiel:	<code>duration(CAL0032012-08-15T00:00:00/2014-09-15T12:00:00) : P2Y1MT12H</code>
Exceptions:	‘Not supported for ordinal systems‘ im Falle eines Ordinalsystems
<b>durationJulDay (Time) : Double</b>	
Beschreibung:	Implementierung der ISO19108-Funktion <i>duration</i> . Gibt die Dauer des <i>Time</i> -Objekts in Dezimaltagen als Double zurück. Bei Komplexen -1.
Beispiel:	<code>durationJulDay(CAL0032012-08-15T00:00:00/2014-09-15T12:00:00) : 761.5 [Tage]</code>
Exceptions:	‘Not supported for ordinal systems‘ im Falle eines Ordinalsystems

### Erweiterte ISO19108-Funktionen

Diese Operationen weiten die Funktionen des zeitlichen Schemas auf die komplexen Typen aus (Siehe Tabelle 3-11).

Tabelle 3-11: Erweiterte ISO19108-Funktionen

	<i>Time</i> × <i>Time</i> → Text	<i>Time</i> × <i>Time</i> → Double	<i>Time</i> → Text	<i>Time</i> → Double
<b>multiDuration (Time) : Text&lt;8601 duration&gt;</b>				
Beschreibung:	Aufbauend auf der bereits genannten <i>duration</i> Funktion. Erweitert diese für MultiPeriod-Objekte und gibt die Summe der Dauern der einzelnen Perioden wieder. Ähnlich wie <i>duration</i> auch mit einer Variante, welche die Dauer in Dezimaltagen zurückgibt.			
Beispiel:	<code>multiDuration (CAL0012012-09-15T00:00:00/2012-10-15T00:00:00, 2013-11-15T00:00:00/2013-12-15T00:00:00): P2M</code>			
Exceptions:	‘Not supported for ordinal systems‘ im Falle eines Ordinalsystems			
<b>multiDistance (Time, Time) : Text&lt;8601 duration&gt;</b>				
Beschreibung:	Gibt auch für Multi-Objekte die minimale Distanz wieder. Auch hier sollte eine weitere Variante die Dauer in Dezimaltagen wiedergeben.			
Beispiel:	<code>multiDistance ((CAL0012012-01-01T00:00:00,2013-01-01T00:00:00, 2014-01-01T00:00:00), CAL0012012-02-01T00:00:00) : P1M</code>			
Exceptions:	‘Not supported for ordinal systems‘ im Falle eines Ordinalsystems			

Einer Erweiterung der Funktion *relativePosition* ist hier ausgespart. Sind Komplexe involviert, wären die zwölf Rückgabewerte nicht mehr eindeutig bestimmbar. Möglich ist eine

Berechnung mit der temporalen *Bounding Box* der komplexen Objekte. Alternativ könnte eine Hierarchie auf den Rückgabewerten definiert und derjenige mit dem höchsten Rang wiedergegeben werden.

### Operatoren

Mit Operatoren sind hier syntaktische Erweiterungen gemeint. Dies sind Zeichen die zwischen (Infix-Notation) oder vor einem Argument stehen (Präfix-Notation) und dadurch eine bestimmte Operation auslösen (GRUMM UND SOMMER 2011 S.102). Um die Arbeit mit der *Time*-Struktur zu erleichtern, sollten auch Operatoren für diese definiert werden. Diese könnten dann die Syntax der Anwendung ergänzen. Da dies ohne Vereinbarungen seinen Zweck jedoch nicht erfüllt, sind beispielhaft nur zwei Operatoren beschrieben. Diese rufen bereits bekannte Funktionen auf:

= :  $Time = Time \rightarrow equal(Time, Time) : Boolean$

- :  $Time - Time \rightarrow distanceJulDay(Time, Time) : Double$

Hier bestehen noch zahlreiche Möglichkeiten, insbesondere im Sinne von qualitativen Vergleichen, die boolesche Werte liefern. Ein Beispiel ist die ‚ist-kleiner‘-Abfrage vieler numerischer Typen. Hier ließen sich leicht Abfragen integrieren, die testen ob die relative Position zweier Instanzen einem bestimmten Zustand entspricht<sup>32</sup>.

### 3.4.2 Möglichkeiten raum-zeitlicher Analysen

Die bis hierhin gezeigten Funktionen können als Basis für weitreichendere Analysefunktionen dienen. Diese Analysefunktionen sollen es ermöglichen, bestimmte Aspekte der Dynamik der modellierten Objekte unter Nutzung der konzipierten Struktur auszuwerten. Dazu operieren sie auf der Objektebene.

#### Fortführungsbeziehungen

Die Ermittlung der Fortführungsbeziehungen der ISO19108 zwischen verschiedenen Objekten soll als ein erstes Beispiel dienen. Dabei wird angenommen, dass die Objekte einer Klasse angehören und in einer Relation mit einem Geometriotyp sowie *Time* abgelegt sind. Es existieren sich räumlich überlagernde Objekte, die dann jedoch keine überlappende zeitliche Gültigkeit aufweisen<sup>33</sup>. Um nun die Vorgänger für ein konkretes Objekt zu finden, ist die zeitliche Dimension ebenso wie die räumlichen Dimensionen zu berücksichtigen. Durch die Funktion *relativePosition* können diejenigen Objekte gefunden werden, deren Gültigkeit endete, als die des Bezugsobjekts begann. Durch die Funktionen des Geometrietyps können aus dieser Menge wiederum diejenigen selektiert werden, die das Bezugsobjekt räumlich überlagern. Umfasst die Lösungsmenge mehr als ein Objekt, besteht eine

<sup>32</sup> Bei POSTGIS 2012b sind ähnliche Operatoren für die räumliche Geometrie definiert.

<sup>33</sup> Durch einen *Trigger* könnte die Konsistenz gewährleistet werden.

$TM\_FeatureFusion$ -, andernfalls eine  $TM\_FeatureSubstitution$ -Beziehung zwischen der Menge und dem Ausgangsobjekt (Siehe Abbildung 3-6).

$I = \{ i \mid i \text{ ist Instanz} \}$	$O \subset I$	$O = \{ o \}$	Klasse I und ein Objekt
$T \subset I$	$T = \{ t \mid relativePosition(t, o) = MetBy \}$		Temporaler Aspekt
$X = T \cap O$			Räumlicher Aspekt <sup>34</sup>
Menge $X$ steht in einer $TM\_FeatureSubstitution$ oder $TM\_FeatureDivision$ -Beziehung zu $o$ .			

Abbildung 3-6: Ablauf bei der Ermittlung der ISO-Fortführungsbeziehungen.

Ähnlich lassen sich auch die Nachfolge-Objekte ermitteln und feststellen, ob es sich um eine  $TM\_FeatureDivision$ - oder eine  $TM\_FeatureFusion$ -Beziehung handelt. So ließe sich durch eine geeignete Prozedur die Historie eines Ortes voll automatisch rekonstruieren. Auf diesem Wege ließe sich beispielsweise die Anzahl der Veränderungen in einer gegebenen Zeitspanne, der Zustand zu einem bestimmten Zeitpunkt oder sogar der durchschnittliche Gradient eines thematischen Aspekts bestimmen<sup>35</sup>. Bei einem solchen iterativen Aufruf der Funktion ist zu beachten, dass immer nur der Teil der Objektgeometrie eingeht, der tatsächlich den gleichen Raum wie das Ausgangsobjekt einschließt. So lassen sich die objektorientierten Pseudofunktionen der Tabelle 3-12 definieren, auf denen weitere Abfragen aufbauen können

Tabelle 3-12: Raum-zeitliche Pseudofunktionen

<b>predecessors () : Menge von Geoobjekten</b>	
Beschreibung:	Findet alle Vorgänger des aufrufenden Objekts und gibt sie zurück, direkt oder über einen Schlüssel.
Beispiel:	Germany.predecessors (): 'Germany', 'GDR'
<b>successors () : Menge von Geoobjekten</b>	
Beschreibung:	Findet analog zu predecessors die Nachfolger des Objekts.
Beispiel:	Rainforest.successors(): 'Soybean field'

Durch weitere Prozeduren ließen sich dann die Objektfortführungsbeziehungen der ISO19108 feststellen und die für den iterativen Aufruf notwendigen geometrischen Prozesse durchführen.

### Räumlich-zeitliche Abfragen

Die Selektionen zur Ermittlung der Fortführungsbeziehungen basieren auf der Vorgabe bestimmter Kriterien, welche die Objekte in der geometrischen und temporalen Dimension erfüllen müssen. Durch die Formulierung weiterer Bedingungen könnten analog

<sup>34</sup> Der Schnittmengenoperator soll hier einer räumlichen *Intersection* entsprechen. Die Reihenfolge der temporalen und räumlichen Operationen ist logisch beliebig, der Beginn mit der Zeitdimension ist vermutlich jedoch performanter.

<sup>35</sup> So könnte die durchschnittliche Wahlbeteiligung eines Ortes auch über wechselnde Verwaltungsgrenzen bestimmt werden. Eine weitere Möglichkeit wäre die Bestimmung eines Durchschnittswertes für eine Analysezelle aus variierenden Isolinien-Polygonen.

verschiedene sinnvolle Teilmengen aus einer oder mehreren Klassen abgefragt werden. WORBOYS UND DUCKHAM kategorisieren die Abfragen nach den einbezogenen Objektdimensionen und unterscheiden hier die geometrische, temporale und thematische Dimension. Eine weitere Differenzierung beruht auf dem Unterschied zwischen punktuellen und einen Abschnitt einbeziehenden Anfragen. Sie formulieren drei einfache Fragen, die eine Geotemporaldatenbank beantworten können sollte:

- *What was the state of the world or database at a particular location and given time?*
- *Where was the world or database in a particular state at a given time?*
- *When was the world or database in a particular state at a given location?* (WORBOYS UND DUCKHAM 2004 S.380).

Hier werden Festsetzungen in zwei Dimensionen getroffen und der Zustand in der übrigen Dimension ist gesucht. Durch die oben konzipierten Funktionen ließen sich die temporalen Bedingungen in geeigneter Weise formulieren. Beispielsweise kann durch *simultaneous* leicht ermittelt werden, ob zwei Zeitgeometrien gleichzeitig sind, unabhängig davon ob es sich um Zeitpunkte, Zeitspannen oder komplexere Formen handelt. So könnten, aufbauend auf den konzipierten Funktionen, die ersten beiden Fragen beantwortet werden. Die dritte Frage ließe sich durch die Übermittlung der *Time*-Instanz des gefundenen Objekts beantworten. Erfüllen mehrere Objekte die thematischen und räumlichen Bedingungen, könnten die temporale *Bounding Box*, ein Multiobjekt oder die einzelnen *time*-Instanzen elementar übergeben werden.

### Analyse raum-zeitlicher Ereignisse

Die allgemeinen Aussagen zu den räumlich-zeitlichen Abfragen sollen hier an einem Beispiel für einen Anwendungsfall spezialisiert werden. So könnte die Analyse eines räumlich und zeitlich punktuell modellierten dynamischen Phänomens gewünscht sein. Dies könnte z.B. ein bestimmtes Wetterereignis sein oder ein Kriminaldelikt. Dabei wird davon ausgegangen, dass jedes Auftreten des Ereignisses einer Instanz in einer entsprechenden Klasse entspricht. Hier können dann verschiedene Informationen von Interesse für die Anwender sein. Exemplarisch könnte die Frage nach der maximalen Häufung der Ereignisse bezogen auf ein konkretes Intervall an einem bestimmten Ort sein. Also der Form:

- In welchem Kalendermonat gab es die meisten Unwetter?
- An welchem Wochentag ereignen sich die meisten Raubüberfälle?

Um diese Fragen zu beantworten, könnten Prozeduren aus der temporalen *Bounding Box* der betreffenden Objekte und dem gegebenen Intervall ein Histogramm erstellen und die gesuchten Maxima hier abgreifen. Häufig wird es dabei zweckmäßig sein, für die Histogrammbildung einen zyklischen Zeitverlauf anzunehmen. Die beiden Beispielfragen zeigen dies. Durch die Einbindung räumlicher Verschneidungen mit einer Bezugsgeometrie, z.B. Verwaltungsbezirken, könnte dies separat für jedes dieser Bezugsobjekte erfolgen. So könnte auch ermittelt werden, wo solche Maxima liegen. Daneben wären noch weitere Bedingungen mit einzubeziehen. Beispielsweise sollte es möglich sein, neben dem

Histogrammintervall auch eine Zeitspanne als Zeitraum der Auswertung festzulegen. Dann könnte der Monat mit den meisten Unwettern für jede verfügbare Jahresdekade ermittelt werden. Die Bedingungen könnten vermutlich weiter parametrisiert werden und so auch universellere Funktionen konzipiert werden.

## 4. Beschreibung der Implementierung

In diesem Abschnitt erfolgt die Beschreibung der prototypischen Implementierung der obigen Konzeption. Als Muster für die Umsetzung wurde das objektrelationale DBMS PostgreSQL mit PostGIS Erweiterung gewählt. Im Folgenden wird zunächst auf die gewählte Möglichkeit der Einbindung eines abstrakten Datentyps in PostgreSQL eingegangen. Dem schließt sich eine Beschreibung des physischen Entwurfs der Datenstruktur an. Danach wird erläutert, welche Referenzsysteme und *Time*-Typen umgesetzt wurden und wie diese durch die Struktur repräsentiert werden. Darauf aufbauend werden die wichtigsten umgesetzten Funktionen dargestellt. Am Ende dieses Abschnittes wird die erfolgte Bearbeitung von Testdaten beschrieben.

PostgreSQL setzt strikt den ANSI-SQL:2008-Standard um und PostGIS implementiert die relevanten OGC-Standards, was sich auch in der Verifizierung durch die OSGeo zeigt. Darüber hinaus ist der strikte Open Source Charakter beider Projekte zu betonen, sowie die zahlreichen Möglichkeiten, die PostgreSQL zur Anpassung und Erweiterung des Systems bietet (POSTGRESQL 2012a; POSTGIS 2012c).

### 4.1 Abstrakte Datentypen in PostgreSQL

PostgreSQL bietet verschiedene Wege an, die vorhandenen Datentypen um eine eigene Struktur zu ergänzen. Diese basieren auf dem Konzept der abstrakten Datentypen. Gemeinsam ist den Möglichkeiten, dass die gewünschte Struktur über einen *CREATE TYPE* Befehl der Datenbank bekannt gemacht wird<sup>36</sup> (POSTGRESQL 2012b). Für eine adäquate Umsetzung des Konzeptes ist die Option am besten geeignet, bei der ein benutzerdefinierter Basisdatentyp mit variabler Länge definiert wird. Eine Übersicht über die weniger geeigneten Alternativen in PostgreSQL ist im Anhang B enthalten. Die physische Struktur der Daten wird bei dieser Variante durch extern definierte Funktionen bestimmt, gewöhnlich in der Programmiersprache C.

---

<sup>36</sup> Dieser ist Teil des SQL3 Standards (oder auch SQL:1999), hier wurden Konzepte der Objektorientierung auf Relationale Datenbanken übertragen. Es wird dann von Objektrelationalen oder erweiterten relationalen DBMS gesprochen. Im Zusammenhang mit benutzerdefinierten Typen (oder ADTs) gilt das Prinzip der Kapsellung. Die Funktionen auf dem Typ werden in dem DBMS durch ein *Interface* bereitgestellt und die genauen Abläufe bleiben verborgen. Die Funktionen können dann auch extern definiert und lediglich eingebunden werden (ELMASRI UND NAVATHE 2002 S.484-487).

“[...] *Creating a new base type requires implementing functions to operate on the type in a low-level language, usually C.*” (POSTGRESQL 2012c).

Obligatorisch sind dabei zwei Funktionen, eine die eine eingegebene Zeichenkette in die interne Struktur umwandelt und eine zweite, die aus der internen Struktur wieder eine Zeichenkette erzeugt (Siehe Abbildung 4-1). Die Zeichenkette ist dabei als die Nutzersicht der Daten zu sehen, die interne Struktur definiert, wie die Daten im Speicher abgelegt werden. Die Operationen, welche die genannten Funktionen durchführen, sollten dabei verständlicherweise invers zueinander sein. Über ein bloßes Datenspeichern geht die Funktionalität, für einen so neu geschaffenen Datentyp, zunächst nicht hinaus. Alle gewünschten Funktionen, Typumwandlungen, Operatoren etc. müssen explizit definiert werden. Diese werden dann entweder über die *CREATE TYPE* oder weitere Anweisungen in der Datenbank mit dem Datentyp verknüpft. Voraussetzung hierfür ist die Einbindung der C-Funktionen in PostgreSQL in Form einer dynamischen Bibliothek (POSTGRESQL 2012c).

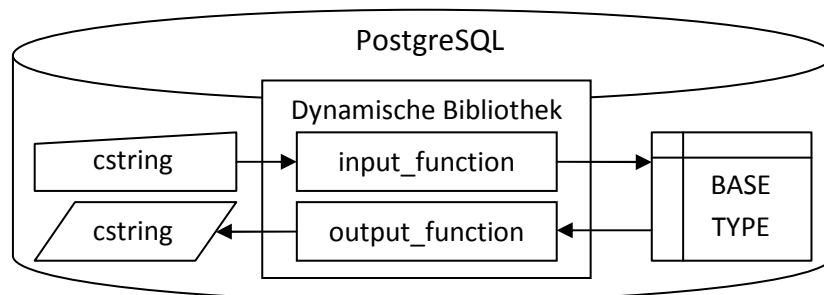


Abbildung 4-1: Einbindung Abstrakter Datentypen in PostgreSQL.

#### 4.1.1 Physischer Entwurf

Die interne Struktur der Daten entspricht üblicherweise einer C-Struktur. Für einen Typ variabler Größe muss das erste Feld dabei exakt vier Bytes lang sein, was durch einen *Character Array* mit vier Elementen umgesetzt werden sollte. In diesem Feld wird die Länge der Instanz gespeichert. Es sollte nicht direkt manipuliert werden, sondern nur über dafür vorgesehene interne PostgreSQL-Funktionen (POSTGRESQL 2012c). Neben dem vorgegebenen Feld für die Länge der Instanz müssen noch weitere Informationen in die Struktur abgebildet werden. Dies sind das externe zeitliche Referenzsystem, die Geometrieform der Zeit und die diese bildenden konkreten Zeitpunkte im internen Referenzsystem, also dem Julianischem Tag.

Die unterschiedlichen Referenzsysteme, wie auch die Geometrieformen, sollen dabei intern bestimmten natürlichen Zahlen zugewiesen werden. Hierfür werden zwei entsprechende Attribute vorgesehen, die jeweils eine Größe von einem Byte haben sollen, so dass hier 256 verschiedene Referenzsysteme und Geometrieformen unterschieden werden können.

Insbesondere bei den Geometrieformen wird diese hohe Anzahl verschiedener Zustände nicht erforderlich sein<sup>37</sup>. Für den Prototyp ist sie auch für die Referenzsysteme ausreichend. In der Konzeption wurde vorgegeben, dass die Zeitpositionen, unabhängig vom Bezugssystem, intern als Julianisches Datum gespeichert werden. Dafür soll als Kompromiss aus Genauigkeit und Speicherbedarf, ein *Double* verwendet werden, der 64Bit belegt<sup>38</sup> (KLIMA UND SELBERHERR 2007 S.54). Das Strukturfeld ist dabei als Array mit einem Element angelegt. Wenn mehr als eine Zeitposition gespeichert wird, werden entsprechend viele Elemente angehängt und die resultierende gesamte Größe der Struktur, in dem dafür vorgesehenen ersten Feld der Struktur, gespeichert. Der physische Entwurf des Datentyps ist in Abbildung 4-2 abgebildet. Der Datentyp trägt in der Umsetzung den Arbeitstitel *geotime*.

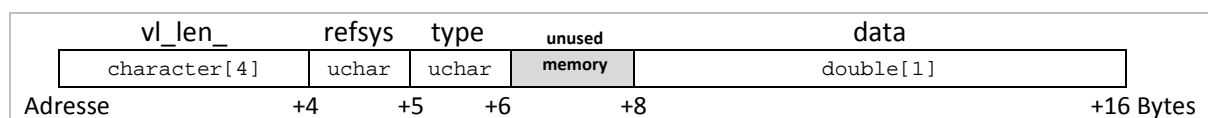


Abbildung 4-2: Die interne Struktur.

## 4.2 Umgesetzte Referenzsysteme und Typen

In diesem Abschnitt wird erläutert, welche Referenzsysteme und *Time*-Objekte in der prototypischen Umsetzung zur Verfügung stehen.

### 4.2.1 Referenzsysteme

Es wurden drei Kalendersysteme sowie je ein Beispiel zu den temporalen Koordinatensystemen und den Ordinalsystemen in die Implementierung integriert. Der Bezug zu einem anderen System als dem Gregorianischen Kalender kann wie in Abschnitt 3.3 konzipiert angegeben werden. Die jeweiligen Schlüssel und internen Werte im refsys-Feld sind in Tabelle 4-1 aufgelistet. Die Umsetzung ist so angelegt, dass sie um weitere Systeme ergänzt werden kann. Die betreffenden Anforderungen werden knapp mit jeder Art der Referenzsysteme genannt.

<sup>37</sup> Die Verwendung von Bitfeldern oder Bit-Operatoren würde jedoch keine Speicherplatzeinsparung bringen, da der nachfolgende Gleitkommawert im Speicher ausgerichtet wird und somit ohnehin ein nicht genutzter Speicherbereich enthalten ist. Ein schließen dieser Lücken ist möglich, würde aber zu Lasten des Laufzeitverhaltens gehen und wird daher nicht weiter in Betracht gezogen (WOLF 2010 S.287).

<sup>38</sup> Die Genauigkeit der Werte ist für die Gegenwart – in Sekunden ausgedrückt – präziser als der Millisekunden-Bereich. Dies erscheint für die möglichen Anwendungen als ausreichend. Mit zunehmendem Abstand vom Nullpunkt des Julianischen Datums sinkt die Genauigkeit.



Tabelle 4-1: Implementierte Referenzsysteme.

System	Art	Schlüssel	refsys Wert
Gregorianische Kalender und UT	CAL	GRE, gre, 001	2
Julianische und Gregorianische Kalender mit UT	CAL	JUG, jug, 002	0
Julianische Kalender mit UT	CAL	JUL, jul, 003	1
Julianisches Datum	TCS	JDA, jda, 001	100
Geologische Epochen	ORD	GEO, geo, 001	200

### Kalendersysteme

Das Standardsystem ist der Gregorianische Kalender. Wird kein System angegeben, werden die Zeitpositionen auf dieses System bezogen. Es können auch Zeitpositionen vor der tatsächlichen Einführung dieses Kalenders referenziert werden. Ein weiteres Kalendersystem ist der Julianische Kalender, der für historische Daten ebenfalls eine globale Bedeutung hat. In zahlreichen Staaten<sup>39</sup> wurde dieses System erst Anfang des 20. Jahrhunderts vom moderneren Gregorianischen Kalender abgelöst. Das dritte System vereint die beiden genannten in der Weise, dass auf den 4. Oktober 1582 direkt der 15. Oktober 1582 folgt. Bis zu dieser Zeitspanne gelten die Regeln des Julianischen, danach die des Gregorianischen Kalenders. Dies entspricht der historischen Umstellung in zahlreichen europäischen Ländern. Daher lassen sich hier entsprechende Daten korrekt angeben, ohne dass manuelle Transformationen notwendig sind.

Die Gültigkeit eines Kalenderdatums wird entsprechend den jeweiligen Schaltjahresregeln geprüft. Die Angabe von Zeitpositionen vor dem Ursprung des Julianischen Datums ist mit den Kalendersystemen nicht möglich. Die verwendeten Transformationsalgorithmen sind in diesem Bereich nicht korrekt. Die Kalender sind hier zunächst einheitlich mit dem UT-Zeitsystem implementiert worden. Es wird also eine einheitliche Tageslänge von 86.400 Sekunden unterstellt, Schaltsekunden werden nicht berücksichtigt.

Die Erweiterung um zusätzliche Kalendersysteme ist möglich, erfordert jedoch einen Algorithmus, um die Daten in das Julianische Datum zu transformieren und umgekehrt. Für zahlreiche, auch historische, Kalendersysteme existieren solche Algorithmen.

### Zeitliche Koordinatensysteme

Als temporales Koordinatensystem steht das intern ohnehin verwendete Julianische Datum auch direkt auf der Nutzerebene zur Verfügung. Die Erweiterung um weitere zeitliche Koordinatensysteme ist auf einfache Art möglich. Die Transformation zwischen einem solchen und dem Julianischem Datum könnte lediglich eine Verschiebung des Ursprungs sowie einen Maßstabsfaktor erfordern.

---

<sup>39</sup> Populäre Beispiele sind die Gebiete Chinas, der ehemaligen Sowjetunion, die Türkei und Griechenland (RICHARDS 1998 S.248f).

## Ordinalsysteme

Beispielhaft für die Ordinalsysteme wurden die geologischen Epochen umgesetzt, wie sie in der ISO19108 angegeben sind (Iso 2002a S.29). Die Ergänzung um weitere Ordinalsysteme erfordert lediglich eine geordnete Auflistung der Bezeichnungen der *TM\_OrdinalEra*-Objekte.

### 4.2.2 Time-Typen

In die Implementierung wurden die in Teil 3. konzipierten Typen integriert, mit Ausnahme von den unterminierten regelmäßigen Multiobjekten. Zeitpunkte und Zeitspannen werden wie vorgegeben gebildet und die Information über den konkreten Typ im *type* Feld abgelegt. Die Gestalt der Kompositionsobjekte ergibt sich ebenfalls aus der Konzeption, diese Instanzen haben einen eigenen *type* Wert. Da davon ausgegangen wird, dass die Typen *Instant* oder *Period* besonders häufig genutzt werden, sind sie als separate Objekte implementiert. So wird eine Bestimmung der Anzahl der enthaltenen Primitive durch Algorithmen in weiteren Funktionen überflüssig, wodurch sich ein besseres Laufzeitverhalten erreichen lässt.

Für die sich regelmäßig wiederholenden *Instant*-Objekte sind Informationen über den Beginn, die Kardinalität und die Dauer zwischen der Gültigkeit zu speichern. Für entsprechende *Period*-Objekte ist zusätzlich die Dauer der Gültigkeit zu erfassen. Wie bereits erläutert, ist es erforderlich, eine Kalendersystem-Dauer in den Kalendereinheiten zu halten. Daraus ergeben sich für eine Dauer im Allgemeinen fünf ganzzahlige Werte sowie ein Gleitkommawert<sup>40</sup>. Um nicht auch für die ganzzahligen Werte den Speicher eines *Double* aufzuwenden, werden diese Werte in einem Speicherbereich mit der Größe von zwei *Double* zusammengefasst und in dieser Form gespeichert. Zusammen mit dem Sekundenwert ergeben sich so drei *Double* für eine Kalenderdauer<sup>41</sup>. Bei einem Zugriff werden die Werte entsprechend rekonstruiert. Die Kardinalität wird zusammen mit der ersten Dauer gespeichert. So ergibt sich für regelmäßige *Instant*-Objekte eine Größe von vier *Double*-Werten und für *Periods* eine Größe von entsprechend sieben Werten. Bezogen auf ein temporales Koordinatensystem kann auch eine Dauer durch einen einzelnen Gleitkommawert angegeben werden. Zusammen mit der Kardinalität und dem Zeitpunkt des Beginns ergibt sich eine Größe von drei (*Instant*) bzw. vier (*Period*) Elementen im *data*-Feld. Solche Objekte sollen in einem Ordinalsystem hier nicht möglich sein. In der Tabelle 4-2 sind die umgesetzten Objekte noch einmal zusammengefasst.

---

<sup>40</sup>Ganzzahlig sind Jahr, Monat, Tag, Stunde, Minute. Die Sekunde soll auch mit Dezimalstellen angegeben werden können.

<sup>41</sup>Alternativ könnten die terminierten regelmäßigen Objekte zur Speicherung auch in gewöhnliche Multiobjekte überführt werden. Hier ist Laufzeitverhalten gegen Speicherverbrauch abzuwägen. Der Vorteil der erfolgten Art und Weise liegt in der direkten Übertragbarkeit auf die unterminierten Typen. Diese könnten in gleicher Weise gespeichert werden und bei einem Zugriff wäre dann nur noch die momentane Kardinalität festzustellen.

Tabelle 4-2: Implementierte Typen.

Typ	type Wert	Elemente in data
TM_Instant	1	1
TM_Period	2	2
MultiInstant	3	Anzahl Primitive
MultiPeriod	4	Anzahl Primitive · 2
RegularMultiInstant - <i>limited</i>	5	(CAL-> 4) (TCS-> 3)
RegularMultiPeriod - <i>limited</i>	6	(CAL-> 7) (TCS-> 4)

### 4.3 Umgesetzte Funktionen

In diesem Teil wird auf die wichtigsten umgesetzten Funktionen eingegangen. Prinzipiell muss zwischen den in der dynamischen Bibliothek implementierten Prozeduren und der Sicht aus der Datenbank auf diese unterschieden werden. Hier steht die Datenbanksicht im Vordergrund. Vereinzelt wird auf die internen Abläufe eingegangen. Darüber hinaus wurden Prototypen für die in Abschnitt 3.4.2 aufgezeigten möglichen raum-zeitlichen Analysen umgesetzt, welche am Ende des Abschnitts erläutert werden. Zum Zwecke der Veranschaulichung wird vereinzelt die Datensicht in pgAdmin3 abgebildet, einer grafischen Oberfläche für PostgreSQL. Weitere umgesetzte Funktionen, die in diesem Teil nicht enthalten sind, werden im Anhang C beschrieben.

#### 4.3.1 Basisfunktionen

Diese Funktionen werden im DBMS lediglich deklariert und sind vollständig in der Programmiersprache C implementiert. Die Quelltextfragmente in Abbildung 4-3 sollen dies Prinzip der externen Implementierung veranschaulichen. Das Beispiel zeigt eine mögliche Funktion, die zwei *Integer*-Werte addiert und das Ergebnis zurückgibt.

<pre>CREATE FUNCTION integer_plus_integer(integer, integer)   RETURNS integer   AS 'DIRECTORY/example.so'   LANGUAGE C STRICT;</pre>	<p>Deklaration in PostgreSQL <i>Erweiterter SQL-Syntax</i></p>
<pre>Datum integer_plus_integer(PG_FUNCTION_ARGS); /* C-Deklaration */  PG_FUNCTION_INFO_V1(integer_plus_integer); /* PostgreSQL Macro (notwendig für SQL-Einbindung) */ Datum integer_plus_integer(PG_FUNCTION_ARGS) { /* Implementierung */   int int_arg1 = PG_GETARG_INT32(0);   int int_arg2 = PG_GETARG_INT32(1);   int int_return;   int_return = int_arg1 + int_arg2;   PG_RETURN_INT32(int_return); }</pre>	<p>Implementierung in C <i>C-Syntax mit PostgreSQL Macros</i> Wird kompiliert in <b>example.so</b> * möglicher Speicherüberlauf ignoriert</p>

Abbildung 4-3: Eine simple Funktion als externe C-Funktion in PostgreSQL.

Grundlegend für die Verwendung der Struktur sind die obligatorischen Funktionen zur Erzeugung einer Instanz aus der textuellen Darstellung und umgekehrt (Siehe Tabelle 4-3).

Tabelle 4-3: Umgesetzte Funktionen – Ein- und Ausgabe des ADT *geotime*.

<b>Funktion:</b>	<b><i>geotime_in</i>(cstring) : geotime</b>	<b><i>geotime_out</i>(geotime) : cstring</b>
<b>Implementiert:</b>	Time(Text) : Time	Text(Time) : Text
<b>Beschreibung:</b> Die Funktion <i>geotime_in</i> implementiert den Konstruktor für die <i>geotime</i> -Objekte. Die gegebene Zeichenkette wird analysiert und das Bezugssystem, der Geometrietyt, sowie die angegebenen Zeitpositionen werden extrahiert. Dabei wird syntaktische und semantische Korrektheit sichergestellt. Die Informationen werden, wie oben angegeben, in der Datenstruktur abgelegt. Die Zeichenkette wird dabei nur einmal kopiert und folgend nur durch Zeiger manipuliert. So konnte das notwendige günstige Laufzeitverhalten erreicht werden. Die Funktion <i>geotime_out</i> produziert die konzipierte textuelle Darstellung aus den Daten einer <i>geotime</i> -Instanz.		

Durch die Implementierung der Import- und Export-Funktion und deren Deklaration in PostgreSQL kann auch der ADT *geotime* eingebunden werden. Dies erfolgt über die Anweisung in Abbildung 4-4:

```
CREATE TYPE geotime (
    internallength = VARIABLE, -- Typ variabler Länge
    input = geotime_in,       -- Eingabefunktion
    output = geotime_out,     -- Ausgabefunktion
    alignment = double );    -- Wird im Speicher wie Double ausgerichtet
```

Abbildung 4-4: PostgreSQL-Anweisung – Deklaration des ADT *geotime*

Von besonderer Bedeutung für eine Analyse der Dynamik sind die durch die ISO19108 vorgegebenen Funktionen. Durch die Tabelle 4-4 werden die Funktionen zur Bestimmung der relativen Position zweier Primitive beschrieben.

Tabelle 4-4: ISO19108-Funktionen – Relative Position zweier Primitive.

<b>Funktion:</b>	<b><i>tm_relative_position</i>(geotime, geotime) : cstring</b>
	<b><i>tm_relative_position_integer</i>(geotime, geotime) : integer</b>
<b>Implementiert:</b>	<i>relativePosition</i> (Time, Time) : Text<relative position text> <i>relativePositionInteger</i> (Time, Time) : Integer<relative position integer>
<b>Beschreibung:</b> Durch <i>tm_releative_position</i> wird wie vorgegeben die Position des ersten <i>geotime</i> -Objekts relativ zum zweiten Objekt ermittelt. Die Prozedur basiert auf qualitativen Vergleichen der Werte des <i>data</i> -Arrays der jeweiligen Instanzen. Die Art und Weise richtet sich nach dem <i>type</i> -Feld der Objekte. Zurückgegeben wird die passende Zeichenkette aus der Konzeption. Auf denselben internen Prozeduren basiert die Funktion <i>tm_relative_position_integer</i> , welche jedoch den entsprechenden Zahlenwert des konzipierten Aufzählungstyps zurückgibt.	

Die Abbildung 4-5 zeigt Beispiele für die *relativePosition*-Funktionen. Die dritte und vierte Spalte der Relation wurden durch diese Funktionen aus den Werten der ersten beiden ermittelt.

timeobject1 geotime	timeobject2 geotime	string text	code integer
CALGRE002012-09-03T16:33:22.212Z	CALGRE002012-09-17T12:00:00.000Z		
CALGRE002009-07-01T00:00:00.000Z/002009-10-01T00:00:00.000Z	CALGRE002009-10-01T00:00:00.000Z/002012-10-01T00:00:00.000Z	Before	0
CALJUG001582-10-04T00:00:00.000Z/002012-09-03T16:33:22.212Z	CALJUG001582-10-15T00:00:00.000Z	Meets	1
TCSJDA002456157.075530	TCSJDA002456156.090000/002456157.075530	Contains	6
ORDGEOCretaceous	ORDGEOPaleocene	Ends	9
ORDGEOTriassic	ORDGEOPaleocene	MetBy	11
CALGRE002012-05-14T19:27:22.200Z	CALGRE002012-07-14T19:27:22.200Z	After	12
		Before	0

Abbildung 4-5: PgAdmin3 – Die *relativePosition*-Funktionen

Die Funktionen in Tabelle 4-5 sind die umgesetzten *distance*-Funktionen der ISO19108.

Tabelle 4-5: ISO19108-Funktionen – Zeitlicher Abstand zweier Primitive.

<b>Funktion:</b>	<b>tm_distance_string</b> (geotime, geotime) : cstring <b>tm_distance_julday</b> (geotime, geotime) : double precision
<b>Implementiert:</b>	distance (Time t1, Time t2) : Text<8601 duration> distanceJulDay (Time t1, Time t2) : Double
<b>Beschreibung:</b>	Diese Funktionen ermitteln den geringsten Abstand zwischen zwei Zeitgeometrien. Von <i>tm_distance_string</i> wird das Ergebnis als kalenderbezogene Dauer im ISO8601-Format zurück gegeben. Die Funktion nutzt die relative Position der Objekte und entscheidet logisch, aus welchen <i>data</i> -Elementen die kürzeste zeitliche Distanz zu berechnen ist. Die Differenz wird hier aus den Kalenderdaten der Objekte gebildet. Die verwandte Funktion <i>tm_distance_julday</i> gibt unmittelbar die Differenz aus den <i>data</i> -Elementen zurück.

Die Abbildung 4-6 soll der Veranschaulichung der *distance*-Funktionen dienen. Durch diese wurden die Werte der dritten und vierten Spalte aus denen der ersten beiden berechnet.

timeobject1 geotime	timeobject2 geotime	distance_value double precision	distance_iso8601_string text
CALGRE002012-09-03T16:33:22.212Z	CALGRE002012-09-17T12:00:00.000Z		
CALGRE002009-07-01T00:00:00.000Z/002009-10-01T00:00:00.000Z	CALGRE002009-10-01T00:00:00.000Z/002012-10-01T00:00:00.000Z	13.8101595831104	P13DT19H26M37.788S
		0	POS

Abbildung 4-6: PgAdmin3 – Die *distance*-Funktionen

Durch die ISO19108-Funktionen lassen sich verschiedene temporale Bedingungen in PostgreSQL-Abfragen in geeigneter Weise formulieren.

In PostgreSQL besteht die Möglichkeit Funktionen zu überladen. Dadurch kann eine Information vom Nutzer in gleicher Weise für verschiedene Eingabeparameter abgefragt werden. Ein Beispiel soll hier die Funktion *temporal\_bbox* sein (Siehe Tabelle 4-6).

Tabelle 4-6: Überladene Funktion – Temporale Bounding Box.

<b>Funktion:</b>	<b>temporal_bbox</b> (geotime, geotime) : geotime <b>temporal_bbox</b> (geotime[]) : geotime <b>temporal_bbox</b> (geotime set) : geotime	Zwei Primitive geotime-Array geotime-Set
<b>Implementiert:</b>	temporalBoundingBox (Time) : Time	
<b>Beschreibung:</b>	Diese Funktion sucht die minimalen und maximalen Zeitpunkte der <i>geotime</i> -Objekte und gibt sie gemäß dem <i>Bounding Box</i> -Konzept als <i>Period</i> zurück.	

Je nach Eingabeparametern werden hier unterschiedliche interne Prozeduren aufgerufen. So kann die temporale *Bounding Box* nun für zwei Instanzen, die Elemente eines Arrays oder umfassend für alle Werte eines *geotime*-Attributs einer Relation einheitlich abgefragt werden.

Diese und die weiteren, im Anhang C aufgeführten Funktionen, dienen der Erfassung, Verwaltung und Analyse der verschiedenen konzipierten Zeitobjekte. Wird die zeitliche Gültigkeit von Geoobjekten durch ein Attribut vom Typ *geotime* modelliert, lassen sich durch diese Funktionen verschiedene Aspekte der Dynamik der abstrahierten Phänomene analysieren.

### 4.3.2 Aufbauende PostgreSQL-Funktionen

Aufbauend auf den oben beschriebenen Basisfunktionen wurden Prototypen für räumlich-zeitliche Analysen umgesetzt. Die Implementierung dieser Funktionen erfolgte in der PostgreSQL-Programmiersprache PL/pgSQL. Diese baut auf SQL auf und erweitert diese um verschiedene Aspekte. Insbesondere werden die üblichen Kontrollstrukturen angeboten (POSTGRESQL 2012d).

Als Testdaten für diese Funktionen dienten verschiedene, unter öffentlichen Lizenzen stehende, Datensätze im *Shapefile*-Format. Diese sind in der Tabelle 4-7 charakterisiert und in Abbildung 4-7 ist die Geometrie der Objekte dargestellt.

Tabelle 4-7: Charakterisierung der verwendeten Testdaten

<i>Shapefile</i>	Beschreibung	Quelle
cshape.shp	<b>244 Objekte.</b> Enthält historische und aktuelle Staatsgebiete seit dem Ende des zweiten Weltkriegs. Die Geometrie ist vom Typ Multipolygon, die <i>valid time</i> durch Start und Endzeitpunkt gegeben.	WEIDEMANN, N. B. 2010
statesp020.shp	<b>53 Objekte.</b> Statische Karte. Enthält die 50 Bundesstaaten der USA, Washington D.C., Puerto Rico und den zu den USA gehörigen Teil der Virgin Islands. Geometrie ist vom Typ Polygon.	NATIONALATLAS.GOV 2012
tornadx020.shp	<b>52.007 Objekte.</b> Enthält Tornados die sich zwischen 1950 und 2008 auf amerikanischem Staatsgebiet ereigneten. Diese sind als zeitlich und räumlich punktuelle Objekte modelliert.	
airprtx010g.shp	<b>940 Objekte.</b> Enthält die auf amerikanischem Staatsgebiet gebauten öffentlichen Flughäfen, Wasserflughäfen sowie Helikopter-Landeplätze mit mehr als 250Passagieren pro Jahr. Es ist jeweils der Eröffnungstag der Einrichtung angegeben. Ähnlich wie tornadx020.shp zeitlich und räumlich punktuell Modell.	



Abbildung 4-7: Geometrie der verwendeten Testdaten

Unter Verwendung von PostGIS-Funktionen wurden die Datensätze in eine PostgreSQL Datenbank importiert. Hier ließen sich die Zeitangaben durch simple SQL-Anweisungen in *geotime*-Attribute abbilden. Die Details werden entsprechend erläutert werden.

In Tabelle 4-8 sind die Prototypen für die Funktionen zur Bestimmung der Fortführungsbeziehungen erläutert.

Tabelle 4-8: Raum-zeitliche Funktionen – Fortführungsbeziehungen

Funktion:	<b>get_predecessor</b> (integer, tablename) : SETOF integer <b>get_successor</b> (integer, tablename) : SETOF integer
Implementiert:	predecessors () : Menge von Geoobjekten successors () : Menge von Geoobjekten
Beschreibung:	Die Funktion <i>get_predecessor</i> findet wie konzipiert die Vorgänger des Objektes mit der ID <i>integer</i> aus der Relation <i>tablename</i> . Es werden die IDs der entsprechenden Objekte zurück gegeben. Analog gibt die Funktion <i>get_successor</i> die Nachfolger des Objektes zurück. Um die gesuchten Objekte zu bestimmten, Kombinieren diese Funktionen <i>tm_relative_position_integer</i> mit verschiedenen PostGIS-Funktionen.

Anhand des CShape-Datensatzes ließ sich die Funktionalität von *get\_predecessor* und *get\_successor* testen. Die zeitliche Gültigkeit, gegeben durch Start- und Endzeitpunkt, konnte durch die konzipierten *Period*-Objekte elementar modelliert werden. Die Abbildung 4-8 enthält einen Ausschnitt aus einer erstellten PostgreSQL-Sicht. Diese Sicht ruft für alle Objekte des Datensatzes die beiden Funktionen auf. Zur Anschauung wurden nicht unmittelbar die zurückgegebenen Zahlenwerte in die Sicht eingefügt, sondern die zugehörigen Ländernamen abgegriffen und in einem entsprechenden Array ausgegeben.

	id integer	country_name character varying	predecessors character varying(00)[]	successor character varying(00)[]	validtime geotime
7	184	Congo	{}	{}	CALJUG001960-08-17
8	189	Canada	{Canada}	{}	CALJUG001948-06-3
9	190	Germany	{"Germany Federal Republic", "Germany Democratic Republic"}	{}	CALJUG001990-10-0
10	195	Yugoslavia	{}	{Croatia, Slovenia, Yugoslavia}	CALJUG001945-12-3
11	197	Egypt	{Egypt}	{Egypt}	CALJUG001958-01-2
12	199	China	{China}	{}	CALJUG001949-11-1
13	200	Pakistan	{Pakistan}	{Pakistan}	CALJUG001949-12-3

Abbildung 4-8: PgAdmin3 – Vorgänger und Nachfolger verschiedener Staaten - Ausschnitt.

Umfasst die Ergebnismenge nur ein Objekt, liegt eine *TM\_Substitution* vor und die Veränderungen betreffen nur die thematische Dimension. Wird die leere Menge zurückgegeben, sind für das Objekt keine Nachfolger bzw. Vorgänger vorhanden.

Die in Tabelle 4-9 beschriebenen Funktionen dienen der Untersuchung der Realisierbarkeit der Konzepte zur Analyse raum-zeitlicher Ereignisse.

Tabelle 4-9: Raum-zeitliche Funktionen – Bezugsobjekt mit den häufigsten Ereignissen

<b>Funktion:</b>	<b>when_maximum</b> (integer, tablename) : TABLE(geotime, bigint, geometry) <b>when_where_maximum</b> (integer, tablename1, tablename2) : TABLE(geotime, character varying[255], bigint)
<b>Implementiert:</b>	-
<b>Beschreibung:</b>	Diesen Funktionen werden eine Ganzzahl und der Name einer Relation übergeben. Die Relation enthält zeitlich und räumlich punktuelle Ereignisse. Gemeinsam ist beiden Funktionen, dass sie die zeitliche <i>Bounding Box</i> über die Ereignisse in Zeitspannen zerlegen, die Anzahl wird dabei durch den <i>Integer</i> -Parameter vorgegeben. Ausgehend von diesen Zeitspannen werden Relationen erstellt, die als Rückgabewerte dienen. Die Funktion <i>when_maximum</i> gibt für jede dieser Zeitspannen die Anzahl der gültigen Ereignisse als <i>bigint</i> sowie die Geometrie dieser Ereignisse als <i>geometry</i> , vom Typ <i>MULTIPOINT</i> , zurück. Im Wesentlichen wird so ein einfaches Histogramm erstellt. Der Funktion <i>when_where_maximum</i> wird zusätzlich der Name einer zweiten Relation übergeben. Diese wird als statisch aufgefasst und muss Polygonobjekte enthalten. Die Histogrammbildung der Ereignisse erfolgt dann separat für jedes Objekt der zweiten Relation. Dabei werden nur die Ereignisse gezählt, die dieses Objekt räumlich einschließt. Darauf aufbauend wird für jede Zeitspanne das Flächenobjekt ermittelt, in dem die meisten Ereignisse stattfanden. In der von dieser Funktion zurückgegebenen Relation werden für jede Zeitspanne der Name des Polygonobjekts mit den meisten Ereignissen sowie die Anzahl der Ereignisse innerhalb der Zeitspanne und des Objekts angegeben.

Unter Verwendung der Datensätze *Tornado* und *Airport* ließen sich die Ansätze zur Ereignisanalyse testen. Durch die Funktion *when\_where\_maximum* wurden diese Ereignis-Layer mit dem *State*-Datensatz verschnitten und wie oben angegeben ausgewertet. So konnte ermittelt werden, dass sich, bezogen auf die USA, die meisten Tornados seit 1955 in Texas ereigneten und die meisten Flughäfen in Alaska gebaut wurden. Abbildung 4-9 zeigt die zurückgegebene Relation für *Tornados* und zehn Histogrammintervalle.

	period geotime	state character varying	count bigint
1	CALJUG001950-01-03T00:00:00.000Z/001955-11-27T07:11:60.000Z	Kansas	318
2	CALJUG001955-11-27T07:11:60.000Z/001961-10-20T14:23:60.000Z	Texas	542
3	CALJUG001961-10-20T14:23:60.000Z/001967-09-13T21:35:60.000Z	Texas	598
4	CALJUG001967-09-13T21:35:60.000Z/001973-08-07T04:47:60.000Z	Texas	952
5	CALJUG001973-08-07T04:47:60.000Z/001979-07-01T11:59:60.000Z	Texas	801
6	CALJUG001979-07-01T11:59:60.000Z/001985-05-24T19:11:60.000Z	Texas	916
7	CALJUG001985-05-24T19:11:60.000Z/001991-04-18T02:23:60.000Z	Texas	723
8	CALJUG001991-04-18T02:23:60.000Z/001997-03-11T09:35:60.000Z	Texas	1020
9	CALJUG001997-03-11T09:35:60.000Z/002003-02-02T16:47:60.000Z	Texas	917
10	CALJUG002003-02-02T16:47:60.000Z/002008-12-26T23:59:60.000Z	Texas	864

Abbildung 4-9: PgAdmin3 – Am häufigsten von Tornados betroffene Bundesstaaten der USA.

Abbildung 4-10 zeigt die Rückgabe der simpleren *when\_maximum* Funktion mit den *Airport*-Objekten. So konnte festgestellt werden, dass die meisten Flugplätze während der Zeit des zweiten Weltkrieges und unmittelbar danach gebaut wurden. Das Geometrieattribut enthält alle für die Zeitspanne gültigen Flughäfen.



	period geotime	count bigint	the_geom geometry
1	CALJUG001939-07-11T02:24:00.000Z/001943-04-13T19:12:00.000Z	269	010400000000
2	CALJUG001943-04-13T19:12:00.000Z/001947-01-15T12:00:00.000Z	125	010400000007L
3	CALJUG001947-01-15T12:00:00.000Z/001950-10-19T04:48:00.000Z	98	010400000006Z
4	CALJUG001935-10-07T09:36:00.000Z/001939-07-11T02:24:00.000Z	97	010400000006i
5	CALJUG001962-01-27T07:12:00.000Z/001965-10-31T00:00:00.000Z	38	0104000000066

Abbildung 4-10: PgAdmin3 – Histogramm aus *Instant*-Objekten - Ausschnitt.

## 5. Fazit und Ausblick

Ziel dieser Bachelorarbeit ist die Konzeption und Implementierung eines Datenschemas für die Modellierung der Dynamik von Geoobjekten. Dabei sollte das Konzept der ISO19108 einbezogen und umgesetzt werden. In diesem Teil werden die Ergebnisse zusammengefasst und kritisch eingeschätzt.

Durch die Vorgaben der ISO19108 ließen sich in der Konzeption Objekte klassifizieren, durch die verschiedenste Zeitgeometrien modelliert werden können. Dem Standard folgend wurde die Zeit zunächst isoliert von den übrigen Objektdimensionen betrachtet. Durch das Modell lässt sich die Dynamik von Objekten erfassen, funktionelle Abhängigkeiten zwischen Raum und Zeit wurden in die Betrachtungen jedoch nicht einbezogen.

In der Konzeption wurde bereits vorweggenommen, dass die Art und Weise der Einbindung der Bezugssysteme rein pragmatisch erfolgte. So konnte gezeigt werden, dass die integrierte Verwendung verschiedener Systeme möglich ist. Durch die Einführung des Julianischen Datums, als internes Bezugssystem, konnten die weiteren Betrachtungen weitgehend unabhängig vom jeweiligen System erfolgen. Für die praktische Verwendung wäre es jedoch zweckmäßig, die Zuordnung zu einem Bezugssystem auf der Relations- oder Datenbankebene zu ermöglichen. Dies schließt auch eine Beschreibung der Charakteristika eines jeden Systems auf der Nutzerebene mit ein. Dies sind Informationen, die als Metadaten der Datensätze für eine praktische Verwendung abrufbar sein müssen. Realisierbar erscheint auch die Definition von Bezugssystemen durch den Anwender. Insbesondere ein temporales Koordinatensystem oder Ordinalsystem ließe sich, anhand einfacher Parameter, durch die Nutzer hinreichend beschreiben.

Für die klassifizierten Objekte wurde eine eindeutige textuelle Darstellung konzipiert. Aufbauend auf der ISO8601 lassen sich standardkonforme Zeichenketten für Zeitpunkte, Zeitspannen oder auch mehrere Zeitpunkte oder Zeitspannen definieren. Für die Beschreibung der konzipierten regelmäßigen Objekte lassen sich keine oder nicht eindeutige Entsprechungen in der ISO8601 finden. Daher wurde hier eine Erweiterung der Syntax festgelegt. Die Angabe des Bezugssystems ließ sich nur durch eigens definierte Codes realisieren. Infolge dessen ist die Syntax nicht ohne weiteres für eine allgemeine Informationsübermittlung geeignet. Wenn die Bezugssysteme grundsätzlich anders verwaltet werden würden, könnte die Beschreibung eines Zeitobjekts hiervon losgelöst

erfolgen. Die Informationen über das Bezugssystem könnten dann auf einer anderen Ebene abgefragt werden.

Für die modellierten Objekte ließen sich verschiedene Operationen definieren. Von besonderer Bedeutung sind die elementaren Analysefunktionen, zu denen auch die Funktionen der ISO19108 zählen. Durch diese lassen sich temporale Informationen wie die relative Position, Distanz oder Gleichzeitigkeit über die Objekte gewinnen. Darüber hinaus wurden weitere Funktionen konzipiert, welche die Erzeugung, die Manipulation und den Zugriff auf die Daten ermöglichen. Diese dienen dem praktischen Umgang mit den Objekten. Es wurden Möglichkeiten zur Analyse der Dynamik von Geoobjekten unter Einbeziehung der geometrischen und temporalen Dimension aufgezeigt. Mit eingeschlossen wurden hier die Fortführungsbeziehungen der ISO19108.

Der Funktionsumfang kann noch um zahlreiche Möglichkeiten ergänzt werden, die nicht Gegenstand dieser Arbeit waren. Beispielsweise könnten die Ereignis-Histogramme als stochastische Zufallsprozesse aufgefasst werden und entsprechende Berechnungen durchgeführt werden. So ließen sich Aussagen über die Korrelation, Periodizität und den Trend eines Ereignisses treffen. Solche Funktionen wären ein stark spezialisiertes Expertenwerkzeug, doch die Darstellungen in den Grundlagen legen nahe, dass hier ein Bedarf besteht. Ähnlich spezialisierte Werkzeuge haben sich in GIS schon bewährt, die Sichtlinienanalyse sei als ein Beispiel genannt.

Auf der konzeptuellen Ebene wäre noch eine intensive Auseinandersetzung mit der Behandlung von Systemen und Informationen unterschiedlicher zeitlicher Granularität erforderlich.

Die grundsätzliche Umsetzbarkeit der Konzeption konnte durch die erfolgte Implementierung nachgewiesen werden. Dem Rahmen dieser Arbeit entsprechend, ist diese Implementierung nicht vollständig. Der Großteil der konzipierten Objekte kann durch die alphanummerische Darstellung angelegt und verwaltet werden. Die noch ausstehende Einbindung der unterminierten regelmäßigen Phänomene wurde bedacht. Dem Nutzer stehen verschiedene Bezugssysteme zur Verfügung. Praktisch ist es möglich, weitere Bezugssysteme in der erfolgten Art und Weise einzubinden. Wie bereits erwähnt, sollte das Konzept in dieser Hinsicht jedoch überarbeitet werden. In jedem Fall ist noch die Ergänzung um UTC als Uhrzeitsystem notwendig, speziell für den Gregorianischen Kalender. Hier könnte die Anzahl der erfolgten Schaltsekunden nach dem Prinzip einer Lookup-Tabelle intern abgefragt werden.

Auch die konzipierten Operationen konnten prototypisch umgesetzt und deren Funktionalität validiert werden. Die Ansätze für die raum-zeitlichen Analysen ließen sich ebenfalls realisieren. Insbesondere diese raum-zeitlichen Funktionen würden, hinsichtlich der gewählten Ein- und Ausgabeparameter, noch ein großes Optimierungspotential bieten. Insgesamt wären noch weitere Funktionen zu ergänzen, die Vorhandenen zu perfektionieren und vor allem eingehend zu testen. Beispielsweise sollte auch die Bearbeitung von Mengen, in Form von *SETs* oder *Arrays*, umfassend umgesetzt werden.

Durch die Möglichkeiten in PostgreSQL ergeben sich noch weitere sinnvolle Ergänzungen des Funktionsumfangs. Von besonderer Bedeutung für die praktische Verwendung wäre die Definition der *Generalized Search Tree*-Funktionen für die Datenstruktur. Damit wären unmittelbar verschiedene Indexierungsverfahren auf dem Typen möglich. Dies würde vermutlich das Laufzeitverhalten verschiedenster Abfragen deutlich optimieren. Die genannten Funktionen sind jedoch auch Grundlage der Berechnung zahlreicher SQL-Anweisungen wie *ORDER BY* oder *EQUALS* in dem DBMS. Bisher ist nur ein Hashindex-Operator umgesetzt, der lediglich *EQUALS* ermöglicht. Durch die Benutzerfreundlichkeit von PostgreSQL scheint auch die Realisierung eines GiST-Indexes möglich:

*“One advantage of GiST is that it allows the development of custom data types with the appropriate access methods, by an expert in the domain of the data type, rather than a database expert.”* (POSTGRESQL 2012e).

Darüber hinaus sollte die Datenstruktur auch mit dem Datentyp *bytea* verbunden werden. Dieser dient dem binären Datenaustausch in PostgreSQL.

Während der Umsetzung wurden aufgrund der ausführlichen Dokumentation durchweg positive Erfahrungen mit PostgreSQL gesammelt. Die Einbindung und Wartung eines benutzerdefinierten Datentyps scheint, nach kurzer Einarbeitung, ein mit einigem Aufwand zu realisierendes Vorhaben zu sein.

Die in dieser Arbeit erfolgte Weise der Zeitmodellierung ist prinzipiell für die Erfassung, Verwaltung und Analyse der Dynamik von Geoobjekten geeignet. Eine praxistaugliche Implementierung würde eine Überarbeitung der Konzeption, hinsichtlich der oben genannten Punkte, und eine vollständige Umsetzung erfordern. Es ist davon auszugehen, dass in dem Zyklus aus Entwickeln und Testen noch weitere Modifikationen notwendig würden.

Eine solche vollständige, möglichst fehlerfreie Implementierung ist mit einigem Aufwand verbunden. Jedoch könnte als Ergebnis eine geschlossene Bibliothek vorliegen, welche die erforderlichen Operationen zur Analyse der Dynamik beinhaltet. In der Datenbank würde die *valid time* einheitlich durch den abstrakten Datentyp modelliert werden. Den Nutzern könnten die Funktionen auf diesem Typ, weitgehend gekapselt von der unterschiedlichen internen Behandlung der Objekte, angeboten werden.

Hierauf könnten weitere Schichten aufbauen, die Einbindung in ein GIS oder einen Geodatendienst wären denkbar. Die Einschätzung, ob sich ein profunder praktischer Nutzen aus der Datenstruktur erzielen lässt, ist unmittelbar an die Realisierbarkeit derartiger Schnittstellen gebunden. Eine Prüfung dessen wäre, anknüpfend an die Ergebnisse dieser Arbeit, erforderlich.

## Quellenverzeichnis

- ADV (2009): *Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens (GeoInfoDok) – Hauptdokument – Version 6.0.1.*
- ADV (2012): *AdV – AFIS-ALKIS-ATKIS-Modell*, <http://www.adv-online.de/icc/extdeu/broker.jsp?uMen=0a170f15-8e71-3c01-e1f3-351ec0023010>, (06.09.2012).
- BARTELME, N. (2005): *Geoinformatik – Modelle – Strukturen – Funktionen*, 4. vollständig überarbeitete Auflage, Berlin, Heidelberg, Springer.
- BILL, R. (2010): *Grundlagen der Geo-Informationssysteme*, 5. völlig neu bearbeitete Auflage, Berlin, Offenbach, Wichmann – VDE Verlag GMBH.
- BIPM (2006): *The International System of Units (SI) – 8<sup>th</sup> edition.*  
[http://www.bipm.org/utis/common/pdf/si\\_brochure\\_8\\_en.pdf](http://www.bipm.org/utis/common/pdf/si_brochure_8_en.pdf)
- ELMASRI, R.; NAVATHE, S. B. (2002): *Grundlagen von Datenbanksystemen*, 3. überarbeitete Auflage, München, Pearson Studium.
- ESRI (2012a): Arcgis Resources - Help 10.1 2012 – Desktop – Mapping – Time – Temporal data management - *How time is supported in spatial data*,  
[http://resources.arcgis.com/en/help/main/10.1/index.html#/How\\_time\\_is\\_supported\\_in\\_spatial\\_data/005z00000004000000/](http://resources.arcgis.com/en/help/main/10.1/index.html#/How_time_is_supported_in_spatial_data/005z00000004000000/) (06.09.2012).
- ESRI (2012b): Arcgis Resources - Help 10.1 2012 – Desktop – Mapping – Time – Temporal data management – *Supported field formats*,  
[http://resources.arcgis.com/en/help/main/10.1/index.html#/Supported\\_field\\_formats/005z00000006000000/](http://resources.arcgis.com/en/help/main/10.1/index.html#/Supported_field_formats/005z00000006000000/) (06.09.2012).
- ESRI (2012c): Arcgis Resources - Help 10.1 2012 – Desktop – Mapping – Time – Visualizing temporal data – *Setting the time properties on data*,  
[http://resources.arcgis.com/en/help/main/10.1/index.html#/Setting\\_time\\_properties\\_on\\_data/005z000000q0000000/](http://resources.arcgis.com/en/help/main/10.1/index.html#/Setting_time_properties_on_data/005z000000q0000000/) (06.09.2012).
- ESRI (2012d): Arcgis Resources - Help 10.1 2012 Extensions – *What is Tracking Analyst?*,  
[http://resources.arcgis.com/en/help/main/10.1/index.html#/What\\_is\\_Tracking\\_Analyst/006300000001000000/](http://resources.arcgis.com/en/help/main/10.1/index.html#/What_is_Tracking_Analyst/006300000001000000/) (06.09.2012).
- FURIA, C. A.; MANDRIOLI, D.; MORZENTI, A.; ROSSI, M. (2010): *Modeling Time in Computing: A Taxonomy and a Comparative Survey*. In: ACM Computing Surveys, Vol. 42, No. 2.
- GEOSERVER (2012): *GeoServer – Welcome*, <http://geoserver.org/display/GEOS/Welcome>, (06.09.2012).
- GEOSOLUTIONS (2011a): *Geosolutions – Blog – Time and elevation support in GeoServer*  
<http://geo-solutions.blogspot.de/2011/06/time-and-elevation-support-in-geoserver.html>  
(06.09.2012).
- GEOSOLUTIONS (2011b): *Geosolutions – Blog – Animate your maps with GeoServer Animator Tool*, <http://geo-solutions.blogspot.de/2011/11/animate-your-maps-with-geoserver.html>  
(06.09.2012).
- GIANCOLI, D. C. (2006): *Physik*, 3. aktualisierte Auflage, München, Pearson Studium.

- GRASER, A. (2011): *Visualisierung raum-zeitlicher Daten in Geoinformationssystemen am Beispiel von Quantum GIS mit „Time Manager“-Plug-In*, [http://www.fossgis.de/konferenz/2011/programm/attachments/186\\_TimeManager\\_Abstract.pdf](http://www.fossgis.de/konferenz/2011/programm/attachments/186_TimeManager_Abstract.pdf) (06.09.2012).
- GRUMM, H. P.; SOMMER M. (2011): *Einführung in die Informatik*, 9.Auflage, München, Oldenbourg Wissenschaftsverlag GmbH.
- IERS (2010): Home – Glossary – *Universal Time*.  
[http://www.iers.org/nn\\_10780/IERS/EN/Service/Glossary/ut1.html?\\_\\_nnn=true](http://www.iers.org/nn_10780/IERS/EN/Service/Glossary/ut1.html?__nnn=true) (06.09.2012).
- Iso (2002a): ISO19108:2002 Geographic information – *Temporal schema*.
- Iso (2002b): ISO19101:2002 Geographic information – *Reference model*.
- Iso (2005): ISO 19109:2005 Geographic information – *Rules for application schema*.
- Iso (2004): DIN ISO8601:2004 Datenelemente und Austauschformate – Informationsaustausch – *Darstellung von Datum und Uhrzeit*.
- Iso (2006): ISO 19108:2002/Cor.1:2006.
- Iso (2007): ISO 11404:2007 Information technology – *General-Purpose Datatypes*.
- KEMP, K. K. (Hrsg.) (2008): *Encyclopedia of Geographic Information Science*, 1.Auflage, Los Angeles, London, New Delhi, Singapore, Sage Publications.
- KLIMA, R.; SELBERHERR, S. (2007): *Programmieren in C*, 2.Auflage, Wien, Springer.
- NATIONALATLAS.GOV (2011): nationalatlas.gov – *Map Layers*, <http://nationalatlas.gov/maplayers.html> (06.09.2012).
- MAPSERVER (2006): MapServer – Documentation – OGC Support and Configuration – *WMS Time*, [http://www.mapserver.org/ogc/wms\\_time.html](http://www.mapserver.org/ogc/wms_time.html), (06.09.2012).
- MAPSERVER (2012): MapServer – *Welcome to MapServer*, <http://www.mapserver.org/index.html>, (06.09.2012).
- MEEUS, J. (1991): *Astronomical Algorithms*, Richmond, 1.Auflage, Willmann-Bell.
- OGC (2006): OpenGIS® *Web Map Server Implementation Specification*, Version 1.3.0.
- OGC (2007): OpenGIS® *Geography Markup Language (GML) Encoding Standard*, Version 3.2.1.
- OGC (2009): The OpenGIS® *Abstract Specification - Topic 5: Features*, Version 5.0.
- OGC (2010): OpenGIS® *Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option*, Version 1.2.1.
- OGC (2011): OpenGIS® *Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture*, Version 1.2.1.
- PEUQUET, D. J. (2001): *Making Space for Time: Issues in Space-Time Data Representation*. In: *Geoinformatica*, Volume 5, Number 1.
- PEUQUET, D. J. (2002): *Representations of Space and Time*, New York, The Guilford Press.

- POSTGIS (2012a): PostGIS Documentation – PostGIS 2.0.2SVN Manual – 4.1.1 *OpenGIS WKB and WKT*, [http://postgis.refractions.net/documentation/manual-2.0/using\\_postgis\\_dbmanagement.html#OpenGISWKBWKT](http://postgis.refractions.net/documentation/manual-2.0/using_postgis_dbmanagement.html#OpenGISWKBWKT) (06.09.2012).
- POSTGIS (2012b): PostGIS Documentation – PostGIS 2.0.2SVN Manual – 8. *PostGIS Reference*, <http://postgis.refractions.net/documentation/manual-2.0/reference.html> (06.09.2012).
- POSTGIS (2012c): *PostGIS – What is PostGIS?*, <http://www.postgis.org/> (07.09.2012).
- POSTGRESQL (2012a): PostgreSQL – *About*, <http://www.postgresql.org/about/> (07.09.2012).
- POSTGRESQL (2012b): PostgreSQL 9.1.5 Documentation – *CREATE TYPE*, <http://www.postgresql.org/docs/9.1/interactive/sql-createtype.html> (06.09.2012).
- POSTGRESQL (2012c): PostgreSQL 9.1.5 Documentation – *User defined Types*, <http://www.postgresql.org/docs/9.1/interactive/xtypes.html> (06.09.2012).
- POSTGRESQL (2012d): PostgreSQL 9.1.5 Documentation – *PL/pgSQL Procedural Language*, <http://www.postgresql.org/docs/9.1/interactive/plpgsql-overview.html> (07.09.2012).
- POSTGRESQL (2012e): PostgreSQL 9.1.5 Documentation – *GiST Indexes – Introduction*, <http://www.postgresql.org/docs/9.1/interactive/gist-intro.html> (09.09.2012).
- POSTGRESQL (2012f): PostgreSQL 9.1.5 Documentation – *C-Language Functions – Base Types in C-Language Functions*, <http://www.postgresql.org/docs/9.1/interactive/xfunc-c.html#XFUNC-C-BASETYPE> (06.09.2012).
- RICHARDS, E. G. (1998): *Mapping Time – The Calendar and its History*, New York, Oxford University Press.
- USNO (2012): U.S. Naval Observatory – Time Service Department – *Leap Seconds*, <http://tycho.usno.navy.mil/leapsec.html>
- VOSSEN, G. (2008): *Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme*, 5. überarbeitete und erweiterte Auflage, München, Oldenbourg Wissenschaftsverlag GmbH.
- WEIDEMANN, N. B.; DOREEN, K.; GLEDITSCH, K. S. (2010): *The Geography of the International System: The CShapes Dataset*, <http://nils.weidmann.ws/projects/cshapes/shapefile> (06.09.2012).
- WEFER, G. (Hrsg.) (2010): *Dynamische Erde – Zukunftsaufgaben der Geowissenschaften – Strategieschrift*, 1. Auflage, Geokommission der Deutschen Forschungsgemeinschaft.
- WOLF, J. (2010): *Grundkurs C*, 1. Auflage, Bonn, Galileo Press.
- WORBOYS, M.; DUCKHAM, M. (2004): *GIS – A Computing Perspective*, 2. Auflage, Boca Raton, CRC Press.

## Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tage eingereichte Bachelorarbeit mit dem Titel *Realisierung der Zeitkomponente einer Geodatenbank durch einen ISO19108 konformen Datentyp* vollkommen selbstständig und nur unter Benutzung der in der Arbeit angegebenen Hilfsmittel angefertigt habe. Dies schließt die auf der CD-ROM befindlichen Daten sowie den Anhang mit ein.

Dresden, den 17.09.2012

---

Peter Broßheit

# Anhang

## Anhang A | CD-ROM

Liste der enthaltenen Ordner und Dateien – Nur oberstes Verzeichnis:

/BA_PeterBrosseit.pdf	Digitale Version dieser Bachelorarbeit (pdf).
/geotime-0.1/	Linux <i>Makefile</i> . Weitere Informationen in der <i>Info.pdf</i> .
/testdata_conditioning/	SQL-Dateien zum prozessieren der Testdaten. Werden von <i>geotimeImportScript</i> genutzt.
/testdata_shapefiles/	Die Testdaten im Shapefile-Format.
/src/	Der C-Quellcode. Redundant mit einem entsprechenden Verzeichnis in <i>/geotime-0.1/</i> .
/Abstract.pdf	Die dieser Arbeit vorstehende Kurzfassung in englischer und deutscher Sprache.
/CREATE_DATABASE.sql	Entfernt(!) die Datenbank ‚geotime‘ und legt sie neu an. Wird von <i>geotimeImportScript</i> genutzt.
/example_tables.sql	Weitere SQL-Anweisungen für das <i>geotimeImportScript</i> .
/geotime.sql	Die geotime-SQL-Datei. Enthält die Anweisungen zur Erzeugung des Typen und der verknüpften Funktionen. Wird von <i>geotimeImportScript</i> genutzt.
/geotimeImportScript	Linux-Skript. Importiert die Testdaten und legt weitere Tabellen mit Testdaten an. Siehe <i>Info.pdf</i> .
/Info.pdf	Weitere Informationen zu den Daten der CD. Insbesondere zu der Installation von <i>geotime</i> und dem Anlegen einer Test-Datenbank.
/testdataImportScript	Linux-Skript. Siehe <i>Info.pdf</i>



## Anhang B | Ergänzung zu ‚Abstrakte Datentypen in PostgreSQL‘

Hier werden die weiteren Möglichkeiten – neben der gewählten – zur Einbindung eines benutzerdefinierten Datentyps in PostgreSQL dargestellt.

Der Nutzer kann prinzipiell eine zusammengesetzte Struktur, einen Aufzählungstyp oder einen eigenen, elementaren Basisdatentyp registrieren. Ein zusammengesetzter Typ ist lediglich eine Zusammenfassung einer konstanten Menge von Attributen bestimmter elementarer Datentypen. Die PostgreSQL Dokumentation setzt einen solchen Typ zur Anschaulichkeit einer Zeile einer bestimmten Tabelle gleich. Ein Aufzählungstyp umfasst eine feste Menge geordneter Werte, wobei eine Instanz des Typs immer genau einen dieser Werte annimmt. Diese beiden Varianten sind aufgrund ihrer statischen Natur und den noch relativ geringen Modifikationsmöglichkeiten nicht geeignet um eine adäquate Datenstruktur anzulegen (POSTGRESQL 2012b).

Neben der gewählten internen Form variabler Länge sind noch zwei weitere Arten von Basisdatentypen möglich. Die erste basiert auf der Übergabe des Wertes und einer festen Größe. Das heißt bei jedem Zugriff werden die gesamten Daten kopiert, daher ist hier auch, Hardware abhängig, eine maximale Größe von 4 bis 8 Bytes zulässig. Aufgrund der bei komplexen Zeittypen hohen Informationsmenge ist diese Form hier nicht geeignet. Die zweite Möglichkeit bringt ebenfalls eine feste Größe mit sich, hier werden allerdings nicht die gesamten Daten übergeben, sondern lediglich die Speicheradresse, an der die Daten zu finden sind. Problematisch ist hier allerdings die konstante Größe. Dies würde bedeuten, dass jede Instanz der Datenstruktur soviel Speicher beansprucht, wie der größtmögliche Komplex benötigt. Die Lösung dieses Problems ist die dritte Variante, in der die Übergabe ebenfalls durch die Speicheradresse erfolgt, aber die Struktur eine variable interne Länge haben kann (POSTGRESQL 2012f). Diese Option wurde in der dieser Arbeit beiliegenden Implementierung gewählt.

## Anhang C | Weitere Funktionen der prototypischen Umsetzung

Hier werden weitere Funktionen, ähnlich wie in Abschnitt 4.3, in tabellarischer Form dargestellt. Auf die Funktionsbeschreibungen folgt jeweils ein Beispiel für eine Abfrage mit dem entsprechenden Rückgabewert. Die SQL-Abfragen wurden in dieser Weise getestet. Zunächst sei ein Beispiel für die Verwendung der Funktion *temporal\_bbox* ergänzt:

```
SELECT temporal_bbox( testdata_airports.validtime ) FROM testdata_airports;
CALGRE001928-04-01T00:00:00.000Z/002003-06-01T00:00:00.000Z
```

<i>Funktion:</i>	<b>tm_duration_string</b> (geotime) : cstring <b>tm_duration_julday</b> (geotime) : double precision
------------------	---

<i>Implementiert:</i>	duration (Time) : Text<8601 duration> durationJulDay (Time) : Double
-----------------------	---

<i>Beschreibung:</i>	Diese Funktionen geben die Dauer einer <i>Period</i> zurück. Ähnlich wie die <i>distance</i> -Funktionen als String im ISO8601-Format oder als Dezimaltag.
----------------------	--

```
SELECT tm_duration_julday('CALJUG1582/1583')      355
SELECT tm_duration_julday('CALGRE1582/1583')     365
SELECT tm_duration_string('2001-3-12T8:59:59.999+1/2003-10-8T10:20:0+1')
                                                                 P2Y6M26DT1H20M0.001
```

**Funktion:** **transform\_refsys**(geotime, cstring) : geotime

**Implementiert:** transformRefSys (Time, Text) : Time

**Beschreibung:** Diese Funktion transformiert die gegebene *geotime*-Instanz in das angegebene Bezugssystem und gibt das Ergebnis zurück. Transformationen von oder in Ordinalsysteme sind nicht möglich. Intern erfordern die Transformationen lediglich eine Änderung des Wertes im refsys-Feld der Struktur.

```
SELECT transform_refsys( 'TCSJDA2450548.670257/2893489.465744', 'CALGRE')
                                                                 CALGRE001997-04-10T04:05:10.205Z/003210-01-01T23:10:40.282Z
```

**Funktion:** **two\_instants\_to\_period**(geotime, geotime) : geotime

**Implementiert:** period (Time, Time) : Time

**Beschreibung:** Diese Funktion bildet aus zwei gegebenen *Instants* eine *Period* und gibt diese zurück.

```
SELECT two_instants_to_period('CALJUL-200-3-21','CALJUL200-4-12')
                                                                 CALJUL-00200-03-21T00:00:00.000Z/000200-04-12T00:00:00.000Z
```

**Funktion:** **is\_simultaneous**(geotime, geotime) : boolean

**Implementiert:** simultaneous (Time, Time) : Boolean

**Beschreibung:** Diese Funktion gibt *TRUE* zurück, wenn die Objekte einen gemeinsamen Zeitpunkt teilen, andernfalls *FALSE*.

```
SELECT is_simultaneous(
(SELECT temporal_bbox(validtime) FROM testdata_airports),
(SELECT temporal_bbox(validtime) FROM testdata_cshape) )      t
```

**Funktion:** **Typumwandlungen**

geotime <-> **date**

geotime <-> **timestamp with time zone**

geotime <-> **timestamp without time zone**

geotime -> **text** *Assignment*

**Implementiert:** date (Time) : Date                    time (Date) : Time

**Beschreibung:** Es stehen Typumwandlungen zwischen den drei aufgelisteten Typen und *geotime* zur Verfügung. Die Umwandlungen werden auf den jeweiligen internen Strukturen durchgeführt. Ein weiterer Cast empfiehlt die Umwandlung eines *Strings* in *geotime* durch die Eingabefunktion, wenn *geotime* erwartet wird, z.B. in einigen der sonstigen hier gezeigten Funktionen.

```
SELECT geotime('2012-01-02 12:00:00.001'::timestamp without time zone)
                                                                 CALGRE002012-01-02T12:00:00.001Z
```

**Funktion:** **Operatoren**

geotime = geotime    -> boolean

geotime - geotime    -> double precision

**Implementiert:** Time = Time → equal (Time, Time) : Boolean

Time – Time → distanceJulDay (Time, Time) : Double

**Beschreibung:** Das Gleichheitszeichen überprüft die binäre Gleichheit der Objekte. Dafür wird eine weitere, nur für diesen Zweck, umgesetzte Funktion aufgerufen. Das Minuszeichen ruft die

Funktion <code>tm_distance_julday</code> auf.	
<code>SELECT '2014,2015,2016,2017'::geotime = '2012-2-29T10:30:45.329'::geotime</code>	f
<code>SELECT '2014/2016'::geotime - '2012-2-29T10:30:45.329'::geotime</code>	671,561975358985

<b>Funktion:</b>	<b><code>regular_multi_to_multi</code></b> (geotime) : geotime
<b>Implementiert:</b>	<code>regularMultiToMulti</code> (Time) : Time
<b>Beschreibung:</b>	Diese Funktion bildet aus einem gegebenen regelmäßigen MultiObjekt ein einfaches MultiObjekt und gibt dieses zurück. Durch diese Funktion ließen sich einfache und regelmäßige MultiObjekte in vielen weiteren Funktionen einheitlich behandeln. Nur für Kalendersysteme umgesetzt.
<code>SELECT regular_multi_to_multi('R3/2012-9-16T12:20/PT20M/PT40M')</code>	<code>CALGRE002012-09-16T12:20:00.000Z/002012-09-16T12:40:00.000Z,</code> <code>002012-09-16T13:20:00.000Z/002012-09-16T13:40:00.000Z,</code> <code>002012-09-16T14:20:00.000Z/002012-09-16T14:40:00.000Z</code>
<code>SELECT regular_multi_to_multi('R3/2012-9-16T/P1M')</code>	<code>CALGRE002012-09-16T00:00:00.000Z,002012-10-16T00:00:00.000Z,</code> <code>002012-11-16T00:00:00.000Z</code>
<code>SELECT regular_multi_to_multi('R3/2012-9-16T/P30D')</code>	<code>CALGRE002012-09-16T00:00:00.000Z,002012-10-16T00:00:00.000Z,</code> <code>002012-11-15T00:00:00.000Z</code>

<b>Funktion:</b>	<b><code>get_start_instant</code></b> (geotime) : geotime <b><code>get_end_instant</code></b> (geotime) : geotime
<b>Implementiert:</b>	<code>startInstant</code> (Time) : Time
<b>Beschreibung:</b>	Gibt den Beginn bzw. das Ende einer <i>Period</i> als <i>Instant</i> zurück.
<code>SELECT get_start_instant((SELECT temporal_bbox(validtime) FROM testdata_airports))</code>	<code>CALGRE001928-04-01T00:00:00.000Z</code>
<code>SELECT get_end_instant((SELECT temporal_bbox(validtime) FROM testdata_airports))</code>	<code>CALGRE002003-06-01T00:00:00.000Z</code>

<b>Funktion:</b>	<b><code>add_julian_day</code></b> (geotime, double precision) : geotime
<b>Implementiert:</b>	<code>addDuration</code> (Time,Text<8601 duration>) : Time
<b>Beschreibung:</b>	Interpretiert den gegebenen Gleitkommawert als Dezimaltag und addiert diese Dauer zu dem gegebenen <i>Instant</i> .
<code>SELECT add_julian_day('2012',30.25)</code>	<code>CALGRE002012-01-31T06:00:00.000Z</code>

<b>Funktion:</b>	<b><code>add_instant</code></b> (geotime) : geotime
<b>Implementiert:</b>	-
<b>Beschreibung:</b>	Fügt ein <i>Instant</i> -Objekt zu einer <i>MultiInstant</i> hinzu.
<code>SELECT add_instant('2012,2016','2013')</code>	<code>CALGRE002012-01-01T00:00:00.000Z,002016-01-01T00:00:00.000Z,</code> <code>002013-01-01T00:00:00.000Z</code>