

Perzeptive Syntagmen für Bewegbilddokumente

Dissertation

Zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
vorgelegt im Fachbereich
Elektrotechnik, Informationstechnik, Medientechnik
an der
Bergischen Universität Wuppertal

von
Marko Hedler

Wuppertal, April 2008

Die Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20080538

[<http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20080538>]

Referent: Univ.-Prof. Dr. Karl-Heinrich Schmidt
Korreferent: Univ.-Prof. Dr. Peter Urban
Tag der mündlichen Prüfung: 16. September 2008

Zusammenfassung

Die vorliegende Dissertation beschäftigt sich mit der Analyse und formalisierten Abbildung von Metadaten für Bewegtbilddokumente. Dies geschieht anhand einer Neuentwicklung von Syntagmen zur Strukturierung von Teildokumenten im Film. Der Schwerpunkt liegt dabei auf Syntagmen, die erstmals auch Dokumententeile unterhalb der Einstellungsebene berücksichtigen.

Bisher kann eine Menge von Einstellungen nur dann syntagmatisch klassifiziert werden, wenn sie als Ganzes räumlich zusammenhängend ist. Dies ist für die logische Assoziation von Einstellungen allerdings nicht zwingend notwendig. Ein Beobachter kann trotz eines fehlenden räumlichen Zusammenhangs Segmente als zusammengehörig auffassen. Hier wird durch neu entwickelte perzeptive Syntagmen die Aggregation von Einstellungen ohne räumliche Zusammenhangsbedingungen geleistet.

Das Ergebnis ist eine formalisierbare Beschreibung perzeptiver filmischer Syntagmen, einbettbar in das Metadatenformat MPEG-7.

Abstract

This thesis researches the analysis and the formalised metadata mapping of moving data. Therefore a new syntagma for structuring document subparts in movies is being developed. The dissertation focuses on syntagmata which for the first time consider subparts of a document below the shot level.

Until now, a set of shots could only be syntagmatically classified, if they were spatially connected. However, this is not mandatory for the logical association of all kinds of shots. Despite the missing spatial connection, an observer can classify segments as if they belonged together. With the new development of a perceptive syntagma, shots lacking a spatial connection can now be aggregated.

The result is a formalized description of perceptive syntagmata for moving data which can be embedded in the metadata format MPEG-7.

Inhaltsverzeichnis

Zusammenfassung	3
Abstract.....	3
Danksagung	7
1 Einleitung.....	9
1.1 Problembeschreibung und Aufgabenstellung	9
1.2 Lösungsweg und Ergebnis	10
1.3 Aufbau der Arbeit	11
 TEIL I – Dokumententheoretische Grundlagen	
2 Dokumente und Metadaten	15
2.1 Das Dokument	15
2.2 Metadaten	18
2.3 Bibliothekarische Metadaten	20
2.4 Metadaten für Teildokumente	24
3 XML – Extensible Markup Language	29
3.1 Grundlagen	29
3.2 Syntaktischer Aufbau	30
3.3 DTD – Document Type Definition	34
3.4 Zusammenfassung	41
4 XML-Schema	43
4.1 Referenz auf ein Schema	43
4.2 Definition eines XML-Schemas	44
4.3 Definition von Elementen	44
4.4 Definition von Elementmodellen	45
4.5 Definition von Attributen	49
4.6 Notationen und Entities	50
4.7 Zusammenfassendes Beispiel	51
5 MPEG-7	53
5.1 Gliederung von MPEG-7	53
5.2 Die Segment Entity Description Tools des MDS	57
5.3 Die Segment Attribute Description Tools des MDS	67
5.4 Die Segment Decomposition Tools	72
5.5 Die Segment Relation Description Tools	77
5.6 Zusammenfassung	82

TEIL II – Syntagmatische Analyse

6	Basale Syntagmatik von Einstellungen und narrative Hierarchisierung.....	85
6.1	Narratologie	85
6.2	Strukturierte cinematographische Dokumente.....	87
6.3	Monochronie.....	89
6.4	Narrative Hierarchisierung.....	94
6.5	Die Erzählebenen von Branigan	95
7	Perzeptive Syntagmen	101
7.1	Zur speziellen Syntax von Perzeptionssegmenten.....	102
7.2	Zur situationstheoretischen Semantik visueller Perzeption.....	104
7.3	Filmische Serialisierung perzeptiver Relationen	112
7.4	Perzeptiv fokalisierbare Strukturen	117

TEIL III – Realisierung der Ergebnisse mit MPEG-7

8	Abbildung von Syntax und Semantik mit MPEG-7.....	127
8.1	Beschreibung von Perzeptionssituationen in MPEG-7	127
8.2	Syntaktische Anordnungsformen für Perzeption.....	138
9	Beispiele der dramaturgischen Umsetzung von Perzeption mit MPEG-7	149
9.1	Beispiel 1	149
9.2	Beispiel 2	152
9.3	Beispiel 3	156
9.4	Beispiel 4	162
10	Schlussbetrachtung	167
11	Literatur	169

Danksagung

Diese Arbeit entstand an Schreibtischen in Bernhausen, Melbourne, Möhringen, im Stuttgarter Westen und an der Hochschule der Medien in Vaihingen; auf zahlreichen Zugfahrten nach Wuppertal, Essen und Bielefeld; im Café Thalia in Wuppertal und am Wohnzimmertisch „Am Ellerbrocks Hof“.

Besonderer Dank gilt an dieser Stelle meinem Betreuer und Mentor Univ.-Prof. Dr. Karl-Heinrich Schmidt, der seine uneingeschränkte Begeisterung für das Thema an mich weitergegeben hat. Ihm verdanke ich den Feinschliff in der Welt des wissenschaftlichen Arbeitens, den er mir mit großem Aufwand – und letztendlich mit Erfolg – verpassen konnte.

Bei Univ.-Prof. Dr. Peter Urban möchte ich mich für die Übernahme des Korreferats bedanken.

Dank gebührt auch all denjenigen, die mich während der letzten sechs Jahre an der Hochschule der Medien unterstützt haben; allen voran Prof. Dr. Wolfgang Faigle für sein immer offenes Ohr und seine großen Bemühungen, Promotionsvorhaben zu unterstützen. Danken möchte ich der Adolf W. Döpfert-Stiftung, insbesondere ihrem Vorsitzenden Prof. Dr. Rainer Nestler, für die Unterstützung dieser Forschungsarbeit.

Dank auch an die vielen Kollegen im Studiengang Medieninformatik, besonders Herrn Prof. Dr. Edmund Ihler, für die gewährten zeitlichen Freiräume, um dieses Projekt voranzutreiben.

Dank an die Deutsche Bahn, die mich immer sicher (wenngleich auch nicht immer pünktlich), komfortabel und reichhaltig mit Speisen und Getränken versorgt an mein Ziel brachte.

Meinen Eltern möchte ich danken für die uneingeschränkte Unterstützung während des Studiums und während meiner Zeit als Wissenschaftlicher Mitarbeiter sowie für das Vertrauen, dass es „schon für irgendetwas gut“ sein werde.

Ohne die Unterstützung meiner Lebensgefährtin Simone wäre das alles nicht möglich gewesen. Danke.

1 Einleitung

Wissenschaftliche Einrichtungen, kommerzielle Interessenten und auch der „Normalbürger“ nutzen das World Wide Web (WWW) und dessen zahlreiche Dokumente (Text- oder multimediale Dokumente) als Publikationsmedium. Die daraus entstehende Informationsflut wäre ohne die Möglichkeiten des „Herausfilterns“ von Informationen für den Menschen nicht zu bewältigen. Da die Dokumenteninhalte von Menschen gestaltet und dafür bestimmt sind, wiederum von Menschen aufgenommen zu werden, ergibt sich folgende Problematik: Obwohl die Dokumente maschinenlesbar sind, ist im Allgemeinen eine Maschinenverstehbarkeit nicht so weit gegeben, als dass von Menschen leicht erkennbare Informationseinheiten genauso leicht in maschinellen Modellierungen berücksichtigt werden können. Das bedeutet insbesondere, dass der inhaltliche Zusammenhang von multimedialen (Teil-)Dokumenten oft keinen Einfluss auf eine maschinelle Unterstützung bei der Informationserhebung (z. B. bei der Bewertung der Relevanz von Suchergebnissen) hat.

1.1 Problembeschreibung und Aufgabenstellung

Eine mögliche Lösung bietet die Modellierung des inhaltlichen Zusammenhangs von (Teil-)Dokumenten durch Metadaten. Allgemein versteht man unter Metadaten („Daten über Daten“) strukturierte Daten, mit deren Hilfe eine Informationsressource beschrieben werden kann.

Voraussetzungen für eine „Maschinenverständlichkeit“ sind im Allgemeinen eine (wohlgeformte) Syntax und eine (validierbare) Semantik. Beide sind allerdings kein Selbstzweck: Bei deren Festlegung ist es wichtig, dass ein möglichst einheitliches Datenaustauschformat für Ressourcenbeschreibungen entsteht.

Oft besteht ein Metadatenschema aus einer festen oder variablen Anzahl von Attributen, die entweder für einen bestimmten Anwendungsbereich gedacht sind oder anwendungsunabhängig Informationsressourcen beschreiben. Bekanntester Vertreter dieser letzten Kategorie ist der Standard der DublinCore Metadata Initiative, der im Wesentlichen eine bibliothekarische Sicht auf Informationsressourcen widerspiegelt. Dieser wird komplementiert durch eine neue Generation von anwendungsunabhängigen – also generischen – Sprachen speziell in der XML-Welt, die Möglichkeiten zur Definition eigener Attribute vorsehen. Deren formale Repräsentation wird durch Sprachen wie zum Beispiel TopicMaps, RDF/RDF-S, DAML+OIL und OWL ermöglicht, die auf einer wohlgeformten Syntax aufbauen. Die Verwendung solcher Metadatensprachen soll es maschinellen Agenten erleichtern, Informationseinheiten inhaltssensitiv zu erheben.

Bei der Festlegung dieser Sprachen wurde nicht differenziert zwischen unterschiedlichen Inhaltsarchitekturen der Ressourcen. Grundsätzlich ist es für alle genannten Repräsentationsschemata einerlei, ob sie sich auf Textressourcen, Bilder oder Tondokumente beziehen.

In dieser Arbeit soll medienspezifisch – nicht generell – ausschließlich die inhaltliche Beschreibung von Bewegbilddaten einen Schritt weitergebracht werden, indem in Analogie zur Strukturierung von Volltextdokumenten „natürliche“ Einheiten identifiziert werden, die dann sinnvoll mit Metadaten versehen werden können.

Der MPEG-7-Standard zeigt bereits, wie Metadaten auf Bewegbilddokumente angewandt werden können. Diese Metadaten stützen sich sowohl auf bereits im Video- und Audiosignal codierte Informationen wie zum Beispiel erkennbare Objekte, Ton oder Farbe, als auch auf nicht explizit im Signal verankerte Informationen.

Eine Vielzahl von Merkmalen für die Bild- und Tonbeschreibung von Dokumenten sind im MPEG-7-Standard vorgegeben. Außerdem kann MPEG-7 kontrolliert erweitert werden und erlaubt es, jeden beliebigen Dokumententeil nach vorgegebenen oder eigenen Kriterien zu beschreiben. Es liegt ein riesiges Metadatenschema vor, das die Werkzeuge für die Beschreibung von Informationseinheiten bereitstellt. Kriterien für die Festlegung „natürlicher“ Informationseinheiten, die vom Menschen leicht erfasst werden können, liefert MPEG-7 jedoch nicht. Die Festlegung solcher Standardeinheiten speziell für Bewegbilddaten erfordert eine Syntagmatik von Bewegbilddaten, in der die Teile eines Dokuments, die sich für den Menschen als Einheiten der Beschreibung mit Metadaten anbieten, als Syntagmen identifiziert werden.

In dieser Arbeit werden aufbauend auf MPEG-7, aber völlig unabhängig davon, Syntagmen festgelegt, die als Einheiten für die Beschreibung mit Metadaten benutzt werden können. Diese sind dann analog zu großen Einheiten von Volltextdokumenten (wie sie zum Beispiel in DocBook oder in der TEI beschrieben werden) „natürliche“ Träger von Metadaten.

1.2 Lösungsweg und Ergebnis

Im Rahmen dieser Arbeit wird für oben stehende Aufgabenstellung der folgende Lösungsweg gewählt:

Zu Beginn werden die Sprache XML sowie deren Regelwerke DTD und XML-Schema dahingehend untersucht, inwieweit sie sich dafür eignen, Filmdokumente zu strukturieren und entsprechende Metadaten dafür einzubetten. In einem weiteren Schritt erfolgt die genaue Darstellung der Möglichkeiten des Metadatenstandards MPEG-7 bezüglich der Abbildung der Strukturen von Filmdokumenten. Dabei wird gezeigt,

dass MPEG-7 zwar die deskriptiven Möglichkeiten besitzt, Strukturen von Filmdokumenten abzubilden, jedoch für nicht im Videosignal codierte Metadaten keinerlei vordefinierte Metadatensätze enthält.

Aufgrund intensiver Literaturarbeit wird für die filmische Repräsentation von Perzeptionsvorgängen ein spezielles Syntagma erarbeitet. Perzeptionsvorgänge sind ein entscheidendes Mittel einer inhaltlichen Verknüpfung von Bewegbilddaten. Die „filmische“ Repräsentation eines vollständigen Perzeptionsvorgangs ist im Allgemeinen nicht im Videosignal verankerbar. Ziel ist es, Bedingungen zu definieren, die bei der filmischen Repräsentation von Perception gelten müssen. Unter Berücksichtigung der gewonnenen Randbedingungen wird ein perzeptives Syntagma herausgearbeitet, das als Basis für die zukünftige Klassifizierung von Segmenten dient. Eine entscheidende neue Erkenntnis ist dabei, dass sich syntagmatische Einheiten ergeben, die, anders als bisherige, auch unterhalb der Einstellungsebene Dokumentteile einbeziehen.

Die formale Beschreibung des gewonnenen perzeptiven Syntagmas erfolgt schließlich durch dessen Einbettung in MPEG-7.

1.3 Aufbau der Arbeit

Im Anschluss an diese Einleitung umfasst der erste Teil der Arbeit eine Einführung in Metadaten von strukturierten und unstrukturierten Dokumenten.

Danach wird das auf die Bedürfnisse dieser Arbeit zugeschnittene Handwerkszeug für die Beschreibung von Bewegbilddaten bereitgestellt. Dies basiert auf der Extensible Markup Language (XML) und XML-Schema (XSD). Daran anschließend wird die Metadatensprache MPEG-7 vorgestellt mit einem Schwerpunkt auf der Multimedia Description Language (MDS). Den Abschluss von Teil I bildet die ausführliche Übersicht über die einzelnen hier relevanten XML-Schema-Fragmente von MPEG-7 sowie deren Beschreibung.

In Teil II folgt die syntagmatische Analyse für Bewegbilddaten, in denen Perzeptionsvorgänge repräsentiert sind. Die Definition eines geeigneten Syntagmas sowie dessen Randbedingungen folgen im Anschluss.

Teil III bildet den Abschluss mit der Einbettung der gewonnenen Syntagmen in MPEG-7, sowie der Analyse von empirischem Material.

TEIL I – Dokumententheoretische Grundlagen

2 Dokumente und Metadaten

Strukturen von Dokumenten haben einen entscheidenden Einfluss darauf, wie und wie gut Dokumente archiviert und wieder gefunden werden können. Dabei spielt die Auswertbarkeit des Inhaltes eine wichtige Rolle. Dokumente werden daher häufig mit Zusatzinformationen (Metadaten) versehen, um z. B. die Systematik der Archivierung zu verbessern. Nicht alle Dokumenttypen bieten jedoch die Möglichkeit, Inhalte so zu strukturieren, dass diese direkt für verschiedene Anwendungen verwendbar sind. Filmdokumente können zwar strukturierte Dokumente sein, bieten jedoch unter Umständen nur eingeschränkte Möglichkeiten, diese Strukturen bereitzustellen und für Anwendungen nutzbar zu machen.

In diesem Kapitel wird zunächst der Begriff des Dokuments eingeführt. Abschnitt 2.2 beleuchtet ausführlich unterschiedliche Typen von Metadaten und beschreibt Anforderungen an eine gemeinsame Metadatensprache für beliebige Dokumente. In Abschnitt 2.3 wird der Standard der Metadateninitiative Dublin Core eingeführt und auf verschiedene Anwendungsmöglichkeiten bei unterschiedlichen Dokumenttypen hin bewertet. Abschnitt 2.4 beinhaltet eine Analyse des Einsatzes von Metadaten für Binärdokumente. Zum Abschluss wird die Argumentation der Notwendigkeit für den Bedarf an Strukturstandardisierungen und Metadatenformaten für Filmdokumente gezeigt.

2.1 Das Dokument

„Unter Dokumenten im allgemeinsten Sinne verstehen wir Bündel von Informationen, die in der Absicht erzeugt worden sind, dass sie von Menschen wahrgenommen werden“ [Wilhelm, Heckmann 1996, S. 1]. Beispiele dafür sind Tonbeiträge, Filme, Bilder und Textdokumente, die sowohl in analoger als auch in digitaler Form vorliegen können¹.

Aufgrund der digitalen Speicherung und Verarbeitung von Dokumenten ist es möglich, diese nicht nur auf Papier, sondern auch mit elektronischen Mitteln darzustellen. Dokumente lassen sich unter drei Gesichtspunkten betrachten (s. Abbildung 1); diese sind:

- die logische Dokumentsicht (logical view),
- die Layoutsicht (layout view),
- die Inhaltssicht (content view)².

¹ Im Weiteren wird der Begriff „Dokument“ mit der Einschränkung verwendet, dass dieses computergestützt erzeugt und bearbeitet werden kann.

² In Klammern jeweils der englische Begriff, wie er in der Dokumentnorm [ISO/IEC 8612] spezifiziert ist.

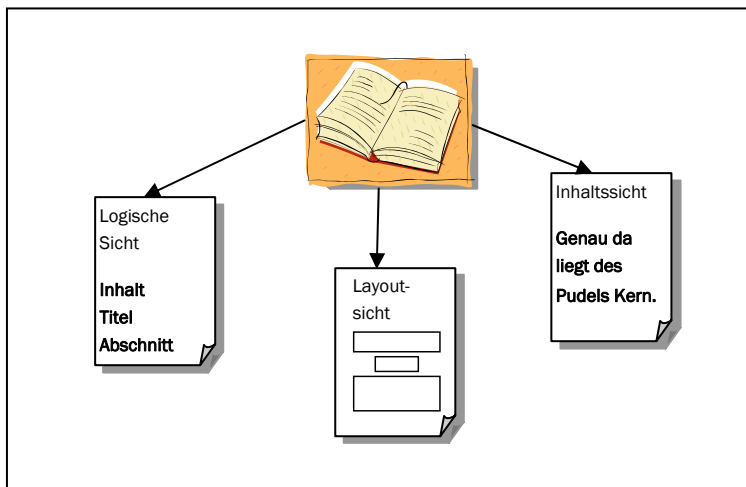


Abbildung 1: Verschiedene Sichtweisen auf ein Dokument.

Die *logische* Struktur gibt die Organisation des Inhalts eines Dokuments vor. Dabei wird noch nicht auf das Erscheinungsbild des Dokuments eingegangen. Beispiele für logische Einheiten von Dokumenten sind bei Textdokumenten Absätze, Kapitel oder Fußnoten. Bei Nichttextdaten wie z. B. einem Spielfilm ist eine logische Einheit eine Einstellung oder eine Szene.

Damit ein Dokument vom Betrachter wahrgenommen werden kann, bedarf es der Visualisierung oder einer sonstigen Präsentation des Dokuments; z. B. bei Bewegbilddokumenten am Bildschirm oder bei Text auf Papier. Diese Darstellung setzt ein *Layout* eines Dokuments voraus. Sie führt zu einer zu einem bestimmten Zeitpunkt messbaren Repräsentation eines Dokuments.

Der *Inhalt* eines Textdokuments besteht beispielsweise aus einer Sequenz von Zeichen, die zu Wörtern oder Sätzen zusammengefasst werden können. Bei Videodokumenten besteht der Inhalt aus rein binären Informationen, deren interne Struktur lediglich den Vorgaben des Dateiformatstandards genügen muss.

Nicht jedes Dokument kann ohne genaue Analyse als strukturiert bezeichnet werden. Die Eigenschaft der Strukturiertheit ist abhängig von der logischen Aufteilung der einzelnen Teile eines Dokuments. So ist z. B. ein Textdokument nur dann als strukturiertes Dokument zu bezeichnen, wenn (z. B. durch Auszeichnungen) mindestens zwei voneinander getrennte Dokumentteile identifiziert werden können. Dies kann beispielsweise bei einem Textdokument durch die Aufteilung des Gesamttextes in zwei oder mehrere

Kapitel erfolgen. Ein Dokument, das unterhalb eines „Wurzelementes“ lediglich Text enthält, ist daher nicht als strukturiertes Dokument anzusehen³.

Die Inhaltssicht betrifft das typische Beobachterinteresse an einem Dokument: Der „Betrachter“ möchte den ihm präsentierten „Inhalt“ aufnehmen. Für die logische Sicht und die Layoutsicht, die im Weiteren im Vordergrund stehen, werden gemäß [Schmidt, Strauch 2002] und [Schmidt 2004] die folgenden Annahmen und Definitionen benutzt⁴:

Sowohl die Layoutsicht als auch die logische Sicht modellieren ein Dokument als Baum. Dies führt für beide Sichten zu drei verschiedenen Typen von Objekten gemäß [ISO/IEC 8612]:

1. dem Wurzelknoten eines Dokuments; dieser heißt *document (logical oder layout) root*;
2. einem oder mehreren Endknoten; diese heißen *basic (logical oder layout) objects*;
3. einem oder mehreren Knoten zwischen Wurzelknoten und Endknoten; diese heißen *composite (logical oder layout) objects*.

Jeder Layoutprozess erzeugt aus dem logischen Baum einen Layoutbaum. Im filmischen Fall ist der Layoutprozess also zuständig für die Serialisierung der Inhaltstücke, die über einen filmischen Ausgabedatenstrom mit einem Player dargestellt werden können.

Hinsichtlich der logischen Struktur eines Dokuments werden im Weiteren nur sogenannte strukturierte Dokumente behandelt; alle anderen sind für die Behandlung von komplexen narrativen Strukturen in dieser Arbeit unerheblich. Es ergibt sich als Definition gemäß [Schmidt, Strauch 2002] und [Schmidt 2004]:

Ein Dokument *D* heißt *strukturiert*, wenn seine logische Struktur mindestens zwei *composite logical objects* enthält.

³ An dieser Stelle sei anzumerken, dass den in XML ausgezeichneten Dokumenten sehr häufig die Eigenschaft der Strukturiertheit unterstellt wird.

⁴ Eine ausführliche Behandlung der logischen Sicht und der Layoutsicht auf ein cinematographisches Dokument erfolgt in [Schmidt 2004].

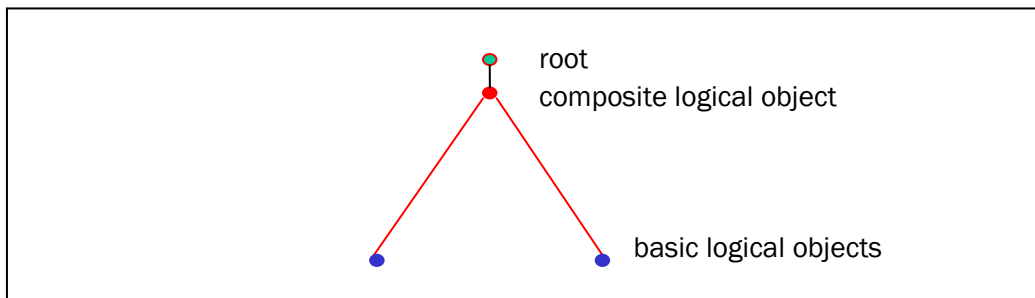


Abbildung 2: Beispiel für ein nicht strukturiertes Dokument.

Das in Abbildung 2 repräsentierte Dokument ist nicht strukturiert, da alle basic logical objects Nachfolger nur eines einzigen composite logical object sind. Ein Beobachter muss nach Identifikation des ganzen Dokuments keine über das composite logical object hinausgehende Aggregationsleistung für die einzelnen basic logical objects vollbringen.

Für ein Musikmedium sind dies (bei einer im Allgemeinen größeren Anzahl von basic logical objects („Stücken“) übliche Strukturen.

Neben den eigentlichen strukturierten Informationseinheiten eines Dokuments sind oft noch andere, zusätzliche Informationen von inhaltlicher Bedeutung. Solche Informationen bezeichnet man als Metadaten eines Dokuments.

2.2 Metadaten

Unter Metadaten („Daten über Daten“) versteht man (im Allgemeinen strukturierte) Daten, mit deren Hilfe eine Informationsressource beschrieben und dadurch unter anderem auch besser auffindbar gemacht wird. Der Begriff existierte zwar bereits vor dem Web-Zeitalter, fand aber vor allem im Zusammenhang mit elektronischen Informationssystemen seine Verbreitung. Von Tim Berners-Lee, Direktor des World Wide Web Consortiums (W3C), stammt die folgende Definition: „Metadaten sind maschinenlesbare Informationen über elektronische Ressourcen oder andere Dinge“ [Lee 1997]. Metadaten werden im Folgenden als Informationen über Dokumente oder Dokumentteile aufgefasst.

Aufgrund der Entwicklung des Internets und des Bestands an jedermann frei zugänglicher Information kommt der genaueren Suche nach bestimmten Informationen eine immer größere Bedeutung zu. Ein Beispiel für den Einsatz von Metadaten ist die Codierung der in einem wissenschaftlichen Aufsatz verwendeten Schlüsselwörter. Weiterhin von Interesse sind z. B. auch die Zusammenfassung des Inhalts eines solchen Dokuments (Summary), Angaben über den Autor, das Dateiformat oder das Erstellda-

tum. Diese Metadaten beziehen sich auf Informationen rund um das Dokument, nicht jedoch spezifisch auf dessen Inhalt und schon gar nicht auf die Organisation des Inhaltes im Dokument. Liegen diese Metadaten dann noch in einer strukturierten Form vor, wäre eine Suche nach speziellen Dokumenteigenschaften möglich und deshalb eine bessere Auffindbarkeit dieser Dokumente hinsichtlich der in den Metadaten repräsentierten Eigenschaften gewährleistet. Da es sich bei solchen Metadaten im Allgemeinen nicht um die eigentlichen Inhaltsdaten des zu beschreibenden Dokuments handelt, sondern um eine „Bibliothekssicht“, bezeichnet man sie als „bibliothekarische Metadaten“ (siehe Abschnitt 2.3).

Ähnlich wie bei Textdokumenten ist auch die Durchsuchbarkeit von Videodokumenten eine für Metadaten prädestinierte Anwendung. Über eine Klassifikation von Schlagwörtern können in einem Videodokument wichtige Informationen, wie z. B. Tag und Uhrzeit der Aufnahme, Dauer und Inhalt in Form von Metadaten elektronisch gespeichert werden. Diese Informationen können mit Hilfe einer Datenbank für die Suche zugänglich gemacht werden. Genauso wie im vorher beschriebenen Beispiel handelt es sich hierbei ebenfalls im Allgemeinen um „bibliothekarische Metadaten“.

Neben der reinen Annotation von ganzen Dokumenten können Metadaten auch zur Verbesserung der Suche nach Dokumentteilen dienen oder zusätzlich Relationen zwischen diesen Dokumentteilen abbilden (siehe Abschnitt 2.4). Liegt z. B. ein Textdokument in einem geeigneten strukturierten Format vor, so können neben dem gesamten Dokument auch Dokumentteile, wie Abschnitte, Sätze oder Wörter, mit Metadaten versehen werden. Ein passendes Beispiel dafür ist wiederum der wissenschaftliche Aufsatz, der am Ende des Dokuments verschiedene Literaturangaben beinhalten kann. Werden nun diese Literaturangaben mit Metadaten versehen, so können diese kontextspezifisch durchsucht und z. B. die Namen aller zitierten Autoren ermittelt werden. Voraussetzung für diese Art von Metadaten für Dokumentteile ist ein strukturierbares Dokument.

Die obigen Beispiele zeigen den Einsatz von Metadaten beim Suchen und Auffinden von Informationen. Damit solche und ähnliche Lösungen weltweit funktionieren können, bedarf es einer gemeinsamen Sprache zur Formulierung von Metadaten sowie einheitlicher Richtlinien, wie diese erstellt werden sollen. R.B. Guha beschreibt in [Guha 1995] solche Richtlinien:

- Die Repräsentation, Manipulation und Speicherung von Metadaten sollte unabhängig von Medientyp oder Datenformat des Dokuments sein.
- Metadaten sollten maschinenverarbeitbar sein. Während Dokumente hauptsächlich von Menschen genutzt werden, sind Metadaten vor allem zur Verarbeitung durch Maschinen bestimmt.

- Die Sprache, mit der Metadaten repräsentiert werden, sollte möglichst starke Strukturierungen zulassen. Je stärker strukturiert Metadaten sind, desto genauer lässt sich der Inhalt einer Ressource beschreiben.
- Die Erstellung von Metadaten sollte von deren Nutzung eindeutig trennbar sein. Das bedeutet, dass Programme, die Metadaten erzeugen, andere sein können als diejenigen, die Metadaten benutzen.
- Eine Sprache für Metadaten sollte Metadaten selbst wiederum als Dokument behandeln können und dadurch eine Annotation durch Metadaten zulassen.
- Metadaten sollten sich – egal aus welcher Quelle sie stammen – zusammenführen lassen.
- Es sollte ein Standardvokabular zur Nutzung von Metadaten existieren.

Die Nutzung einer einheitlichen Sprache für die Repräsentation von Metadaten könnte dazu führen, dass verschiedene Anwendungen zum Auffinden von Informationen gleichermaßen genutzt werden können. Dies wäre wiederum ein Anreiz dafür, überhaupt Metadaten zu erstellen. Ziel ist es also, eine größere Menge von Metadaten in einer einheitlichen Sprache zu erstellen.

Die folgenden Abschnitte 2.3 und 2.4 zeigen solche standardisierten Sprachen zur Beschreibung und Einbettung von Metadaten.

2.3 Bibliothekarische Metadaten

Bibliothekarische Metadaten beschreiben vorwiegend Zusatzinformationen zu einem Dokument und gehen kaum (oder gar nicht) auf dessen eigentlichen Inhalt ein. Für die Anwendung solcher bibliothekarischer Metadaten hat sich der Standard der Dublin Core Metadata Initiative (DCMI)⁵ weltweit etabliert. 1995 rief das Online Computer Library Center (OCLC) zusammen mit dem National Center for Supercomputing (NCSA) den Metadatenstandard Dublin Core (DC) ins Leben. Der Begriff „Core“ steht für einen ausgearbeiteten Kernsatz von zunächst 13 Elementen, die später auf 15 erweitert wurden. Die optional verwendbaren Elemente dienen der Beschreibung von elektronischen Dokumenten.

Ziele der Dublin Core Initiative sind gemäß [Guha 1995]:

- Einfachheit,
- Austauschbarkeit von Metadaten,

⁵ Die DCMI basiert auf der Grundlage einer internationalen Zusammenarbeit von Museumsfachleuten, Bibliothekaren, Vertretern der öffentlichen Verwaltung und kommerziellen Unternehmen sowie Informatikern und Netzwerkspezialisten.

- Verbesserung der Ergebnisse bei der Dokumentendurchsuchbarkeit,
- Entwicklung maschinell verarbeitbarer Semantiken.

Der Dublin Core beschreibt kein eigenes Dateiformat für die Speicherung von Metadaten, sondern macht lediglich Vorgaben für die Angabe der Metadaten. Für jedes zu beschreibende Dokument soll die Angabe von maximal 15 Eigenschafts/Wert-Paaren zur Beschreibung von Informationsressourcen angegeben werden können. Diese lassen sich in vier Kategorien einteilen, die zusammen mit ihren Elementen näher beschrieben werden:

Kategorie „Beiträger und Rechteinhaber“:

Die Eigenschaft DC.PUBLISHER trägt als Inhalt Informationen über den oder die Ersteller der Ressource. Werte können also der Name des Verlegers, ein Herausgeber, eine Person oder eine Organisation sein. Der DC.CREATOR ist eine Person oder Organisation, die für den Inhalt der Ressource verantwortlich ist; üblicherweise ist der Wert der Name des Autors. Mit der Eigenschaft DC.CONTRIBUTORS werden zusätzliche Personen oder Organisationen aufgeführt, die einen Beitrag zu der Ressource geleistet haben. Informationen über Rechte am Inhalt oder an der Ressource selbst werden mit der Eigenschaft DC.RIGHTS annotiert. Typischerweise ist für den Inhalt dieses Elementes eine URI zu einem Urhebervermerk oder ein „Rights-Management“-Vermerk über die rechtlichen Bedingungen zu verwenden.

Kategorie „Vernetzung“:

Die Eigenschaft DC.SOURCE trägt Informationen über Rechte am Inhalt oder an der Ressource selbst. Typischerweise ist für den Inhalt dieses Elementes eine URI zu einem Urhebervermerk oder ein „Rights-Management“-Vermerk über die rechtlichen Bedingungen zu verwenden. Durch das Element DC.RELATION werden Verbindungen zu verwandten oder ähnlichen Ressourcen hergestellt. Für die Anwendung dieses Elementes empfiehlt sich der Einsatz der dazugehörigen Qualifier im qualifizierten Dublin Core.

Kategorie „Lebenszyklus“:

Mit dem Element DC.DATE wird das Datum eines Ereignisses innerhalb der Lebensdauer der Ressource angegeben. Typischerweise ist dies das Datum der Erzeugung der Ressource oder das Datum, an dem die Ressource zugänglich gemacht wurde. Hier wird empfohlen, ein einheitliches Format zu verwenden.

Kategorie „Inhalt“:

Das Element DC.TITEL gibt den Titel der Quelle, also den vom Verfasser, Urheber oder Verleger vergebenen Namen der Ressource an. Das Thema der Ressource bzw. Stichwörter oder Phrasen, die den Inhalt zusammenfassen, werden durch das Element DC.SUBJECT beschrieben. In DC.COVERAGE werden Angaben zur räumlichen Bestimmung und zeitlichen Gültigkeit der Ressource eingetragen. Typischerweise sind dies geografische Koordinaten oder Zeitstempel. Textliche Beschreibungen des Ressourceninhalts werden durch DC.DESCRPTION annotiert. Eine solche Beschreibung kann ein Abstract, ein Inhaltsverzeichnis, eine Inhaltsbeschreibung für grafische Ressourcen oder ein Freitext über den Inhalt der Ressource sein.

Kategorie „Technische Daten“:

Eine eindeutige Referenz zu einer Ressource wird durch DC.IDENTIFIER beschrieben. Hier wird eine Zeichenkette oder Zahl eingetragen, die diese Ressource eindeutig identifiziert, z.B. URIs. Physikalische oder digitale Erscheinungsformen der Ressource werden mit DC.FORMAT angegeben. Typischerweise wird hier der Medientyp oder das Format der Ressource verwendet. Beispiele sind Text/HTML, ASCII, eine Postscript-Datei, eine ausführbare Anwendung, JPEG-Bilddateien, usw. Für dieses Element wird die Verwendung eines kontrollierten Vokabulars empfohlen. Durch DC.TYPE wird die Art oder das Genre des Inhaltes der Ressource beschrieben. Werte für dieses Element können sein: Homepage, Roman, Gedicht, Arbeitsbericht, technischer Bericht, Essay, Wörterbuch, usw. In DC.LANGUAGE wird die Sprache des Ressourceninhalts eingetragen.

Der DC beschreibt lediglich die Eigenschaften der zu codierenden Metadaten, nicht jedoch das Format der Einbettung. Für verschiedene Dateiformate haben sich deshalb auch verschiedene Einbettungsmöglichkeiten ergeben.

Die wohl am weitesten verbreitete Anwendung des DC ist dessen Einbettung in HTML-Dokumente. Durch die Aufteilung des HTML-Formates in Head- und Body-Bereich gestaltet sich die Einbettung des DC denkbar einfach. Die DTD von HTML lässt im Head-Bereich des Dokuments das Element `Meta` mehrfach zu. Das folgende Beispiel zeigte die Annotation einer Webseite mit dem DC:

```
<head>
  <meta name="DC.Title" content="Backen mit Hildegard">
  <meta name="DC.Creator" content="Johanna Lang">
  <meta name="DC.Subject" content="Backrezepte">
  <meta name="DC.Description" content="Backrezepte nach Hildegard von Bingen">
  <meta name="DC.Publisher" content="Back und Co">
  <meta name="DC.Contributor" content="Simone Hoffmann">
  <meta name="DC.Date" content="2006-01-15">
  <meta name="DC.Type" content="Text">
  <meta name="DC.Format" content="text/html">
```

```
<meta name="DC.Identifier" content="http://www.prohildegard.de">
<meta name="DC.Language" content="de">
<meta name="DC.Coverage" content="Freudenstadt">
<meta name="DC.Rights" content="Alle Rechte liegen beim Autor">
</head>
```

Durch die Optionalität der einzelnen DC-Elemente wird ein HTML-Dokument mit Metadaten angereichert und kann so mit Hilfe einer geeigneten Suchmaschine besser indexiert werden.

Neben der Möglichkeit Webseiten zu annotieren, wird der DC vor allem im Bereich von Textdokumenten, wie z. B. bei Zeitschriftenartikeln, Monographien oder Dokumentationen eingesetzt. Bekanntester Vertreter der Strukturierung solcher Dokumente in XML ist der Dokumenttyp „DocBook“, der von der Organization for the Advancement of Structured Information Standards (OASIS) gepflegt wird. Wegen der großen Verbreitung von DocBook für die Erstellung von Publikationen bietet der Standard auch die Annotation von DC-Metadaten an⁶.

```
<book status="text" lang="de">
  <bookinfo>
    <title>Backen mit Hildegard</title>
    <author>
      <firstname>Johanna</firstname>
      <surname>Lang</surname>
    </author>
    <subjectset>
      <subject>
        <subjectterm>Backrezepte</subjectterm>
      </subject>
    </subjectset>
    <abstract>
      <para>Backrezepte nach Hildegard von Bingen</para>
    </abstract>
    <publisher>
      <publishername>Back und Co</publishername>
    </publisher>
    <othercredit>
      <firstname>Simone</firstname>
      <surname>Hoffmann</surname>
    </othercredit>
    <date>2006-01-15</date>
    <biblioid>http://www.prohildegard.de/book.xml</biblioid>
    <bibliocoverage>Freudenstadt</bibliocoverage>
    <copyright>
      <year>2006</year>
      <holder>Autor</holder>
    </copyright>
  </bookinfo>
  ...
</book>
```

⁶ Zur Codierung der DC Elemente in Docbook dient das tabellarische Mapping von DC-Elementen und den korrespondierenden Docbook-Elementen, das unter <http://www.docbook.org/specs/cs-docbook-docbook-4.2.html#d0e652> zu finden ist.

Die Codierung von DC-Metadaten sowohl in HTML als auch in der XML-Anwendung „DocBook“ benutzt als Einbettungsformat das jeweilige Dateiformat des Dokumentinhaltes. Dies funktioniert deshalb so problemlos, weil es sich bei beiden Formaten um Textformate handelt, die mit relativ geringem Aufwand editiert werden können. Anders verhält sich die Codierung von DC-Metadaten bei Nicht-Textdokumenten wie z. B. Video- oder Audiodaten. Die Einbettung solcher Metadaten in das Binärformat des Dokumentinhalts funktioniert nur in Übereinstimmung mit dem jeweiligen Codierungsstandard des Datenformats und unter Zuhilfenahme eines geeigneten Einbettungswerkzeugs.

Für die Annotation von Nicht-Textdokumenten im speziellen, aber auch zur Annotation aller eindeutig identifizierbarer Ressourcen, kann das Format RDF (Resource Description Framework) zur Einbettung der DC-Elemente verwendet werden.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description about="http://www.prohildegard.de/imagevideo.mpg">
    <dc:title>Backen mit Hildegard</dc:title>
    <dc:creator>Johanna Lang</dc:creator>
    <dc:subject>Imagevideo</dc:subject>
    <dc:description>Imagevideo von Prohildegard</dc:description>
    <dc:publisher>Back und Co</dc:publisher>
    <dc:contributor>Simone Hoffmann</dc:contributor>
    <dc:date>2006-01-20</dc:date>
    <dc:type>Video</dc:type>
    <dc:format>mpg</dc:format>
    <dc:identifier>http://www.prohildegard.de</dc:identifier>
    <dc:language>de</dc:language>
    <dc:coverage>Freudenstadt</dc:coverage>
    <dc:rights>Alle Rechte liegen beim Autor</dc:rights>
  </rdf:Description>
</rdf:RDF>
```

Das Format RDF (hier in XML serialisiert) ermöglicht es durch das Element `Description` und das Attribut `about`, auf beliebige Ressourcen Bezug zu nehmen und diese z. B. mit DC-Metadaten zu versehen.

2.4 Metadaten für Teildokumente

Textdokumente besitzen gegenüber binären Dokumenten den großen Vorteil, dass deren Inhalte automatisch indiziert werden können, so dass z. B. ein Wortindex generiert werden kann. Handelt es sich zudem noch um strukturierte (z. B. in XML codierte) Dokumente, so ergibt sich häufig die Besonderheit, dass potenzielle Metadaten schon im Dokument selbst enthalten sind und ohne Probleme extrahiert werden können. Als Beispiel sei hier der bereits erwähnte Fachartikel genannt, der neben den inhaltlichen Informationen im Fließtext auch die Namen der zitierten Autoren beinhaltet.

Durch die im Allgemeinen offene Architektur von strukturierten Textdokumenten ist es zudem möglich, auf unterschiedlichen Strukturebenen unterschiedliche Metadaten zu annotieren. Als Beispiel dient das folgende mit DocBook strukturierte Dokument.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE part PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN"
"http://www.oasis-open.org/docbook/xml/4.2/docbookx.dtd">
<part>
  <title>Grundlagen</title>
  <chapter>
    <title>Einführung</title>
    <sect1>
      <title>Geschichte von XSL-FO</title>
      <para>Als 1990 erstmals...</para>
    </sect1>
  </chapter>
  <article class="specification">
    <title>XSL-FO Spezifikation</title>
    <sect1>
      <title>Primer</title>
      <para>Ein Primer beschreibt...</para>
    </sect1>
  </article>
</part>
```

Ein Buch kann in DocBook aus verschiedenen Teilen (`part`) bestehen, diese selbst wiederum Kapitel (`chapter`) oder Artikel (`article`) umfassen. Durch diesen Aufbau des Dokuments ergibt sich der Bedarf nach unterschiedlichen Metadaten für Teildokumente. Obwohl `chapter` und `article` lediglich Strukturen von Dokumentteilen kennzeichnen, lässt DocBook unterschiedliche Möglichkeiten für die Annotation von Metadaten zu. Während ein Kapitel in DocBook nicht genauer spezifiziert werden kann, ermöglicht DocBook jedoch die detaillierte Klassifikation eines Artikels. Der Inhalt des `class`-Attributs gibt den genauen Typ eines Artikels an. Dieses Beispiel zeigt sehr deutlich den Bedarf nach unterschiedlichen Metadaten für unterschiedliche Dokumentteilstrukturen.

Wesentlich schwieriger gestaltet sich eine solche Metadaten-Annotation für Nicht-Textdokumente wie z. B. Videos, Bilder oder Audiodateien. Binärdaten enthalten als Inhalt zwar die für eine Suche relevanten Informationen, jedoch nicht so aufbereitet, dass sie von einer textuellen Suchanfrage genutzt werden können. Die dazu nötigen Metadaten werden also nicht in den Daten des Inhalts selbst vorgehalten, sondern müssen beschreibend außerhalb oder in einem festgelegten Abschnitt innerhalb des Binärdokuments definiert sein. Hilfreich an dieser Stelle kann wiederum ein extern vorgehaltenes Metadatendokument sein. Sollen nun aber strukturabhängig Metadaten annotiert werden, so bedarf es zuerst einer Klassifikation dieser Strukturen innerhalb von Binärdokumenten, vergleichbar der Strukturierung eines DocBook-Dokuments in Teile, Kapitel und Artikel. Für Filmdokumente existieren dazu nur erste

Klassifikationsschemata (vgl. [Hunter 1999], [Schmidt, Strauch 2002] und [Metz 1972]). Abbildung 3 zeigt einen ersten Ansatz zur Strukturierung von Videodokumenten und zur Annotation der einzelnen Strukturelemente mit Metadaten.

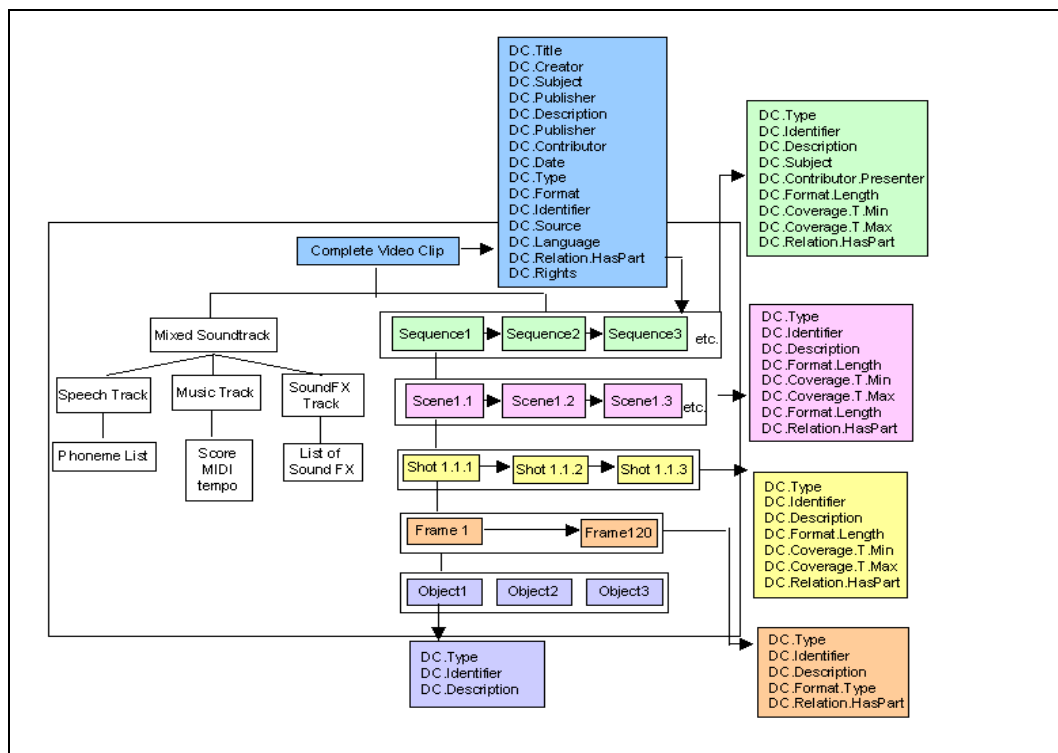


Abbildung 3: Hierarchische Strukturen eines Videodokuments mit DC-Metadaten (angelehnt an [Hunter 1999, dort Abb. 1]).

Die für diese Arbeit wichtige strukturelle Gliederung der Video-Informationen besteht aus den Teilen Sequenz (Sequence), Szene (Scene), Einstellung (Shot), Frame und Objekt (Object)⁷. Vom obersten Strukturelement des „Complete Video Clips“ bis hin zur atomaren Einheit eines Objektes wird von Hunter alles mit dem DC-Metadatenatz beschrieben. Damit wird auch für ein Videodokument der DC nutzbar gemacht, wie dies bereits für DocBook schon gezeigt wurde (allerdings natürlich ohne Einbettung in die Dokumenteninstanz). Insgesamt ist dies aber deshalb noch nicht ausreichend, da über die einfachen bibliothekarischen Metadaten wie z. B. DC.Type oder DC.Description in einem Videodokument noch mehr Informationen relevant sein

⁷ Die von Hunter verwendeten Begriffe der „Scene“ und „Sequence“ sind in ihrer Struktur und Verwendung von den in Abschnitt 6.3 definierten Begriffe der Syntagmen der Szene und Sequenz zu unterscheiden.

können, als mit DC beschrieben werden können. Beispielsweise müssen bei der Beschreibung eines Objektes völlig andere Metadaten zu Grunde gelegt werden als für die Beschreibung einer Szene, da diese aus komplett anderen Substrukturen aufgebaut ist und grundsätzlich verschiedene Eigenschaften hat.

An den vorangegangenen Beispielen zeigen sich für die Annotation von Metadaten für Videodokumente zwei grundlegende Forschungsaufgaben. Zum einen bedarf es eines verbindlichen Strukturschemas für den Aufbau von Videodokumenten. Eine Überarbeitung und Formalisierung von Filmgrammatiken wurde von Schmidt und Strauch in [Schmidt, Strauch 2002] und [Schmidt 2004] vorgenommen. Die Frage, ob ein Schema in dieser Form ausreicht, ist elementarer Bestandteil von Teil II. Zum anderen ist eine detailliertere Möglichkeit nötig, Unterstrukturen von Filmdokumenten mit Metadaten zu versehen, als dies mit dem DC möglich ist. Gut geeignet dafür ist ein Metadatenformat, das zum einen eigene Definitionen von filmspezifischen Eigenschaften liefert, zum anderen leicht erweiterbar ist, um nicht starr auf vordefinierte Strukturen vertrauen zu müssen. Der Standard MPEG-7 bietet diese Möglichkeiten (siehe Abschnitt 4.6).

3 XML – Extensible Markup Language

XML ist einer der bedeutendsten Web-Standards des World Wide Web Consortiums (W3C) und wird in verschiedensten Anwendungsgebieten für die Präsentation und den Austausch von Daten verwendet. Nachdem die grundlegende Idee und die syntaktischen Grundlagen dieser Sprache in Abschnitt 3.1 vorgestellt werden, folgt in Abschnitt 3.2 die Beschreibung des syntaktischen Aufbaus von XML gemäß der W3C-Spezifikation. Abschnitt 3.3 führt dann die Regelsprache DTD (Document Type Definition) ein. Dabei werden die sprachlichen Mittel für die Beschreibung sowohl von Text- als auch von Nicht-Text-Strukturen, um die es in dieser Arbeit geht, an einem Beispiel vorgestellt.

3.1 Grundlagen

XML ist seit 1996 ein Standardisierungsprojekt des W3-Consortiums und eine Weiterentwicklung der umfangreichen und schwer zu implementierenden SGML-Spezifikation aus dem Jahre 1986. XML hat sich als universelle Metasprache zum Austausch, Retrieval und zur Verwaltung von Dokumenten entwickelt.

XML ist eine leicht verarbeitbare formale Metasprache zur Erzeugung von Auszeichnungssprachen, ohne jedoch selbst eine Auszeichnungssprache zu sein. Das bedeutet, dass XML kein Vokabular zur Auszeichnung von Dokumentinhalten bereitstellt. Zusätzlich regelt die XML-Spezifikation die Anwendung dieser Auszeichnungssprachen, also die Auszeichnung konkreter Dokumente. So gehören zum Sprachumfang einerseits die Deklarationsregeln zur Definition von Dokumenttypen, andererseits die Regeln für die Auszeichnung von Dokumenten. Die Deklarationsregeln (Dokumenttyp-Definitionen (DTD) beschreiben, wie ein Dokument des entsprechenden Typs auszuzeichnen ist. Die Auszeichnungsregeln schreiben vor, mit welchen Mitteln dies zu tun ist.

Die Flexibilität und Modularität von XML führte zu zahlreichen Entwicklungen im Bereich der XML-Anwendungen, die Erdmann in zwei Gruppen einteilt (vgl. [Erdmann 2001, S. 15]):

- XML für die Datenrepräsentation,
- XML für das Publishing.

Anwendungen von XML zur Datenrepräsentation finden sich beispielsweise im Bereich des E-Commerce. Die Grundidee ist dabei, dass Anwendungen z. B. aus dem B2B-Bereich Daten austauschen. XML dient dabei als ideale Schnittstellen- und Protokollsprache.

Der zweite Einsatzbereich von XML liegt im Bereich des elektronischen Publizierens. Hier stellen die verwendeten XML-Dateien den wesentlichen Inhalt und die Grundstruktur des zu publizierenden Informationsbestandes (siehe Kapitel 2). Die Trennung von Inhalt und Layout spielt dabei eine große Rolle, da dadurch eine medienneutrale Datenhaltung nach dem Single-Source-Prinzip grundsätzlich möglich ist. Egal für welches Ausgabemedium publiziert wird, erfolgt die Generierung des fertigen Dokuments aus einer einzigen Quelle. Die Anpassung der Layout-Komponente erfolgt durch Stylesheets, die während des Publikationsvorgangs auf den Quelldatenbestand angewandt werden.

Zusätzlich zur Anwendung in den Bereichen Datenrepräsentation und Publishing wird XML auch zur reinen Annotation von Dokumenten verwendet. Verbreitete Metadaten-sprachen nutzen XML als Serialisierungssprache (siehe Kapitel 2 und Abschnitt 4.6).

3.2 Syntaktischer Aufbau

Die XML 1.0 Spezifikation [Bray et al. 2006] definiert die Syntax von XML-Dokumenten und legt fest, dass ein XML-Dokument aus einer Folge von Unicode-Zeichen besteht. Unicode ist ein ISO-Standard (ISO/IEC 10646-1:2000) zur Vereinheitlichung von Zeichencodierungen. Durch den Einsatz von Unicode ist es möglich, einheitlich auf verschiedene Alphabete und Sonderzeichen zuzugreifen. Dies führt besonders beim Einsatz im Web zu einer hohen Akzeptanz von XML weltweit.

Jedes XML-Dokument enthält in Unicode encodierte Daten und Metadaten. Die Ausdrücke in spitzen Klammern „<...>“ repräsentieren Metadaten, durch die Inhalte näher beschrieben werden können. Damit sowohl der Beginn als auch das Ende der Inhaltsbeschreibung durch diese Metadaten abgegrenzt werden können, werden in XML sogenannte Start- und Endetags verwendet. Insgesamt gilt gemäß [Bray et al. 2006]⁸:

```
[40] STag ::= '<' Name (S Attribute)* S? '>'
```

```
[42] ETag ::= '</' Name S? '>'
```

Start- und Endetags können also Inhalte strukturieren und näher beschreiben, aber auch leer sein. Produktion 39 bildet aus Starttag, Endetag und Inhalt ein so genanntes „Element“:

```
[39] element ::= EmptyElemTag | STag content Etag
```

⁸ Zur besseren Übersicht werden die Nummern der Produktionen so verwendet, wie sie sich auch in der XML-Spezifikation [Bray et al. 2006] finden.

Ein Start-Tag (z. B. `<von>`) kennzeichnet den Beginn eines Elementes, ein Ende-Tag (z. B. `</von>`) dessen Ende.

Der gesamte Inhalt zwischen Start- und Ende-Tag wird gemäß Produktion 39 als Content des Elementes bezeichnet. Besitzt ein Element keinen Content, so kann man ein Empty-Element-Tag verwenden, bei dem das Ende-Tag weggelassen wird und dafür am Ende des Start-Tags ein „/“ eingefügt wird. Ein Beispiel dafür ist das Empty-Element-Tag `
`, das sich auch in XHTML (als eine der wichtigsten XML-Anwendungen) findet.

Die Inhalte eines XML-Dokuments folgen einem Dokumentenprolog und sind eingerahmt durch ein besonderes XML-Element, das schon in der sogenannten Doctype-Deklaration als Dokument-Typ deklariert werden kann. Dieses sogenannte Root-Element (Wurzelement) markiert typischerweise auch das Ende des Dokuments. Danach können noch beliebige Kommentare, Prozessoranweisungen und Weißräume folgen. Insgesamt gilt gemäß [Bray et al. 2006]:

```
[1] document ::= prolog element Misc*
[27] Misc    ::= Comment | PI | S
```

Ein Prolog beginnt optional mit einer XML-Deklaration, die typischerweise folgendermaßen aussieht: `<?xml version="1.0" encoding="UTF-8"?>`. Hier werden die Versionsnummer der verwendeten XML-Spezifikation sowie die Codierung der Textzeichen angegeben⁹.

Im Anschluss an die XML-Deklaration kann eine Dokument-Typ-Deklaration (DOCTYPE) folgen, die einen Verweis auf eine externe Dokument-Typ-Definition (DTD) enthält oder diese beinhaltet. Damit beschreibt sie eine Grammatik, die zur syntaktischen Überprüfung eines XML-Dokuments verwendet werden kann (Bedeutung und Syntax siehe Abschnitt 3.3.1). Die Dokument-Typ-Deklaration leistet auch die Verknüpfung zwischen Deklarationsregeln und Daten über den Namen des oben angegebenen Root-Elementes.

```
[22] prolog    ::= XMLDecl? Misc* (doctypedec1 Misc*)?
[28] doctypedec1 ::= '<!DOCTYPE' S Name (S ExternalID)?
                  (' [' (markupdec1 | DeclSep)* ']' S?)? '>'
```

Auf den Prolog folgt der eigentliche Inhalt des XML-Dokuments, bestehend aus Daten und Metadaten. Das Root-Element kann und wird im Allgemeinen weitere Elemente enthalten, die wiederum Elemente enthalten können usw., was zu einer Baumstruk-

⁹ UTF-8 steht für Unicode Transformation Format mit 8 Bit.

tur eines XML-Dokuments führt. Für den Inhalt eines XML-Dokumentes innerhalb eines Elementes, beginnend mit dem Wurzelement, gilt¹⁰:

```
[43] content ::= CharData? ((element | Reference | CDSect |
                             PI | Comment) CharData?)*
```

Im folgenden Beispiel findet sich zunächst die XML-Deklaration und eine Dokument-Typ-Deklaration, wobei hier mit dem Schlüsselwort `System` und dem nachfolgenden „`email.dtd`“ ausgedrückt wird, dass lokal (und sogar im selben Verzeichnis) in einem File `email.dtd` eine DTD zur Prüfung dieses Dokumentes bereit steht. Man beachte, dass der Name des Doctypes `email` dem Namen des Root-Elementes in spitzen Klammern entspricht. Die weiteren Elemente lauten: `datum`, `von`, `an`, `betreff` und `text`.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email SYSTEM "email.dtd">
<email>
  <von>hough@johnnson.com</von>
  <an>amueller@wein.de</an>
  <betreff>Weinstein</betreff>
  <text>
    Lieber Albert, der 88er Rasteaux hat sehr viel Depot, was sagst du dazu?<br/>
    Grüße<br/>
    Hough<br/>
  </text>
</email>
```

Eine zusätzliche Struktur innerhalb eines Start-Tags kann durch sogenannte Attribute definiert werden. Attribute haben die Form:

```
[41] Attribute ::= Name Eq AttValue
```

Im folgenden Beispiel enthält das Element `datum` ein Attribut mit dem Namen `tag`, das wiederum den Attribut-Wert „`06.06.2006`“ enthält. Die verschiedenen Arten von Attributen werden in Abschnitt 3.3.2 beschrieben.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email SYSTEM "email.dtd">
<email>
  <datum tag="06.06.2006" zeit="12:10"/>
  <von>hough@johnnson.com</von>
  <an>amueller@wein.de</an>
  <betreff>Weinstein</betreff>
  <text>
    Lieber Albert, der 88er Rasteaux hat sehr viel Depot, was sagst du dazu?<br/>
    Grüße<br/>
    Hough<br/>
  </text>
</email>
```

¹⁰ Produktion 39 stellt zusammen mit Produktion 40 die einzige rekursive Regel in der XML-Norm dar.

In der Produktion 43 benennt „CharData“ die Möglichkeit, Textdaten einem XML-Dokument als Inhalt direkt hinzuzufügen. Diese werden zwischen den Start- und Ende-Tags notiert. Für den Text können bis auf wenige Ausnahmen alle Zeichen aus dem jeweiligen Standard, der beim `encoding` angegeben wird, verwendet werden. Einige Zeichen jedoch werden von XML z. B. zur Kennzeichnung von Tags und Entities verwendet, was zu einer besonderen Auszeichnung dieser speziellen Zeichen durch sogenannte Zeichen-Entities führt. Soll beispielsweise das Zeichen `<` (kleiner) als Inhalt und nicht als Markup für ein Tag verwendet werden, so muss es als Entity `<` (less than) repräsentiert werden. Gleiches gilt für das Zeichen `&`, das in XML als Entity `&` (ampersand) repräsentiert werden muss.

Soll der Inhaltsbereich eines Elementes trotzdem derartige, eigentlich nicht erlaubte, Zeichen enthalten, so können gemäß Produktion 43 sogenannte CDATA-Abschnitte benutzt werden, um die Interpretation dieser Zeichen als Markup zu verhindern. CDATA-Abschnitte entsprechen folgender Syntax:

```
[18] CDSect ::= CDStart CData CDEnd
[19] CDStart ::= '<![CDATA['
[20] CData   ::= (Char* - (Char* ']]>' Char*))
[21] CDEnd   ::= ']]>'
```

Folgendes Beispiel ist erst durch einen CDATA-Abschnitt gemäß der XML-Spezifikation syntaktisch korrekt:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email SYSTEM "email.dtd">
<email>
  <datum tag="06.06.2006" zeit="12:10"/>
  <von>hough@johnnson.com</von>
  <an>amueller@wein.de</an>
  <betreff>Weinstein</betreff>
  <text>
    Lieber Albert, der 88er Rasteaux hat sehr viel Depot, was sagst du dazu?<br/>
    Grüße<br/>
    Hough<![CDATA[Fa. Wine AG & Co]]><br/>
  </text>
</email>
```

Das erste Beispiel wurde nun um einen CDATA-Abschnitt erweitert, der ein Zeichen enthält, das außerhalb des CDATA-Abschnitts zu einem Syntaxfehler führen würde.

Ebenso können gemäß Produktion 43 Kommentare (Comments) und Prozessoranweisungen (Processing Instructions, PIs) als Bestandteil eines XML-Dokuments verwendet werden. Kommentare (Syntax: `<!-- ... -->`) dienen der Kommentierung von XML-Dokumenten, beispielsweise zum leichteren Verständnis der Strukturen durch Anmerkungen, und sind für den Menschen als Leser des Dokuments bestimmt.

Die Spezifikation legt für Kommentare folgende Produktionsregel fest:

```
[15] Comment ::= '<!--' ((Char - '-' ) | ('-' (Char - '-')))* '-->'
```

Innerhalb der Kommentarzeichen sind also beliebige Zeichen zugelassen, mit Ausnahme von zwei aufeinander folgenden Trennstrichen.

Durch Prozessoranweisungen (PI's) werden für bestimmte Anwendungen spezielle Anweisungen innerhalb eines XML-Dokuments verankert.

Die Syntax einer PI lautet:

```
[16] PI ::= '<?' PITarget (S (Char* - (Char* '?>' Char*))?) '?>'  
[17] PITarget ::= Name - (('X' | 'x') ('M' | 'm') ('L' | 'l'))
```

PIs werden von „<?“ und „?>“ eingeschlossen. Eine PI enthält einen Zielbereich, der die Zugehörigkeit zur Applikation, für die diese PI bestimmt ist, regelt. Außerdem enthält sie einen Wertebereich, in dem die eigentliche Instruktion stehen kann. Für den Namen der Applikation sind jedoch alle Namensbildungen, die das Wort „XML“ in irgendeiner Form enthalten, ausgeschlossen.

Eine XML-Instanz bietet an verschiedenen Stellen die Möglichkeit, Nicht-Text-Daten (Daten, die nicht direkt vom Menschen gelesen werden sollen) unterzubringen: z. B. in den Werten eines Attributs oder in CDATA-Abschnitten. Die Möglichkeit, Nicht-Text-Daten kontrolliert in XML-Dokumente einzuarbeiten, bietet das Entity-Konzept in DTDs, das im Folgenden erläutert wird.

3.3 DTD – Document Type Definition

Im vorherigen Abschnitt wurde der syntaktische Aufbau von XML-Dokumenten beschrieben. Genügt ein XML-Dokument dabei den Regeln der XML-Spezifikation, so bezeichnet man es als wohlgeformt im Sinne syntaktischer Korrektheit.

Zusätzlich zur syntaktischen Korrektheit ist es möglich, Regeln zu formulieren, denen XML-Dokumente als Instanzen entsprechen müssen. Eine wichtige Regelsprache für XML-Dokumente, die bereits in XML 1.0 formuliert ist, erlaubt es, dafür Dokument-Typ-Definitionen (DTDs) festzulegen. Enthält ein Dokument einen Verweis auf eine DTD (oder enthält dieses Dokument selbst die DTD) und genügt dieses Dokument den in der DTD enthaltenen Regeln, so wird es auch als „valide“ Instanz bezeichnet. Validität schließt dabei Wohlgeformtheit ein.

Eine DTD deklariert Elemente, Attribute, Entities und Notationen in einer an die EBNF (Erweiterte Backus-Naur-Form) angelehnte Syntax. Die grundlegenden Regeln zu deren Festlegung werden im Folgenden in der angegebenen Reihenfolge erläutert.

DTDs erhalten für Elemente, Attribute, Entities und Notationen die Schlüsselworte

- ELEMENT,
- ATTLIST,
- ENTITY,
- NOTATION.

Davon sind die ersten beiden (ELEMENT und ATTLIST) für die Festlegung der logischen Struktur verantwortlich. Das Schlüsselwort ENTITY wird verwendet, um dokumentenintern und -extern ein Text- oder binäres Objekt anzulegen, auf das Bezug genommen werden kann. Alle Textentities sind aus Sicht einer XML-Instanz im Prinzip vermeidbar. Binäre Entities hingegen können nicht ersetzt werden und ergänzen XML als reines Textformat, da mit ihnen wenigstens Bezüge auf Binärdaten möglich sind. Mit dem Schlüsselwort NOTATION gibt es die Möglichkeit, auf kontrollierte Art und Weise Binärdatenarchitekturen mit „ihren“ Anwendungen zu verknüpfen.

3.3.1 Elemente

Der Großteil der Struktur einer XML-DTD wird durch die Definition von Elementen gebildet, die den beiden folgenden Produktionen entsprechen:

```
[45] elementdecl ::= '<!ELEMENT' S Name S contentspec S? '>'  
[46] contentspec ::= 'EMPTY' | 'ANY' | Mixed | children
```

Die Elementdeklaration beginnt mit „<!ELEMENT“, gefolgt vom Namen des Elementes sowie dessen Inhaltsmodell, das angibt, welche Struktur in einer zugehörigen XML-Instanz unterhalb des deklarierten Elementes zulässig ist.

Im einfachsten Fall ist das Element als leer definiert. Das Inhaltsmodell besteht dann aus dem Schlüsselwort EMPTY. Soll das Element hingegen Kindelemente enthalten, so umfasst die DTD eine Definition zum Auftreten und zur Anordnung dieser Kindelemente durch die Verwendung einer Elementgruppendefinition mit runden Klammern sowie Kommata zur Abtrennung der Elemente, wie in folgendem Beispiel:

```
<!ELEMENT email (datum, von, an, betreff, text)>
```

Das Element `email` besteht aus einer festgelegten Reihenfolge der Elemente `von`, `an`, `betreff` und `text`. Für diese Elemente werden dann wiederum eigene Regeln definiert.

Neben den Regeln für die sequenzielle Anordnung von Elementen durch Kommata lässt sich in DTDs der Oder-Operator (geschrieben als senkrechter Strich „|“) verwenden. Während bei der Verwendung von Kommata die Reihenfolge des Auftretens der jeweiligen Elemente geregelt ist, definiert der Oder-Operator, dass die abgetrennten Elemente wahlweise vorkommen dürfen.

Zur Steuerung der Häufigkeit des Auftretens von Elementen oder Elementgruppen hält die DTD drei Häufigkeitsoperatoren (Quantoren) bereit:

- Das Fragezeichen „?“ deklariert ein Element oder eine Elementgruppe als optional.
- Das Pluszeichen „+“ definiert die Wiederholung desselben Elementes. Dabei muss das Element mindestens einmal im XML-Dokument auftreten.
- Der Stern „*“ erlaubt die Wiederholung ein und desselben Elementes. Dabei muss das Element nicht zwingend im XML-Dokument auftreten.

Die Anwendung von Häufigkeitsoperatoren auf Elementgruppen sowie die Verschachtelung von Elementgruppen lässt eine große Vielfalt an komplexen Inhaltsmodellen für Elemente zu.

Gemäß Produktion 46 kann der Inhalt eines Elementes auch aus einem gemischten Inhalt (Mixed Content) bestehen. Ein Elementtyp hat einen gemischten Inhalt, wenn ein entsprechendes Element Textdaten mit optionalen Kindelementen enthalten darf. Im einfachsten Fall besteht die Deklaration von gemischten Inhalten nur aus Text (#PCDATA: „parsed character data“), wie in diesem Beispiel:

```
<!ELEMENT betreff (#PCDATA)>
```

Insgesamt ergibt sich als möglicher Element-Bestandteil einer DTD für das E-Mail-Beispiel:

```
<!ELEMENT email (datum, von, an, betreff, text)>
<!ELEMENT datum EMPTY>
<!ELEMENT von (#PCDATA)>
<!ELEMENT an (#PCDATA)>
<!ELEMENT betreff (#PCDATA)>
<!ELEMENT text (#PCDATA|br)*>
<!ELEMENT br EMPTY>
```

Eine letzte Möglichkeit besteht in der Verwendung von `ANY` als Inhaltsmodell. Hierbei sind alle in der DTD definierten Elemente sowie `#PCDATA` in beliebiger Reihenfolge erlaubt.

Insgesamt bietet die Elementstruktur eines XML-Dokumentes mit dem Schlüsselwort `EMPTY` eine Standard-Möglichkeit, Nicht-Text-Daten an ein Element anzubinden. Die Verknüpfung erfolgt dann mit einem Attribut aus einer zugehörigen Attributliste.

3.3.2 Attribute

Einen zweiten Bestandteil von DTDs bildet die Deklaration von Attributen. Eine Listen-Deklaration für Attribute eines Elementes beginnt mit `<!ATTLIST . . .>`, gefolgt vom Namen des korrespondierenden Elementes sowie der Attributdefinition.

```
[52] Attlist ::= '<!ATTLIST' S Name AttDef* S? 'V>'
```

Eine Attributdefinition

```
[53] AttDef ::= S Name AttType S DefaultDecl
```

besteht aus einem der Attributtypen CDATA, ID, IDREF, IDREFS, ENTITY, ENTITIES, NMTOKEN, NMTOKENS einem Enumerationstyp oder aus dem Typ NOTATION. CDATA steht dabei für beliebige Zeichenketten. Ein ID-Attribut verleiht einem Element eine eindeutige Identifikation. Der Wert eines ID Attributs kann daher nur ein einziges Mal in einem XML-Dokument als Attributwert vorkommen. Ein Attribut vom Typ IDREF darf nur einen Wert besitzen, der mit einem Wert von in der Instanz verwendeten ID-Attributen korrespondiert. IDREFS kann mehrere dieser Werte durch Whitespace getrennt aufnehmen und dadurch gleichzeitig auf mehrere ID's verweisen. Durch die Angabe eines Attributs vom Typ ENTITY darf in der Dokumentinstanz nur auf einen durch eine Entity vordefinierten Wert verwiesen werden. Mit dem Typ ENTITIES ist die Wertemenge nicht nur auf eine einzelne Entity beschränkt, sondern kann mehrere beinhalten. Der Typ NMTOKEN stellt eine Einschränkung der Zeichendaten des Typs CDATA dar. Attributwerte vom Typ NMTOKEN beschränken sich auf Buchstaben, Ziffern und die Zeichen Punkt, Doppelpunkt, Bindestrich, Unterstrich. Sie dürfen keine Leerzeichen enthalten. NMTOKENS kann mehrere Werte vom Typ NMTOKEN beinhalten.

Attribute können auch durch Auflistung aller zulässigen Attributwerte (eingeschlossen in runde Klammern und getrennt durch das Oder-Zeichen „|“) als sogenannte Enumeration spezifiziert werden. Dabei müssen die Attributwerte den Vorgaben eines Typs NMTOKEN entsprechen und dürfen daher z.B. keine Leerzeichen enthalten. Neben den „normalen“ Werten einer Enumeration lassen sich mit dem Attributtyp NOTATION auch Attributwerte als Enumeration spezifizieren, denen eine besondere Bedeutung zukommt. Dabei dürfen nur Werte verwendet werden, die durch das Schlüsselwort NOTATION (siehe Abschnitt 3.3.4) in der DTD näher spezifiziert sind¹⁴.

Die dritte Komponente einer Attributdeklaration ist gemäß Produktion 53 die Default-Deklaration. Diese legt fest, ob ein Attribut zwingend (#REQUIRED) oder nur optional anzugeben ist (#IMPLIED), oder ob ein Default-Wert angenommen wird.

¹⁴ Durch die kombinierte Verwendung des Schlüsselwortes NOTATION mit dem Attributtyp NOTATION ist es möglich, binäre Inhaltsarchitekturen mit speziellen Metadaten zu versehen.

Das erweiterte Beispiel der E-Mail-DTD zeigt die Verwendung von Attributen beim Element `datum`:

```
<!ELEMENT datum EMPTY>
<!ATTLIST datum
  tag CDATA #REQUIRED
  zeit CDATA #REQUIRED
>
```

Die Attribute `tag` und `zeit` sind erforderlich und dürfen beliebigen Text beinhalten.

3.3.3 Entities

Zur physikalischen Strukturierung von Dokumenten bietet die XML-Spezifikation die Definition von Speicherungseinheiten, sogenannten Entities, an. Mit Hilfe von Entities lässt sich ein Dokument so aufteilen, dass die Struktur und der eigentliche Inhalt oder Inhaltsteil in verschiedenen Dokumenten oder Dokumentbereichen getrennt gespeichert werden können. „Vertikal“ können Entities in zwei Arten unterteilt werden: in Parameter-Entities, deren Gültigkeit sich auf die DTD beschränkt, und allgemeine Entities. Parameter-Entities dienen der Modularisierung innerhalb der DTD und sind hier von untergeordnetem Interesse. Allgemeine Entities werden in DTDs deklariert und in interne und externe Entities unterteilt.

Interne Entities sind Speicherungseinheiten für Text, wobei es für den eigentlichen Inhalt kein separates physikalisches Speicherobjekt (Datei) gibt. Der Inhalt der Entity wird in der DTD vorgegeben, wie in

```
<!ENTITY hj "Hough Johnson">.
```

Die Verwendung in der Dokumentinstanz erfolgt durch eine Referenz. Die Syntax des Aufrufs innerhalb der Instanz beginnt mit dem „Und“-Zeichen „&“, gefolgt vom Namen der Entity und einem Semikolon „;“ als Abschluss. Das folgende Beispiel zeigt den Aufruf einer Entity innerhalb einer Instanz:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email SYSTEM "email.dtd">
<email>
  <datum tag="06.06.2006" zeit="12:10"/>
  <von>hough@johnnson.com</von>
  <an>amueller@wein.de</an>
  <betreff>Weinstein</betreff>
  <text>
    Lieber Albert, der 88er Rasteaux hat sehr viel Depot, was sagst du dazu?<br/>
    Grüße<br/>
    &hj;<![CDATA[Fa. Wine AG & Co]]><br/>
  </text>
</email>
```

Bei externen Entities wird der Inhalt der Entity in einem separaten physikalischen Speicherobjekt (Datei) vorgehalten. Inhalte einer externen Entity können Texte, aber insbesondere auch Bilder oder andere binäre Objekte sein. Aufgrund dieser Unter-

schiedlichkeit der Inhaltsstücke unterteilt man externe Entities zusätzlich in parsbare Entities (mit der Deklaration `PEDecl`, Parsed Entity Declaration) und nichtparsbare Entities (mit der Deklaration `GEDecl`, General Entity Declaration) gemäß der folgenden Produktionen:

```
[70] EntityDecl ::= GEDecl | PEDecl
[71] GEDecl    ::= '<!ENTITY' S Name S EntityDef S? '>'
[72] PEDecl    ::= '<!ENTITY' S '%' S Name S PEDef S? '>'
[73] EntityDef ::= EntityValue | (ExternalID NDataDecl?)
[74] PEDef     ::= EntityValue | ExternalID
```

Parsbare Entities enthalten als Inhalt parsbaren Text, der referenziert und integraler Bestandteil des Dokuments wird. Bei nicht parsbaren Entities können Informationsstücke von Dokumenten mit binärem Inhalt, z. B. Bilder, Audiodokumente oder Videos, referenziert werden. Im einfachsten Fall kann eine Referenzierung durch ein Attribut mit einem mit `CDATA` getypten Verweis auf ein externes Dokument verwendet werden, wie folgendes Beispiel, angelehnt an die `DocBook-DTD`, zeigt:

```
<!ATTLIST graphic
    ...
    fileref    CDATA    #IMPLIED
    ...
>
```

Diese einfache Referenz legt jedoch lediglich den Pfad zu einer externen Quelle. Sie macht weder eine Aussage über den Typ des externen Informationsstücks noch über dessen Inhalt.

Eine bessere Möglichkeit zur Referenzierung externer Binärdaten besteht darin, für den Typ des Referenzattributs das Schlüsselwort `ENTITY` zu verwenden, um dadurch nur durch Entities deklarierte externe Inhalte zu referenzieren.

```
<!ATTLIST graphic
    ...
    entityref  ENTITY  #IMPLIED
    ...
>
```

Die Einschränkung, nur auf bereits deklarierte Entities zu verweisen, ermöglicht eine Kontrolle darüber, welche Inhalte referenziert werden. Eine weitere Entity könnte z. B. eine Liste aller verwendeten Grafiken beinhalten. Das Element `graphic` dürfte dann nur diese bereits vordefinierten Grafiken referenzieren. Eine gewisse Kontrolle über die Verwendung externer Dokumente, nicht jedoch über deren Typ und Inhalt, ist dadurch gewährleistet.

3.3.4 Notationen

Durch das Schlüsselwort NOTATION werden in DTDs Namen für binäre Inhaltsarchitekturen vergeben und diese mit Anwendungen verknüpft. Die Nutzung dieser Typisierung muss in der XML-verarbeitenden Applikation erfolgen.

Eine Notation wird beispielsweise wie folgt in einer DTD deklariert:

```
<!NOTATION GIF PUBLIC "-//CompuServe//NOTATION Graphics Interchange Format 87a//EN">
<!ELEMENT graphics EMPTY>
<!ATTLIST graphics fileref NOTATION GIF>
```

Der Aufruf in der Dokumentinstanz sieht dann folgendermaßen aus:

```
<graphics fileref="figure.gif"/>
```

Dem Attribut `fileref` wird hier die Notation GIF zugeordnet. Die weitere Verwendung der Notation, in diesem Fall die Angabe über das verwendete Grafikformat, wird vom Parser ignoriert. Die Auswertung dieser Information bleibt dann der eigentlichen Anwendung vorbehalten.

Die größtmögliche Kontrolle über den Datentyp bietet die Referenzierung über eine Entity mit NDATA. Gemäß Produktion 73 kann eine Entity-Deklaration einer nicht-parsbaren Entity mit Hilfe eines System-Identifiers auf ein lokal vorliegendes externes Dokument verweisen und dieses zusätzlich mit einer NDATA-Deklaration versehen, wie das folgende Beispiel zeigt:

```
<!NOTATION GIF PUBLIC "-//CompuServe//NOTATION Graphics Interchange Format 87a//EN">
<!ELEMENT graphics EMPTY>
<!ATTLIST graphics entityref ENTITY #REQUIRED>
<!ENTITY my_figure SYSTEM "icon100.gif" NDATA GIF>
```

Das Attribut `entityref` verweist nun nicht wie das Attribut `fileref` im obigen Beispiel auf eine Pfadangabe zur Referenzierung des externen Binärdokumentes, sondern schränkt die Auswahl auf bereits definierte Entities ein. Somit entsteht eine Kontrolle über die verwendeten externen Datenquellen, und es existiert eine (jedoch durch den Parser nicht geprüfte) Angabe zum Typ des referenzierten Dokuments.

Es werden also in rudimentärer Form deskriptive Metadaten zur Inhaltsarchitektur der verwendeten Entities bereitgestellt. Allerdings bieten DTDs dazu nicht mehr als einen Namen, der gegebenenfalls noch mit einem öffentlichen Schlüssel verknüpft wird. Darüber hinaus kann dieser Name zusätzlich mit einer Anwendung verknüpft werden.

3.4 Zusammenfassung

Durch die Beschreibung der Strukturierungsmöglichkeiten von DTDs wurde das Beispiel der XML-Instanz erweitert. Es ergibt sich abschließend folgende Beispielinstantz:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email SYSTEM "email.dtd">
<email>
  <datum tag="06.06.2006" zeit="12:10"/>
  <von>hough@johnnson.com</von>
  <an>amueller@wein.de</an>
  <betreff>Weinstein</betreff>
  <text>
    Lieber Albert, der 88er Rasteaux hat sehr viel Depot, was sagst du dazu?<br/>
    Grüße<br/>
    Hough<![CDATA[Fa. Wine AG & Co]]><br/>
  </text>
  <graphics entityref="my_figure"/>
</email>
```

Als Beispiel der Strukturbeschreibung einer E-Mail ergibt sich folgende DTD:

```
<!NOTATION GIF PUBLIC "-//CompuServe//NOTATION Graphics Interchange Format 87a//EN">

<!ENTITY my_figure SYSTEM "icon100.gif" NDATA GIF>
<!ENTITY hj "Hough Johnson">

<!ELEMENT email (datum, von, an, betreff, text, graphics)>
<!ELEMENT datum EMPTY>
<!ATTLIST datum
  Tag CDATA #REQUIRED
  Zeit CDATA #REQUIRED
>
<!ELEMENT von (#PCDATA)>
<!ELEMENT an (#PCDATA)>
<!ELEMENT betreff (#PCDATA)>
<!ELEMENT text (#PCDATA|br)*>
<!ELEMENT br EMPTY>
<!ELEMENT graphics EMPTY>
<!ATTLIST graphics entityref ENTITY #REQUIRED>
```

XML stellt reichhaltige Konstrukte zur Referenzierung und Prüfung von Textdaten zur Verfügung. Die Typisierung und die Inhaltsbeschreibung von binären Inhaltsstücken können jedoch nur mit großen Einschränkungen vorgenommen werden und beschränken sich im Wesentlichen auf die Benennung binärer Inhaltsarchitekturen, einer rudimentären Semantik für diese Namen durch die Angabe öffentlicher Schlüssel und auf die Möglichkeit einer funktionalen Verknüpfung mit Anwendungen.

4 XML-Schema

Die DTD, wie sie in der XML-Spezifikation definiert ist, stellt nicht die einzige vom W3C verabschiedete Regelsprache für XML-Dokumente dar. Die Verschiebung des Fokus von XML-Anwendungen hin zu maschinenles- und -interpretierbaren Dokumenten, Protokollen und Datenbanken machten eine datenzentrierte Regelsprache nötig. Die Gründe dafür liegen in den Beschränkungen und Unzulänglichkeiten von DTDs als Regelsprache für datenzentrierte Dokumente, gegeben durch

- die unzureichende Unterstützung von Datentypen,
- die beschränkte Unterstützung von Kardinalitäten,
- die mangelnde Erweiterbarkeit des Typsystems,
- die fehlende Unterstützung von Namensräumen und
- eine mangelnde XML-Konformität der Schemasyntax.

Die Spezifikation der XML-Schema Description Language (XSD) liegt als W3C Recommendation vor (vgl. [Thompson et al. 2004]). XML-Schema stellt eine vollständig in XML-Syntax formulierte Grammatik zur Definition von XML-Strukturen dar.

Nachfolgend werden die wesentlichen Bestandteile von XML-Schema eingeführt und anhand von Beispielen erläutert, soweit sie für das Themengebiet dieser Arbeit relevant sind¹².

4.1 Referenz auf ein Schema

Jedes XML-Schema bildet ein eigenständiges XML-Dokument, das wie bei der Verwendung einer DTD von der XML-Instanz referenziert werden muss. Die Einbindung eines XML-Schemas in die XML-Instanz ist – anders als bei DTDs – nicht vorgesehen. Die Referenzierung zwischen der XML-Instanz und dem XML-Schema wird durch die Angabe des Attributs `schemaLocation` (bzw. `noNamespaceSchemaLocation`) innerhalb des Root-Elementes in der XML-Instanz definiert¹³.

¹² Die XSD-Spezifikation gliedert sich gemäß [Thompson et al. 2004] in drei große Teile: Part 0 (Primer) mit ausführlichen Beispielen der in Part 1 und Part 2 definierten Schemata; Part 1 (Structures) zur Definition von Inhaltsmodellen für Elemente und Attribute; Part 2 (Datatypes) zur Bereitstellung eines Standardsatzes von Datentypen und zur Definition von Einschränkungen von Datentypen.

¹³ Wird durch das XML-Schema kein Namespace für die enthaltenen Deklarationen definiert, muss die Referenz auf das Schema-Dokument in `noNamespaceSchemaLocation` angegeben werden.

Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?>
<mein_dokument
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:SchemaLocation="dokument.xsd"/>
```

Im obigen Beispiel wird zuerst der Namespace der Schema-Instanzen an das Präfix `xsi` gebunden, um zu ermöglichen, dass Elemente und Attribute aus der XML-Schema-Spezifikation in die Instanz eingebunden werden können. Dann wird das XSD-Attribut `SchemaLocation` im Root-Element benutzt.

Durch die Definition von XML-Schema als weitere Sprache zur Definition von Instanzstrukturen (neben DTDs) bedarf es auch einer Neudefinition der Validität von Dokumenten. Deshalb wird ein XML-Dokument, das hinsichtlich seiner Struktur einem XML-Schema konform ist, auch als „schema valid“ bezeichnet.

4.2 Definition eines XML-Schemas

Da jedes XML-Schema ein XML-Dokument ist, gelten für XML-Schema-Dokumente alle Regeln für den syntaktischen Aufbau gemäß der XML-Spezifikation.

Das Root-Element eines XML-Schema-Dokuments ist das Element `schema`. Durch eine Reihe von Attributen dieses Elementes werden zusätzliche Eigenschaften des Schema-Dokuments festgelegt, die für alle Definitionen innerhalb dieses Dokuments gültig sind. Minimal findet sich also:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"> ... </xsd:schema>
```

Im Beispiel wird der Namespace `xsd` definiert, zu dem die Elemente des XML-Schemas gehören. Zusätzlich zum Namespace der Elemente zur Schemadefinition kann auch der Namespace derjenigen Elemente definiert werden, für die das Schema erstellt wird (target namespace). Durch die Angabe einer URI für das Attribut `targetNamespace` erhalten alle durch das XML-Schema deklarierten Elemente und Attribute den angegebenen Namespace.

4.3 Definition von Elementen

Grundsätzlich wird jedes Element durch das XML-Schema-Element `element` definiert. Eine einfache Form der Definition eines Elementes zeigt folgendes Beispiel:

```
<xsd:element name="text" type="string"/>.
```

Diese Definition deklariert ein Element `text` mit dem Datentyp `string`. Die Auswahl des Typs eines zu erzeugenden Elementes kann unter anderem aus der Liste der

durch Part 2 der Spezifikation festgelegten vordefinierten Typen erfolgen (vgl. [Thompson et al. 2004]). Zusätzlich kann der Typ als Wert auch jeden beliebigen, durch den Schema-Autor definierten Datentyp annehmen.

Die XML-Schema-Spezifikation definiert zudem noch weitere Eigenschaften für Elemente, die durch Attribute des Elementes `element` definiert werden können. Die folgende Auflistung beschreibt diejenigen Attribute, die im MPEG-7-Standard (siehe Kapitel 4.6) Verwendung finden. Diese sind im Einzelnen:

- `id`: Eindeutige Identifikation eines Elementes,
- `maxOccurs`: Maximale Zahl der gültigen Vorkommen eines Elementes in der Instanz,
- `minOccurs`: Minimale Zahl der Vorkommen eines Elementes in der Instanz,
- `name`: Angabe des Namens eines Elementes,
- `type`: Typangabe eines Elementes (Spezifikation Part 2).

Das folgende Beispiel verdeutlicht einige dieser Attribute.

```
<xsd:element name="von" type="xsd:string" minOccurs="1" maxOccurs="1"/>
```

Die Definition des Elementes `von` benutzt `string` als einfachen Datentyp (Simple Type) und kann genau einmal in der Instanz vorkommen.

4.4 Definition von Elementmodellen

Die Definition von Inhaltsmodellen und die daraus resultierende Verwendung der bereits definierten Elemente erfolgt durch den Einsatz eines Typsystems.

4.4.1 Einfache Typen

Für unstrukturierten Inhalt bietet das Schlüsselwort `#PCDATA` bei der Definition von DTDs lediglich Zeichenketten als Inhaltstyp eines Elementes an. XML-Schema Part 2 macht dieser Schwäche ein Ende und legt insgesamt 44 vordefinierte Datentypen, sogenannte atomare Datentypen, fest. Um diese nun in einem XML-Schema verwenden und auch einem Element zuweisen zu können, kann das Element `simpleType` verwendet werden. Ein solcher einfacher Typ ist dadurch gekennzeichnet, dass er weder Kindelemente noch Attribute besitzt.

Neben den atomaren Datentypen von XML-Schema können einfache Datentypen durch Ableitung von bestehenden einfachen und atomaren Datentypen erzeugt werden. Insbesondere erreicht man dies durch Einschränkung eines bestehenden Da-

tentyps. Weiterhin sieht die XML-Schema-Spezifikation die Bildung von Vereinigungstypen und Listen vor.

Die einschränkende Ableitung eines atomaren Datentyps erzeugt einen neuen Datentyp als `simpleType`. Nachfolgendes Beispiel zeigt eine Aufzählung (`enumeration`) optionaler Werte:

```
<xsd:simpleType name="werktag">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Montag"/>
    <xsd:enumeration value="Dienstag"/>
    <xsd:enumeration value="Mittwoch"/>
    <xsd:enumeration value="Donnerstag"/>
    <xsd:enumeration value="Freitag"/>
    <xsd:enumeration value="Samstag"/>
  </xsd:restriction>
</xsd:simpleType>
```

Die Ableitung des neuen Datentyps `werktag` erfolgt ausgehend von dem im `base`-Attribut definierten Datentyp (hier `string`). Ausgehend von einem `String` als Basisdatentyp kann nun eine Werteliste gebildet werden, deren Einträge als Inhalt für ein Element des Datentyps `werktag` gültig sind.

Neben exklusiv verwendeten Aufzählungen in Form einer `enumeration` können auch Listen von Werten als Datentyp für eine Instanz Verwendung finden.

```
<xsd:simpleType name="tag_id">
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="31"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="tagesliste">
  <xsd:list itemType="tag_id"/>
</xsd:simpleType>
```

Ausgehend von einem bereits definierten Datentyp `tag_id` kann die Verwendung als Aggregationstyp durch das Element `list` mit einem Verweis auf `tag_id` benutzt werden.

Neben der Einschränkung von Datentypen und deren Ableitung durch Auflistung kann die Vereinigung von Datentypen durch das Element `union` gebildet werden. Vereinigungstypen werden aus einer Menge von definierten Ausgangstypen gebildet, wobei der neu erzeugte Datentyp nun die Vereinigungsmenge aller definierten Datentypen darstellt.

4.4.2 Kindmodelle

Wie bereits eingangs erwähnt, können einfache Typen keine Kindelemente und Attribute besitzen. Der Typ `complexType` hingegen besitzt diese Möglichkeiten.

In der DTD wird das sequenzielle Auftreten von Kindelementen (siehe Abschnitt 3.3.1) in einer bestimmten Reihenfolge durch ein Komma als Operator angegeben. XML-Schema stellt für die Definition von Sequenzen von Elementen innerhalb eines Complex Types das Element `sequence` bereit. Alle in einer Sequenz definierten Elemente werden als Kindelemente des Elementes `sequence` in XML-Schema definiert.

Entsprechend der Definition der Auswahl von Elementen in DTDs (siehe Abschnitt 3.3.1) hält die XML-Spezifikation das Element `choice` zur Definition von Oder-Verknüpfungen bereit. Für die Definition einer Oder-Verknüpfung mit Mehrfachauswahl existiert das Element `all`. Diese drei Typen zur Reihenfolgedefinition lösen die Klammerung innerhalb von DTDs ab.

Im nachfolgenden Beispiel wird die bekannte Struktur des E-Mail-Beispiels erweitert. Parallel zur Struktur einer E-Mail sollen nun auch Informationen über die Postadresse des Empfängers abgelegt werden. Das folgende Beispiel stellt einen Ausschnitt aus der Definition einer Postadresse dar:

```
<xsd:element name="adresse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"/>
      <xsd:element name="vorname" type="xsd:string"/>
      <xsd:element name="strasse" type="xsd:string"/>
      <xsd:element name="stadt" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Innerhalb des Elementes `adresse` sind die entsprechenden Kindelemente in das Element `sequence` eingebettet. Genau wie in einer DTD müssen in der entsprechenden XML-Instanz die Elemente exakt in der hier verwendeten Reihenfolge vorkommen.

XML-Schema bietet die Wiederverwendung von Inhaltsmodellen ohne konkrete Einbettung in ein Element an. Dies begünstigt die Wiederverwendung von Strukturen eines Inhaltsmodells, ohne dabei ein konkretes Element zu benennen, was im folgenden Beispiel deutlich wird:

```
<xsd:element name="adresse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="name" type="name_typ"/>
      <xsd:element name="strasse" type="xsd:string"/>
      <xsd:element name="stadt" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="name_typ">
  <xsd:sequence>
    <xsd:element name="vorname" type="xsd:string"/>
    <xsd:element name="familienname" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```



```

    </xsd:sequence>
</xsd:complexType>

```

Die Referenzierung des Complex Types erfolgt zum einen über die Angabe eines Namens für den Typ, zum anderen elementseitig über die Typ-Angabe durch das `type`-Attribut¹⁴.

4.4.3 Ableitung

Zur gezielten Wiederverwendung von bereits definierten Strukturen sind die vorgestellten Möglichkeiten in ihrer Verwendung noch zu starr. Einmal angelegte Strukturen in Form von Complex Types sollten optimalerweise auch in abgeänderter Form, also eingeschränkt oder erweitert, verwendbar sein. Deshalb werden in XML-Schema zwei verschiedene Varianten zur Ableitung von Typen definiert:

- Ableitung durch Einschränkung und
- Ableitung durch Erweiterung.

Bei der Ableitung durch Einschränkung erhält der abgeleitete Typ eine engere Definition als der Supertyp. Die Verwendung der Ableitung durch Einschränkung eines Complex Types kommt zudem einer kompletten Neudefinition des Inhaltsmodells gleich, weshalb deren Verwendung zu sehr aufwendigen Schemaevolutionen führen kann (vgl. [Obasanjo 2003]).

Anders verhält es sich bei der Ableitung durch Erweiterung. Hier erhält der abgeleitete Typ eine zusätzliche Definition zum Supertyp. Bestehende Inhaltsmodelle des Supertyps können wieder verwendet werden, wie das folgende Beispiel zeigt.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:complexType name="adresse">
    <xsd:sequence>
      <xsd:element name="Name" type="xsd:string"/>
      <xsd:element name="Vorname" type="xsd:string"/>
      <xsd:element name="Strasse" type="xsd:string"/>
      <xsd:element name="PLZ" type="xsd:string"/>
      <xsd:element name="Ort" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="adresse_2">
    <xsd:complexContent>
      <xsd:extension base="adresse">
        <xsd:sequence>

```

¹⁴ XML-Schema bietet die Möglichkeit, einen Datentyp so zu deklarieren, dass dieser nicht direkt instanziiert werden kann. Wird ein Datentyp als „abstract“ deklariert, so muss er zuerst von einem anderen Datentyp beerbt werden, um konkret verwendet werden zu können. Der MPEG-7 Standard macht ausgiebigen Gebrauch von abstrakten Datentypen und Ableitungen, um umfangreiche Typhierarchien aufzubauen.

```
        <xsd:element name="Postfach" type="xsd:string"/>
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:element name="elem_adr" type="adresse"/>
<xsd:element name="elem_adr_2" type="adresse_2"/>
</xsd:schema>
```

Bei der Ableitung durch Erweiterung ergänzt man bereits vorhandene komplexe Typen um weitere Elemente oder Attribute. Die Einschränkung bei dieser Technik besteht darin, dass man die vorgegebene Reihenfolge (hier `sequence`) nicht ändern kann. Es lassen sich also lediglich Elemente oder Attribute an den bestehenden `complexType` anhängen.

4.5 Definition von Attributen

Die Deklaration von Attributen erfolgt in XML-Schema durch das Element `attribute`. Durch verschiedene Attribute des Elementes `attribute` lässt sich die Funktionalität von Attributen so definieren, wie sie bereits in der DTD formuliert ist. Die Standardattribute einer Attributdeklaration sind:

- `name`: Name des Attributs,
- `id`: Eindeutige Kennzeichnung,
- `default`: Vorgabewert,
- `fixed`: Konstante Belegung,
- `type`: Typ des Attributs (definiert durch die Typdefinition eines SimpleTypes),
- `form`: Qualifizierter oder unqualifizierter Namesraumpräfix,
- `ref`: Verweis auf eine Attributdefinition,
- `use: Verwendung`: `optional`, `required`, `prohibited`.

Bei der Definition von Attributen kann zur Definition des Datentyps auf die gleiche Syntax zurückgegriffen werden wie bei den Elementinhalten. Auch hier ist die Ableitung, Vereinigung und Aggregation von Datentypen für Attributwerte möglich. Gegeben sei die Strukturdefinition des E-Mail-Beispiels und daraus im speziellen das Instanzfragment zur Definition des Datums:

```
<datum tag="06.06.2006" zeit="12:10"/>
```

Die Abbildung dieser Struktur mit XML-Schema beginnt zunächst mit der bekannten Definition des Elementes „Datum“.

```
<xsd:element name="datum"/>
```

Da es sich bei `datum` nicht um einen atomaren und auch nicht um einen einfachen Datentyp handeln kann, da ein Element mit Attributen vorliegt, wird zur Definition des

Datentyps das Element `complexType` verwendet. Darin eingebettet findet sich die Definition der Attribute `tag` und `zeit`.

```
<xsd:element name="datum">
  <xsd:complexType>
    <xsd:attribute name="tag" use="required" type="xsd:string"/>
    <xsd:attribute name="zeit" use="required" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
```

Neben den bereits aus den Elementdefinitionen bekannten Attributen `name` und `type` wird bei Attributdefinitionen zusätzlich deren verpflichtende Verwendung durch das Attribut `use` geregelt. Der Wert „required“ stellt dabei eine Verpflichtung für das Auftreten dieses Attributes in der XML-Instanz dar.

4.6 Notationen und Entities

Mit Notationen und Entities sind die Möglichkeiten, mit reinem XML bei der Festlegung von Dokumentenstrukturen "oberhalb" der Instanzen Metadaten für referenzierbare Binärdaten bereitzustellen, erschöpft.

Diese Möglichkeiten gibt es auch beim Einsatz von XML-Schema als Schemasprache. Das Element `notation` bietet – wie bei der Definition in einer DTD – die Möglichkeit, Typen von Binärdaten zunächst zu spezifizieren und dann später zu referenzieren.

```
<xsd:notation name="GIF"
  public="-//CompuServe//NOTATION Graphics Interchange Format 87a//EN"/>
```

Die obige Deklaration beschreibt die Notation eines Binärdatentyps, in diesem Fall GIF. Die Referenz auf die deklarierte Notation kann nun den Typ des Binärdokumentes festlegen.

```
<xsd:element name="bild">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:hexBinary">
        <xsd:attribute name="typ">
          <xsd:simpleType>
            <xsd:restriction base="xsd:NOTATION">
              <xsd:enumeration value="GIF"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:attribute>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

<bild typ="GIF">
ffdf8ffe000104a46494600010200006400640000ffec00114475636b7900010004000000410000ffee000e41646
f62650064c000000001ffdb0084000504040404040504050405070906050506090a08080908080a0d0a0b
0b0b0b0a0d0c0c0c0d0c0c0c0f0f11110f0f171600
</bild>
```

Das Element `bin` kann durch die Angabe des Datentyps `hexBinary` Binärdaten in die Instanz einbetten und auch das Encoding des einzubettenden Typs prüfen¹⁵. Die Typisierung der Binärdaten kann in diesem Beispiel über das Attribut `typ` erfolgen, das wiederum auf die Notation des Typs verweist. Die Angaben über das Format der Binärdaten sind jedoch schema-intern. Dies bedeutet, dass durch die Validierung einer XML-Instanz gegen ein XML-Schema das in der Notation festgelegte spezifische Datenformat nicht geprüft werden kann.

In XML-Schema übernehmen die Datentypen `xsd:entity` und `xsd:entities` die Rolle der Attributtypen `ENTITY` und `ENTITIES` aus der Definition von DTDs (siehe Abschnitt 3.3.3). Die Deklaration der eigentlichen Entities muss bei der Verwendung von XML-Schema dann jedoch in der Dokumentinstanz erfolgen. Für diese Definition ist typischerweise eine Entity-Deklaration als internes Subset aufzunehmen.

4.7 Zusammenfassendes Beispiel

Es ergibt sich nun abschließend die Struktur des E-Mail-Beispiels, definiert in XML-Schema, wie folgt:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:notation name="GIF"
  public="-//CompuServe//NOTATION Graphics Interchange Format 87a//EN"/>
  <xsd:element name="email">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="datum"/>
        <xsd:element ref="von"/>
        <xsd:element ref="an"/>
        <xsd:element ref="betreff"/>
        <xsd:element ref="text"/>
        <xsd:element ref="graphics"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="datum">
    <xsd:complexType>
      <xsd:attribute name="tag" type="xsd:string" use="required"/>
      <xsd:attribute name="zeit" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="von" type="xsd:string"/>
  <xsd:element name="an" type="xsd:string"/>
  <xsd:element name="betreff" type="xsd:string"/>
  <xsd:element name="text">
    <xsd:complexType mixed="true">
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="br"/>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

¹⁵ Neben der hexadezimalen Codierung von Binärdaten gibt es in XML-Schema zudem die Möglichkeit, Binärdaten mit dem Datentyp `base64` zu codieren.

```

</xsd:element>
<xsd:element name="br">
  <xsd:complexType/>
</xsd:element>
<xsd:element name="graphics">
  <xsd:complexType>
    <xsd:attribute name="entityref" type="xsd:ENTITY" use="required"/>
  </xsd:complexType>
</xsd:element>
</xsd:schema>

```

Aufgrund der unterschiedlichen Verwendung von Entities in DTD und XML-Schema ergibt sich auch eine geänderte XML-Instanz des E-Mail-Beispiels:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email [
<!ENTITY my_figure SYSTEM "icon100.gif" NDATA GIF>
]>
<email xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="email.xsd">
  <datum tag="06.06.2006" zeit="12:10"/>
  <von>hough@johnnson.com</von>
  <an>amueller@wein.de</an>
  <betreff>Weinstein</betreff>
  <text>
    Lieber Albert, der 88er Rasteaux hat sehr viel Depot, was sagst du dazu?<br/>
    Grüße<br/>
    Hough<![CDATA[Fa. Wine AG & Co]]><br/>
  </text>
  <graphics entityref="my_figure"/>
</email>

```

Durch die Möglichkeit, XML-Elemente und Attribute reichhaltig zu typisieren, ergibt sich ein entscheidender Vorteil beim Einsatz von XML-Schema im Vergleich zu DTDs. Für die Beschreibung von Binärdaten bieten die Datentypen `base64` und `hexbinary` die kontrollierte Einbettung in XML. Allerdings geschieht dies naturgemäß nicht auf unterschiedlichen hierarchischen (Granularitäts-)Ebenen, sondern immer in einem Stück.

Aber auch solche binären Inhaltsstücke können mit „Bordmitteln“ nur eingeschränkt typisiert werden. Weder die Regelsprache DTD noch XML-Schema können mit ihrem definierten Vokabular binäre Inhaltsarchitekturen reichhaltig kennzeichnen. Dafür müssen Metadaten-sprachen für Multimediadokumente bereitgestellt werden, die selbst Anwendungen der vorgestellten Schemasprachen sind. Idealerweise stellen sie ein Vokabular und eine Grammatik bereit, die genau für diesen Anwendungszweck der reichhaltigen Beschreibung von Binärdaten auf unterschiedlichen Granularitätsebenen zugeschnitten sind. Der auf XML-Schema basierte Metadatenstandard MPEG-7 versucht, diesen Anforderungen zu genügen.

5 MPEG-7

Parallel zu den Entwicklungen im Bereich Video- und Autokompression (MPEG-1, MPEG-2 und MPEG-4) hat die Moving Picture Experts Group (MPEG) einen Standard entwickelt, der – im Unterschied zu den vorher genannten Entwicklungen – nicht die Kompression von Multimediadokumenten beschreibt, sondern eine Sprache zur standardisierten Darstellung von Metadaten dieser Dokumente entwickelt. MPEG-7 ist also ein Standard zur Beschreibung von Dokumentinhalten. Er dient vor allem als Basis für viele Anwendungen im Bereich des Retrievals, der Archivierung und Verwaltung von Audio- und Videodokumenten.

5.1 Gliederung von MPEG-7

Der MPEG-7-Standard [ISO/IEC 15938] besteht aus den folgenden Teilen:

MPEG-7 Systems	Dieser Teil beschreibt das binäre Speicherformat von MPEG-7.
MPEG-7 Description Definition Language (DDL)	Dieser zweite Teil beschreibt eine Sprache, die es erlaubt, neue Description Schemes (DS) sowie neue Deskriptoren (D) zu erzeugen. Weiterhin erlaubt die DDL die Modifikation und die Erweiterung bereits existierender Description Schemes (DS).
MPEG-7 Visual	Der dritte Teil ist eine Sammlung von grundlegenden Strukturen und Beschreibungen für Eigenschaften aus dem visuellen Umfeld, wie z. B. Farbe, Texturen, Form, Bewegung, usw.
MPEG-7 Audio	Dieser vierte Teil ist eine Sammlung von Strukturen und Beschreibungen für Eigenschaften aus dem Bereich Audio, wie z. B. spektrale und temporale Eigenschaften eines Audiosignals, sowie abstrakte Eigenschaften zur Erkennung von Melodien, Instrumenten, Tonarten, usw.
MPEG-7 Multimedia Description Schemes (MDS)	Dieser fünfte Teil umfasst eine generische Bibliothek von Beschreibungsstrukturen für Multimediadokumente.

MPEG-7 Reference Software	Der sechste Teil dient mit seiner Experimentation Model (XM) Software zur Simulation von MPEG-7-Deskriptoren und Description Schemes. Diese ist eine Beispielimplementierung zur Erzeugung und Verarbeitung von Deskriptoren.
MPEG-7 Conformance Testing	Dieser siebte Teil enthält Richtlinien und Verfahren für das Testen von MPEG-7-Implementierungen.

Tabelle 1: Aufbau des MPEG-7-Standards.

5.1.1 Deskriptoren (Ds) und Description Schemes (DSs)

Eigenschaften eines audiovisuellen Inhaltsstückes werden durch einen Deskriptor (D) oder mehrere Deskriptoren (Ds) repräsentiert. Ein Deskriptor beschreibt die Syntax und die Semantik der Beschreibung von audiovisuellen Eigenschaften und kann als die Zusammenfassung mehrerer Werte für ein Merkmal betrachtet werden. Ein Deskriptor für das visuelle Merkmal „Farbe“ könnte z. B. den Farbraum sowie die Verteilung der Farbwerte auf die verschiedenen Farbkanäle definieren.

Für viele Audio- und Video-Beschreibungen gibt es bereits eine große Menge an vordefinierten Deskriptoren. Diese werden in Low- und High-Level-Deskriptoren eingeteilt. Low-Level-Deskriptoren sind insbesondere dafür geeignet, eine automatisierte Beschreibung von Metadaten für Multimediadokumente zu erzeugen, da sie ohne menschliches Eingreifen aufgrund von im Dokument vorhandenen Messwerten eingesetzt werden können. Ein Beispiel für einen Low-Level-Deskriptor ist der Farbwert eines bestimmten Filmsegments zu einer bestimmten Zeit. Auch High-Level-Deskriptoren beschreiben Metadaten aufgrund von im Dokument vorhandenen Kriterien (z.B. die Deklaration eines Segmentes als Person). Allerdings ist hier häufig ein menschlicher Eingriff erforderlich, da bestimmte Entscheidungen nicht automatisch getroffen werden können.

MPEG-7 Description Schemes (DSs) erweitern Deskriptoren dadurch, dass sie Beziehungen zwischen Deskriptoren und bereits definierten Description Schemes zu komplexeren Einheiten zusammenstellen können.

5.1.2 Beschreibungswerkzeuge für Strukturen

Da in dieser Arbeit nur syntagmatische Fragestellungen von Multimediadokumenten behandelt werden, wird im Folgenden derjenige Teil des Multimedia Description Schemes (MDS) genauer beleuchtet, der generisch den Zugriff auf Strukturen von

Multimediadokumenten zur Verfügung stellt. Mit Hilfe dieser Strukturen können dann später Syntagmen gebildet werden.

Für den Zugriff auf diese Strukturen stellt der Teil „Structural Aspects“ des MDS Beschreibungswerkzeuge, sogenannte „Tools“, zur Verfügung (siehe Tabelle 2). Diese erlauben es beispielsweise, temporale Segmente von Audio- und Videoinformationen oder auch zeitlich unabhängige Segmente wie z. B. Bilder zu beschreiben.

Es existieren vier Unterkategorien dieser Beschreibungswerkzeuge:

Segment Entity Description Tools (siehe Abschnitt 5.2)	Enthält Description Schemes für die Definition von räumlichen und zeitlichen Segmenten von Multimedia-Inhalten.
Segment Attribute Description Tools (siehe Abschnitt 5.3)	Enthält Description Schemes für die Beschreibung verschiedener Eigenschaften von Segmenten.
Segment Decomposition Tools (siehe Abschnitt 5.4).	Enthält Description Schemes zur Beschreibung der Zerlegung von Segmenten mit Multimediainhalt.
Segment Relation Description Tools (siehe Abschnitt 5.5).	Enthält Description Schemes zur Beschreibung von Beziehungen zwischen den Segmenten.

Tabelle 2: Übersicht über die wichtigsten MPEG-7 Tools.

Den Zusammenhang zwischen diesen einzelnen Tools zeigt Abbildung 4. Die Werkzeuge der Segment Entity Description Tools regeln den grundsätzlichen Aufbau von Segmenten. Die Segment Relation Tools und die Segment Decomposition Tools erlauben eine Beschreibung der Beziehungen zwischen diesen Segmenten.

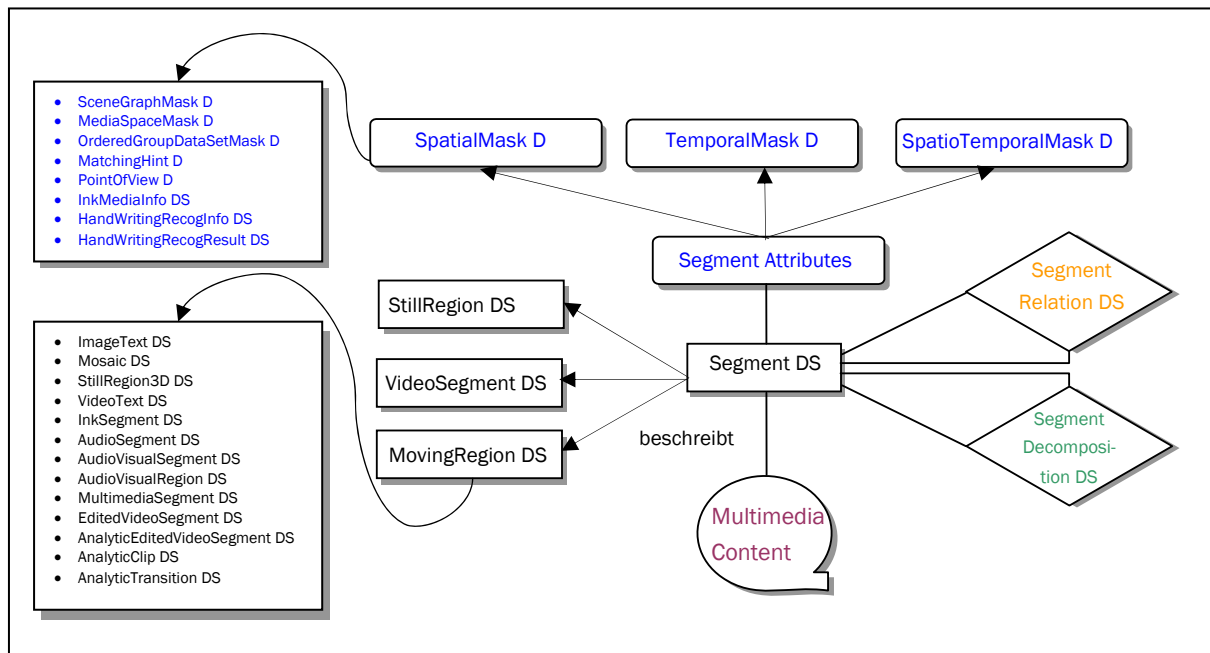


Abbildung 4: Zusammenhang zwischen den einzelnen Tools¹⁶ (Abbildung angelehnt an [van Beek et al. 2001, dort Abb. 12]).

Da für syntagmatische Fragestellungen von cinematographischen Dokumenten unter anderem Moving-Regions betrachtet werden müssen, liegt im Folgenden der Schwerpunkt auf den Beschreibungen der Segmenttypen der Still-Region, des Video-Segmentes und der Moving-Region. Alle übrigen zu den Segment Entity Description Tools gehörigen Description Schemes bleiben unberücksichtigt. Abbildung 5 zeigt den Zusammenhang zwischen diesen drei Segmenttypen.

Durch die Zerlegung von Video-Segmenten können neue Video-Segmente oder auch Moving-Regions gebildet werden. Moving-Regions können in sich selbst oder in Still-Regions zerlegt werden. Still-Regions können nur in sich selbst zerlegt werden. Eine ausführliche Beschreibung dieser Möglichkeiten findet sich in Abschnitt 5.4.2.2.

¹⁶ Zur Darstellung der Zugehörigkeit der jeweiligen Deskriptoren und Description Schemes zu den Beschreibungswerkzeugen wurden unterschiedliche Farben gewählt: schwarz: Segment Entity Description Tools; grün: Segment Decomposition Tools; orange: Segment Relation Description Tools; blau: Segment Attribute Description Tools. Der Vollständigkeit halber wurden hier alle Deskriptoren und Description Schemes aus den Segment Entity Description Tools und den Segment Attribute Description Tools aufgelistet (siehe große Boxen links). In den folgenden Abbildungen wird auf diese vollständige Darstellung verzichtet.

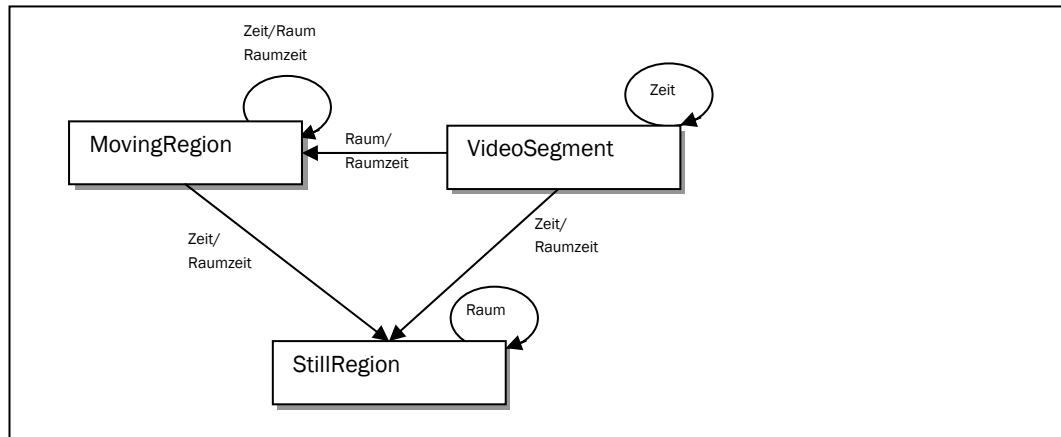


Abbildung 5: Aufteilung und Zerlegungshierarchie der Segmenttypen (Abbildung angelehnt an [van Beek et al. 2001, dort Abb. 32]).

Im Folgenden werden nun in der Reihenfolge der Tabelle 2 die drei wichtigsten Beschreibungswerkzeuge des MDS vorgestellt. Die Segment Entity Description Tools finden sich in Abschnitt 5.2, die Segment Attribute Description Tools in Abschnitt 5.3, die Segment Decomposition Tools in Abschnitt 5.4 und die Segment Relation Description Tools in Abschnitt 5.5.

5.2 Die Segment Entity Description Tools des MDS

Die Segment Entity Description Tools des MDS regeln den grundsätzlichen Aufbau von Segmentbeschreibungen. Zentral ist dabei der abstrakte Typ des Segment DS. Von diesem können allein 16 konkrete Segmenttypen abgeleitet werden. Für den Rahmen dieser Arbeit sind jedoch lediglich drei konkrete Segmenttypen von Bedeutung (s. Abbildung 6):

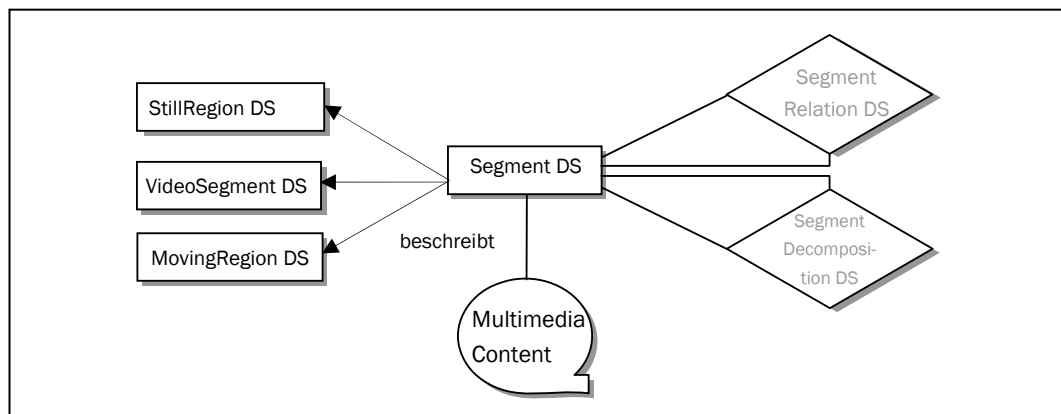


Abbildung 6: Die Segment Entity Description Tools im Gesamtzusammenhang. (Abbildung angelehnt an [van Beek et al. 2001, dort Abb. 12]).

Dies sind im Einzelnen das StillRegion DS zur Segmentierung von Bereichen innerhalb eines stehenden Bildes, das VideoSegment DS zur Segmentierung von ganzen Frames und Mengen von Frames und das MovingRegion DS zur Segmentierung von raumzeitlichen Bereichen innerhalb einer Videosequenz.

5.2.1 Segment DS

Das Segment DS ist ein abstrakter Typ eines Description Schemes, von dem alle spezielleren Segment DSs abgeleitet sind. Das Segment DS definiert grundlegende Eigenschaften von Segmenten, die auch für alle abgeleiteten Typen gültig sind, wie z. B. das Erstellungsdatum eines Segmentes oder dessen Verwendung.

Das Segment DS besteht aus einem einzigen abstrakten Segment-Typ. Er ist wie folgt definiert:

```
<!-- ##### -->
<!-- Definition of Segment DS -->
<!-- ##### -->

<complexType name="SegmentType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <choice minOccurs="0">
          <element name="MediaInformation" type="mpeg7:MediaInformationType"/>
          <element name="MediaLocator" type="mpeg7:MediaLocatorType"/>
        </choice>
        <element name="StructuralUnit" type="mpeg7:ControlledTermUseType"
          minOccurs="0"/>
        <element name="CreationInformation"
          type="mpeg7:CreationInformationType" minOccurs="0"/>
        <element name="UsageInformation" type="mpeg7:UsageInformationType"
          minOccurs="0"/>
        <element name="TextAnnotation" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <complexContent>
              <extension base="mpeg7:TextAnnotationType">
                <attribute name="type" use="optional">
                  <simpleType>
                    <union memberTypes="mpeg7:termReferenceType string"/>
                  </simpleType>
                </attribute>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element name="MatchingHint" type="mpeg7:MatchingHintType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="PointOfView" type="mpeg7:PointOfViewType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="Relation" type="mpeg7:RelationBaseType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Mit den sprachlichen Mitteln von XML-Schema, die in Kapitel 4 dargestellt wurden, ergeben sich für den komplexen Datentyp `SegmentType` folgende syntaktische Merkmale.

Der Datentyp `SegmentType` leitet sich aus dem Datentyp `DSType` ab, der eine abstrakte Beschreibung eines Description Schemes liefert. Weiter besteht der Datentyp `SegmentType` alternativ oder auch gar nicht aus den Elementen `MediaInformation` oder `MediaLocator`. Die Inhaltstypen dieser beiden Elemente werden wiederum innerhalb von MDS als komplexer Datentyp zur Verfügung gestellt.

An diese Oder-Gruppe schließen sich die drei optionalen Elemente `StructuralUnit`, `CreationInformation` und `UsageInformation` an.

Ebenfalls optional ist das Element `TextAnnotation`, das mehrfach auftreten kann. `TextAnnotation` stellt eine Ableitung durch Erweiterung des Datentyps `TextAnnotationType` dar. Das Element `TextAnnotation` definiert zusätzlich zu allen Element- und Attributdefinitionen aus dem Datentyp `TextAnnotationType` das optionale Attribut `type`. Das Inhaltsmodell des Attributes `type` wiederum besteht aus der Vereinigung des atomaren Datentyps `string` mit dem in MPEG-7 definierten Datentyp `termReferenceType`.

Als letzte Elemente der Sequenz schließen sich die drei optionalen, aber auch mehrfach verwendbaren Elemente `MatchingHint`, `PointOfView` und `Relation` an.

Auffällig ist zunächst, dass alle diese Elemente optional sind; das heißt, dass der Segmenttyp im Minimalfall nicht über den `DSType` hinausgeht.

Das Segment DS beschreibt ein Segment eines multimedialen Inhaltsstücks mit den folgenden Eigenschaften gemäß [ISO/IEC 15938-5, S. 253ff.]:

- Das Element `MediaInformation` nimmt direkten Bezug auf das Segment und beschreibt die Struktur von Informationen über das Medium des Segmentes. Solange kein neues Element `MediaInformation` für dieses Segment spezifiziert ist, bleibt die Information (z. B. das Dateiformat) rekursiv für alle Subsegmente des vorliegenden Segmentes gültig.
- Das Element `MediaLocator` kann anstelle von `MediaInformation` benutzt werden. Dieses kann auch ein `MediaLocator`-Element beinhalten, um physikalisch vorliegende Segmente zu referenzieren.
- Das Element `StructuralUnit` beschreibt die Rolle, die das Segment unter Berücksichtigung des gesamten Multimedialinhalts spielt. Das Element `StructuralUnit` ist vom Type `ControlledTermUseType`, der durch das

Attribut `href` auf einen Wert vom Typ `termReferenceType` referenziert und somit Zugriff auf die Werte eines `ClassificationSchemes` hat. Beispiele von Werten für `StructuralUnit` können beispielsweise klassifikatorische Begriffe wie „Szene“ oder „Sequenz“ sein, wie im Anschluss an diese Auflistung noch gezeigt wird.

- Das Element `CreationInformation` beinhaltet Metadaten über den Prozess der Erstellung des Segmentes. Beispiel für eine `CreationInformation` sind Personendaten des Autors, Zeitpunkt und Ort (Koordinaten) der Aufzeichnung, Freitextbeschreibungen, usw.
- Das Element `UsageInformation` beinhaltet Metadaten zu Rechten und zur Verfügbarkeit des zu beschreibenden Segmentes. Die hier definierten Metadaten vererben sich wiederum in mögliche Sub-Segmente.
- Mit `TextAnnotations` können freisprachliche Annotationen zum vorliegenden Segment in die Metadaten eingefügt werden. Mit dem Attribut `type` kann zusätzlich noch der Typ oder der Zweck der Anmerkung spezifiziert werden. Beispiele hierfür sind: „Beschreibung der Szene“, „Beschreibung der verwendeten Farben“, usw.
- Das Element `MatchingHint` beschreibt die Bedeutung der Audio- und Video-Deskriptoren und `Description-Schemes` innerhalb des Segment `DS` für den Zweck der Priorisierung bei Suchanfragen.
- Das Element `PointOfView` beschreibt die Wichtigkeit des Segmentes in Abhängigkeit von einer bestimmten Blickrichtung der Kamera.
- Das Element `Relation` beschreibt die Beziehungen zwischen dem spezifizierten Segment und anderen Segmenten. Diesem Element kommt bei der Beschreibung der Segmentbeziehungen in Abschnitt 5.5 eine besondere Bedeutung zu.

Sowohl das Element `StructuralUnit` als auch das Element `Relation` verwenden kontrollierte Vokabulare in Form von sogenannten `ClassificationSchemes`. Ein Klassifikationsschema stellt eine Reihe von Termini zur Verfügung, die ein Vokabular für eine bestimmte Anwendung oder Domäne ergeben. Diese Termini werden nun in Beschreibungen von z. B. Segmentrelationen durch Referenzierung auf das Klassifikationsschema und den Terminus verwendet. Diese Art der Referenzierung erlaubt es, zukünftige Entwicklungen und Erweiterungen neuer Vokabulare zu unterstützen.

Klassifikationsschemata werden innerhalb einer MPEG-7 Instanz definiert oder importiert. Beispielhaft sei hier ein Klassifikationsschema für mögliche Werte des Elementes `StructuralUnit` definiert:

```
<Description xsi:type="ClassificationSchemeDescriptionType">
  <ClassificationScheme uri="urn:mpeg:mpeg7:cs:StructuralUnitCS">
    <Term termID="szene"></Term>
    <Term termID="sequenz"></Term>
    <Term termID="planszene"></Term>
  </ClassificationScheme>
</Description>
```

Durch die Angabe der URI des Klassifikationsschemas sowie durch die zusätzliche Angabe des jeweiligen Terminus lassen sich die Werte eines Schemas wie folgt verwenden:

```
<VideoSegment id="ConversationVS">
  <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:szene"/>
  <MediaTime>
    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT1M00S</MediaDuration>
  </MediaTime>
</VideoSegment>
```

Der Einsatz eines solchen Vokabulars für temporale Segmentrelationen wird in Abschnitt 5.5 erläutert.

5.2.2 StillRegion DS

Das StillRegion DS erweitert das oben beschriebene Segment DS. Eine Still-Region beschreibt den zweidimensionalen räumlichen Bereich eines Bildes oder Video-Frames. Eine Besonderheit besteht darin, dass eine Still-Region aus einer bestimmten Anzahl von räumlichen Komponenten bestehen kann, die nicht zwingend räumlich miteinander verbunden sein müssen. Die Komposition dieser räumlichen Komponenten zu einer Still-Region wird in Abschnitt 5.4.2.1 beschrieben.

Das StillRegion DS besteht nur aus einem einzigen `StillRegionType`, der eine Erweiterung von `SegmentType` darstellt. Er ist wie folgt definiert:

```
<!-- ##### -->
<!-- Definition of StillRegion DS -->
<!-- ##### -->

<complexType name="StillRegionType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <element name="SpatialLocator" type="mpeg7:RegionLocatorType"
          minOccurs="0"/>
        <element name="SpatialMask" type="mpeg7:SpatialMaskType" minOccurs="0"/>
        <choice minOccurs="0">
          <element name="MediaTimePoint" type="mpeg7:mediaTimePointType"/>
          <element name="MediaRelTimePoint" type="mpeg7:MediaRelTimePointType"/>
          <element name="MediaRelIncrTimePoint" type="mpeg7:MediaRelIncrTimePointType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="VisualDescriptor" type="mpeg7:VisualDType"/>
          <element name="VisualDescriptionScheme" type="mpeg7:VisualDSType"/>
          <element name="GridLayoutDescriptors" type="mpeg7:GridLayoutType"/>
        </choice>
        <element name="SpatialDecomposition"
          type="mpeg7:StillRegionSpatialDecompositionType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
        minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </extension>
</complexContent>
</complexType>
```

Die Beschreibung eines zweidimensionalen Bildes erfolgt durch folgende Elemente gemäß [ISO/IEC 15938-5, S. 255]:

- Das optionale Element `SpatialLocator` beschreibt die räumliche Lokalisierung der Still-Region.
- Die `SpatialMask` beschreibt den Aufbau von sich räumlich nicht überlappenden, zusammenhängenden Komponenten, die zusammen eine Still-Region bilden. Wird dieses Element nicht verwendet, besteht die Still-Region aus dem Bereich, der mit dem `SpatialLocator`-Element definiert ist.
- Der `MediaTimePoint` bestimmt den Punkt der Still-Region auf der Zeitachse, wenn diese Still-Region einen Frame in einem Video darstellt.
- Der `MediaRelTimePoint` bestimmt den relativen Zeitpunkt der Still-Region auf der Zeitachse, wenn diese einen Frame in einem Video darstellt.
- Der `MediaRelIncrTimePoint` bestimmt einen Zeitpunkt der Still-Region relativ zu einer bestimmten Basis, die z. B. auf der Zählung einzelner Frames beruht.
- Die Elemente `VisualDescriptor` und `VisualDescriptionScheme` verweisen auf vordefinierte Deskriptoren zur visuellen Beschreibung einer Still-Region.
- Ein `GridLayoutDescriptor` beschreibt die visuellen Eigenschaften von Sub-Regions, die von einer Zerlegung der Still-Region durch eine Grid-Decomposition stammen. Ein Grid-Layout-Descriptor beschreibt nur rechteckige Still-Regions.
- Das Element `SpatialDecomposition` beschreibt die räumliche Zerlegung der Still-Region in eines oder mehrere Teil-Segmente.

5.2.3 VideoSegment DS

Wie das `StillRegion DS` leitet sich auch das `VideoSegment DS` von dem abstrakten `Segment DS` ab. Das `VideoSegment DS` beschreibt ein zeitliches Intervall von Videodaten. Ein Beispiel für ein Video-Segment ist eine Folge von Video-Frames oder auch eine komplette Filmsequenz. Weiterhin kann ein Video-Segment zeitlich zusammenhängend oder nicht-zusammenhängend sein. Das `VideoSegment DS` besteht aus einem einzigen `VideoSegmentType`, der eine Erweiterung des `SegmentType` darstellt. Er ist wie folgt definiert:

```

<!-- ##### -->
<!-- Definition of VideoSegment DS -->
<!-- ##### -->
<complexType name="VideoSegmentType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <element name="MediaTime" type="mpeg7:MediaTimeType" minOccurs="0"/>
        <element name="TemporalMask" type="mpeg7:TemporalMaskType" minOccurs="0"/>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="VisualDescriptor" type="mpeg7:VisualDType"/>
          <element name="VisualDescriptionScheme" type="mpeg7:VisualDSType"/>
          <element name="TimeSeriesDescriptors" type="mpeg7:TimeSeriesType"/>
          <element name="Mosaic" type="mpeg7:MosaicType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="SpatialDecomposition"
            type="mpeg7:VideoSegmentSpatialDecompositionType"/>
          <element name="TemporalDecomposition"
            type="mpeg7:VideoSegmentTemporalDecompositionType"/>
          <element name="SpatioTemporalDecomposition"
            type="mpeg7:VideoSegmentSpatioTemporalDecompositionType"/>
          <element name="MediaSourceDecomposition"
            type="mpeg7:VideoSegmentMediaSourceDecompositionType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

Der Datentyp `VideoSegmentType` beschreibt einen zeitlichen Abschnitt eines Videos mit den folgenden bisher noch nicht eingeführten Elementen:

- Das Element `MediaTime` beschreibt die zeitlichen Eigenschaften eines Video-Segmentes durch die Festlegung einer Startzeit und einer Dauer. Ist das Video-Segment nicht zusammenhängend, so sollte die Dauer des Video-Segmentes gleich der Dauer des kleinsten angenommenen zusammenhängenden zeitlichen Intervalls im kompletten Video-Segment sein. Besteht also ein Video-Segment aus zwei Komponenten mit einer Dauer von zwei Sekunden und drei Sekunden sowie einer zeitlichen Lücke von einer Sekunde, so hat das kleinste angenommene zeitlich zusammenhängende Intervall, das beide Segmente beinhaltet, eine Dauer von sechs Sekunden.
- Das Element `TemporalMask` beschreibt den Aufbau von zeitlich nicht überlappenden, aber zusammenhängenden Komponenten, die zusammen ein zeitlich nicht zusammenhängendes Video-Segment ergeben. Fehlt die `TemporalMask`, so besteht das Segment aus einer einzelnen Komponente. Das Beispiel aus dem obigen Listenpunkt setzt also eine Temporal-Mask voraus.
- Das Element `TimeSeriesDescriptors` beschreibt eine temporale Sequenz von visuellen Eigenschaften eines Video-Segmentes.
- Das optionale Element `Mosaic` erlaubt die Beschreibung eines Video-Segmentes mit Hilfe eines einzigen Bildes (Keyframe).

- Sollen Segmente in weitere Teilsegmente zerlegt werden, so erreicht man dies durch die Elemente der Dekomposition. Dabei steht das Element `SpatialDecomposition` für die Zerlegung in räumliche Teilsegmente, das Element `TemporalDecomposition` für die Zerlegung in zeitliche Teilsegmente, das Element `SpatialTemporalDecomposition` für die Zerlegung in raumzeitliche Elemente und das Element `MediaSourceDecomposition` für die Zerlegung von Multimediadokumenten in die verschiedenen vordefinierten Medientypen – beispielsweise die Zerlegung eines audiovisuellen Inhaltsstücks in seine Bestandteile Video und Audio.

Das folgende Beispiel zeigt die Verwendung des `VideoSegment DS` zur Beschreibung eines Video-Segmentes mit MPEG-7:

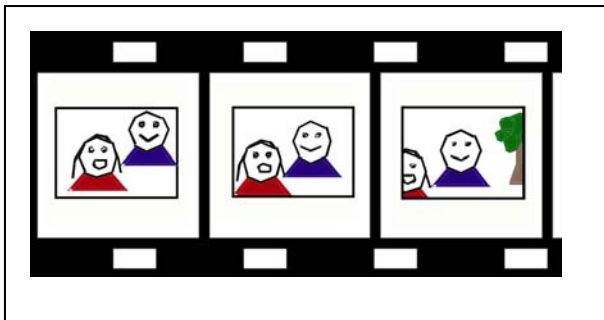


Abbildung 7: Beispiel eines Videosegmentes.

```
<VideoSegment id="ConversationVS">
  <MediaLocator>
    <MediaUri>movie.mpg</MediaUri>
  </MediaLocator>
  <TextAnnotation>
    <FreeTextAnnotation>Unterhaltung</FreeTextAnnotation>
  </TextAnnotation>
  <MediaTime>
    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT1M30S</MediaDuration>
  </MediaTime>
</VideoSegment>
```

Das Video-Segment `ConversationVS` beschreibt den Inhalt des Binärfiles `movie.mpg`. Durch die Angaben des Elementes `MediaTime` beschränkt sich die Definition des Segmentes und damit auch die Annotation von Metadaten auf die ersten 90 Sekunden des Videos¹⁷.

¹⁷ Die Angabe des Zeitrahmens erfolgt nach [ISO/IEC 8601].

5.2.4 Das MovingRegion DS

Das MovingRegion DS erweitert den abstrakten Datentyp des Segment DS und beschreibt einen zweidimensionalen raumzeitlichen Bereich von Videodaten. Eine Moving-Region beschreibt dabei eine beliebige Menge an Pixeln in einer beliebigen Menge an Video-Frames. Auch eine Moving-Region kann in Raum und/oder Zeit zusammenhängend oder nicht zusammenhängend sein. Eine solche Unterteilung nimmt der Deskriptor der SpatioTemporalMask vor. Das MovingRegion DS besteht aus einem MovingRegionType, der eine Erweiterung des Typs SegmentType darstellt. Er ist wie folgt definiert:

```
<!-- ##### -->
<!-- Definition of MovingRegion DS -->
<!-- ##### -->
<complexType name="MovingRegionType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <element name="SpatioTemporalLocator"
          type="mpeg7:SpatioTemporalLocatorType" minOccurs="0"/>
        <element name="SpatioTemporalMask" type="mpeg7:SpatioTemporalMaskType"
          minOccurs="0"/>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="VisualDescriptor" type="mpeg7:VisualDType"/>
          <element name="VisualDescriptionScheme" type="mpeg7:VisualDSType"/>
          <element name="TimeSeriesDescriptors" type="mpeg7:TimeSeriesType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="SpatialDecomposition"
            type="mpeg7:MovingRegionSpatialDecompositionType"/>
          <element name="TemporalDecomposition"
            type="mpeg7:MovingRegionTemporalDecompositionType"/>
          <element name="SpatioTemporalDecomposition"
            type="mpeg7:MovingRegionSpatioTemporalDecompositionType"/>
          <element name="MediaSourceDecomposition"
            type="mpeg7:MovingRegionMediaSourceDecompositionType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Neben den bereits in StillRegion DS und VideoSegment DS verwendeten Elementen verwendet der MovingRegionType zusätzlich folgende Elemente:

- Das Element des SpatioTemporalLocator beschreibt die raumzeitliche Lokalisierung einer Moving-Region. Handelt es sich um eine nicht-zusammenhängende Moving-Region, definiert der SpatioTemporalLocator die zusammenhängende Komponente, die die Teile der Moving-Region einschließt.
- Das Element der SpatioTemporalMask beschreibt die Zusammensetzung von raumzeitlich nicht überlappenden und in sich verbundenen Komponenten, die zusammen eine nicht-verbundene Moving-Region er-

geben. Existiert keine `SpatioTemporalMask`, so besteht das Segment aus einem einzelnen in Raum und Zeit verbundenen Bereich, wie er mit dem `SpatioTemporalLocator` definiert wurde.

Das folgende Beispiel zeigt die Verwendung des `MovingRegion` DS zur Beschreibung einer Moving-Region in MPEG-7.

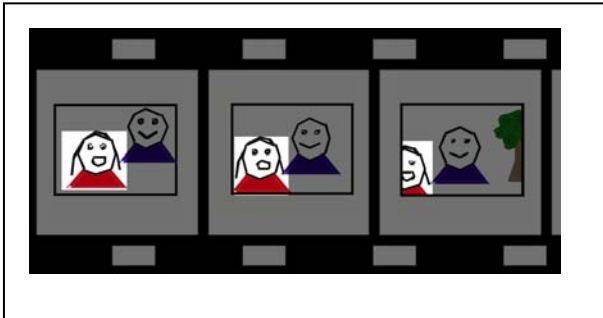


Abbildung 8: Beispiel für eine Moving Region.

Das Beispiel zeigt eine Moving-Region, die einen zeitlichen und räumlichen Ausschnitt aus dem Gesamtvideo repräsentiert.

```
<MovingRegion id="GirLMR">
  <MediaLocator>
    <MediaUri>movie.mpg</MediaUri>
  </MediaLocator>
  <TextAnnotation>
    <FreeTextAnnotation>Unterhaltung</FreeTextAnnotation>
  </TextAnnotation>
  <MediaTime>
    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT1M30S</MediaDuration>
  </MediaTime>
  <SpatioTemporalLocator>
    <FigureTrajectory type="rectangle">
      <MediaTime>
        <MediaTimePoint>T00:00:00</MediaTimePoint>
        <MediaDuration>PT0M30S</MediaDuration>
      </MediaTime>
      <Vertex>
        <WholeInterval>
          <MediaDuration>PT30S</MediaDuration>
        </WholeInterval>
        <InterpolationFunctions>
          <KeyValue type="startPoint">0.1</KeyValue>
          <KeyValue type="startPoint">0.2</KeyValue>
        </InterpolationFunctions>
      </Vertex>
      <Vertex>
        <WholeInterval>
          <MediaDuration>PT30S</MediaDuration>
        </WholeInterval>
        <InterpolationFunctions>
          <KeyValue type="startPoint">0.3</KeyValue>
          <KeyValue type="startPoint">0.4</KeyValue>
        </InterpolationFunctions>
      </Vertex>
    </FigureTrajectory>
  </SpatioTemporalLocator>
</MovingRegion>
```

```

    <WholeInterval>
      <MediaDuration>PT30S</MediaDuration>
    </WholeInterval>
    <InterpolationFunctions>
      <KeyValue type="startPoint">0.5</KeyValue>
      <KeyValue type="startPoint">0.6</KeyValue>
    </InterpolationFunctions>
  </Vertex>
</FigureTrajectory>
</SpatioTemporalLocator>
</MovingRegion>

```

Die Moving-Region `Girl1MR` beschreibt einen rechteckigen Ausschnitt aus dem Binärfile `movie.mpg`. Durch die Angabe des Elementes `MediaTime` beschränkt sich die Definition des Segmentes und damit die Annotation von Metadaten auf die ersten 30 Sekunden des Videos. Der Bewegungsablauf des Rechtecks wird gemäß ISO/IEC 15938-3 (MPEG-7-Visual) definiert, das nicht mehr Bestandteil des MDS ist. Das Element `FigureTrajectory` beschreibt hierbei einen rechteckigen Ausschnitt. Dieser wird gemäß Standard durch drei Ecken (`Vertex`) spezifiziert. Die Einheit der Zeitperiode ist über die ISO 8601 [ISO/IEC 8601] geregelt. Dabei ist der Datentyp von `MediaDuration` ein abgeleiteter Typ des Typs `basicDuration` aus dem XML-Schema Standard. Die Syntax der Angabe stellt sich wie folgt dar:

(-) PnDTnHnMnSnNnF

Dabei gilt für die Angabe der Tage `nD`, für Stunden `nH`, für Minuten `nM`, `nS` für Sekunden, `nN` für die Anzahl der Sekundenteile und `nF` für die Anzahl der Teile, die eine Sekunde hat (üblicherweise 60).

Über die Elemente `KeyValue` wird eine Ecke des Rechtecks in einem zweidimensionalen Raum bestimmt.

5.3 Die Segment Attribute Description Tools des MDS

Die Segment Attribute Description Tools des MDS definieren eine große Menge an verschiedenen Eigenschaften von Segmenten. Von Bedeutung für diese Arbeit sind vor allem die Definitionen von Masken für die bereits vorgestellten Segmenttypen. Sie dienen dem Ausblenden von Segmentbestandteilen.

Dies sind im Einzelnen: Der `SpatialMask D` zur Maskierung von räumlichen Bereichen innerhalb eines Segments, der `TemporalMask D` zur Maskierung von zeitlichen Bereichen innerhalb eines Segments sowie der `SpatialTemporalMask D` zur Maskierung von raumzeitlichen Bereichen innerhalb eines Segments (siehe Abbildung 9).

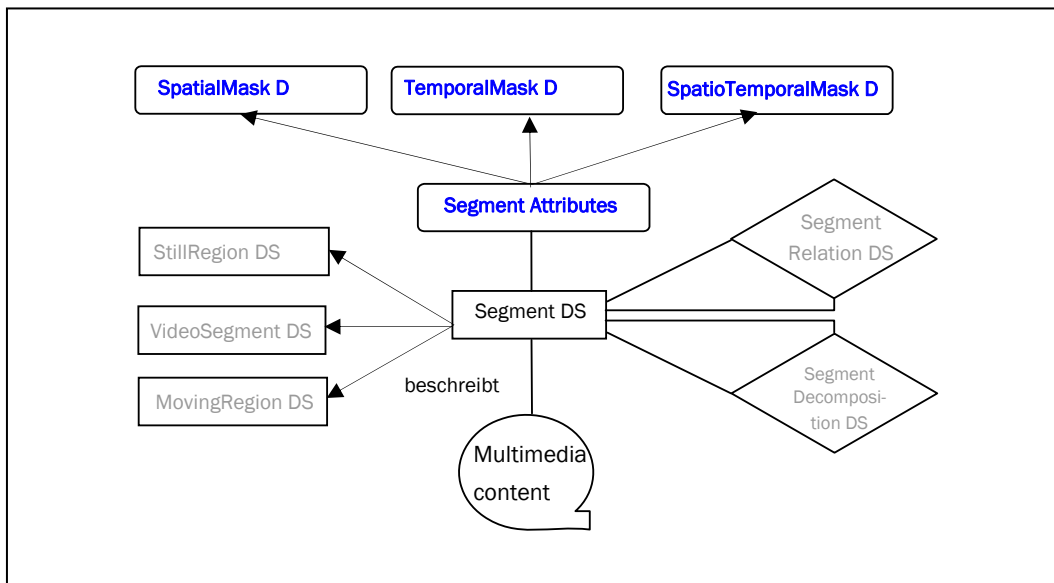


Abbildung 9: Die Segment Attribute Description Tools im Gesamtzusammenhang (angelehnt an [van Beek et. al 2001, dort Abb. 26]).

Im Folgenden werden diese drei konkreten Deskriptoren detailliert erläutert.

5.3.1 Zusammenhängende Segmente

Segmente, die aus multimedialen Inhalten bestehen, sind nicht immer zusammenhängend – weder zeitlich noch räumlich. Die Deskriptoren `TemporalMask`, `SpatialMask` und `SpatioTemporalMask` beschreiben die Zusammensetzung von Komponenten zu Segmenten, die dann als zusammenhängend oder nicht zusammenhängend angesehen werden können. Die Eigenschaft des Zusammenhangs von Segmenten definiert sich wie folgt:

- Ein zeitlich zusammenhängendes Segment entspricht einem zeitlich fortlaufenden¹⁸ Segment über einem bestimmten Zeitintervall.
- Ein räumlich zusammenhängendes Segment entspricht einem räumlich zusammenhängenden Bereich wie z.B. einer Menge von Pixeln, die einen Bereich in einem Bild beschreiben.
- Ein räumlich und zeitlich zusammenhängendes Segment entspricht einem Bereich, der zum einen zeitlich zusammenhängend und zu jedem Zeitpunkt räumlich zusammenhängend ist.

¹⁸ Übersetzt aus dem Englischen: contiguous

Erfährt ein Segment eine geeignete Maskierung – egal ob temporal, spatial oder spatio-temporal – so erhält man ein Segment, das aus zusammenhängenden Komponenten besteht. Das eigentliche Ausgangssegment bleibt dabei erhalten. Eine echte Zerlegung in Einzelsegmente findet an dieser Stelle nicht statt. Für eine solche Zerlegung existieren eigene Description Schemes, die in Abschnitt 5.4 beschrieben werden.

Abbildung 10 zeigt zeitlich und räumlich auf unterschiedliche Art und Weise zusammenhängende Segmente:

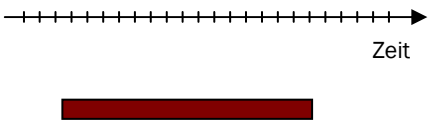
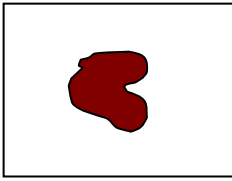
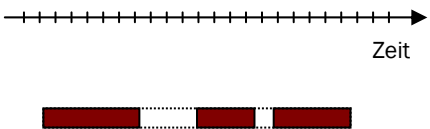
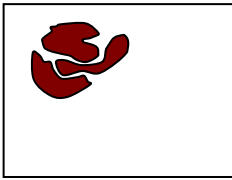
Zeitliches Segment	Räumliches Segment
 <p data-bbox="327 1019 790 1086">a) Segment zusammengesetzt aus einer zusammenhängenden Komponente</p>	 <p data-bbox="826 1019 1292 1086">b) Segment zusammengesetzt aus einer zusammenhängenden Komponente</p>
 <p data-bbox="327 1344 790 1411">c) Segment zusammengesetzt aus drei zusammenhängenden Komponenten</p>	 <p data-bbox="826 1344 1292 1411">d) Segment zusammengesetzt aus drei zusammenhängenden Komponenten</p>

Abbildung 10: a) zeigt ein zeitlich zusammenhängendes Segment. b) zeigt ein räumlich zusammenhängendes Segment. c) zeigt ein zeitlich nicht zusammenhängendes Segment, das jedoch aus einzelnen zeitlich zusammenhängenden Segmenten besteht. d) zeigt ein räumlich nicht zusammenhängendes Segment, das aus drei einzelnen räumlich zusammenhängenden Segmenten besteht (Abbildung angelehnt an [van Beek 2001 et. al, dort Abb. 15]).

Die verschiedenen Eigenschaften zur Maskierung von Segmenten werden mit den im Folgenden beschriebenen Deskriptoren realisiert.

5.3.2 Räumliche, zeitliche und raumzeitliche Masken

Der SpatialMask D gehört zum StillRegion DS und beschreibt nicht-überlappende, zusammenhängende räumliche Teilbereiche einer Still-Region. Er ist im MDS wie folgt spezifiziert:

```
<!-- ##### -->
<!-- Definition of SpatialMask D -->
<!-- ##### -->
<complexType name="SpatialMaskType">
  <sequence>
    <element name="SubRegion" type="mpeg7:RegionLocatorType"
      minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Das Element SubRegion beschreibt die räumliche Lokalisierung eines jeden Teilbereichs in der Spatial-Mask.

Der TemporalMask D beschreibt sich nicht-überlappende und verbundene zeitliche Teilintervalle. Er ist im MDS wie folgt definiert.

```
<!-- ##### -->
<!-- Definition of TemporalMask D -->
<!-- ##### -->
<complexType name="TemporalMaskType">
  <sequence>
    <element name="SubInterval" type="mpeg7:MediaTimeType"
      minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Das Element SubInterval beschreibt den Startzeitpunkt und die Dauer eines jeden zeitlichen Sub-Intervalls.

Der SpatioTemporalMask D beschreibt sich nicht überlappende, zusammenhängende raumzeitliche Teilbereiche eines Segmentes. Er ist im MDS wie folgt definiert:

```
<!-- ##### -->
<!-- Definition of SpatioTemporalMask D -->
<!-- ##### -->
<complexType name="SpatioTemporalMaskType">
  <sequence>
    <element name="SubRegion" type="mpeg7:SpatioTemporalLocatorType"
      minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Das Element der SubRegion beschreibt die Lokalisierung des jeweiligen raumzeitlichen Teilsegments.

5.3.3 Beispiel einer zeitlichen Maske

Ausgehend von dem bereits in Abschnitt 5.2.3 eingeführten Beispiel eines Video-Segmentes wird nun im Folgenden eine zeitliche Maske verwendet, um ein Segment aus einzelnen Komponenten zu bilden, ohne dieses dabei in Subsegmente zu zerlegen. Den Ausgangspunkt liefert wieder die Definition eines Video-Segmentes:

```
<VideoSegment id="ConversationVS">
  <MediaLocator>
    <MediaUri>movie.mpg</MediaUri>
  </MediaLocator>
  <TextAnnotation>
    <FreeTextAnnotation>Unterhaltung</FreeTextAnnotation>
  </TextAnnotation>
  <MediaTime>
    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT1M30S</MediaDuration>
  </MediaTime>
</VideoSegment>
```

Durch die zeitliche Maske, die dem Video-Segment zugewiesen wird, entsteht eine zeitliche Lücke des Video-Segmentes gemäß Abbildung 10c.

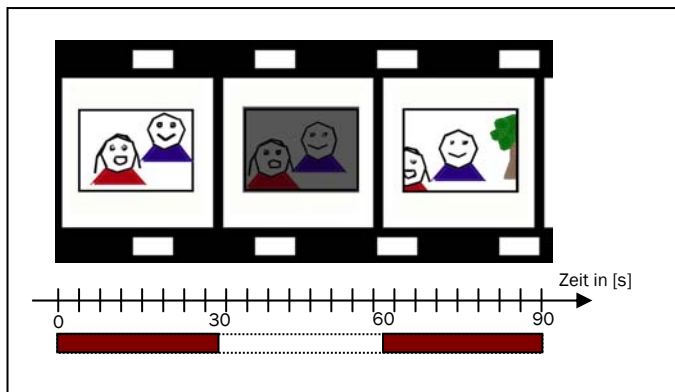


Abbildung 11: Beispiel für ein Videosegment mit zeitlicher Maske.

```
<VideoSegment id="ConversationVS">
  <MediaLocator><MediaUri>movie.mpg</MediaUri></MediaLocator>
  <TextAnnotation>
    <FreeTextAnnotation>Unterhaltung</FreeTextAnnotation>
  </TextAnnotation>
  <MediaTime>
    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT1M30S</MediaDuration>
  </MediaTime>
  <TemporalMask>
    <SubInterval>
      <MediaTimePoint>T00:00:00</MediaTimePoint>
      <MediaDuration>PT0M30S</MediaDuration>
    </SubInterval>
    <SubInterval>
      <MediaTimePoint>T00:00:61</MediaTimePoint>
      <MediaDuration>PT0M30S</MediaDuration>
    </SubInterval>
  </TemporalMask>
</VideoSegment>
```


In mehreren Sub-Intervallen werden durch die bereits bekannten Elemente `MediaTimePoint` und `MediaDuration` der Startpunkt und die Dauer der Intervalle festgelegt, die dann in ihrer Gesamtheit das Segment bilden.

5.4 Die Segment Decomposition Tools

Die Segment Decomposition Tools des MDS regeln die mögliche Zerlegung von Segmenten in Teilsegmente (s. Abbildung 12). Von großer Bedeutung für diese Arbeit sind vor allem die Decomposition Tools für die bereits vorgestellten Segmenttypen.

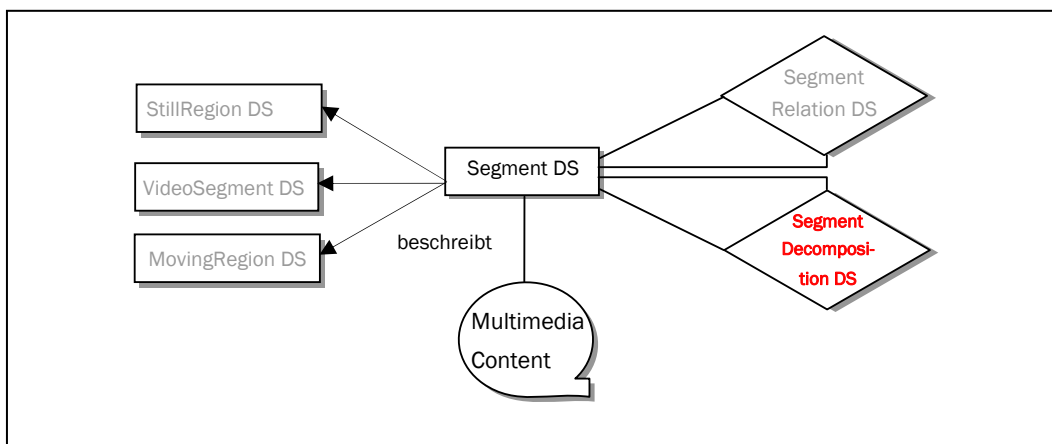


Abbildung 12: Die Segment Decomposition Tools im Gesamtzusammenhang (Abbildung angelehnt an [van Beek et al. 2001, dort Abb. 12]).

Nachfolgend wird die Zerlegung der bereits vorgestellten Segmenttypen Still-Region, Video-Segment und Moving-Region gezeigt.

5.4.1 Abstrakte Datentypen zur Segmentzerlegung

Ähnlich dem Aufbau der Datentypen zur Definition des `Segment DS` sind auch die Datentypen für die Segmentzerlegung mittels einer Typhierarchie definiert. Am Anfang steht wieder ein abstrakter Datentyp (`SegmentDecompositionType`), von dem alle anderen Datentypen für die Segmentzerlegung abgeleitet sind:

```
<!-- ##### -->
<!-- Definition of SegmentDecomposition DS -->
<!-- ##### -->
<complexType name="SegmentDecompositionType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <attribute name="criteria" type="string" use="optional"/>
      <attribute name="overlap" type="boolean" use="default" value="false"/>
      <attribute name="gap" type="boolean" use="default" value="false"/>
    </extension>
  </complexContent>
</complexType>
```

Das `SegmentDecomposition DS` beschreibt die hierarchische Zerlegung von Segmenten. Die abgeleiteten Subklassen sind selbst wieder abstrakte Datentypen, die die nach Raum, Zeit, Raumzeit und Datenquelle durchführbare Zerlegung in Segmente beschreiben. Der `SpatialSegmentDecompositionType` ist wie folgt definiert:

```
<!-- ##### -->
<!-- Definition of SpatialSegmentDecomposition DS -->
<!-- ##### -->

<complexType name="SpatialSegmentDecompositionType" abstract="true">
  <complexContent>
    <extension base="mpeg7:SegmentDecompositionType" />
  </complexContent>
</complexType>
```

Die anderen drei `SegmentDecomposition`-Types für temporale, spatio-temporale und Media-Source Dekomposition werden analog gebildet. Zur Semantik des `SegmentDecompositionType` ist festzuhalten:

- Das Attribut `criteria` beschreibt Merkmale, die für die Zerlegung des Segmentes verwendet worden sind. Dies kann z. B. die Farbe sein.
- Das Attribut `overlap` gibt an, ob die Segmente, die dieser Zerlegung entstammen, in Raum und/oder Zeit überlappen.
- Das Attribut `gap` gibt an, ob die Segmente im Bezug auf das Elternsegment, die aus dieser Zerlegung entstammen, Lücken lassen. Die Zerlegung eines Segmentes hat Lücken, wenn die Vereinigung der wieder zusammengesetzten Kind-Teilsegmente nicht exakt mit dem Elternsegment korrespondiert. Per default ist dieser Wert `false`.

5.4.2 Überlappungen und Lücken bei der Segmentzerlegung

Bei der Zerlegung von Segmenten in Teilsegmente ist zu beachten, dass sich die Grenzen aller zerlegten Kindsegmente nicht über die Grenzen der Elternsegmente hinaus ausdehnen dürfen. Beispielsweise dürfen die Kindsegmente einer zeitlichen Zerlegung eines Videos in Ihrer Gesamtheit nicht über die Grenzen des vorgegebenen zeitlichen Intervalls des Elternsegmentes hinausragen. Umgekehrt kann jedoch eine Zerlegung eines Segmentes so erfolgen, dass entweder Lücken oder Überlappungen zwischen den einzelnen Kindsegmenten entstehen. Die Zerlegung eines Segmentes in Kindsegmente kann daher folgende Verhältnisse zwischen den Kindsegmenten zur Folge haben:

- keine Lücken, keine Überlappungen;
- keine Lücken, Überlappungen;
- Lücken, keine Überlappungen;
- Lücken und Überlappungen.

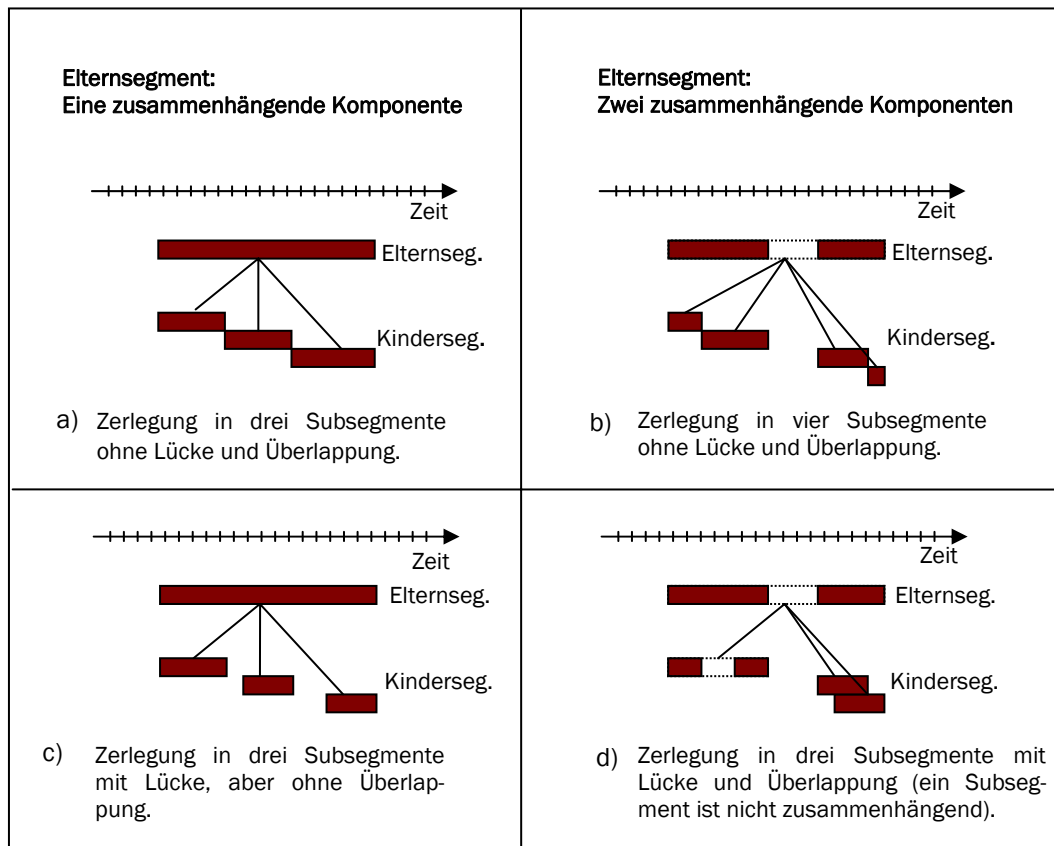


Abbildung 13: Zerlegungsmöglichkeiten von Segmenten (Abbildung angelehnt an [van Beek et al. 2001, dort Abb. 32]).

Abbildung 13a und Abbildung 13b zeigen Segmentzerlegungen ohne Lücken und ohne Überlappungen. In beiden Fällen ergibt die Vereinigung der Kind-Teilsegmente exakt das Elternsegment. Zudem ist die Schnittmenge aller Paare von Kindelementen leer. In Abbildung 13b ist das Elternsegment selbst nicht zusammenhängend.

Abbildung 13c zeigt die Zerlegung mit Lücken, aber ohne Überlappung. In Abbildung 13d ist der Sachverhalt etwas komplexer. Hier handelt es sich um ein nicht zusammenhängendes Elternsegment, das in drei Kindsegmente zerlegt wird. Das erste Kindsegment ist selbst wieder ein nicht zusammenhängendes Segment. Die beiden anderen Kindsegmente sind einzelne zusammenhängende Segmente. Das Beispiel zeigt weiter, dass die Tatsache, ob ein Segment zusammenhängend ist oder nicht, nichts mit dessen Zerlegung zu tun hat¹⁹.

¹⁹ Ein nicht zusammenhängendes Segment kann so zerlegt werden, dass zwischen den Teilsegmenten keine Lücken entstehen; ein zusammenhängendes Segment kann so zerlegt werden, dass zwischen den Teilsegmenten Lücken entstehen.

Weiterhin kann ein Segment nicht nur innerhalb eines Typs zerlegt werden. Es ist z. B. möglich, ein Video-Segment in eine Still-Region und in eine Moving-Region zu zerlegen. Genauso kann eine Moving-Region in eine Still-Region und in eine Moving-Region zerlegt werden (siehe Abbildung 5).

5.4.2.1 DS zur Zerlegung einer Still-Region

Der Deskriptor zur Zerlegung einer Still-Region erbt von dem bereits vorgestellten `SpatialSegmentDecompositionType`. Dieser Deskriptor beschreibt die Zerlegung einer Still-Region in eines oder mehrere räumliche Teilsegmente.

```
<!-- ##### -->
<!-- Definition of StillRegionSpatialDecomposition DS -->
<!-- ##### -->
<complexType name="StillRegionSpatialDecompositionType">
  <complexContent>
    <extension base="mpeg7:SpatialSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="StillRegion" type="mpeg7:StillRegionType"/>
        <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

Das Element `StillRegion` beschreibt die aus der Zerlegung entstandenen Still-Regions. Existiert bereits eine Beschreibung der zerlegten Still-Region, soll lässt sich die Referenz in `StillRegionRef` angeben.

5.4.2.2 DS zur Zerlegung eines Video-Segmentes

Der Deskriptor zur Zerlegung eines Video-Segmentes erbt von dem bereits vorgestellten `SegmentDecompositionType`. Dieser Deskriptor beschreibt die Zerlegung eines Video-Segmentes in eines oder mehrere Teilsegmente auf Basis der Zeit, des Raumes, der Raumzeit oder der Medienquelle.

Die Zerlegung eines Video-Segmentes erlaubt als Kindsegmente die Segmenttypen Moving-Region, eine Mischung aus Still-Regions und Video-Segmenten, eine Mischung aus Moving-Regions und Still-Regions sowie die Zerlegung in Video-Segmente selbst.

```
<!-- ##### -->
<!-- Definition of VideoSegmentSpatialDecomposition DS -->
<!-- ##### -->
<complexType name="VideoSegmentSpatialDecompositionType">
  <complexContent>
    <extension base="mpeg7:SpatialSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="MovingRegion" type="mpeg7:MovingRegionType"/>
        <element name="MovingRegionRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

```

        </extension>
    </complexContent>
</complexType>

<!-- ##### -->
<!-- Definition of VideoSegmentTemporalDecomposition DS -->
<!-- ##### -->
<complexType name="VideoSegmentTemporalDecompositionType">
    <complexContent>
        <extension base="mpeg7:TemporalSegmentDecompositionType">
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="VideoSegment" type="mpeg7:VideoSegmentType"/>
                <element name="VideoSegmentRef" type="mpeg7:ReferenceType"/>
                <element name="StillRegion" type="mpeg7:StillRegionType"/>
                <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

<!-- ##### -->
<!-- Definition of VideoSegmentSpatioTemporalDecomposition DS -->
<!-- ##### -->
<complexType name="VideoSegmentSpatioTemporalDecompositionType">
    <complexContent>
        <extension base="mpeg7:SpatioTemporalSegmentDecompositionType">
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="MovingRegion" type="mpeg7:MovingRegionType"/>
                <element name="MovingRegionRef" type="mpeg7:ReferenceType"/>
                <element name="StillRegion" type="mpeg7:StillRegionType"/>
                <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

<!-- ##### -->
<!-- Definition of VideoSegmentMediaSourceDecomposition DS -->
<!-- ##### -->
<complexType name="VideoSegmentMediaSourceDecompositionType">
    <complexContent>
        <extension base="mpeg7:MediaSourceSegmentDecompositionType">
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="VideoSegment" type="mpeg7:VideoSegmentType"/>
                <element name="VideoSegmentRef" type="mpeg7:ReferenceType"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

```

Mit diesen Zerlegungsmöglichkeiten können für Bewegtbilddaten syntagmatische Konzepte sowohl auf Einstellungsebene als auch unterhalb der Einstellungsebene unter Einbeziehung wenigstens einer Moving-Region entwickelt werden.

5.4.2.3 DS zur Zerlegung einer Moving-Region

Der Deskriptor zur Zerlegung einer Moving-Region beschreibt deren Zerlegung in Teilstücke nach Raum, Zeit, Raumzeit und Medientyp.

```

<!-- ##### -->
<!-- Definition of MovingRegionTemporalDecomposition DS -->
<!-- ##### -->
<complexType name="MovingRegionTemporalDecompositionType">
  <complexContent>
    <extension base="mpeg7:TemporalSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="MovingRegion" type="mpeg7:MovingRegionType"/>
        <element name="MovingRegionRef" type="mpeg7:ReferenceType"/>
        <element name="StillRegion" type="mpeg7:StillRegionType"/>
        <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

Analog zu Abbildung 5 kann eine Moving-Region wiederum in Moving-Regions oder in Still-Regions zerlegt werden.

5.5 Die Segment Relation Description Tools

Die Segment Relation Description Tools des MDS regeln die Beziehungen zwischen Segmenten und deren Eigenschaften. Diese können zeitlicher oder räumlicher Natur sein. Darüber hinaus lassen sich auch Eigenschaften spezifizieren, die weder räumliche noch zeitliche Beziehungen ausdrücken (s. Abbildung 14).

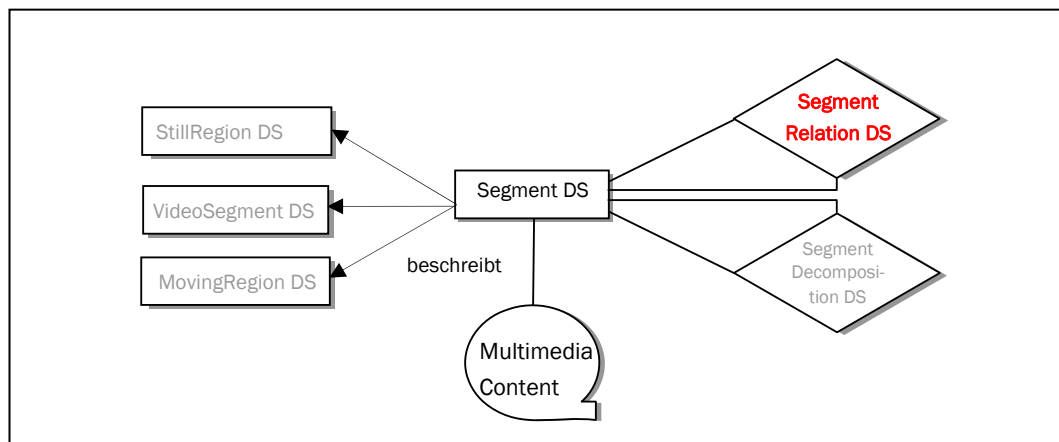


Abbildung 14: Die Segment Relation Description Tools im Gesamtzusammenhang (Abbildung angelehnt an [van Beek et al. 2001, dort Abb. 12]).

Nachfolgend werden nun die möglichen zeitlichen Beziehungen zwischen Segmenten in MPEG-7 vorgestellt.

5.5.1 Repräsentation von Segmentrelationen

Der Datentyp `RelationType` regelt die formale Einbindung von Relationstypen allgemein, wie die folgende Typdefinition zeigt:

```

<!-- ##### -->
<!-- Definition of RelationType DS -->
<!-- ##### -->
<complexType name="RelationType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <attribute name="type" type="mpeg7:termReferenceType" use="optional"/>
      <attribute name="source" use="optional">
        <simpleType>
          <list itemType="anyURI"/>
        </simpleType>
      </attribute>
      <attribute name="target">
        <simpleType>
          <list itemType="anyURI"/>
        </simpleType>
      </attribute>
      <attribute name="directed" type="boolean" use="optional" default="true"/>
      <attribute name="strength" type="mpeg7:zeroToOneType" use="optional"
        default="1.0"/>
    </extension>
  </complexContent>
</complexType>

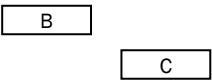
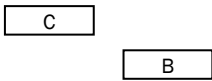
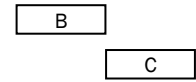
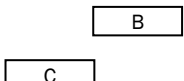
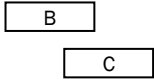
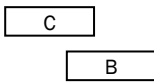
```

Zur Semantik des Datentyps `RelationType` ist festzuhalten:

- Der Wert des Attributs `type` referenziert einen Begriff aus einem Klassifikationsschema. Dieses Schema kann wiederum Namen von Relationen beinhalten, die z.B. zeitliche oder räumliche Relationen beschreiben.
- Die Werte der Attribute `source` und `target` sind Argumente der zu bildenden Relation und verweisen auf die an der Relation beteiligten Segmente. Ist eine Relation in die Segmentdefinition des Quellsegmentes eingebettet, so entfällt die Angabe des Attributs `source`.
- Das Attribut `directed` beschreibt, ob die Relation gerichtet ist. Per default besitzt die Relation eine Richtung.
- Ist eine Gewichtung der Relation gewünscht, kann diese durch das Attribut `strength` definiert werden.

Für diese Arbeit sind zeitliche Relationen besonders wichtig. Die Zeitlogik von Allen basiert auf Zeitintervallen, die zueinander in Relation gesetzt werden (vgl. [Allen 1983]). Dadurch können relativ bezogene Aussagen über Zeitperioden getroffen werden. Die Allen'schen zeitlichen Relationen mit zwei zusätzlichen Relationen (`strictContains` und `strictDuring`) finden sich bei MPEG-7 als Typen im Klassifikationsschema `TemporalRelation` mit folgenden Definitionen²⁰:

²⁰ B und C stellen dabei zeitliche Segmente dar. Die Attribute `.ende` und `.start` bezeichnen den Zeitpunkt des Segmentbeginns bzw. -endes.

<i>Name der Relation</i>	<i>Definition</i>	<i>Beispiel</i>
before	B kommt vor C, wenn: $B.ende < C.start$	
after	B kommt nach C, wenn: $B.start > C.ende$	
meets	B trifft auf C, wenn: $B.ende = C.start$	
metBy	B wurde von C getroffen, wenn: $B.start = C.ende$	
overlaps	B überlappt C, wenn $B.start < C.start$ UND $B.ende > C.start$ UND $B.ende < C.ende$	
overlappedBy	B wird von C überlappt, wenn; $B.start > C.start$ UND $B.start < C.ende$ UND $B.ende > C.ende$	
during	B findet während C statt, wenn: $(B.start > C.start$ UND $B.ende \leq C.ende)$ ODER $(B.start \geq C.start$ UND $B.ende < C.ende)$	Ein Beispiel aus: strictDuring, starts oder finishes
contains	B beinhaltet C, wenn: $(B.start < C.start$ UND $B.ende \geq C.ende)$ ODER	Ein Beispiel aus: strictContains, startedBy oder finishedBy

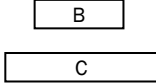
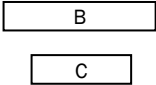

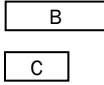
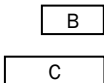
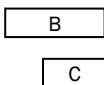
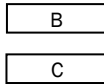
<i>Name der Relation</i>	<i>Definition</i>	<i>Beispiel</i>
	(B.start =< C.start UND B.ende > C.ende)	
strictDuring	B findet während C statt, wenn: B.start > C.start UND B.ende < C.ende	
strictContains	B beinhaltet C, wenn: B.start < C.start UND B.ende > C.ende	
starts	B startet mit C, wenn B.start = C.start UND B.ende < C.ende	
startedBy	B started durch C, wenn B.start = C.start UND B.ende > C.ende	
finishes	B beendet C, wenn: B.start > C.start UND B.ende = C.ende	
finishedBy	B wird beendet durch C, wenn: B.start < C.start UND B.ende = C.ende	
equal	B ist gleich C, wenn: B.start = C.start UND B.ende = C.ende	

Tabelle 3: Relationseigenschaften nach der Zeitlogik von Allen (vgl. [ISO/IEC 15938-5, S. 284f.]).

Um die zeitlichen Relationen in der narrativen Welt (also in der Welt der Figuren) verwenden zu können, wird das Element `Relation` in einen Container `Semantic` eingebettet, wie das folgende Beispiel zeigt:

Gegeben sei wieder das bereits bekannte Videodokument `movie.mpg` mit einer Laufzeit von 90 Sekunden, das jedoch diesmal mit zwei Videosegmenten `ConversationVS` und `TreeVS` modelliert ist.

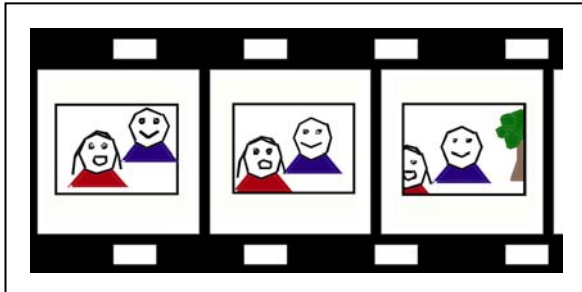


Abbildung 15: Zwei Videosegmente in zeitlicher Folge montiert.

Zuerst werden nun wieder die einzelnen Video-Segmente definiert:

```
<Video id="S">
  <MediaLocator>
    <MediaUri>movie.mpg</MediaUri>
  </MediaLocator>
  <MediaTime>
    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT1M30S</MediaDuration>
  </MediaTime>
  <TemporalDecomposition>
    <VideoSegment id="ConversationVS">
      <MediaTime>
        <MediaTimePoint>T00:00:00</MediaTimePoint>
        <MediaDuration>PT1M00S</MediaDuration>
      </MediaTime>
    </VideoSegment>
    <VideoSegment id="TreeVS">
      <MediaTime>
        <MediaTimePoint>T00:01:01</MediaTimePoint>
        <MediaDuration>PT0M30S</MediaDuration>
      </MediaTime>
    </VideoSegment>
  </TemporalDecomposition>
</Video>
```

Innerhalb der Definition des Segmentes `Conversation VS` kann nun die Segmentrelation wie folgt eingebettet werden:

```
<Semantic>
  <Label>Zeitliche Relation Meets</Label>
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets" target="#TreeVS"/>
</Semantic>
```

Für die beiden Video-Segmente wird mit Hilfe des Elementes `Relation` eine Beziehung modelliert, die durch die Eigenschaften gemäß der Zeitlogik von Allen im Attribut `type` genauer spezifiziert wird.

5.6 Zusammenfassung

MPEG-7 bietet einen vollständigen Ansatz, um Multimedia-Dokumente zu klassifizieren und mit Metadaten auszustatten. Der auf XML basierende Standard definiert sich aus der Metasprache DDL. Basierend auf dem W3C-Standard XML-Schema liefert die DDL eine offene Architektur und ermöglicht somit eine gute Erweiterbarkeit des Standards.

MPEG-7 liefert die formale Basis, um Multimediadokumente standardisiert beschreiben zu können. Zentral ist hierbei die Definition von Segmenten. Durch die große Anzahl der Deskriptoren bietet MPEG-7 die Möglichkeit, sehr genau viele Details einer Bildsequenz zu beschreiben. Zudem ermöglicht der Standard die zeitliche oder räumliche Zerlegung eines Segmentes in weitere Subsegmente und damit in Segmentbäume. Dadurch können Relationen zwischen Segmenten mit Eigenschaften beschrieben und somit ein Beziehungsnetzwerk zwischen diesen Segmenten aufgebaut werden.

TEIL II – Syntagmatische Analyse

6 Basale Syntagmatik von Einstellungen und narrative Hierarchisierung

MPEG-7 liefert die grundlegenden technischen Werkzeuge zur Beschreibung von Multimediadokumenten. Der Standard definiert jedoch naturgemäß keine spezifischen Dokumentstrukturen für die von ihm abgedeckten speziellen multimedialen Inhaltsarchitekturen, beispielsweise cinematographische Dokumente.

Für den detaillierteren Zugriff auf entsprechende Dokumente sind solche Strukturen allerdings unverzichtbar. Zu ihrer Definition wird nun für den Fall des Films zuerst auf Arbeiten zu Filmgrammatiken eingegangen, um den Stand der Forschung auch außerhalb der Ingenieurwissenschaften einzubeziehen. Die nachfolgenden Abschnitte geben einen knappen Blick in die Filmtheorie und Filmgrammatik. Dabei wird in Abschnitt 6.1 ein allgemeiner narrativer Ausgangspunkt gewählt. In der Narratologie ist mittlerweile deutlich herausgearbeitet worden: Narration „itself is a deep structure quite independent of its medium“ [Chatman 1992, S. 403]. Das hat zur Folge, dass Fragen zur Struktur eines Dokumentes generisch und spezifisch – also für eine konkrete Inhaltsarchitektur – behandelt werden können. Zur Untersuchung der spezifischen Struktur werden in Kapitel 6.2 strukturierte cinematographische Dokumente eingeführt. Entscheidend ist dabei, dass mit diesem Dokumententyp ein zeitbasiertes Ausgabemedium gegeben ist. Die basalen syntagmatischen Strukturen dieses Dokumententyps werden in Kapitel 6.3 mit der Redefinition der klassischen Filmgrammatik durch Schmidt und Strauch eingeführt (vgl. [Schmidt, Strauch 2002], [Schmidt 2004]). Kapitel 6.4 beschreibt die Möglichkeit narrativer Hierarchisierungen. Damit stehen insbesondere auch in MPEG-7 repräsentierbare Syntagmen zur Klassifikation wesentlicher Unterstrukturen von Filmen bereit. In Kapitel 6.5 werden anschließend die Arbeiten von E. Branigan dahingehend untersucht, ob und welche Strukturen im klassischen Spielfilm auch unterhalb der Einstellungsebene für das Verständnis des Beobachters wichtig sind.

6.1 Narratologie

Unter Narratologie versteht man die umfassende Wissenschaft von der Erzählung. Sie untersucht alle Arten, wie eine Geschichte erzählt werden kann, ohne sich von vornherein medienspezifisch auf Texte, Filme, Musik, gesprochene Sprache oder Bilder einzuschränken.

G rard Genette als einer der ersten Narratologen differenziert in [Genette 1994] drei verschiedene Auffassungen einer Erz hlung²¹:

- (Auffassung 1: Inhalt + Erz hlakt): Als Erz hlung wird der m ndliche oder schriftliche Diskurs (discours) bezeichnet, der von einem Ereignis (oder einer Reihe von Ereignissen) berichtet.
- (Auffassung 2: Inhalt): Erz hlung ist eine repr sentierbare Abfolge von realen oder fiktiven Ereignissen. Dabei werden nur Handlungen und Situationen mit einbezogen, ohne dabei auf das sprachliche oder sonstige Medium R cksicht zu nehmen, das diese Handlung vermittelt (vgl. [Genette 1994, S. 15]).
- (Auffassung 3: Erz hlakt): In einem weiteren Sinn kann man unter Erz hlung auch den reinen Akt, die Art und Weise der Erz hlung verstehen, ohne dabei genauer auf deren Inhalt einzugehen.

Genette bezeichnet in einem Postscriptum seines Buches seine eigene zweite Definition als eine Definition von Geschichte als Gesamtheit erz hlbarer Ereignisse. Weiterhin unterscheidet er Narration von Erz hlung. W hrend er die Narration als den narrativen Akt (siehe Auffassung 3) bezeichnet, sei Erz hlung das Produkt dieses Akts, das diesen selbst z. B. als geschriebener Text im menschlichen Ged chtnis  berdauern kann (siehe Auffassung 1, vgl. [Genette 1994, S. 199])²².

Allgemeine Voraussetzung f r das Vorhandensein einer Erz hlung ist die Existenz einer minimalen narrativen Struktur (Erz hlstruktur) zwischen Teilen eines Dokuments. Diese ist definiert durch den zeitlichen Anfang, das zeitliche Ende und die dazwischen liegende Durchf hrung eines einen Zustand ver ndernden Vorgangs (Ausgangssituation, Ver nderung, Endsituation). Narrativ ist also ein Dokument oder ein Teildokument dann, wenn in ihm mindestens eine minimale narrative Struktur und somit ein Ereignis repr sentiert wird. David Bordwell unterscheidet zwei verschiedene Narrationsarten: die diegetische und die mimetische Narration (vgl. [Bordwell 1986, S. 3]).

- Mimetische Narration: Das Geschehen wird durch Figuren innerhalb der Geschichte erz hlt (der Erz hler spricht durch die Figuren); Typische Beispiele daf r sind Drama und Theater.

²¹ Auffassung 1 schlie t Auffassung 2 und Auffassung 3 ein.

²²  ber die Anordnung von Narration und Erz hlung schreibt Genette: „In ihrer ersten m ndlichen oder schriftlichen Auspr gung ist die Erz hlung mit der Narration v llig zeitgleich“ (vgl. [Genette 1994, S. 199]).

- Diegetische Narration: Das Geschehen wird durch die Worte eines Erzählers vermittelt, die beim Zuhörer die Vorstellung der Geschichte entstehen lassen (vgl. [Hickethier 1993, S. 102], der Erzähler spricht, nicht die Figuren); Typische Beispiele hierfür sind Hörspiel und Lesung.

Die Erzählung im Film weist die Besonderheit auf, dass hier die Figuren der mimetischen Narration durch die Erzählerperspektive der diegetischen Narration ergänzt werden können (vgl. [Hickethier 1993, S. 102]).

Film erzählt also gleichzeitig sowohl durch diegetische als auch durch mimetische Narration. Hickethier begründet diese Verbindung wie folgt: Film, Fernsehen und Video „erzählen, indem sie zeigen, und sie erzählen durch Sprechen über etwas, was sie nicht zeigen“ [Hickethier 1993, S. 102]. Weiterhin unterscheidet man zwei Grundphänomene der Narration im Film: Chronologie (zeitlicher Ablauf) und Story (Inhalt). Der Filmemacher bestimmt den zeitlichen Ablauf einer Story und steuert so, wie der Zuschauer später die Geschichte wahrnimmt. Alles, was ein Zuschauer wahrnimmt und wie er es wahrnimmt, ist also im Voraus bestimmt und konstruiert²³.

6.2 Strukturierte cinematographische Dokumente

Wegen ihrer möglichen narrativen Tiefe bieten Filme (aus syntagmatischer Sicht) im Allgemeinen eine sehr reichhaltige logische Dokumentstruktur. Dies gilt im Besonderen für den Spielfilm. Denn hier werden einzelne Einstellungen nicht nur nach bestimmten Kriterien hintereinander gezeigt, sondern so angeordnet, dass eine Geschichte entstehen kann.

Die für die logische Strukturierung von Filmdokumenten benötigten narrativ basalen Syntagmen sind in [Schmidt, Strauch 2002] und [Schmidt 2004] beschrieben und werden entsprechend im Folgenden skizziert.

Damit ein Filmdokument so zerlegt werden kann, dass alle in ihm repräsentierten narrativen Zusammenhänge auch logisch sichtbar werden, sind alle Einstellungen eines filmischen Dokuments zunächst den basic *logical* objects (als „Inhaltsstück“ oder „content portion“, hier gekennzeichnet mit „⊗“) zuzuordnen (s. Abbildung 16).

²³ Diese Konstruktion der Geschichte (Gestaltung, Perspektive und Anordnung der Geschehnisse) nimmt der Zuschauer auf und versucht daraus das eigentliche Geschehen zu rekonstruieren. Einerseits liefert ihm die Erzählung Strukturen, die Verbindungen zwischen dem Wahrgenommenen herstellen. Zum anderen stellt auch der Zuschauer selbst aus dem Gesehenen und Gehörten wieder Verbindungen her. „Nicht der Film vermittelt Bedeutung, sondern der Zuschauer erkennt aufgrund bestimmter Bedingungen in ihm Bedeutung“ [Hickethier 1993, S. 100].

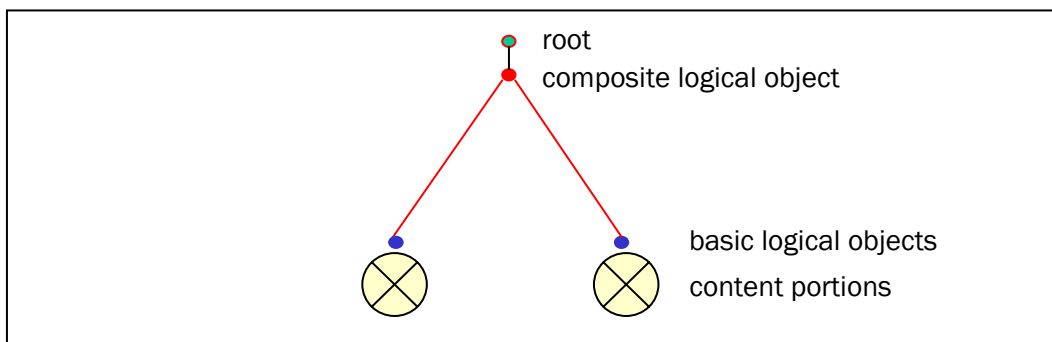


Abbildung 16: Zuordnung von content portions zu basic logical objects.

„Eine einzelne Einstellung ist ein Datensatz, der einer vorgegebenen Menge von audiovisuellen Normen von Bewegtbilddaten genügt“ [Schmidt, Strauch 2002, S. 12f.]. Daraus ergibt sich die folgende Definition gemäß [Schmidt 2004]:

Ein Dokument *D* heißt *cinematographisch*, wenn

1. jedem basic logical object mindestens eine cinematographische content portion zugeordnet werden kann,
2. ein Layoutprozess *L* existiert, der für mindestens eine audiovisuelle content portion eines jeden basic logical object des Dokuments ein basic layout object generiert,
3. *L* die Menge aller Einstellungen des Dokuments einer strengen zeitlichen Ordnung unterwirft.

Die Begrifflichkeiten „cinematographisches Dokument“ und „Film“ werden hier explizit unterschieden. Für den menschlichen Beobachter ist ein cinematographisches Dokument, beziehungsweise Teile davon, nur dann als solches wahrnehmbar, wenn es „aufgeführt“ wird (Imaging). Denn dann sind die gezeigten Einstellungen in ihrer Reihenfolge einmalig. Die weitere Begriffsbildung von Teilen eines cinematographischen Dokuments umfasst deshalb nicht nur dessen Logik, sondern auch dessen Layout. Für layoutierte Teile eines cinematographischen Dokuments gilt gemäß [Schmidt 2004]:

Ein *Segment* in einem cinematographischen Dokument ist eine nichtleere Teilfolge in der layoutierten zeitlichen Folge aller Einstellungen für einen gegebenen Layoutprozess²⁴.

²⁴ Diese Definition bezieht sich explizit auf Segmente von cinematographischen Dokumenten. Im Unterschied zur Segmentdefinition von MPEG-7 (siehe Abschnitt 5.2 und 7.1) fällt diese Definition wesentlich strenger aus, bleibt aber mit MPEG-7 kompatibel.

Gemäß dieser Definition ist es möglich, dass ein Beobachter einen Teil eines Films auch dann betrachtet, wenn eine logische oder zeitliche Lücke existiert.

Ein Segment kann, muss aber nicht Teil der Darstellung eines Syntagmas sein. Ein Syntagma klassifiziert Teilbäume der logischen Struktur eines Dokuments, die mindestens ein composite logical object enthalten und durch einen Layoutprozess auf mindestens ein Segment abgebildet werden (vgl. [Schmidt 2004, S. 102]). Im Weiteren wird daher von Segmenten eines Syntagmas gesprochen.

Die folgende Abbildung zeigt den Fall, bei dem lediglich ein einzelner Pfad von einem composite logical object zu einem basic logical object vorhanden ist.

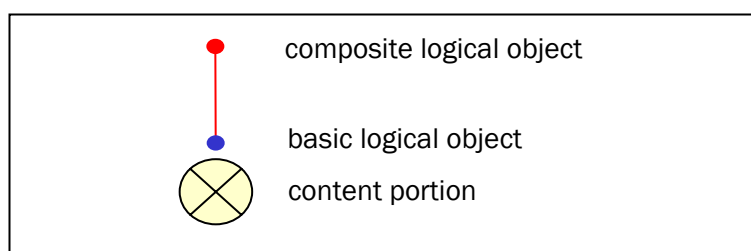


Abbildung 17: Pfad von einem composite logical object zu einem basic logical object.

Bezogen auf die Metz'sche Große Syntagmatik lässt sich eine solche Struktur entweder als Einfügung oder als Planszene einem Syntagma zuordnen²⁵.

Wird innerhalb eines Layoutprozesses eine einzelne Einstellung keinem Syntagma zugeordnet, so ist sie in die Kategorie „Planszene“ einzuordnen. Planszenen wurden in [Schmidt, Strauch 2002] eingeführt und sind der Ausgangspunkt der weiteren Untersuchungen²⁶.

6.3 Monochronie

Eine *Planszene* bildet einen zeitlichen Ablauf homomorph ab. Dadurch ist dem Beobachter im Allgemeinen der zeitliche Ablauf eindeutig klar. Was bedeutet dies nun für nicht triviale Segmente und deren Syntagmen?

Ein Beobachter B aus einer Beobachtermenge B kann Dokumententeile nur dann chronologisch aggregieren, wenn er einzelnen content portions eine (absolute oder relative

²⁵ Einfügungen werden in der vorliegenden Arbeit lediglich als filmisches Analogon von Parenthesen behandelt. Eine Theorie der Einfügungen kann es naturgemäß vermutlich erst am Ende einer syntagmatischen Theorie des Films geben (vgl. [Schmidt, Strauch 2002, S. 70f.]).

²⁶ Die *Planszene* heißt bei Metz noch „Plansequenz“. Die Kategorie der „Plansequenz“ wird hier nicht benutzt, wie in [Schmidt, Strauch 2002] ausführlich begründet wurde.

ve) diegetische Zeit zuordnen und eine Relationierung dieser Zeiten durchführen kann (siehe Abbildung 18).

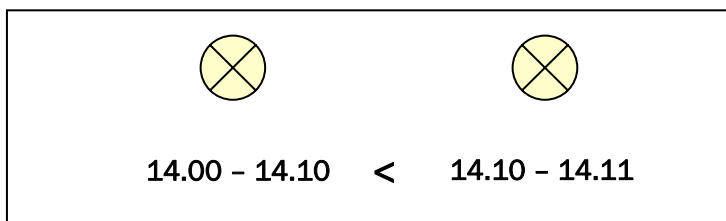


Abbildung 18: Zeitliche Relation zweier content portions.

Es kann angenommen werden, dass sich die diegetische Zeit jeder Einstellung in der Menge aller abgeschlossenen, nicht entarteten Intervalle $I(M)$ eines geeigneten zeitlichen Messraums M repräsentieren lässt (vgl. [Schmidt 2004])²⁷. Um nun die zeitlichen Verhältnisse zwischen mehreren Einstellungen innerhalb der Diegese zu beschreiben, wird im Folgenden die Allen'sche Zeitlogik benutzt (vgl. [Allen, Hayes 1985]). Für die weitere Analyse dieser zeitlichen Verhältnisse sind die folgenden Fälle zu berücksichtigen:

1. diegetisches Aufeinanderfolgen „ohne Lücken“ (wie in einer Szene (s.u.)),
2. diegetisches Aufeinanderfolgen „mit Lücken“ (wie in einer Sequenz (s.u.)),
3. diegetische Gleichzeitigkeit/Kopräsenz,
4. diegetische Überlappung,
5. kein zeitliches Verhältnis zueinander.

Seien T und T' zwei Einstellungen mit den diegetischen Intervallen I und I' . Der erste Fall des diegetischen Aufeinanderfolgens „ohne Lücken“ wird mit der Allen'schen Basisrelation $\text{meets}(I, I')$ modelliert; der zweite Fall des Aufeinanderfolgens „mit Lücken“ durch $\text{before}(I, I')$ bzw. der Inversen $\text{after}(I, I')$; der dritte Fall der Gleichzeitigkeit mit der Relation $\text{contains}(I, I')$. Diese Relation verlangt, dass die beiden Intervalle ineinander enthalten sind (vgl. [Davis 1990, S. 149]). Der vierte Fall der diegetischen Überlappung wird mit der zeitlichen Relation $\text{overlaps}(I, I')$ modelliert. Der fünfte Fall hat keinen Bezug zur chronologischen Syntagmatik. Hier müssen für ein Dokument mehrere zeitliche Zusammenhangskomponenten vorgesehen werden, die keinen gemeinsamen Ursprung haben. Um diesen Fall zu modellieren, müsste man das Dokument in mehrere chronologische Teildokumente aufteilen und anschließend (getrennt voneinander) die ersten vier Fälle zur Klassifikation heranziehen.

²⁷ Zur Definition eines Messraumes vgl. [Davis 1990, S. 147].

Eine Beobachtermenge B eines cinematographischen Dokuments lässt sich daher gemäß [Schmidt 2004] folgendermaßen definieren:

Ein Teilbaum der logischen Struktur eines Dokuments D heißt für eine Beobachtermenge B *chronologisch*, wenn für jede zweielementige Teilmenge $\{T, T'\}$ der zugehörigen Einstellungen M_D mit diegetischen Zeiten I und I' gilt:

meets(I, I') oder before(I, I') oder contains(I, I') oder overlaps(I, I').

6.3.1 Die Szene

Um eine Szene eindeutig als eine solche klassifizieren zu können, muss eine gedankliche Ersetzungsprobe durchführbar sein: Eine Szene muss als eine einzige, räumlich zusammenhängende Einstellung inszenierbar sein. Dies ist vor dem Hintergrund der Kameraführung zu sehen: In einer Szene wechselt der Kamerastandpunkt diskontinuierlich – im Gegensatz zur einzelnen Einstellung – in der die Kamera von einem Standpunkt aus Objekten im Raum folgt. Bedingung ist dabei, dass die in der Szene abgebildeten Objekte tatsächlich als räumlich zusammenhängend erkennbar sind. Der Beobachter darf diesen Zusammenhang nicht nur annehmen, sondern muss diesen auch durch Anker im Raum erkennen können. Gemäß [Schmidt 2004] muss daher gelten:

Ein Teilbaum der logischen Struktur in einem cinematographischen Dokument D ist syntagmatisch für eine Beobachtermenge B dann eine *Szene*, wenn

1. dem Teilbaum mindestens zwei Einstellungen als content portions zugeordnet sind,
2. das vereinigte Raumbild aller zugehörigen Einstellungen von allen $B \in B$ unter Bezug auf das abgebildete Urbild als zusammenhängend konzeptionalisiert werden kann,
3. die Vereinigung der in den zugehörigen Einstellungen denotierten Zeiten von allen $B \in B$ als zusammenhängend konzeptionalisiert werden kann,
4. ein Layoutprozess existiert, so dass die durch ihn erzeugte Reihenfolge der Einstellungen und der zeitliche Ablauf im Urbild von allen $B \in B$ als homomorph etabliert werden kann,
5. in D keine weitere Einstellung, die die 2., 3. und 4. Anforderung erfüllt, einem basalen logischen Objekt außerhalb des Teilbaums zugeordnet ist.

In Bedingung fünf steckt eine wichtige Eigenschaft der Szene: Sie muss das Kriterium der Maximalität erfüllen. Würde man ihr eine weitere Einstellung hinzufügen, würde es sich nicht mehr um eine Szene handeln. Minimal kann man eine szenische Struktur mit der Darstellung des zeitlichen Zusammenhangs gemäß Abbildung 19 darstellen.

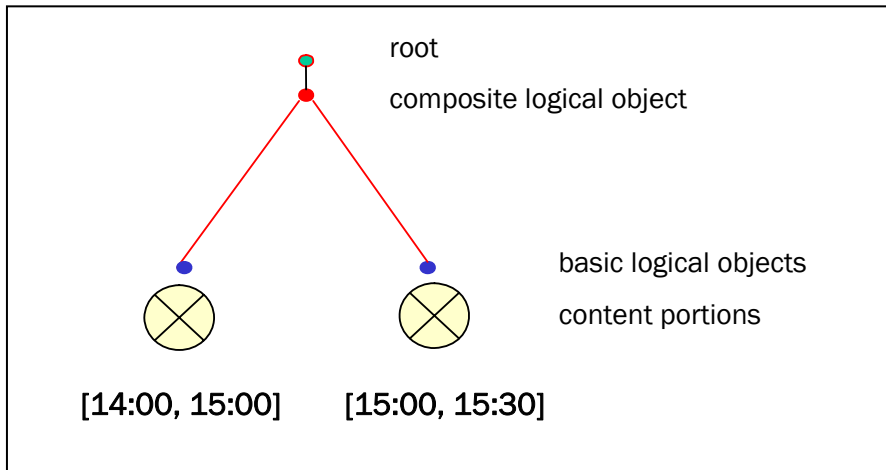


Abbildung 19: Darstellung einer Szene (ohne zeitliche Überlappung).

„Die Definition der Szene erlaubt auch zeitliche Überlappungen die explizit Kriterien kognitiver Adäquatheit einbeziehen“[Schmidt 2004, S.12ff.]. Natürlich können auch künstliche Hintergrundwelten herangezogen werden, solange diese die Formulierung von Zusammenhangsbedingungen erlauben. Die Definition der Szene verwendet keine Bedingungen aus der Handlungstheorie und ist aus diesem Grund weniger restriktiv als die Definitionen, die Handlungseinheiten einbeziehen.

6.3.2 Die Sequenz

Im Gegensatz zur Szene muss bei der Klassifikation einer Sequenz kein zeitlich direktes Aufeinanderfolgen gegeben sein. Gemäß [Schmidt 2004] gilt:

Ein Teilbaum der logischen Struktur in einem cinematographischen Dokument D ist für eine Beobachtermenge B eine *Sequenz*, wenn

1. dem Teilbaum mindestens zwei Einstellungen als content portions zugeordnet sind,
2. das vereinigte Raumbild aller zugehörigen Einstellungen von allen $B \in B$ unter Bezug auf das abgebildete Urbild als zusammenhängend konzeptionalisiert werden kann,
3. die Vereinigung der in den zugehörigen Einstellungen denotierten Zeiten von wenigstens einem $B \in B$ nicht als zusammenhängend konzeptionalisiert werden kann,

4. ein Layoutprozess existiert, so dass die durch ihn erzeugte Reihenfolge der Einstellungen und der zeitliche Ablauf im Urbild von allen $B \in B$ als homomorph etabliert werden kann,
5. in D keine weitere Einstellung, die die 2., 3. und 4. Anforderung erfüllt, einem basic logical object außerhalb des Teilbaums zugeordnet ist.

Gemäß dieser Definition ist es also beispielsweise nicht erlaubt, das gezeigte Einsteigen in ein Auto und das Losfahren in zwei Einstellungen als Sequenz zu klassifizieren, außer der räumliche Zusammenhang ist durch räumliche Anker für den Beobachter erkennbar. Die Minimalanforderung für eine solche Situation ist, dass aus allen Einstellungen Sequenzpaare mit wieder identifizierbaren räumlichen Ankern gebildet werden können (siehe Abbildung 20). Dies muss in der Weise geschehen, dass sich der Raum dieser Einstellungen in deren Gesamtheit als zusammenhängend konzeptionalisieren lässt. Dabei kann eine Sequenz wiederum komplette Szenen beinhalten.

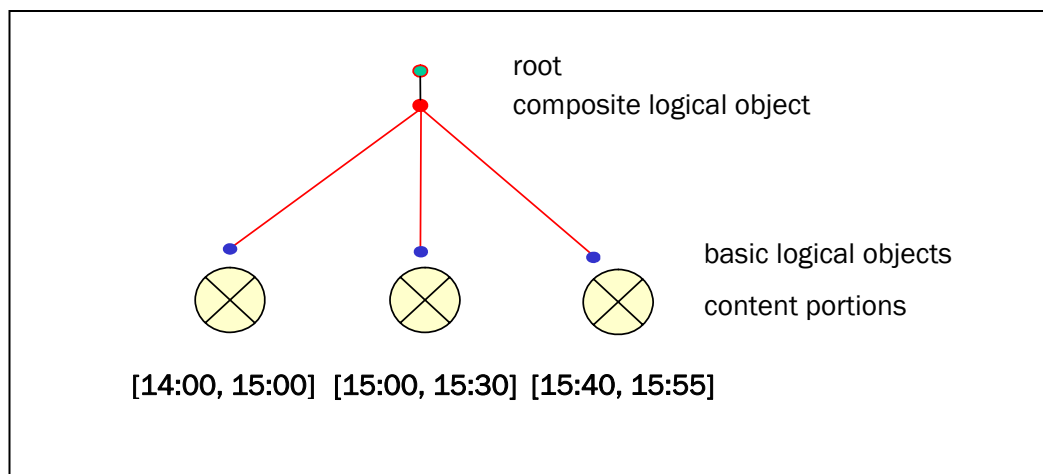


Abbildung 20: Darstellung einer Sequenz.

In Abweichung zu Metz legen Schmidt und Strauch in [Schmidt, Strauch 2002] folgende Definitionen für den narrativen Fall fest²⁸:

Ein (cinematographisch) basales Syntagma ist entweder eine Planszene oder eine Szene oder eine Sequenz.

Ein ganzes cinematographisches Dokument ist für eine Beobachtermenge B narrativ basal, wenn es nur aus basalen Syntagmen besteht.

²⁸ Schmidt und Strauch halten die Syntagmen „Planszene“, „Szene“ und „Sequenz“ als basale Syntagmen zur Klassifikation von „einfachen“ cinematographischen Dokumenten für ausreichend.

Die Verwendung von Planszene, Szene oder Sequenz ist oftmals die einzige mögliche Erzählform. Ein Beispiel hierfür ist die montagelose Aneinanderreihung von Aufnahmen einer Überwachungskamera. Da unterhalb der document logical root nur eine Ebene von composite logical objects erforderlich ist und Zusammenhangsbedingungen lediglich durch die Raumzeit gegeben sind, werden solche Dokumente als narrativ trivial klassifiziert. Erst durch Hierarchisierung zu narrativen Folgen entstehen komplexe narrative Einheiten, wie sie im Folgenden beschrieben sind²⁹.

6.4 Narrative Hierarchisierung

In der folgenden Grafik wird für den Fall der narrativen Hierarchisierung der Bereich der basalen Syntagmen verlassen. Unterhalb der Dokumentenwurzel und oberhalb der eingeführten ersten Montageebene wird nun eine zweite Montageebene eingeführt.

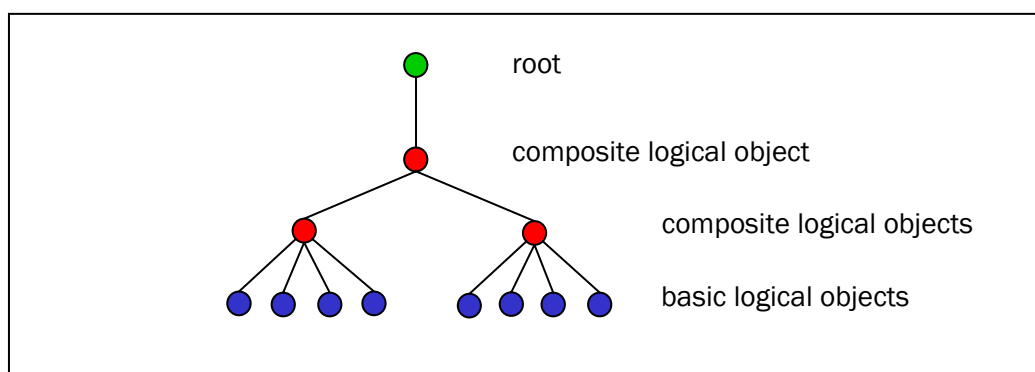


Abbildung 21: Aggregation zweier narrativ basaler Syntagmen zu einer narrativen Folge.

Es ist ein Mindestmaß an zeitlicher Ordnung zwischen den syntagmatisch klassifizierten Syntagmen der ersten Montageebene nötig, um für eine Beobachtermenge B auch die zweite Montageebene syntagmatisch als chronologisch zusammenhängend darzustellen. Für die weiteren Analysen muss daher gemäß [Schmidt 2004] gelten:

Für die syntagmatische Klassifikation eines Teilbaumes der logischen Struktur in einem cinematographischen Dokument D für eine Beobachtermenge B als narrative Folge (erster Ordnung) gilt notwendig, dass

1. die Wurzel dieses Teilbaumes mindestens zwei logische Objekte als Nachfolger enthält,
2. all diese Nachfolger sich von B als basale Syntagmen klassifizieren lassen,

²⁹ Die narrative Folge als Syntagma zweiter Ordnung hatte Metz außerhalb seiner Klassifikation eingeführt und benutzt; er ließ dieses Konstrukt aber definitorisch im Ungefähren.

3. in jedem der zu diesen Nachfolgern gehörigen Segmente eine Einstellung existiert, die für *B* ein zeitliches Verhältnis zu mindestens einer Einstellung in mindestens einem der anderen zugehörigen Segmente hat und
4. zwischen allen zu diesen Nachfolgern gehörigen Segmenten von *B* paarweise kein räumlicher Zusammenhang konzeptionalisiert werden kann.

Gemäß Bedingung eins ist in Abbildung 21 der Knoten der zweiten Montageebene die Wurzel des relevanten Teilbaums. Während Bedingung zwei offenkundig notwendig ist, stellt Bedingung drei sicher, dass es sich um ein chronologisches Syntagma handelt. Bedingung vier garantiert, dass ein basales Syntagma nicht als *narrative Folge* klassifiziert wird. Diese Definition ist die Basis für weitere Analysen.

Die empirische Abgrenzung von narrativen Folgen ist meist unproblematisch. Denn oftmals ist für den menschlichen Beobachter analog zum Raum-Zeit-Kontinuum einer Szene ein narrativer Zusammenhang erkennbar. Dies kann durch das Medium Film besonders einfach dargestellt werden, vor allem wenn auf den raumzeitlichen Zusammenhang verzichtet werden kann, wie dies beispielsweise für den Fall von Perception der Fall ist (siehe Abschnitt 7.2).

Bisher wurde die Syntagmatik nur für ganze Filmbilder untersucht. Häufig ist aber nicht ein ganzes Filmbild, sondern nur ein Teil davon inhaltlich und syntagmatisch relevant. Die filmtheoretischen Grundlagen dafür liefert die bereits einleitend erwähnte Arbeit von E. Branigan [Branigan 1992].

6.5 Die Erzählebenen von Branigan

In diesem Abschnitt werden die „levels of narration“, die Erzählebenen nach Branigan, eingeführt und genauer betrachtet. Um diese besser nachvollziehen zu können, müssen zunächst einige grundlegende Begriffe eingeführt werden. So wird bei einer Erzählung zwischen Fiktion und Nicht-Fiktion unterschieden. Innerhalb der fiktionalen und nicht-fiktionalen Narration kann weiter zwischen diegetisch und nicht-diegetisch unterschieden werden.

Die *diegesis*³⁰ ist für den cinematographischen Fall der gesamte hör- und sehbare Bereich, der für die Figuren auf der Leinwand als wahrnehmbar angenommen werden kann. Es ist die Welt, in der die Charaktere leben und die diese wahrnehmen können. Petra Grimm definiert „diegesis“ wie folgt: „Mit der diegesis (diégèse) ist die

³⁰ Der Begriff diegesis (gleichbedeutend mit dem franz. Begriff diégèse) ist nicht zu verwechseln mit der diegetischen Narration (reine Erzählung, Gegensatz zu Mimesis).

dargestellte Welt des Erzählten (historie) in ihrer rekonstruierbaren Gesamtheit gemeint“ [Grimm 1996, S. 48]. Dementsprechend ist etwas nicht-diegetisch, wenn es nur für den Zuschauer bestimmt ist und nicht für die Figuren im Film. Dies können z. B. Filmmusik oder textuelle Einblendungen sein. Branigan unterscheidet acht Erzählebenen, in denen diese Unterscheidungen eine Rolle spielen³¹:

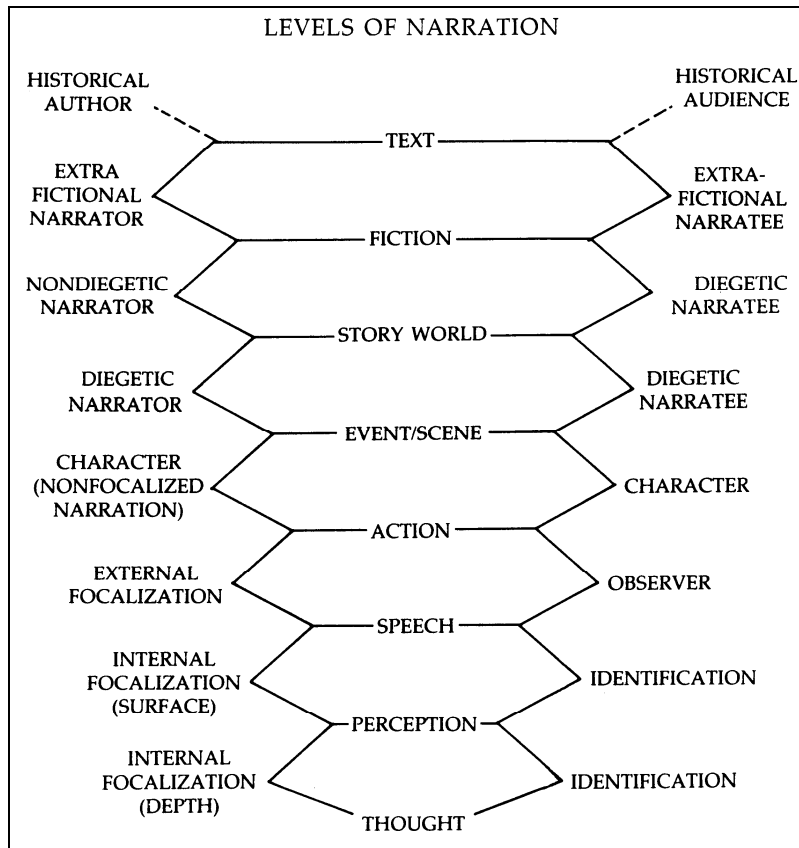


Abbildung 22: Die acht Erzählebenen [Branigan 1992, S. 96].

In der linken Spalte der Abbildung zeigt Branigan die Rolle des Senders (Erzählertyp), rechts die Rolle des Empfängers (Publikum). Ein wichtiges Prinzip der Narrations-ebenen beschreibt Branigan wie folgt:

„In general, several levels of narration will be operating simultaneously with varying degrees of explicitness and compatibility; that is, the spectator may describe the text

³¹ Branigan stützt sich im vorliegenden Text auf die Arbeit von Susan Lanser [Lanser 1981], und erweitert deren sechs Narrationsebenen um zwei weitere.

in several different ways, all of which may be accurate, each within a particular context and for a particular purpose" [Branigan 1992, S. 96].

Im Folgenden wird die Senderseite anhand der „levels of narration“ einzeln durchgearbeitet um anschließend deren syntagmatische Relevanz zu prüfen.

(1) Als *historical author* bezeichnet Branigan den wirklichen Autor, also den „Erzeuger“ einer Erzählung (z. B. Alfred Hitchcock). Der Autor ist mit einer gestrichelten Linie zum Text verbunden, weil wir allein aufgrund des Films nichts über ihn aussagen können. Dies entspricht der Annotation durch bibliothekarische Metadaten, die bereits in Abschnitt 2.3 eingeführt wurden. Der *historical author* würde im Dublin Core mit dem Attribut „author“ annotiert werden. Allerdings steht der *historical author* indirekt in Verbindung mit dem Film, indem ihm ein gewisser Ruf vorausgeht. Alfred Hitchcock beispielsweise ist bekannt durch seine Thriller, seine Art, Spannung zu erzeugen. Würde ein neuer Film von ihm angekündigt, so würde das Publikum indirekt einen spannenden Film erwarten. Ob das dann auch zutrifft, ist natürlich ungewiss.

(2) Eine Ebene tiefer findet sich der *extra-fictional narrator*. Dieser kann explizit oder implizit im cinematographischen Dokument vorkommen. Der explizite *extra-fictional narrator* erzählt außerhalb der Fiktion. So verkörpert z. B. Alfred Hitchcock einen *extra-fictional narrator*, indem er in seinem Film „The wrong man“ selbst (und nicht als eine Figur innerhalb der diegesis) zum Publikum spricht und den Film ankündigt³². Es ist hier deutlich zu erkennen, dass er von seinem Standpunkt außerhalb der Fiktion über die Fiktion (seinen Film) spricht. Der implizite *extra-fictional narrator*³³ stellt eine andere wichtige, nicht an Elemente des Dokumentes verankerbare Komponente einer Narration dar: Wie das Wort „implizit“ ausdrückt, ist hier kein direkt sichtbarer oder sprechender Erzähler am Werk. Es wird nichts explizit gezeigt oder ausgesagt, sondern der Zuschauer kann zwischen den Zeilen bzw. den Bildern lesen und aus einer Situation mögliche Schlüsse auf einen erzählerischen Einfluss ziehen. Diese Erzählform wird bewusst eingesetzt, um etwas über den Film zu vermitteln, ohne es direkt zu sagen oder zu zeigen. Damit eine solche implizite Beziehung zwischen dem Autor und dem Publikum entstehen kann, muss sich der implizite *extra-fictional narrator* gleichzeitig einer (oder mehrerer) anderer Narrationsebenen bedienen. Der implizite *extra-fictional narrator* kann also nicht alleine vorkommen.

(3) Der *nondiegetic narrator* erzählt von einem Standpunkt innerhalb der Fiktion aus über den Inhalt der Geschichte. Dieser Erzähler berichtet so, dass er nur für die Zu-

³² Hitchcock spricht lediglich über den Film, nicht jedoch über dessen Inhalt.

³³ In der Literaturwissenschaft findet sich auch der Begriff *implied author*.

schauer wahrnehmbar ist, nicht jedoch für die Figuren innerhalb der Erzählung. Beispielsweise ist der *nondiegetic narrator* aktiv, wenn eine – für eine Figur innerhalb der *diegesis* – unmögliche Kamerafahrt zu sehen ist (vgl. [Branigan 1992, S. 137]).

(4) Der *diegetic narrator* ist Branigan zufolge ein unsichtbarer Erzähler innerhalb der Welt der Figuren, durch den die Zuschauer die Handlung wahrnehmen. Dieser Erzähler hat allerdings die gleiche Sicht, das gleiche Wissen und die gleichen Möglichkeiten wie die Figuren selbst. Der *diegetic narrator* stellt also nur die unsichtbare Instanz einer Figur dar. Er ist dementsprechend notwendigerweise implizit. So könnte man beispielsweise sagen: „Wenn jemand dabei gewesen wäre, dann hätte er die Handlung genau so gesehen“ [Branigan 1992, S. 95].

Während die oben genannten Narratoren exklusiv für das Publikum erzählen, vermitteln die folgenden vier Erzählertypen die Erzählung durch einen Agenten. Agenten können sichtbare Figuren oder Gegenstände innerhalb der *Diegese* sein. Letztere bezeichnet Branigan als *Fokalisierer* (vgl. [Branigan 1992, S. 101]).

(5) Die Ebene der *nonfocalized narration* beschreibt die reinen Handlungen eines Agenten (einer Figur) als Akteur, ohne dabei auf die Bedeutung dieser Handlung einzugehen oder den mentalen Zustand der Figur wiederzugeben.

Focalized narration (oder kurz: *Focalization*) liegt vor, wenn der Beobachter eine Geschichte durch die Wahrnehmung einer Figur erlebt.

(6) Auf der Ebene der *external focalization* sieht (oder hört) der Zuschauer, was die Figur sieht, jedoch nicht direkt aus der Position der Figur, sondern von außerhalb. Der Zuschauer hat also Anteil an der Aufmerksamkeit der Figur, nicht aber an deren Gefühlen und Gedanken. Dieser Erzähler wird technisch z. B. durch einen sogenannten *eyeline match*³⁴ oder einen *POV-Shot* (*point-of-view-shot*) über die Schulter verwirklicht (vgl. [Grimm 1996, S. 51]). Der Betrachter sieht dabei die Figur und auch das, was die Figur selbst sieht, kennt aber deren Gedanken und Gefühle nicht. Auf dieser Ebene lassen sich leicht Beispiele für die gleichzeitige Präsenz mehrerer Erzählebenen zeigen. Eine Möglichkeit ist die Verwendung der *external focalization* kombiniert mit der Ebene des impliziten *extra-fictional narrators* im folgenden Beispiel: Eine Person liest Zeitung (*external focalization*). Der Zuschauer kann anhand der Blickrichtung erschließen, was der Leser liest und dass dieser weiß, was er liest.

³⁴ Ein *eyeline-match* ist eine Einstellung, die zuerst eine Figur zeigt, die von der Kamera weg in eine bestimmte Richtung blickt. In einer weiteren Einstellung zeigt die Kamera dann, was diese Figur sieht.

(7) Wenn der Zuschauer mit den Augen einer Figur sieht, spricht man von einer *internal focalization* der „Oberfläche“ (*surface*). Salopp formuliert ist diese Ebene „privater“ als die Ebene der *external focalization*. Keine andere Figur kann in Dokumentteilen repräsentierte Sachverhalte so verknüpfen wie diese. Verwirklicht wird die *internal focalization* z. B. durch einen POV-Shot mit den Augen.

(8) Noch tiefer fokussiert ist die Ebene der *internal focalization* der „Tiefe“ (*depth*). Durch einen Erzähler dieser Ebene ist es dem Zuschauer möglich, etwas über die Gedanken der Figur zu erfahren. Dies wird z. B. durch einen POV-Shot mit den Augen und einer gleichzeitigen inneren Stimme erreicht. Weiter kann eine *internal focalization* durch eine extreme Nahaufnahme (*close-up*) präsentiert werden (vgl. [Grimm 1996, S. 55]).

Zunächst kann man sagen, dass alle Narratoren, die „implizit“ sind, syntagmatisch keine Rolle spielen können (aufgrund ihres nur vom Empfänger erschließbaren, aber nicht im Dokument verankerten Status). Auch der „historical author“ kommt innerhalb des Dokuments nicht selbst vor. Bei der bisherigen Diskussion der vier oberen „levels of narration“ ergibt sich, dass nur der explizite *external-fictional narrator* und der *nondiegetic narrator* überhaupt eine syntagmatische Rolle spielen können.

Wie die Bezeichnung „*nondiegetic*“ bereits ausdrückt, bietet diese Ebene die Möglichkeit, außerhalb der Diegese für den Zuschauer zu berichten. Branigan beschreibt dazu eine für die Figuren unmögliche Kamerafahrt als Realisierungsbeispiel. Da Kamerafahrten, egal ob innerhalb der diegetischen Möglichkeiten oder nicht, keine Kriterien dafür enthalten, zwei Einstellungen miteinander zu aggregieren, liefert diese Ebene für die Fragestellungen dieser Arbeit keine neuen Anhaltspunkte.

Ähnliches gilt für den *external-fictional narrator*. Hierbei handelt es sich um eine Figur, die (außerhalb der Fiktion) mit den Möglichkeiten eines Fokalisierers ausgestattet ist. Nimmt ein solcher Narrator etwas wahr, so muss dies an die Ebenen der Fokalisierer delegiert werden. Allein die Tatsache, dass etwas außerhalb der Fiktion spielt, ergibt syntagmatisch keinen Mehrwert.

Nun kann die Ebene der Narratoren verlassen und zur Ebene der Charaktere übergegangen werden.

Auf den vier unteren Ebenen haben wir es mit Charakteren und Fokalisierern zu tun, die in Dokumententeilen als Bestandteil der Diegese repräsentiert sind. Die Charaktere der Ebene fünf sind typischerweise in Dokumentteilen unterhalb der Einstellungsebene repräsentiert. Würde eine Zusammenhangsbedingung zwischen mehreren Einstellungen formuliert, die auf das Vorkommen eines Charakters in all diesen Einstellungen basiert, so würden diese Bedingungen sicherlich „oberhalb“ raumzeitli-

cher Zusammenhangsbedingungen syntagmatisch relevant. Die im Abschnitt 6.3 behandelten basalen Syntagmen der Szene und Sequenz sind typischerweise in dem Sinne aufzufassen, dass über Charakteridentität definierte syntagmatische Einheiten enthalten sein können, aber nicht müssen.

Bei den Ebenen sechs bis acht macht Branigan zunächst keine Aussage zu möglichen syntagmatischen Konsequenzen. Diesen Ebenen gemeinsam ist aber, dass es sich um Annahmen über Geschehnisse im Nahbereich des Fokalisierers handelt. Man kann also erwarten, dass diese drei Ebenen syntagmatisch „oberhalb“ der auf raumzeitlichen Zusammenhangsbedingungen basierenden basalen Syntagmen relevant werden. Filmtechnisch zeigt sich, dass in allen drei Ebenen der POV-Shot als Realisationsmittel angegeben wird. Ein POV-Shot setzt nicht zwangsläufig eine raumzeitliche Einheit voraus. Wir können also annehmen, dass der (medienspezifische) POV-Shot in einem cinematographischen Dokument zu Syntagmen führt, die „oberhalb“ von Szene und Sequenz liegen und eine Brücke zwischen der raumzeitlichen Aggregation und sonstigen narrativen Aggregationen liefert.

Bisher wurden Segmente nur auf Einstellungsebene syntagmatisch behandelt (im Sinne einer Großen Syntagmatik), so dass bei deren Klassifikation der gesamte visuelle Inhalt einer Einstellung zugrunde gelegt wurde. Ausschnitte oder Teile aus dem Filmbild wurden dabei nicht berücksichtigt. Dies kann man bei den narrativen Zusammenhangsbedingungen der unteren drei Ebenen nicht mehr erwarten. Bei der Analyse des POV-Shots wird man also die Große Syntagmatik verlassen müssen und syntagmatisches Neuland (innerhalb des von MPEG-7 definierten Baukastens) für Segmente betreten müssen.

7 Perzeptive Syntagmen

Ein Beobachter kann trotz eines fehlenden räumlichen Zusammenhangs Segmente als zusammengehörig auffassen, wie im folgenden Beispiel: Eine Person, Anna, blickt in einer Einstellung in einen Raum. Eine weitere Einstellung zeigt einen dort liegenden Apfel, von dem der Beobachter annimmt, dass Anna ihn sieht.

Die Klassifikation dieser beiden Einstellungen als Szene (vgl. [Schmidt, Strauch 2002]; vgl. Abschnitt 6.3) ist nicht erlaubt, wenn der räumliche Zusammenhang für einen Beobachter nicht erkennbar ist. In diesem Falle ist aber gleichwohl eine Aggregation durch nicht im Filmsignal codierte Zusammenhangsbedingungen möglich, wenn wenigstens einem der in einer Einstellung repräsentierten Objekte perzeptive Fähigkeiten geeignet zugeschrieben werden. Im obigen Beispiel wäre eine mit Anna gelabelte Pixelmenge ein solches Objekt: Intradiegetisch existiert es in der filmischen Welt (Diegese) als Person und ist mit perzeptiven Fähigkeiten ausgestattet. Dieser Perzeptor Anna selbst ist ferner im Allgemeinen nicht durch eine ganze Einstellung repräsentiert, sondern durch eine Pixelmenge innerhalb einer Einstellung.

Grundsätzlich kann ein Perzeptor im Film in unterschiedlichen Weisen auftreten. Ein intradiegetischer Perzeptor P ist ein in der Diegese existierender Perzeptor und ein sogenannter Fokalisierer. Durch geeignete Filmmontage sind einem Beobachter B die gleichen visuellen Erfahrungen wie P möglich. In diesem Falle liegt eine interne Fokalisierung vor. Im aktuellen Beispiel bedeutet dies, dass sowohl Anna (Perzeptor P) als auch der Beobachter B den Apfel „gleich“ sehen. Zeigt das Perzeptionssegment dagegen Inhalte außerhalb der perzeptiven Möglichkeiten des Fokalisierers, so liegen eine externe Fokalisierung oder andere Verknüpfungen von Perzeptor und Perzept vor.

Bei der internen Fokalisierung wird als Perzept eines Perzeptors von einem Beobachter der Inhalt ganzer Filmbilder angenommen. Im Beispiel der einen Apfel erblickenden Anna ist dies das wenigstens eine Einstellung umfassende Teilsegment, welches zeigt, was die Person Anna „gerade“ sieht. Hier zeigt sich eine wichtige Besonderheit des Films bei der Repräsentation von Perzeptionen: Ein ganzes Segment auf Einstellungsebene kann als Repräsentant einer visuellen Erfahrung funktionieren.

Insgesamt erlaubt die filmische Repräsentation von Perzeption durch die Verknüpfung eines Teilsegmentes, das einen Perzeptor repräsentiert, mit ganzen Einstellungen, die Perzepte repräsentieren, einzelne Segmente unterhalb der Einstellungsebene mit ganzen Einstellungen zu verknüpfen.

Dies wird nun ausgearbeitet, indem zunächst die Syntax und danach die Semantik von Perzeptionssegmenten behandelt wird.

7.1 Zur speziellen Syntax von Perzeptionssegmenten

Für die weitere Analyse und Segmentierung eines Perzeptors ist die Zerlegung einer Einstellung in weitere Teilsegmente erforderlich. Um nun unterhalb der Einstellungsebene Perzeptoren auszeichnen zu können, wird die Zerlegung eines Segmentes in Teilsegmente gemäß MPEG-7 angewendet. Da MPEG-7 in Kapitel 5.2 für diese Zwecke schon ausführlich beschrieben wurde, darf die Beschreibung an dieser Stelle kurz ausfallen.

Definitionsgemäß (vgl. Abschnitt 6.2) ist für jede Segmentierung eines cinematographischen Dokuments eine tiefere hierarchische Segmentierung der Einzelsegmente erlaubt, so dass ein *Segmentbaum* entstehen kann. Unterschieden werden gemäß [Martinez 2002] Temporalsegmente (Video-Segment), Spatialesegmente (Still-Region) und Spatiotemporalsegmente (Moving-Region). Die grundlegenden Eigenschaften dieser Segmente finden sich in Abschnitt 5.2. Im Hinblick auf eine große Syntagmatik, die nur auf ganzen Einstellungen operiert, soll im Weiteren der Begriff des Einstellungssegmentes diejenigen Segmente kennzeichnen, die lediglich aus Einstellungen bestehen.

Bei der Behandlung des räumlichen Zusammenhangs von Perzeptionssegmenten stellt sich insbesondere die Frage, ob eine Moving-Region für die Segmentierung des Perzeptors im Filmbild brauchbar ist. Dabei ist zunächst festzustellen, dass bei der filmischen Darstellung eines Perzeptionsvorgangs immer ein einziger Perzeptor angenommen wird.

Dies würde zuerst auf eine räumlich verbundene Moving-Region schließen lassen, da es sich oft nur um eine einzige Person handelt, die perzipiert. Allerdings wäre es auch denkbar, dass während der Einstellung des Perzeptors ein Teil der perzipierenden Person verdeckt ist, so dass unter Umständen zwei räumliche Fragmente entstehen; oder dass im Falle eines technischen Sehsystems mehrere Komponenten nicht zusammenhängend repräsentiert werden. Unter diesen Umständen muss daher eine räumlich nicht verbundene Moving-Region zugelassen werden, was innerhalb von MPEG-7 problemlos zu realisieren ist (siehe Abschnitt 5.2.4).

Weiter ist zu fragen, ob ein Perzeptor P , wenn er in einer Einstellung im Bild zu sehen ist, auch so lange für den Beobachter sichtbar sein muss, bis die Einstellung zu Ende ist (siehe Abbildung 23).

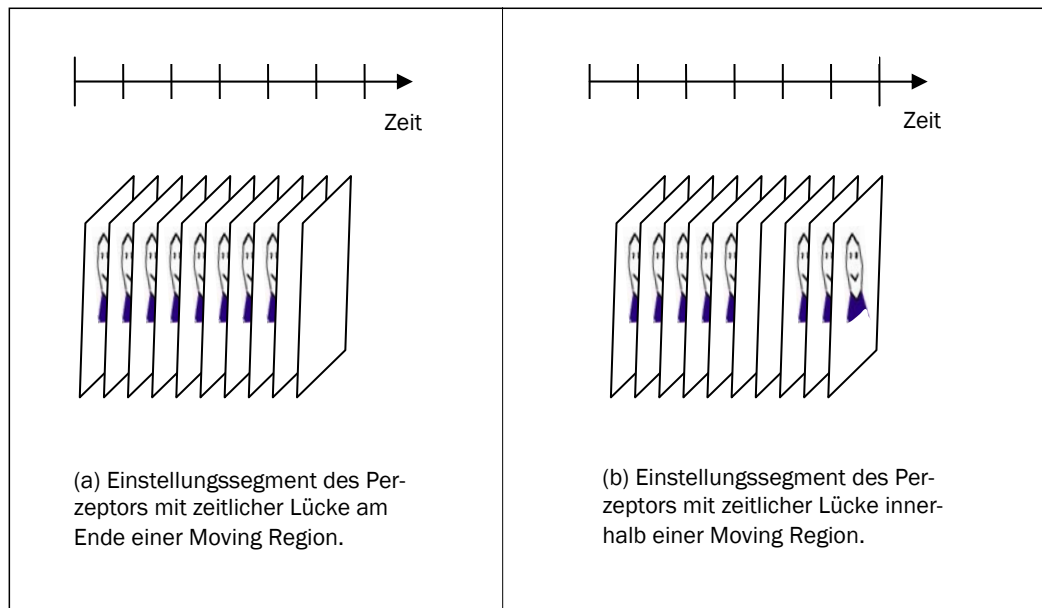


Abbildung 23: Segmente eines Perzeptors mit und ohne zeitliche Lücken.

Unter welchen dramaturgischen Gesichtspunkten der Filmemacher diese Montageform gewählt hat, ist nicht Gegenstand dieser Untersuchungen. In jedem Fall lässt sich der Perzeptor mit einer Moving-Region klassifizieren. Damit ist die Bedingung zur Aggregation dieses Segmentes erfüllt.

Zudem kann die Situation auftreten, dass der Perzeptor gezeigt wird und später noch ein weiteres Mal mit einer zeitlichen Lücke in der gleichen Einstellung wieder auftaucht (siehe Abbildung 23b). Hier lassen sich zwei Varianten herausarbeiten. Zum einen kann nur der zeitlich spätere Teil des Perzeptors als Perzeptorsegment angenommen werden, zum anderen kann die Moving-Region zeitliche Lücken enthalten. Beide Varianten klassifizieren eindeutig ein Perzeptorsegment und müssen daher als eine zeitlich nicht verbundene Moving-Region zugelassen werden, was innerhalb von MPEG-7 zu realisieren ist (siehe Abschnitt 5.2.4).

Für die Auszeichnung des Perzeptors unterhalb der Einstellungsebene ist also ein Segment zu wählen, das im Sinne von MPEG-7 eine Moving-Region labelt. Dabei darf insbesondere die Segmentierung unterhalb der Einstellungsebene zu räumlich und zeitlich nicht verbundenen Segmenten führen.

Um mit diesen syntagmatischen Mitteln den inhaltlichen Zusammenhang von Segmenten unterhalb der Einstellungsebene mit ganzen Einstellungen für den perzeptiven Fall zu untersuchen, werden nun Theoriebildungen aus der Situationstheorie benutzt, die für die Analyse raumzeitlicher Repräsentationen von cinematographischen Dokumenten besonders geeignet sind.

7.2 Zur situationstheoretischen Semantik visueller Perzeption

Die Besonderheiten der Verknüpfung eines Teils einer Einstellung mit einer anderen ganzen Einstellung im Falle visueller Perzeption, werden bestimmt von einem Sehvorgang. Zu dessen Analyse wird zunächst die Situationstheorie visueller Perzeption im Anschluss an [Devlin 1993] behandelt. Anschließend werden die situationstheoretischen Varianten von Perzeptoren für die Festlegung perzeptiver Relationen genutzt, um schließlich Kriterien für die perzeptive Verknüpfung von einem Teil einer Einstellung mit einem Segment auf Einstellungsebene zu gewinnen.

Die Situationstheorie wurde von Barwise und Perry in [Barwise, Perry 1983] eingeführt. Es wird im Folgenden die von Devlin in [Devlin 1993] einer breiteren wissenschaftlichen Öffentlichkeit zugänglich gemachte Variante benutzt, deren Ziel die einheitliche Behandlung von Information und Bedeutung ist. Diese Variante wird benutzt, um unabhängig von linguistischen Fragestellungen die Repräsentation von Perzeption in Bewegtbilddaten zu untersuchen.

Grundsätzlich wird davon ausgegangen, dass die einem Beobachter zugängliche Information von einem Kontext (der Situation) abhängt. Die Welt, in der sich ein Individuum bewegt, setzt sich zusammen aus einer Menge von Ausschnitten (Situationen s), aus denen wiederum Informationen (Infone σ) abgeleitet werden können. Der Sachverhalt „ein Infon σ trifft auf eine Situation s zu“ oder „eine Situation s stützt ein Infon σ “ wird bei [Devlin 1993, S. 49ff.] wie folgt repräsentiert:

$$s \models \sigma .$$

Die umfassendste Situation ist die Welt w .

Oft treffen mehrere Infone auf die gleiche Situation zu:

$$s \models \sigma \text{ und } s \models \sigma' \text{ für } \sigma \neq \sigma' .$$

Grundsätzlich ist ein Infon eine elementare diskrete Informationseinheit. Die allgemeine Form eines Infons σ ist gemäß [Devlin 1993, S. 38]:

$$\langle\langle R, a_1, \dots, a_n, i \rangle\rangle$$

mit R als n -stelliger Relation, a_1, \dots, a_n als Argumentstellen von R und i als sogenannte Polarität des Infons. Die Polarität gibt an, ob ein Infon wahr oder falsch ist. Ein Infon mit der Polarität 1 heißt auch *Faktum*.

Räumliche Lokalisierungen l und zeitliche Lokalisierungen t lassen sich in Infonen explizit auszeichnen, was sie besonders für die Anlage raumzeitlicher Zusammenhänge in Bewegtbilddokumenten tauglich macht. Die übliche Form eines Infons σ ist dann:

$$\ll R, a_1, \dots, a_{n-2}, l, t, i \gg.$$

Für das Beispiel der in einem Raum zu einer bestimmten Zeit einen Apfel erblickenden Anna können wir also als Infon bilden:

$$\sigma \models \ll \text{sehen}, \text{Anna}, \text{Apfel}, l_{\text{sehen}}, t_{\text{sehen}}, 1 \gg$$

mit einer vierstelligen Relation $R=\text{sehen}$, einem Perzeptor an der ersten Argumentstelle (Anna) und einem Perzept an der zweiten Argumentstelle (einem konkreten Apfel), einer zeitlichen Lokalisierung l_{sehen} für den Raum, in dem der Sehvorgang stattfindet, und t_{sehen} für den Zeitpunkt des Sehens.

Durch Verwendung von Parametern in Infonen kann eine einfache Abstraktion erzielt werden. Solche Parameter sind Generalisierungen von Argumenten (wie z. B. Individuen oder räumliche Lokalisierungen).

Grundparameter sind jeweils vom Typ eines Grundtyps. Die Grundtypen sind (vgl. [Devlin 1993, S. 70]):

- der Typ einer Situation (SIT),
- der Typ eines Individuums (IND),
- der Typ einer Relation (REL),
- der Typ einer zeitlichen Lokalisierung (TIM),
- der Typ einer räumlichen Lokalisierung (LOC),
- der Typ eines Infons (INF),
- der Typ eines Typs (TYP),
- der Typ eines Parameters (PAR),
- der Typ einer Polarität (POL).

Hat ein Infon an wenigstens einer Argumentstelle einen Parameter, nennen wir es parametrisch. Als Beispiel betrachten wir das parametrische Infon

$$\sigma \models \ll \text{sehen}, \dot{a}, \text{Apfel}, l_{\text{sehen}}, t_{\text{sehen}}, 1 \gg,$$

wobei \dot{a} ein Parameter vom Typ eines Individuums ist („Jemand sieht am Ort l_{sehen} einen Apfel zum Zeitpunkt t_{sehen} “).

Damit ein solches parametrisches Infon Information liefern kann, bedarf es eines Ankers f , der den Parameter \dot{a} mit einem konkreten Individuum, hier Anna, verbindet.

Ein Anker ist eine Funktion, die auf einer Menge von Grundparametern definiert ist und jedem dieser Grundparameter ein Objekt zuordnet. Im Beispiel bestehe diese Menge von Grundparametern für den Typ IND nur aus \dot{a} und werde über einen Anker f Anna zugeordnet:

$$f[\dot{a}] = Anna.$$

Sobald in einem parametrischen Infon über Anker alle Parameter mit einem Objekt verbunden sind, hat man wieder ein parameterfreies Infon. Im Beispiel würde sich nach Auflösung des Parameters durch den Anker das ursprüngliche Infon

$$\sigma \models \ll \text{sehen}, Anna, \text{Apfel}, l_{\text{sehen}}, t_{\text{sehen}}, 1 \gg$$

ergeben.

Von Infonen lassen sich durch Operationen zusammengesetzte Infone bilden. Sinnvolle Operationen sind die Konjunktion und die Disjunktion von Infonen. Für die Konjunktion gilt (vgl. [Devlin 1993, S. 159f.]):

$$s \models \sigma \wedge \tau \text{ gdw. } s \models \sigma \text{ und } s \models \tau.$$

Für die Disjunktion gilt (vgl. [Devlin 1993, S. 160f.]):

$$s \models \sigma \vee \tau \text{ gdw. } s \models \sigma \text{ oder } s \models \tau \text{ (oder beides).}$$

Für diese Arbeit ist vor allem die Konjunktion wichtig, da ein Agent in einer Szene typischerweise mehrere Infone „ersehen“ kann. Bei visuellen Erfahrungen von Menschen wird davon ausgegangen, dass immer eine Mehrheit von Dingen gesehen wird. Im Beispiel stellt die Aussage „Anna sieht einen Apfel“ nur eine elementare Aussage dar. Da der Apfel sich in einem Raum befindet, kann davon ausgegangen werden, dass der Agent Anna auch den Raum sieht, also den Boden, die Tapeten und Fenster usw. Daraus folgt, dass eine Beschreibung der gesehenen Dinge im Allgemeinen durch Konjunktion elementarer Infone erfolgt. Da jedoch bestimmte Dinge nur nebenbei gesehen werden, konzentrieren wir uns auf die Beschreibung der Infone, die im weiteren Verlauf des cinematographischen Dokuments vom Beobachter B informationell genutzt werden. Wird aus einer Einstellung nur ein Infon informationell genutzt, kann man dessen Betrachtungen auf nicht zusammengesetzte Infone beschränken (vgl. [Devlin 1993, S. 228]).

Perzipiert nun ein Agent etwas, so erzeugt dieses Perzipieren eine Änderung seines Geisteszustandes. Diese visuell erzeugte Änderung ist nach Devlin eine *visuelle Erfahrung* v (vgl. [Devlin 1993, S. 220]). Diese stellt also ein Ereignis dar, das einen Geisteszustand in einen anderen überführt.

Eine Situation s_{vis} ist die Quelle dieser visuellen Erfahrung v , die wir (mit Devlin) auch *visuelle Szene* nennen. Eine visuelle Szene s_{vis} verursacht also eine visuelle Erfahrung, die in einem Agenten eine Zustandsänderung hervorruft.

Hat ein Agent A (wie Anna) eine visuelle Erfahrung v mit zugehöriger visueller Szene s_{vis} , so gilt als visueller Inhalt dieser visuellen Erfahrung die infonische Aussage:

$$s_{vis} \models \sigma,$$

wobei σ ein parameterfreies (sehen) und im Allgemeinen ein zusammengesetztes Infon ist (vgl. oben). Der visuelle Inhalt erfasst, was in s_{vis} von A in dieser speziellen visuellen Erfahrung wirklich (im Allgemeinen parameterfrei) gesehen wird (vgl. [Devlin 1993, S. 226]). Wir sehen nur die Dinge, die wir als visuellen Inhalt einer visuellen Erfahrung haben. Für den visuellen Inhalt der Perzeption von Anna gilt also

$$s_{vis} \models \ll \text{gegenwärtig}, \text{Apfel}, l_{vis}, t_{sehen}, 1 \gg,$$

wobei l_{vis} die räumliche Ausdehnung von s_{vis} ist. Zum Zeitpunkt des Sehens t_{sehen} von Anna ist im „Nahbereich“ von l_{sehen} ein Apfel am Ort l_{vis} lokalisiert. Diese (notwendige) Nähe von l_{vis} zu l_{sehen} ist für die filmische Repräsentation ein entscheidender Unterschied zu intentionalen Zuständen wie beispielsweise Glauben. Perzeption ist immer situiert, salopp formuliert gilt also: Man sieht nur Teile der Situation, in der man räumlich und zeitlich steckt. Dies wird im Folgenden begründet.

Diese Begründung basiert auf dem äußeren Inhalt einer visuellen Erfahrung. Dieser besteht aus dem visuellen Inhalt und zusätzlich aus der Repräsentation der Tatsache, dass die visuelle Erfahrung v von der visuellen Szene s_{vis} erzeugt wird. Allgemein gilt für den *äußeren Inhalt* von v mit der Erzeugungsrelation „ \triangleright “:

$$w \models [(s_{vis} \models \sigma) \wedge (s_{vis} \triangleright v)].$$

Das heißt: „Die visuelle Erfahrung v kommt in ihrem eigenen äußeren Inhalt vor [...]“ [Devlin 1993, S. 231]. Der Teil einer visuellen Szene s_{vis} , der einen visuellen Inhalt $s_{vis} \models \sigma$ einer visuellen Erfahrung begründet, erzeugt auch diese visuelle Erfahrung.

Der äußere Inhalt schließt aus, dass visuelle Inhalte durch Eingebungen zustande kommen. Dies macht den Realitätsbezug einer visuellen Erfahrung aus. Bei Glauben oder Wünschen existieren im Allgemeinen keine äußeren Umstände, die einen solchen Glauben oder Wunsch hervorrufen. „Man glaubt oder wünscht eben einfach“ [Devlin 1993, S. 230].

Bei Perception verhält es sich anders. Kann einem Agenten A die Möglichkeit visueller Perception zugeschrieben werden und befindet sich A in einer bestimmten Situation (z. B. in der visuellen Szene s_{vis}), so hat A defaultmäßig eine visuelle Erfahrung v , da diese bei funktionierender Perception unweigerlich von s_{vis} erzeugt wird. Ein filmisches Beispiel, in dem dies explizit genutzt wird, ist eine Szene aus dem Film *Clockwork Orange*: Der gewalttätige Alex wird aus Rache gezwungen, sich brutale Filmsequenzen anzusehen. Er ist auf einen Stuhl gefesselt und seine Augen werden mit Augenklammern versehen, damit er der Wahrnehmung der Filmsequenzen nicht entkommen kann.

Der äußere Inhalt stellt den Realitätsbezug der filmischen Repräsentation eines Perzeptionsvorganges sicher. Er verlangt nicht von vornherein, dass Perzeptor und Perzept räumlich oder zeitlich etwas miteinander zu tun haben, wenn dies die Verursachungsrelation \triangleright nicht ausdrücklich vorsieht. Die oben schon angesprochene Situiertheit der Perception ergibt sich damit als solche nicht aus dem äußeren Inhalt. Diese folgt aus der Struktur der visuellen Erfahrung, mit der die situationstheoretische Analyse nun abgeschlossen wird.

Ein Agent kann (wie bereits formuliert) nur Dinge in einer Situation sehen, in der er auch steckt. Was hat das mit der Struktur der visuellen Erfahrung zu tun?

Allgemein ist die Struktur $S(I)$ eines mentalen Zustandes I (z. B. Wünschen oder Glauben) wie folgt aufgebaut:

$$S(I) = \langle M, s^{\#}, R^{\#}, a_1^{\#}, \dots, a_n^{\#}, i \rangle,$$

wobei M den psychologischen Modus bezeichnet (z. B. Glaube, Wunsch), $s^{\#}$ den Begriff der Situation angibt, $R^{\#}$ ein Begriff einer Relation ist, $a_1^{\#}, \dots, a_n^{\#}$ Individuenbegriffe sind und i wieder die Polarität angibt (vgl. [Devlin 1993, S. 204]). Dieser Struktur entspricht als äußerer Inhalt des mentalen Zustandes einer (o.B.d.A. elementare) infonischen Aussage

$$s_{vis} \models \langle \langle R, a_1, \dots, a_n, i \rangle \rangle.$$

Die Struktur $S(I)$ existiert bei allen intentionalen Geisteszuständen (vgl. [Devlin 1993, S. 204]) und kann situiert oder nicht situiert sein – je nachdem, ob zur Struktur ein Begriff der Situation im jeweiligen psychologischen Modus angegeben wird oder nicht. Bezieht sich die Situation eines Geisteszustandes auf die aktuelle Situation des Agenten und bildet dieser keinen Begriff dieser Situation, so wird die Situation in der Struktur durch einen Strich repräsentiert:

$$S(I) = \langle M, -, R^{\#}, a_1^{\#}, \dots, a_n^{\#}, i \rangle$$

Beispielhaft dafür ist der Glaube, dass es regnet. Zum einen kann dieser Glaube situiert geglaubt werden: „Ich glaube, dass es regnet“ (hier und jetzt). Das bedeutet nicht, dass es keine Situation gibt, für die geglaubt wird, dass es regnet. Vielmehr bezieht sich dieser Glaube auf die Situation, in der dieser Glaube geglaubt wird. Zum anderen kann sich dieser Glaube explizit auf eine (andere) Situation beziehen wie z. B. „Ich glaube, dass es in Hannover regnet“, wobei dieser Glaube einen Begriff der Situation beinhaltet.

Visuelle Erfahrungen werden daher im Folgenden immer als situiert behandelt. Für die Struktur einer visuellen Erfahrung ergibt sich gemäß [Devlin 1993, S. 230] insgesamt:

$$S(v) = \langle \text{Sieht}, -, R^{\#}, a_1^{\#}, \dots, a_n^{\#}, i, v \rangle$$

Für den äußeren Inhalt einer visuellen Erfahrung und der allgemeinen Struktur $S(I)$ ergeben sich nun zwei Dinge:

1. Der äußere Inhalt $w \models [(s_{vis} \models \sigma) \wedge (s_{vis} \triangleright v)]$ einer visuellen Erfahrung v enthält (wie beschrieben) diese visuelle Erfahrung v . Da v von s_{vis} erzeugt wird, ergibt sich, dass neben dem visuellen Inhalt auch die visuelle Erfahrung in der Struktur der visuellen Erfahrung vorkommt. „Man kann nicht sehen, ohne zu erfahren, dass man sieht“ [Devlin 1993, S. 230].
2. Darüber hinaus ist eine visuelle Erfahrung situiert. Insbesondere kann angenommen werden, dass sich der Agent räumlich in der Nähe des Perzeptes und zeitlich zu ihm kopräsent aufhält.

Damit sind wir am Kern der semantischen Modellierung filmischer Repräsentationen von Perzeptionsvorgängen:

1. Durch den äußeren Inhalt müssen filmische Repräsentationen von Perzeptionsvorgängen „Realitätsadäquatheit“ vermitteln: Das für einen Perzeptor physikalisch mögliche Perzept muss zunächst mit dem repräsentierten Perzept konsistent sein. So dürfen in einem Perzept keine Objekte auftauchen, die in der Diegese nicht existent sein können. Sonst hätten wir es nicht mit filmischen Perzeptionen, sondern z. B. mit Halluzinationen (die keinen äußeren Inhalt haben) zu tun.

Ferner müssen die Repräsentationen der Perzepte kompatibel mit den perzeptiven Möglichkeiten des Perzeptors sein. Das Perzept einer Mikroskopaufnahme kann beispielsweise nur unter Zuhilfenahme eines Vergrößerungsinstrumentes für einen intradiegetischen Perzeptor so gesehen werden. Ein weiteres Beispiel: Ein kurzsichtiger Perzeptor kann ohne geeignete Sehhilfe nur unscharf in die Ferne blicken. Die dazugehörige plausible filmische Repräsentation darf daher keine kantenscharfe Objekte aus der Ferne zeigen, wenn die Sehhilfe nicht getragen wird. Sind solche Bedingungen nicht erfüllt, so kann man das vermeintliche Perzept nicht einem intradiegetischen Perzeptor zuschreiben. Die filmische Repräsentation würde unplausibel.

Schließlich führt die Verursachungsrelation $s_{vis} \triangleright v$ in der filmischen Repräsentation dazu, dass unabhängig davon, ob das Perzept des Agenten dem Beobachter zugänglich ist oder nicht, der Agent als in der Perzeptionssituation ständig visuelle Erfahrungen habend angenommen wird. Die Ungültigkeit dieser Annahme muss (z. B. durch Verbinden der Augen) gezeigt werden.

2. Visuelle Erfahrungen haben eine situierte Struktur. In der filmischen Repräsentation führt dies dazu, dass in der Diegese Perzept und Perzeptor

- zeitlich kopräsent und
- räumlich benachbart

sein müssen. Zeitliche Kopräsenz heißt, dass insbesondere ein Beobachter eine zeitliche Synchronizität zwischen Perzeption und dem Perzept konzeptionalisiert. Räumliche Nachbarschaft bedeutet, dass für einen Beobachter B der Raum des Perzeptes unter Berücksichtigung der perzeptiven Möglichkeiten von P wahrnehmbar sein muss.

Ausgehend davon können nun perzeptive Relationen gebildet werden, die unter Nutzung der bisherigen Ergebnisse zur Modellierung des perzeptiven Zusammenhangs von Einstellungen genutzt werden.

Die Annahmen bzgl. der kognitiven Fähigkeiten von P haben entscheidende Auswirkungen auf die Art und Weise der filmischen Darstellung von Wahrnehmung und Se-

hen. Zur Analyse unterscheiden wir in Anlehnung an Devlin die perzeptiven Relationen *perc* („Wahrnehmen“), *see* und *see_that* (vgl. [Devlin 1993, S. 204]).

(*perc*:) Wenn ein Perzeptor P etwas wahrnimmt, hat er eine visuelle Erfahrung. Kennen wir aber als Beobachter von P nicht den visuellen Inhalt dieser Wahrnehmung, kann man als äußeren Inhalt einer Wahrnehmung lediglich

$$w \models (s_{vis} \triangleright v)$$

ansehen. Dies zeigt sich auch deutlich in der filmischen Serialisierung, die später erläutert wird.

(*see*:) Die bloße Wahrnehmung eines Perzeptors (*perc*) genügt nicht für die Annahme, dass beispielsweise eine Person etwas gesehen und dies auch erkannt hat. Dafür muss für einen Perzeptor P zusätzlich die Annahme gemacht werden, dass P über das reine Wahrnehmen hinaus das Perzept erkennen kann. Damit wird aus dem perzeptiven Agenten ein kognitiver Agent (vgl. [Devlin 1993, S. 34]). In der filmischen Repräsentation wird dann aus dem intradiegetischen Perzeptor ein Agent mit der Fähigkeit des Erkennens.

Wenn ein Perzeptor P etwas sieht (*see*), unterstellen wir als Beobachter also einen visuellen Inhalt:

$$s_{vis} \models \sigma$$

für wenigstens ein Infon σ . Dabei wird keines der Infone, das von s_{vis} gestützt wird und zum visuellen Inhalt gehört, vom Beobachter als notwendig mit dem Intentionalitätsnetzwerk von P verknüpft angenommen (vgl. [Devlin 1993, S. 190]). Betrachten wir die folgende Beschreibung:

„Emma sah Sebastian Coe das olympische 1500-Meter-Finale gewinnen.“

Es wird eine Aussage darüber gemacht, dass Emma eine gewisse visuelle Erfahrung gemacht hat. Allerdings stellt diese Aussage keine Behauptung über das Wissen von Emma hinsichtlich der gesehenen Ereignisse auf. Es kann also sein, dass Emma lediglich etwas wahrgenommen hat, ohne eine Individuation und Konzeptualisierung von Aspekten der Szene durchgeführt zu haben.

(*see_that*:) Wenn ein Perzeptor P etwas sieht und Infone aus dem visuellen Inhalt mit seinem nichtleeren Intentionalitätsnetzwerk verknüpft, sieht er, dass etwas der Fall ist (*see_that*). In Abwandlung des obigen Beispiels betrachten wir die folgende Aussage:

„Emma hat gesehen, dass Sebastian Coe das olympische 1500-Meter-Finale gewonnen hat.“

Neben der visuellen Erfahrung hat Emma Klarheit darüber, welche Gegenstände und Ereignisse sie gerade „kognitiv gesehen“ hat.

Insgesamt kommen also zunächst drei verschiedene Relationen zur Analyse filmischer Syntagmen in Betracht:

1. Die allgemeine Wahrnehmung, für die wir die kanonisch zweistellige Relation *perc* verwenden.
2. Das epistemologisch neutrale Sehen, für das wir die kanonisch zweistellige Relation *see* verwenden. Für diese Relation unterstellen wir, dass Infone aus dem visuellen Inhalt nicht mit dem Intentionalitätsnetzwerk des Perzeptors P verknüpft werden.
3. Das epistemologisch positive Sehen, für das wir die kanonisch zweistellige Relation *see_that* nutzen. Für diese Relation unterstellen wir, das Infone aus dem visuellen Inhalt mit dem Intentionalitätsnetzwerk verknüpft werden.

Wie bereits erläutert, kann einem intradiegetischen Perzeptor unterstellt werden, dass dieser ständig visuelle Erfahrungen hat. Da jedoch dem Beobachter diese visuellen Erfahrungen nicht ständig zugänglich gemacht werden, ergeben sich für die filmischen Realisierungsmöglichkeiten von Perception keinerlei Anhaltspunkte für die Aggregation von Segmenten. Wenn nicht gezeigt wird, dass jemand sieht, ergibt sich auch kein Segment, das zu einem Syntagma gehören könnte. Die Relation *perc* wird daher für syntagmatische Fragestellungen irrelevant.

Kommen wir nun zum Unterschied zwischen *see* und *see_that*, und damit zur Verknüpfung von Infonen mit dem Intentionalitätsnetzwerks. Die alleinige filmische Darstellung eines Perzeptes reicht nicht aus, um eine solche Verknüpfung zu rechtfertigen. So kann es beispielsweise sein, dass dem Beobachter ein Filmbild eines Perzeptes zugänglich gemacht wird, der Perzeptor das Gesehene aber nicht versteht. Wird jedoch eine innere Stimme des Perzeptors dazu genutzt, das Gesehene zu interpretieren, kann eine Verknüpfung mit dem Intentionalitätsnetzwerk angenommen werden. Da sich die Fragestellungen dieser Arbeit aber ausschließlich auf Bildsignale des Films beziehen, kann eine Unterscheidung von *see* und *see_that* in diesem Zusammenhang nicht erfolgen. Für die weiteren Betrachtungen steht der Sachverhalt des Sehens daher für das epistemologisch neutrale Sehen mit der Relation *see*.

7.3 Filmische Serialisierung perzeptiver Relationen

Die Verbindung zwischen einem intradiegetischen Perzeptor und seinem Perzept im Film ist in der klassischen Filmtheorie untersucht worden. Der sogenannte *point-of-view-shot* (*POV-Shot*) wurde (wie in Abschnitt 6.5 dargestellt) erstmals von Edward Branigan in [Branigan 1984, S. 103] eingeführt. In einer neueren Arbeit zur Modifikation der Theorie von Branigan beschreibt Petra Grimm den POV-Shot wie folgt: „Ein Blick ist zu sehen, dann das, worauf er/sie blickt“ [Grimm 1996, S. 52]. Die Struktur eines POV-Shots in einem Segment S mit den Teilsegmenten S' und S'' weist dabei gemäß [Grimm 1996] drei Elemente auf:

1. Segment S': Festlegung eines Orientierungspunktes im Raum durch die Darstellung eines Perzeptors;
2. Übergang durch Schnitt oder Blende;
3. Segment S'': Perzept des zuvor durch Segment S' festgelegten Blickes des Perzeptors.

Es kann also zunächst festgehalten werden: In einem Syntagma, das ein Segment S klassifiziert, in dem eine perzeptive Fokalisierung vorkommt, muss in einem Teilsegment S' ein intradiegetischer Perzeptor von einer Beobachtermenge B erkannt werden können. Weiterhin muss gelten, dass in einem solchen fokalisierenden Segment ein Teilsegment S'' als Perzept von P vorhanden ist und aus mindestens einer Einstellung besteht. Sowohl das Segment S' als auch das Segment S'' können grundsätzlich auch aus mehr als einer Einstellung bestehen.

Eine Fokalisierung durch einen intradiegetischen Perzeptor weist auf Einstellungsebene also folgende logische Grundstruktur auf (s. Abbildung 24):

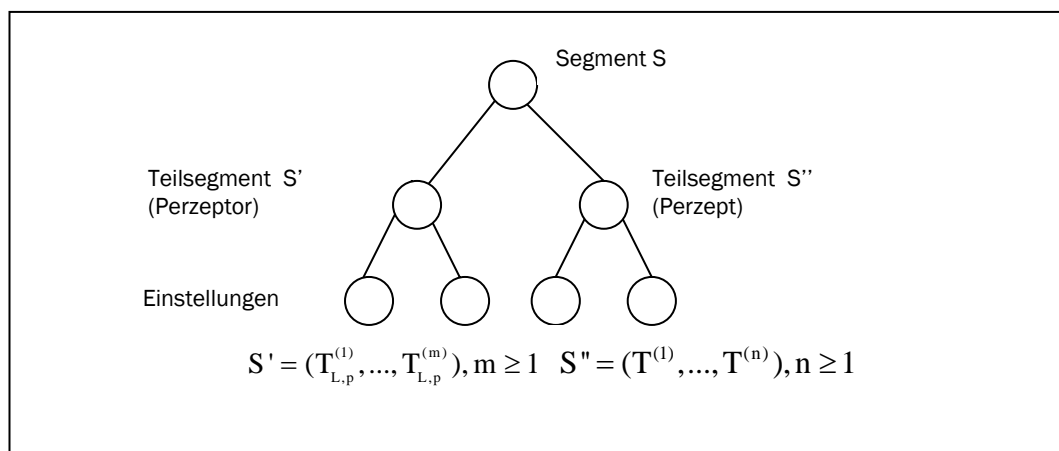


Abbildung 24: Segmentierung von Perception.

Grundsätzlich ist jede Abfolge von Perzeptorsegment S' und Perzeptsegment S'' die serialisierte Repräsentation einer visuellen Erfahrung. Für die Analyse dramaturgischer Varianten auf Einstellungsebene werden zunächst die grundsätzlichen Bedingungen behandelt, denen das Perzeptorsegment S' und das Perzeptsegment S'' unterliegt.

Zunächst wird S' behandelt: Der Perzeptor P wird syntaktisch – wie bereits in Abschnitt 7.1 beschrieben – durch eine Moving-Region repräsentiert. Da die große Syntagmatik nur auf ganzen Einstellungen operiert, wird für die Klassifikation o.B.d.A. die in einem Layout L „letzte“ ganze Einstellung, in der P enthalten ist, zu Grunde gelegt. Wir bezeichnen diese Einstellung in S' nachfolgend mit $T_{L,P}$. Wenigstens eine Einstellung $T_{L,P}$ konstituiert das Perzeptorsegment S' und kommt damit für die Klassifikation in einem perzeptiven Syntagma in Frage. S' besteht minimal aus dieser einen Einstellung.

Nun stellt sich die syntagmatische Frage, ob und unter welchen Umständen S' mehrere Einstellungen enthalten darf.

Dazu drei Beispiele:

Beispiel 1: Eine Person blickt in einer Einstellung aus dem Fenster. Eine zweite Einstellung zeigt das Perzept dieser Person. Eine dritte Einstellung zeigt den Raum, in der sich die Person befindet.

Ohne weitere Information würde ein menschlicher Beobachter hier eine einzige visuelle Erfahrung konzeptionalisieren. Ein syntagmatisch klassifizierender Beobachter würde genau einmal ein perzeptives Syntagma verwenden.

Beispiel 2: Eine Person blickt in einer Einstellung aus dem Fenster. Eine zweite Einstellung zeigt das Perzept der Person. Eine dritte Einstellung zeigt, wie sich die Person vom Fenster wegdreht. Eine vierte Einstellung zeigt, dass die Person wieder zum Fenster läuft und nach draußen blickt. Eine fünfte Einstellung zeigt wieder das Perzept der Person.

Hier wird ein Beobachter im Allgemeinen zwei verschiedene visuelle Erfahrungen mit (zwei) Perzeptionsvorgängen annehmen. Dadurch, dass gezeigt wird, dass die Person sich vom Fenster entfernt, ist für einen Beobachter im Allgemeinen der erste Perzeptionsvorgang abgeschlossen. Es ist zu erwarten, dass in einem solchem Segment zwei Instanzierungen eines perzeptiven Syntagmas bei gleichem Perzeptor realisiert sind.

Beispiel 3: Eine Person blickt in einer Einstellung aus dem Fenster. Eine zweite Einstellung zeigt das Perzept der Person. Eine dritte Einstellung zeigt die immer noch dem Fenster blickende Person. Eine vierte Einstellung zeigt wieder das Perzept der Person.

Hier kann der Beobachter mindestens einen, aber auch zwei Perzeptionsvorgänge annehmen: Das „Sehen“ aus der ersten Einstellung wird in der Dritten ggf. weiter fortgesetzt. Wird dem Beobachter nicht das Gegenteil plausibel gemacht, kann defaultmäßig von B angenommen werden kann, dass es sich um ein und dieselbe visuelle Erfahrung handelt, ergibt sich also eine Instanziierung eines perzeptiven Syntagmas mit der ersten und dritten Einstellung in Segment S' .

Wir können also festhalten: Die Anzahl der Instanziierungen eines perzeptiven Syntagmas in einem Segment hat als untere Schranke die Anzahl der vom Beobachter zugeschriebenen visuellen Erfahrungen eines Perzeptors.

Für eine einzelne visuelle Erfahrung gilt für das Segment $S' = (T_{L,p}^{(1)}, \dots, T_{L,p}^{(m)})$ folgende Anforderung: Die diegetische Zeit jeder Einstellung in S' darf nicht mit der diegetischen Zeit einer anderen Einstellung in S' überlappen. Diese Einschränkung gewährleistet den zeitlichen Fortschritt der angenommenen einen visuellen Erfahrung auch dann, wenn S' und S'' ineinander geschnitten werden. Bei einem mobilen Perzeptor muss darüber hinaus in S' passend Zeit vergehen, damit es P auch möglich ist, an eine bestimmte Position für eine Einstellung in S'' zu kommen.

Im Unterschied zu den zeitlichen Anforderungen sind die räumlichen Anforderungen an S' komplexer. Zunächst könnte man aufgrund der Struktur der visuellen Erfahrung (siehe Abschnitt 7.2) versucht sein, einen Zusammenhang des diegetischen Raumes aller Einstellungen in S' anzunehmen. Das ist aber nicht der Fall: Eine (alternierend geschnittene) Folge von Blicken eines Perzeptors (und zugehörigen Perzepten in S'') ist dann nicht räumlich zusammenhängend, wenn der Perzeptor sich in seinem Raum bewegt oder die Aufnahmen des Perzeptors so sind, dass ein räumlicher Zusammenhang in S' nicht konzeptionalisiert werden kann.

Nun folgt die Analyse von S'' . Ein Perzeptsegment S'' muss auf Einstellungsebene minimal aus einer Einstellung bestehen, wie in Beispiel 1: Einstellung 2 ist dort die einzige Einstellung in S'' . Weiter kann aber S'' aus einer Folge von Einstellungen bestehen, die beliebige zeitliche und räumliche Lücken aufweisen – allerdings unter Wahrung einer entscheidenden Randbedingung: S'' muss zu S' „passen“ in dem Sinne, dass der Beobachter alle Einstellungen in S' und alle Einstellungen in S'' als Repräsentation einer einzelnen Perzeptionssituation s des Perzeptors P konzeptionalisieren kann.

Was heißt das raumzeitlich? Sei angenommen, dass eine zeitlich vollständige filmische Repräsentation von einer einzelnen Perzeptionssituation s angelegt werden soll. Dann muss die layoutierte Abfolge des s repräsentierenden Segmentes S eine zeitlich zusammenhängende Folge an Einstellungen ergeben. Wenn also die Einstellungen in S' zeitlich eine diegetische Lücke aufweisen, muss gelten, dass diese Lücke durch eine Einstellung in S'' geschlossen werden kann, wie in Abbildung 25 skizziert:

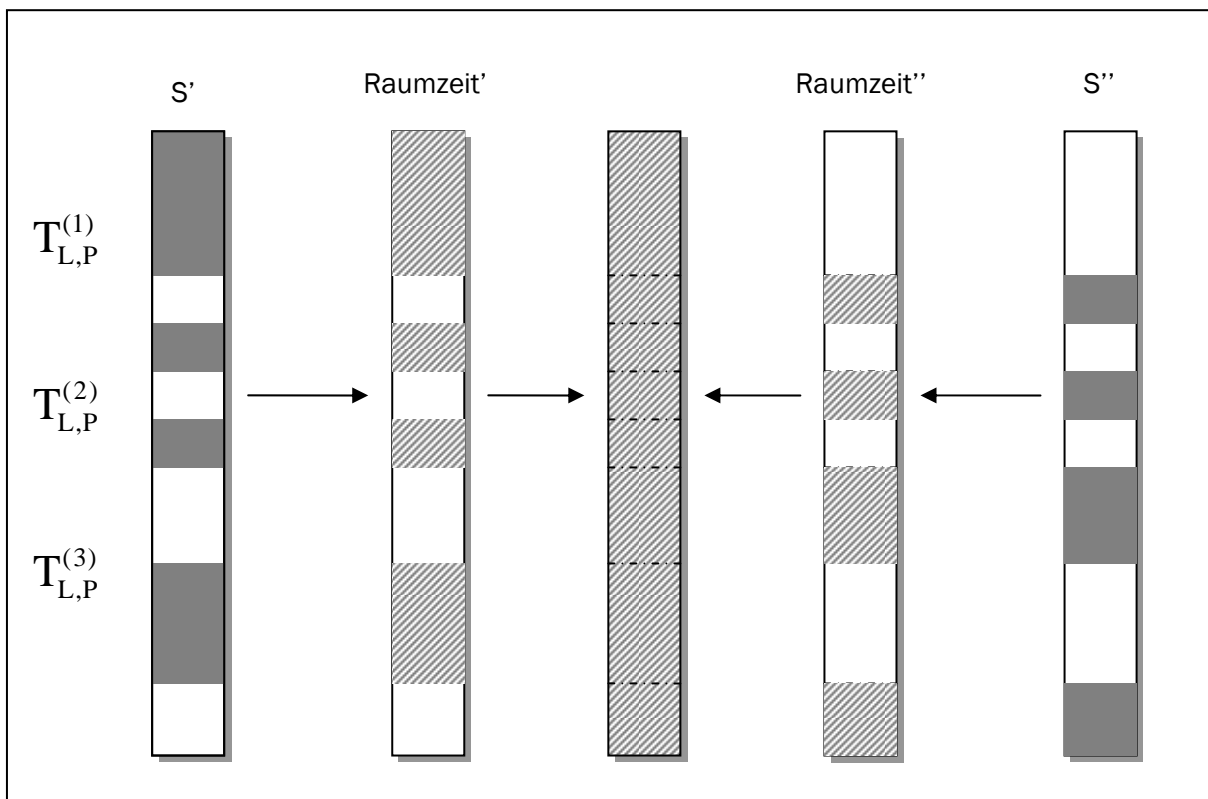


Abbildung 25: Zeitlich vollständige Repräsentation der Perzeptionssituation s durch die Teilsegmente S' und S'' .

Grundsätzlich muss die Repräsentation von s jedoch nicht wie in Abbildung 25 gezeigt zeitlich vollständig sein: Die layoutierte Abfolge eines s repräsentierenden Segmentes S kann eine Folge von Einstellungen mit zeitlicher Lücke beinhalten. Dabei kommt die Struktur der Perzeption ins Spiel. Gemäß der situationssemantischen Analyse wird von einem Beobachter B ständig unterstellt, dass ein intradiegetischer Perzeptor P räumlich und zeitlich bei seinen filmisch repräsentierten Perzepten ist (siehe Abschnitt 7.1). Ist B imstande, unter den gegebenen Voraussetzungen der P unterstellten räumlichen Nähe und zeitlichen Kopräsenz die zeitliche Lücke des gezeigten

Filmbildes zu interpolieren, kann auch in diesem Fall das Segment S als eine visuelle Erfahrung zugehörig klassifiziert werden.

Grundsätzlich müssen die aufgenommenen Einstellungen in S'' mit den diegetischen (insbesondere zeitlichen und räumlichen) Möglichkeiten des Perzeptors P kompatibel sein³⁵ (s. Abschnitt 7.2). So müssen im Rahmen der von B angenommenen räumlichen Nähe und zeitlichen Kopräsenz des Perzeptors die Kamerapositionen aller Einstellungen in S'' jeweils mit den von B angenommenen Positionen des Perzeptors P in S' verträglich sein.

7.4 Perzeptiv fokalisierbare Strukturen

Mit den bisherigen Ergebnissen kann nun festgelegt werden, welche Segmente überhaupt einen Perzeptionsvorgang serialisieren und syntagmatisch entsprechend klassifiziert werden können.

Grundsätzlich kann syntagmatisch bei Annahme eines intradiegetischen Perzeptors P zwischen zwei Fällen unterschieden werden: Es gibt kein oder es gibt ein Segment, das ein Beobachter als Repräsentation des Perzeptes des intradiegetischen Perzeptors ansehen kann.

Ein nahe liegendes Beispiel für den ersten Fall ist eine Darstellung von Anna, wie sie durch ein Schlüsselloch sieht und danach verstört ist. B kann dann nur sagen, dass Anna eine (verstörende) visuelle Erfahrung gemacht hat; er kann sich dabei jedoch nicht auf etwas Dargestelltes beziehen und wird in einer situationssemantischen Modellierung initial als äußeren Inhalt der visuellen Erfahrung v_A von Anna nur $w \models (s_{vis} \triangleright v_A)$ annehmen können.

Es verbleibt der zweite Fall: Es gibt ein nichtleeres Segment als Repräsentation des Perzeptes s_{vis} von P. Dieses Segment kann syntaktisch in mehrere Segmente ohne Zusammenhang zerfallen; diegetisch muss es aber für eine einzelne visuelle Erfahrung s_{vis} vom Beobachter als räumlich und zeitlich zusammenhängend konzeptionalisiert werden können, unabhängig von den syntaktischen Strukturen des zugrunde liegenden Dokuments. Es ergibt sich:

³⁵ Eine Formalisierung dieser perzeptuellen Kompatibilität findet sich in [Davis 1990]. Diese wird hier nicht durchgeführt.

Ein Einstellungssegment S in einem cinematographischen Dokument D heißt *perzeptiv fokalisiert für* eine Beobachtermenge B , wenn

1. S mindestens zwei disjunkte Teilsegmente S' und S'' enthält;
2. jeder Beobachter $B \in B$ in S' wenigstens eine nichtleere Moving-Region als intradiegetischen Perzeptor P so konzeptualisieren kann, dass diesem eine einzelne visuelle Erfahrung v_P zugeordnet werden kann;
3. das Raumzeiturbild von S'' für jeden Beobachter $B \in B$ als eine visuelle Szene s_{vis} einer einzelnen visuellen Erfahrung v für P gelten kann.

Die erste Bedingung schließt, wie in Abschnitt 7.3 begründet, nicht aus, dass S' und S'' in cinematographischen Dokumenten alternierend ineinander geschnitten sind, solange die Einheit der visuellen Erfahrung v nicht gefährdet wird. Zusätzlich ist zu bemerken, dass Einstellungen aus S'' und S' einen räumlichen Zusammenhang bilden können. Perzipiert P etwas im räumlichen Nahbereich und kann ein Beobachter B erkennen, dass ein räumlicher Zusammenhang besteht, dann ist auf Einstellungsebene auch eine Doppelklassifikation von S als perzeptiv fokalisiertes Segment und zusätzlich z. B. als szenisches Segment möglich.

Die zweite Bedingung verlangt, dass im Dokument tatsächlich eine Moving-Region vorhanden ist, die als Repräsentation eines Perzeptors angesehen wird und auch „an“ ist, also beim Beispiel eines Menschen dieser nicht schläft; oder im Falle eines technischen Überwachungssystems, dass es auch tatsächlich überwacht. Die Definition schließt, wie in Abschnitt 7.3 begründet, ferner nicht aus, dass der Perzeptor syntaktisch nicht zusammenhängend ist, solange es sich nur um einen intradiegetischen Perzeptor handelt.

Die dritte Bedingung ordnet P die Möglichkeit von visuellen Erfahrungen in der Weise zu, dass s_{vis} mit P perzeptuell kompatibel ist. Dies berücksichtigt die Beispiele in Abschnitt 7.3, die Annahmen zum perzeptuellen Vermögen von P machen. Diese Bedingung verlangt ferner nicht, dass das Teilsegment S'' maximal gewählt ist: Die (filmische) Repräsentation eines Perzeptionsvorgangs erweist sich für den Perzeptanteil (insbesondere auf Einstellungsebene) als grundsätzlich teilbar. Das mag zunächst etwas überraschen, da sich in linguistischen Analysen solche Verben wie „sehen“ als nicht „kumulativ“ (im Gegensatz zu Verben wie „essen“ oder „lesen“) erweisen. Die Teilbarkeit ist hier aber auch nur eine des filmisch repräsentierten Perzeptanteils. Die Repräsentation der Perzeptionsvorganges ist nicht in gleicher Weise teilbar.

Die obige Definition deckt (im MPEG-7 Sinne) beliebige Segmente eines cinematographischen Dokumentes ab, sagt also auch beim Perzeptanteil nichts über die Syntax des fokalisierten Segmentes aus. Es stellt sich nun die Frage, ob es auch syntaktisch für beliebige Segmente möglich ist, zwischen interner und externer Fokalisierung in einem perzeptiv fokalisiertem Segment zu unterscheiden.

Dazu ist zunächst festzuhalten, dass diese Unterscheidung für die Modellierung der Beobachter notwendig ist: Wenn eine nichtleere Pixelmenge für einen Beobachter als Repräsentation des Perzeptes eines intradiegetischen Perzeptors gilt, bildet er sowohl für sich als auch für den Perzeptor einen äußeren Inhalt mit wenigstens einem σ , so dass $w \models [(s_{vis} \models \sigma) \wedge (s_{vis} \triangleright v)]$ gilt. Nur eine hinreichend große Überschneidung der visuellen Inhalte $s_{vis} \models \sigma_P$ und $s_{vis} \models \sigma_B$ kann überhaupt feststellbar machen, dass intern fokalisiert wird. Schon das wirft die Frage auf, ob grundsätzlich für beliebige Segmenttypen zwischen interner und externer Fokalisierung unterschieden werden kann.

Ferner gilt: Sind der modellierte Beobachter B und der intradiegetische Perzeptor P in einem perzeptiv fokalisiertem Segment an keiner Stelle vertauschbar in dem Sinne, dass für B eine perzeptive (Teil-)Erfahrung v_P von P auch als (Teil-)Erfahrung v_B von B gelten kann, dann liegt sicher keine interne Fokalisierung vor. Der Beobachter sieht (im visuellen Falle) dann nie durch die Augen des Betrachters. Ist dies dagegen für einen Teil des fokalisiertem Segmentes gegeben, können abhängig vom Beobachter und der Syntax des Segmentes unterschiedliche Teile über den vertauschbaren Anteil hinaus intern fokalisiert werden oder nicht: Der Beobachter macht sich gegebenenfalls Teile des eigentlich nicht vertauschbaren „Restes“ zu eigen. Genau diese Unbestimmtheit ist es, die die Branigan'schen Mühen (vgl. Abschnitt 6.5) bei der Auftrennung der Fokalisierungsebenen ausmacht.

Diese Unbestimmtheit wird sich für alle in MPEG-7 behandelten Dokumententypen nicht grundsätzlich klären lassen. Auf der Einstellungsebene von cinematographischen Dokumenten kann man sich aber die Arbeit von Grimm [Grimm 1986] zunutze machen, die eine solche Trennung im Rahmen einer großen Syntagmatik erlaubt.

Branigan beschreibt in seinem Modell der Achsenverhältnisse keine unterschiedliche filmtechnische Umsetzung der internen und externen Fokalisierung und nimmt damit auch nicht deren Abgrenzung vor (vgl. [Grimm 1996, S. 54]). So beschreibt er lediglich den klassischen POV-Shot mit den Augen der Figur als filmische Umsetzung der Ebene der internen Fokalisierung und den POV-Shot über die Schulter als filmische Umsetzung der Ebene der externen Fokalisierung (vgl. [Grimm 1996, S. 54]).

Gemäß Grimm kann zunächst davon ausgegangen werden, dass die Zuordnung zur externen Fokalisierung erst dann gegeben ist, wenn der Fokalisierer zumindest teilweise zu sehen ist. Zudem ist nicht anzunehmen, dass zur internen Fokalisierung ausschließlich der POV-Shot mit den Augen zählt. Selbst wenn die Kameraposition nicht exakt mit dem Standpunkt des Perzeptors übereinstimmt, wird gemäß Grimm dennoch eine „Figurenperspektive ähnlich der mit den Augen“, also eine interne Fokalisierung suggeriert (vgl. [Grimm 1996, S. 55]). Insgesamt unterscheidet Grimm die Zuordnung nach interner und externer Fokalisierung nach den folgenden Einstellungskombinationen, wie Abbildung 26 zeigt:

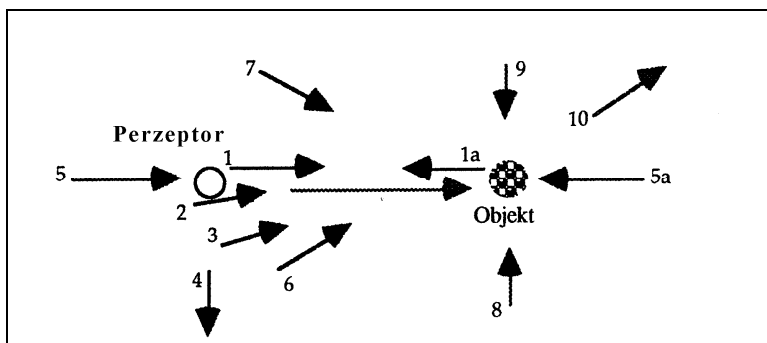


Abbildung 26: Modell der Achsenverhältnisse (angelehnt an [Grimm 1996, S. 56]): Das Objekt kann ein beliebiger Gegenstand sein (auch ein perzipierender Agent). Der lange Pfeil in der Mitte, der das Perzeptor mit dem Objekt verbindet, stellt die Blickrichtung des Perzeptors dar³⁶.

K.1: Klassischer POV-Shot mit den Augen

Die Kameraposition suggeriert dem Beobachter genau die Position des Perzeptors.

K.1a: Klassischer POV-Shot aus der Perspektive des Objekts

Diese Konstellation entspricht K.1, jedoch aus Objektperspektive³⁷.

K.2: Direkt neben dem Perzeptor

Die Kamera befindet sich direkt neben dem Perzeptor. Der Perzeptor selbst ist dabei nicht im Bild zu sehen.

K.3: Ungefähr 30 Grad entfernt vom Perzeptor

Die Kamera befindet sich nun etwas weiter vom Perzeptor entfernt. Der Perzeptor selbst ist dabei nicht im Bild zu sehen.

³⁶ Branigan [Branigan 1992, S. 118] zeichnet in seinem Modell der Achsenverhältnisse anstelle des Perzeptors ein Auge.

³⁷ z.B. dann, wenn das Objekt auch eine Person ist.

K.4: Abweichender POV-Shot

Im Bild ist ein Objekt, auf das der Perzeptor nicht blickt.

K.5: Über eine Schulter

Die Position der Kamera hinter dem Perzeptor bewirkt, dass der Perzeptor noch teilweise im Filmbild zu sehen ist.

K.5a: Wie K.5, jedoch aus der Perspektive des Objekts.

Ist das Objekt ein perzipierender Agent, funktioniert diese Einstellung zugleich als erste Einstellung einer neuen POV-Struktur.

K.6: Diese Kameraposition zeigt das, was der Perzeptor sieht, jedoch nicht aus der Position des Perzeptors. Damit kann also bereits aufgrund der abweichenden Position der Kamera nicht vollständig auf das Perzept des Perzeptors geschlossen werden.

K.7: Wie K.6, nur gespiegelt. Die Kamera zeigt wieder das, was der Perzeptor sieht, jedoch nicht aus der Position des Perzeptors. Der Unterschied zu K.6 besteht darin, dass dem Beobachter suggeriert werden soll, dass, wenn das Objekt eine Person ist, beide (Perzeptor und Objekt) in entgegengesetzte Richtungen blicken oder sich zumindest nicht ansehen.

K.8, 9: Destabilisierende POV-Shots

Im Bezug auf den Perzeptor wird ein falscher Raum impliziert.

K.10: Extrem destabilisierender POV-Shot

Fungiert oft als Übergang zu einer neuen Szene.

Bezugnehmend auf die Perzepte der Personen ergibt sich nun eine mögliche Zuordnung der Kamerakonstellationen zur internen und externen Fokalisierung. Dabei ist anzumerken, dass es sich bei einigen der gezeigten Einstellungen nicht um Perzepte des Perzeptors handeln kann. Die Einstellungen zu den Konstellationen K.4 und K.10 weichen von den perzeptiven Möglichkeiten (Blickrichtung des Perzeptors) ab und können daher nicht als Perzept eines Perzeptors gelten³⁸. Gezeigt wird in diesem Fall ein anderer Raum oder ein anderes Objekt als in Abbildung 26 dargestellt.

Bei den Einstellungen zu den Konstellationen K.6, K.7, K.8. und K.9 wird dem Beobachter zwar das korrekte Objekt gezeigt, die Blickrichtung des Perzeptors stimmt jedoch nicht mit der Kamerakonstellatation überein, sodass nicht auf die Perception eines

³⁸ An dieser Stelle stellt sich die Frage, ob die Grimm'schen Kamerakonstellationen wirklich einen POV-Shot in dem hier definierten Sinne darstellen. Die Antwort darauf soll jedoch anderen Arbeiten aus der Filmnarratologie vorbehalten sein.

Perzeptors geschlossen werden kann. K.1a und K.5a sind bis auf das Bezugsobjekt identisch mit K.1 und K.5 und werden daher nicht gesondert kategorisiert.

Gemäß [Grimm 1996] werden nun die Kamerakonstellationen der internen und externen Fokalisierung wie folgt zugeordnet:

- Zur internen Fokalisierung³⁹ zählen die Konstellationen K.1, K.2, K.3, sofern für den Beobachter das vereinigte Raumzeiturbild aller Einstellungen in S'' als Perzept des intradiegetischen Perzeptor P gelten kann. Der Beobachter sieht S'' und nimmt an, dass der diegetische Perzeptor das Urbild von S'' sieht.
- Zur externen Fokalisierung zählt auf jeden Fall die Konstellation K.5. Der Beobachter sieht S'' und nimmt an, dass der diegetische Perzeptor einen Teil des Urbild von S'' sieht.

Mit diesen Ergebnissen kann nun definiert werden:

Ein Einstellungssegment S in einem cinematographischen Dokument D heißt für eine Beobachtermenge B *perzeptiv intern fokalisiert*, wenn

1. es als Segment perzeptiv fokalisiert ist,
2. für die diegetische Raumposition zwischen dem Perzeptor P in S' und dem Perzept S'' die Achsenverhältnisse K.1, K.2 oder K.3 angenommen werden können.

Ein Einstellungssegment S in einem cinematographischen Dokument D heißt für eine Beobachtermenge B *perzeptiv extern fokalisiert*, wenn

1. es als Segment perzeptiv fokalisiert ist,
2. für die diegetische Raumposition zwischen dem Perzeptor P in S' und dem Perzept S'' die Achsenverhältnisse K.5 angenommen werden können.

Beide Bedingungen sehen für eine Große Syntagmatik ausführlich ausgeschrieben wie folgt aus:

Ein Einstellungssegment S in einem cinematographischen Dokument D heißt für eine Beobachtermenge B *perzeptiv intern fokalisiert*, wenn

³⁹ Grimm verwendet an dieser Stelle den Begriff „Fokussierung“ anstelle von „Fokalisierung“. Es sei angemerkt, dass sich beides auf die Bedeutung des Begriffs „focalization“ im Sinne von Branigan bezieht.

1. es mindestens zwei Einstellungen enthält, von denen eine in einem Teilsegment $S' = (T_1', \dots, T_m')$, $m \geq 1$ und eine weitere in einem Teilsegment $S'' = (T_1'', \dots, T_n'')$, $n \geq 1$ liegt,
2. ein Beobachter $B \in B$ in S' einen intradiegetischen Perzeptor P konzeptualisieren kann,
3. für B das vereinigte Raumzeiturbild aller Einstellungen in S'' als Perzept des intradiegetischen Perzeptor P gelten kann,
4. für die diegetische Raumposition zwischen dem Perzeptor P in S' und dem Perzept S'' die Achsenverhältnisse K.1, K.2, oder K.3 angenommen werden können.

Ein Einstellungssegment S in einem cinematographischen Dokument D heißt für eine Beobachtermenge B *perzeptiv extern fokalisiert*, wenn

1. es mindestens zwei Einstellungen enthält, von denen eine in einem Teilsegment $S' = (T_1', \dots, T_m')$, $m \geq 1$ und eine weitere in einem Teilsegment $S'' = (T_1'', \dots, T_n'')$, $n \geq 1$ liegt,
2. ein Beobachter $B \in B$ in S' einen intradiegetischen Perzeptor P konzeptualisieren kann,
3. ein echter nichtleerer Teil des vereinigten Raumzeiturbild aller Einstellungen in S'' als Perzept des intradiegetischen Perzeptor P gelten kann.
4. für die diegetische Raumposition zwischen dem Perzeptor P in S' und dem Perzept S'' die Achsenverhältnisse K.5 angenommen werden können.

Damit können nun abschließend die beiden perzeptiven Syntagmen definiert werden:

Ein Teilbaum der logischen Struktur in einem cinematographischen Dokument D ist für eine Beobachtermenge B eine *perzeptiv intern fokalisierbare Struktur*, wenn es einen Layoutprozess L gibt, der ein zugehöriges perzeptiv intern fokalisiertes Segment S erzeugt.

Ein Teilbaum der logischen Struktur in einem cinematographischen Dokument D ist für eine Beobachtermenge B eine *perzeptiv extern fokalisierbare Struktur*, wenn es einen Layoutprozess L gibt, der ein zugehöriges perzeptiv extern fokalisiertes Segment S erzeugt.

Perzeptiv extern fokalisierbare Strukturen klassifizieren eine filmisch präsentierte Perzeptionssituation auf andere Weise als perzeptiv intern fokalisierbare Strukturen. Der Beobachter sieht auch in S'' einen Teil des Perzeptes und einen Teil des Perzeptors und dies wiederum typischerweise innerhalb einer Einstellung. Für eine große Syntagmatik ist die Verknüpfung zwischen S' und S'' also nicht eine zwischen Perzeptor und Perzept, sondern eine zwischen einem Agenten in einer Situation, die in S'' als Perzeptionssituation weiter denotiert wird (oft durch Granularitätswechsel in der räumlichen Auflösung mit einem größeren Schwerpunkt auf dem Perzept). Syntagmatisch bleibt man aber in einer szenischen Repräsentation, die man unterhalb der Einstellungsebene mit den Labeln „Perzeptor“ und „Perzept“ weiter segmentieren kann.

Insgesamt ergibt sich: Nur intern fokalisierbare Strukturen erlauben es, über die szenische Bindung von Einstellungen hinaus eine Menge von Einstellungen als perzeptiv, aber nicht als räumlich zusammenhängend zu klassifizieren.

TEIL III – Realisierung der Ergebnisse mit MPEG-7

8 Abbildung von Syntax und Semantik mit MPEG-7

Nach den Untersuchungen aus Kapitel 7 können nun die entwickelten perzeptiven Syntagmen mit Hilfe von MPEG-7 in ein Metadatenschema für Perzeptionsrepräsentationen überführt werden.

Die syntaktische Beschreibung von Segmenten auf und unterhalb der Einstellungsebene wurde in Abschnitt 5.2 vorgestellt. Um jedoch Perzeptionssituationen umfassend abzubilden zu können, müssen neben diesen Segmentbeschreibungen auch semantische Eigenschaften, wie sie in Abschnitt 7.4 erarbeitet wurden, in die Metadaten-sprache mit einfließen. Dies geschieht mit den Werkzeugen der Semantic Description Tools des MDS, die eine Reihe an vordefinierten Beschreibungselementen bereithalten.

Im Abschnitt 8.1 wird ein Metadatenschema für Perzeption erarbeitet. Abschnitt 8.2 enthält eine vollständige Auflistung aller auf der Layoutebene möglichen Fallbeispiele für konkrete Perzeptionssituationen unter Berücksichtigung der zeitlichen Relationen im Layout und in der Diegese.

8.1 Beschreibung von Perzeptionssituationen in MPEG-7

Mit den in Kapitel 5 eingeführten Werkzeugen zur Strukturierung von Filmsegmenten mit MPEG-7 lässt sich ein filmisches Dokument in einzelne Segmente zerlegen. Darüber hinaus lassen sich auch Relationen dieser Segmente modellieren.

In Kapitel 7 wurden verschiedene Bedingungen erarbeitet, die für Perzeptionsdarstellungen im Film erfüllt sein müssen⁴⁰:

1. Es existieren mindestens zwei Segmente auf Einstellungsebene S' und S''.
2. Es existiert ein intradiegetischer Perzeptor P.
3. Für die diegetische Raumposition zwischen P und S'' können die Achsenverhältnisse K.1, K.2, oder K.3 angenommen werden.
4. Das räumliche Verhältnis zwischen Perzeptor und Perzept innerhalb der Diegese muss benachbart sein und dabei den perzeptiven Möglichkeiten des Perzeptors entsprechen. Damit kann das räumliche Verhältnis sowohl mit der „overlaps“-Relation, als auch mit der „disjoint“-Relation modelliert werden. Im

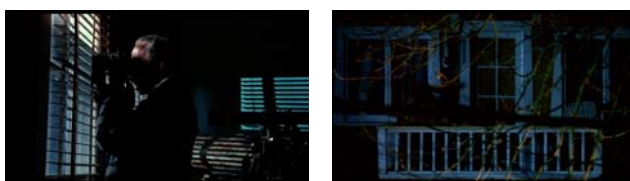
⁴⁰ Diese Bedingungen gelten für den Fall der internen Fokalisierung, da nur diese einen Mehrwert für die Bildung eines Syntagmas im Sinne einer großen Syntagmatik darstellt (siehe Abschnitt 7.4).

Falle der „disjoint“-Relation läge dann eine Bindung von Einstellungen vor, die als perceptiv und gleichzeitig als nicht als räumlich zusammenhängend zu klassifizieren ist.

5. Das zeitliche Verhältnis zwischen Perzeptor und Perzept innerhalb der Diegese muss die Bedingungen der Kopräsenz erfüllen.

Bei genauerer Betrachtung dieser fünf Bedingungen fällt auf, dass sie grundsätzlich in zwei Kategorien eingeteilt werden können. Bedingungen 1 und 2 betreffen Relationen und Eigenschaften auf der Layoutebene, es handelt sich also um strukturelle Bedingungen. Bedingungen 3, 4 und 5 hingegen beschreiben Eigenschaften, die sich auf die Diegese bzw. die diegetische Anordnung von Segmenten beziehen. Mit den in Kapitel 5 eingeführten Möglichkeiten, Filmsegmente in MPEG-7 syntaktisch zu beschreiben, lassen sich jedoch nur die Bedingungen 1 und 2 abbilden.

Im Folgenden wird anhand eines Beispiels die strukturelle Modellierung einer Perzeptionssituation unter Berücksichtigung der Bedingungen 1 und 2 vorgenommen. Im Anschluss daran erfolgt die Vorstellung der semantischen Beschreibungswerkzeuge von MPEG-7, mit denen sich die Bedingungen 3, 4 und 5 formulieren lassen. Exemplarisch soll eine Sequenz des Films „Stakeout“ zur Klassifikation herangezogen werden. In diesem Film beginnt der Polizist Bill mit einem Beschattungsauftrag und versteckt sich in einem Haus. Er will eine Frau beobachten, die sich in dem gegenüberliegenden Haus aufhalten soll. Ein Beispiel für Perzeption liefern die Einstellungen E20 und E21. Bill sieht mit einem Fernglas aus dem Fenster (E20). In E21 zeigt die Kamera das gegenüberliegende Haus aus Bills Perspektive.



E20

E21

Abbildung 27: Perzeptionseinstellungen (E20, E21) aus dem Spielfilm „Stakeout“.

Vom Beobachter kann in E20 ein intradiegetischer Perzeptor konzeptionalisiert werden. Die Kameraposition in E21 lässt auf die Blickrichtung des Perzeptors in E20 schließen, was der Kameraposition K.1 (aus 7.4) entspricht. Der Zuschauer hat den Eindruck, genau das zu sehen, was der Perzeptor sieht. E20 ist somit Bestandteil des Perzeptorsegmentes S_x und E21 Bestandteil des Perzeptorsegmentes S_y . Zudem kann

Bill in Sx als intradiegetischer Perzeptor erkannt und damit als ein Moving-Region-Segment Smr ausgezeichnet werden⁴¹.

Gemäß den Strukturbeschreibungen des MDS können die Einstellungen E20 und E21 zusammen als Gesamtvideosegment S modelliert werden. Die Spieldauer des Segments S beschrieben wird für dieses Beispiel mit 1:11 Minuten angegeben.

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video id="S">
        <CreationInformation>
          <Creation>
            <Title>Segment einer Perzeptionssituation.</Title>
          </Creation>
        </CreationInformation>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT1M11S</MediaDuration>
        </MediaTime>
        ...
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

Im Anschluss daran wird das Gesamtsegment S durch eine temporale Zerlegung in zwei Videosegmente geteilt. Daraus ergibt sich zum einen das Segment Sx, das den Perzeptor mit einer Laufzeit von 55 Sek. beschreibt, zum anderen das Segment Sy zur Beschreibung des Perzeptes mit einer Laufzeit von 16 Sekunden. Diese Zerlegung erfüllt damit Bedingung 1.

```
<TemporalDecomposition>
  <VideoSegment id="Sx">
    <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Planszene"/>
    <CreationInformation>
      <Creation>
        <Title>Ein Mann ist zu sehen.</Title>
      </Creation>
    </CreationInformation>
    <MediaTime>
      <MediaTimePoint>T00:00:00</MediaTimePoint>
      <MediaDuration>PT0M55S</MediaDuration>
    </MediaTime>
    <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets" target="#Sy"/>
    ...
  </VideoSegment>
  <VideoSegment id="Sy">
    <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Planszene"/>
    <CreationInformation>
```

⁴¹ Aufgrund der Einschränkung der Zeichenverwendung bei der Wertebelegung von ID-Attributen in XML-Schema, kann die Notation S' und S'', wie sie in Kapitel 7 eingeführt wurde, nicht direkt in den XML-Code übertragen werden. Damit die Instanzen valide bleiben, wird bei den Code-Beispielen die Notation Sx für das Perzeptorsegment S', Sy für das Perzeptsegment S'' und Smr für die Moving Region des Perzeptors verwendet.

```

    <Creation>
      <Title>Ein Haus ist zu sehen.</Title>
    </Creation>
  </CreationInformation>
  <MediaTime>
    <MediaTimePoint>T00:00:56</MediaTimePoint>
    <MediaDuration>PT1M11S</MediaDuration>
  </MediaTime>
</VideoSegment>
</TemporalDecomposition>

```

Wie bereits in Abschnitt 5.2.1 erläutert, kann die interne Struktur eines Segmentes mit dem Element `StructuralUnit` durch ein kontrolliertes Vokabular genauer beschrieben werden. Dies geschieht in diesem Beispiel durch die Angabe des jeweiligen Syntagmas der gebildeten Videosegmente. Für die Segmente wird angegeben, dass es sich bei S_x und S_y jeweils um eine Planszene handelt. Da S_x und S_y auf der Lay-outebene direkt aufeinander folgend modelliert sind, wird in der Definition von S_x die zeitliche Relation „meets“ modelliert.

Um Perzeption überhaupt möglich zu machen, bedarf es weiterhin der Auszeichnung eines Perzeptors, der als Moving-Region innerhalb von S_x beschrieben werden muss. Durch die raumzeitliche Zerlegung des Segments S_x entsteht das Perzeptorsegment S_{mr} unterhalb der Einstellungsebene. Diese Zerlegung erfüllt damit Bedingung 2.

```

<SpatioTemporalDecomposition>
  <MovingRegion id="Smr">
    <CreationInformation>
      <Creation>
        <Title>Ein Mann als MR.</Title>
      </Creation>
    </CreationInformation>
    <SpatioTemporalLocator>
      <FigureTrajectory type="polygon">
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT0M55S</MediaDuration>
        </MediaTime>
        <Vertex>
          <!-- Beschreibung eines Polygons -->
        </Vertex>
      </FigureTrajectory>
    </SpatioTemporalLocator>
  </MovingRegion>
</SpatioTemporalDecomposition>

```

Wie zuvor erwähnt, beziehen sich die Bedingungen 3, 4 und 5 nicht auf die strukturellen Gegebenheiten von Segmenten, sondern auf deren Semantik. Zunächst werden die für die Beschreibung von Perzeption nötigen semantischen Elemente des MDS vorgestellt und die Bedingungen 3, 4 und 5 für das Beispiel im Anschluss daran modelliert.

Semantische Beschreibungen in MPEG-7 haben als Ausgangspunkt den `SemanticDescriptionType` der sich als Instanz wie folgt darstellt:

```

<Description xsi:type="SemanticDescriptionType">
  <Semantics>
    <Label>
      <Name>Bill beobachtet.</Name>
    </Label>
    <Definition>
      <FreeTextAnnotation>Bill beobachtet mit einem Fernglas ein gegenüberliegendes
      Haus.</FreeTextAnnotation>
    </Definition>
  </Semantics>
</Description>

```

Neben dem `ContentEntityType` zur Beschreibung der Struktur und damit der Syntax von Dokumenten, wird der Datentyp `SemanticDescriptionType` zur Beschreibung der Semantik von Multimediadokumenten verwendet.

```

<complexType name="SemanticDescriptionType">
  <complexContent>
    <extension base="mpeg7:ContentAbstractionType">
      <choice>
        <element name="Semantics" type="mpeg7:SemanticType" minOccurs="1"
        maxOccurs="unbounded"/>
        <element name="ConceptCollection" type="mpeg7:ConceptCollectionType"
        minOccurs="1" maxOccurs="unbounded"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

<complexType name="SemanticType">
  <complexContent>
    <extension base="mpeg7:SemanticBagType"/>
  </complexContent>
</complexType>

```

Als Container einer semantischen Beschreibung ist im MDS der Datentyp des `SemanticType` vorgesehen, der für das Element `Semantics` den Typ angibt und jeweils für eine Beschreibung der narrativen Welt instanziiert wird. Wahlweise kann auch mindestens eine Sammlung von semantischen Konzepten gemäß dem Datentyp `ConceptCollectionType` modelliert werden. Darüber hinaus ließen sich durch den `GraphType` Relationen zwischen semantischen Beschreibungen oder strukturellen Beschreibungen modellieren.

```

<complexType name="SemanticBagType" abstract="true">
  <complexContent>
    <extension base="mpeg7:SemanticBaseType">
      <sequence>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="SemanticBase" type="mpeg7:SemanticBaseType"/>
          <element name="SemanticBaseRef" type="mpeg7:ReferenceType"/>
        </choice>
        <element name="Graph" type="mpeg7:GraphType" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

Der Datentyp `SemanticType` ist selbst eine konkrete Ableitung des abstrakten Datentyps `SemanticBagType`, der, wie der Name bereits ausdrückt, als Container für

semantische Beschreibungen dient. Diese verschiedenen Beschreibungen werden durch den abstrakten Datentyp `SemanticBaseType` dargestellt, von dem sich dann die konkreten Beschreibungswerkzeuge für Semantiken ableiten.

```
<complexType name="SemanticBaseType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="AbstractionLevel" type="mpeg7:AbstractionLevelType" minOccurs="0"/>
        <element name="Label" type="mpeg7:TermUseType" minOccurs="1"
maxOccurs="unbounded"/>
        <element name="Definition" type="mpeg7:TextAnnotationType" minOccurs="0"/>
        <element name="Property" type="mpeg7:TermUseType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="MediaOccurrence" minOccurs="0" maxOccurs="unbounded">
          ...
        </element>
        <element name="Relation" type="mpeg7:RelationType" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Verpflichtend für jede semantische Beschreibung ist die Angabe mindestens eines Namens durch das Element `Label`. Optional können durch die Angabe des Elementes `Definition` eine Freitextbeschreibung oder durch das Element `Property` bestimmte Eigenschaften einer semantischen Beschreibung hinzugefügt werden. Durch die Angabe des Elementes `AbstractionLevel` lässt sich der Grad der Abstraktion der semantischen Beschreibung annotieren. Das Element `MediaOccurrence` kann durch die definierten Kindelemente Bezug auf den Content eines Multimediadokumentes und damit auf dessen Strukturbeschreibungen nehmen. Im Weiteren verwenden wir jedoch für die Abbildung der Semantik auf die Struktur den Datentyp `RelationType`, der optional innerhalb des `SemanticBaseType` mit dem Element `Relation` verwendet werden kann.

Die für die Beschreibung von Perception in dieser Arbeit wichtigen Ableitungen des `SemanticBase DS` sind im Einzelnen:

- Event DS: Darstellung von Ereignissen, in die konkrete Objekte involviert sind.
- SemanticPlace DS: Ort oder Raum in der filmischen Welt.
- SemanticTime DS: Zeitraum oder Zeitpunkt in der filmischen Welt.
- Object DS: Beschreibung tatsächlich vorhandener Objekte.
- AgentObject DS: Erweiterung des Object DS um handlungsfähige Objekte, um z.B. Personen zu beschreiben.

Beginnen wir nun zunächst mit der Erweiterung des eingangs vorgestellten Beispiels um einige semantische Beschreibungen:

```

<Description xsi:type="SemanticDescriptionType">
  <Semantics>
    <Label>
      <Name>Bill beobachtet etwas.</Name>
    </Label>
    <Definition>
      <FreeTextAnnotation>Bill beobachtet mit einem Fernglas ein gegenüberliegendes
      Haus.</FreeTextAnnotation>
    </Definition>
    <SemanticBase xsi:type="EventType" id="EV_perceiving">
      <Label>
        <Name>Bill sitzt mit einem Fernglas vor einem Fenster.</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
      target="#l_perceiving"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
      target="#t_perceiving"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
      target="#Sx"/>
    </SemanticBase>
  </Semantics>
</Description>

```

Ein wichtiger Bestandteil einer semantischen Beschreibung sind Events. Ein Event ist ein Ereignis von zeitlichem und räumlichem Ausmaß und kann sich im Allgemeinen auf mehrere Objekte beziehen. In MPEG-7 werden Ereignisse über das Event DS dargestellt. Beinhaltet ein Ereignis auch Objekte, wird eine Relation zwischen den Objekten und Events hergestellt. Dementsprechend wurde im obigen Beispiel für die semantische Beschreibung des Perzeptorsegmentes Sx, in dem Bill etwas beobachtet, das Event EV_perceiving modelliert.

```

<complexType name="EventType">
  <complexContent>
    <extension base="mpeg7:SemanticBaseType">
      <sequence>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Event" type="mpeg7:EventType"/>
          <element name="EventRef" type="mpeg7:ReferenceType"/>
        </choice>
        <element name="SemanticPlace" type="mpeg7:SemanticPlaceType"
        minOccurs="0" maxOccurs="unbounded"/>
        <element name="SemanticTime" type="mpeg7:SemanticTimeType" minOccurs="0"
        maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

Der Datentyp EventType leitet sich vom SemanticBaseType ab und definiert optional Beschreibungen für Zeit und Ort des zu beschreibenden Events. Alternativ lassen sich die Beschreibungen von Ort und Zeit auch über die Relationen location und time unter Verwendung des RelationTypes modellieren, was eine separate Beschreibung für Ort und Zeit nötig macht.

```

<SemanticBase xsi:type="SemanticPlaceType" id="l_perceiving">
  <Label>
    <Name>Wohnhaus, Main Street 12, 2. Stockwerk</Name>
  </Label>
</SemanticBase>
<SemanticBase xsi:type="SemanticTimeType" id="t_perceiving">
  <Label>
    <Name>21.12.1997, 22:30:00 - 22:30:55</Name>
  </Label>
</SemanticBase>

```

Die dritte Relation des Beispielsevents `EV_perceiving` bildet die semantische Beschreibung des Events auf das Perzeptorsegment `Sx` der Strukturbeschreibung ab. Dies geschieht mit der semantischen Relation `depictedBy`. Analog zum Perzeptorevent lässt sich nun das Perzeptsegment `Sy` durch das Event `EV_percept` beschreiben:

```

<SemanticBase xsi:type="EventType" id="EV_percept">
  <Label>
    <Name>Ein Haus von außen. Die Räume sind nicht beleuchtet.</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
    target="#l_percept"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time" target="#t_percept"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy" target="#Sy"/>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="SemanticPlaceType" id="l_percept">
  <Label>
    <Name>Wohnhaus, Main Street 14</Name>
  </Label>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="SemanticTimeType" id="t_percept">
  <Label>
    <Name>21.12.1997, 22:30:56 - 22:31:11 </Name>
  </Label>
</SemanticBase>

```

Ein Event ist ein Ereignis, das im Allgemeinen verschiedene Objekte betreffen kann, die selbst mit dem `ObjectType` beschrieben werden können. Der `ObjectType` beschreibt alle Objekte, die als Individuum oder in abstrakten Klassen in Zeit und Raum existieren können⁴².

⁴² Die Syntax des `Object DS` und des `Event DS` sind sich sehr ähnlich. Ein `Object DS` beschreibt ein semantisches Objekt in Raum und Zeit für den Zweck, das Objekt mit einer Relation auf ein Segment abzubilden. Ähnliches gilt für das `Event DS`. Die Besonderheit beider DS ist die Möglichkeit der rekursiven Definition. So ist es also möglich, innerhalb eines `Event DS` verschiedene Subevents zu bilden. Ein Event- oder Objektbaum ist daher leicht zu erstellen.

```

<complexType name="ObjectType">
  <complexContent>
    <extension base="mpeg7:SemanticBaseType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element name="Object" type="mpeg7:ObjectType"/>
        <element name="ObjectRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

Abgeleitet vom `ObjectType`, und mit gleichartigem strukturellem Aufbau, existiert der `AgentObjectType`, mit dem personifizierbare Subjekte in der Diegese beschrieben werden können. Zur semantischen Beschreibung der Moving Region `Smr` verwenden wir einen solchen Agenten.

```

<SemanticBase xsi:type="AgentObjectType" id="AO_perceptor">
  <Label>
    <Name>Bill</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy" target="#Smr"/>
  <Relation type="urn:mpeg7_extension:perception:cameraAxisRelationK1"
    target="#EV_percept"/>
  <Agent xsi:type="PersonType">
    <Name>
      <GivenName>Bill</GivenName>
    </Name>
  </Agent>
</SemanticBase>

```

Neben der Abbildung der semantischen Beschreibung des Perzeptors auf die Struktur, kann nun eine weitere Bedingung formuliert werden, die bei Perzeptionssituationen gelten muss. Die Blickrichtung von Bill im Beispiel entspricht der Kameraposition aus E21. Die Kameraposition des POV-Shots kann daher mit K1 klassifiziert werden, was mit der Relation `cameraAxisRelationK1` zwischen `AO_perceptor` und `EV_percept` modelliert wird. Diese Relation beschreibt damit die Bedingung 3.

Abbildung 28 zeigt im Zusammenhang das Verhältnis der verschiedenen semantischen Deskriptoren zueinander.

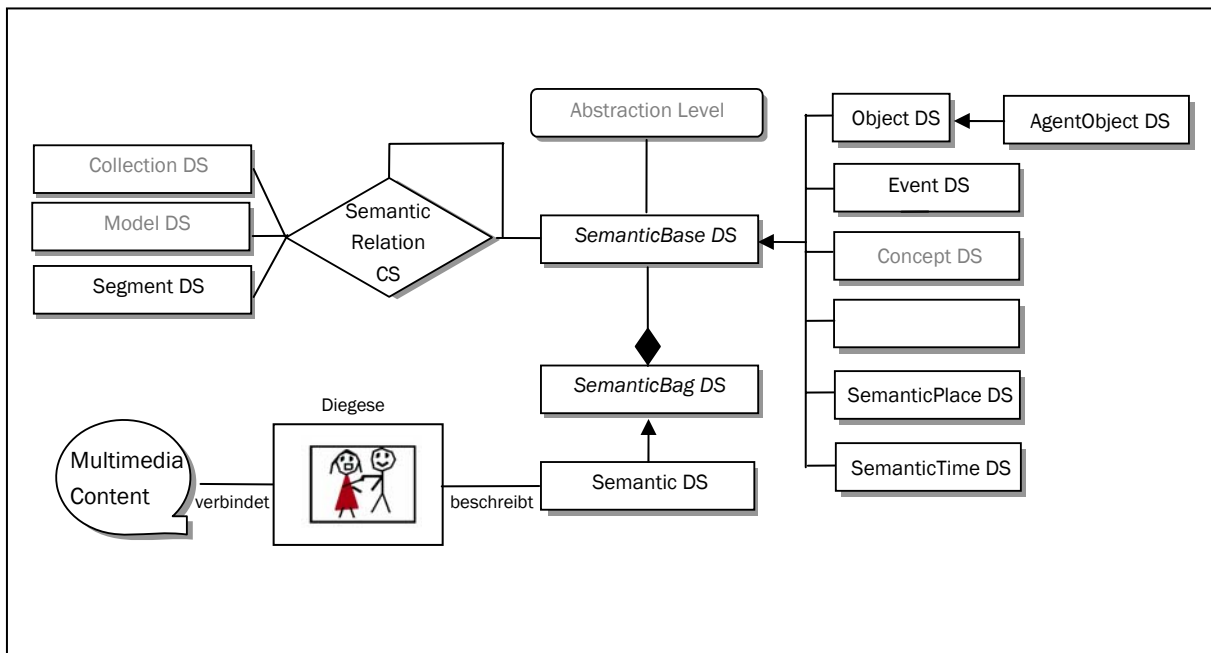


Abbildung 28: Zusammenhang zwischen den semantischen Deskriptoren⁴³.
(Abbildung angelehnt an [ISO/IEC 15938-5, dort Abb. 54])

Die Beschreibung der Relationen durch den `RelationType` nimmt dabei eine Sonderrolle ein. Zum einen dient er dazu, verschiedene semantische Deskriptoren in Relation zueinander zu setzen. Zum anderen schlägt er die Brücke zwischen Semantik und Struktur, um semantische Beschreibungen überhaupt erst auf Segmente abbilden zu können. Dies geschieht, wie in der Beispielinstantz gesehen, mit der Relation `depictedBy`.

Bisher haben wir lediglich die Teilnehmer modelliert, die am eigentlichen Ereignis „Perzeption“ beteiligt sind: den Perzeptor als Agent `AO_perceptor` und das Pezept als Event `EV_percept`. Damit diese nun zusammen das Ereignis „Perzeption“ bilden, muss ein neues Event `EV_perception` modelliert werden.

```
<SemanticBase xsi:type="EventType" id="EV_perception">
  <Label>
    <Name>Bill sieht das Haus.</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
  target="#AO_perceptor"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:stimulusOf"
  target="#EV_percept"/>
</SemanticBase>
```

⁴³ Der graue Text kennzeichnet Beschreibungen, die für diese Arbeit von untergeordneter Bedeutung sind. Darüber hinaus kennzeichnet der kursive Text (analog zur UML) abstrakte Deskriptoren.

Der Agent wird dem Event `EV_perception` mit der Relation `agent` hinzugefügt. Die zweite Beziehung, die bei Perceptionssituationen ständig aktiv ist, ist die Verursachungsrelation ($s_{vis} \triangleright v_A$), die einen Teil des äußeren Inhalts bildet. Die visuelle Szene erzeugt die visuelle Erfahrung (siehe Abschnitt 7.3). Im vorgestellten Beispiel erzeugt die visuelle Szene `EV_percept` die visuelle Erfahrung `EV_perception`. Der zweite Teilnehmer am Perceptionereignis wird daher mit der Relation `stimulusOf` hinzugefügt.

Kommen wir nun zurück auf die eingangs erwähnten semantischen Bedingungen zur Formulierung von Perception: Das räumliche Verhältnis zwischen Perzeptor und Perzept innerhalb der Diegese muss benachbart sein (Bedingung 4). Das zeitliche Verhältnis zwischen Perzeptor und Perzept innerhalb der Diegese muss den Bedingungen der Kopräsenz entsprechen (Bedingung 5).

Da sich im vorliegenden Beispiel `Sx` und `Sy` in der diegetischen Zeit nicht überlappen und auch keine Lücke bilden, kann in der Definition der `SemanticTime` des Perzeptors `t_perceiving` eine zusätzliche zeitliche Relation eingefügt werden, die dieses Verhältnis ausdrückt.

```
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets" target="#t_percept"/>
```

Gleiches gilt für die Beschreibung der `SemanticTime` des Perzeptes `t_percept`, jedoch mit der Umkehrrelation `metBy`:

```
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metBy" target="#t_perceiving"/>
```

Neben den zeitlichen Relationen stehen auch die beiden semantischen Räume `l_perceiving` und `l_percept` in folgendem Verhältnis zueinander:

```
<Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:disjoint" target="#l_perceiving"/>
```

```
<Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:disjoint" target="#l_percept"/>
```

Da sich der Raum `l_perceiving` nicht mit dem Raum des Perzeptes `l_percept` schneidet, wird für die Modellierung beider Räume die Relation `disjoint` verwendet.

Damit sind nun für das Beispiel von Bills Beobachtung alle Bedingungen, die für Perception erfüllt sein müssen, vorgestellt. Ganz allgemein müssen für die Modellierung von Perception folgende modellierungstechnischen Voraussetzungen gelten:

1. Es existieren mindestens zwei Videosegmente `Sx` und `Sy`.
2. Es existiert eine `MovingRegion` `Smr`, die Teil von `Sx` ist und den Perzeptor auszeichnet.

3. Für die Segmente S_x und S_y existiert eine semantische Beschreibung als Event $EV_perceiving$ und $EV_percept$, jeweils versehen mit der Beschreibung der diegetischen Zeit t und des diegetischen Raumes l .
4. Für das Segment S_{mr} existiert eine semantische Beschreibung als Agent-Object $AO_perceptor$.
5. Es existiert eine Relation zwischen $AO_perceptor$ und $EV_percept$, welche die Achsenverhältnisse K.1, K.2, oder K.3 angibt.
6. Das räumliche Verhältnis zwischen $l_perceiving$ und $l_percept$ muss mit den räumlichen Relationen `overlaps` oder `disjoint` angegeben werden.
7. Das zeitliche Verhältnis zwischen $t_perceiving$ und $t_percept$ muss mit einer zeitlichen Relation angegeben werden (siehe Abschnitt 8.2).
8. Das Event $EV_perception$ wird durch die Relation `agentOf` mit dem Perzeptor $AO_perceptor$ und durch die Relation `stimulusOf` mit dem Perzept $EV_percept$ verbunden.

Neben diesen Modellierungsbedingungen gibt es für den Fall der Perzeption verschiedene Möglichkeiten der dramaturgischen Umsetzung. Vor allem die zeitliche Anordnung der Segmente auf der Layoutebene lässt montagetechnisch viele verschiedene Darstellungsformen zu.

Möller-Naß hat bereits 1979 für den Fall der Perzeption die Differenz zwischen der syntagmatischen Analyse und den syntaktischen Anordnungsmöglichkeiten im Film bearbeitet und die Existenz einer endlichen Anzahl solcher Kombinationen beschrieben: Perzeptive Syntagmen „[...] sind Zuordnungssyntagmen, d.h. Kombinationen von Einheiten, die einen vollständigen und nicht reduzierbaren, aber sequenziell noch ungeordneten syntaktisch-semantischen Zusammenhang bilden, der durch verschiedene, aber nicht beliebige syntaktische Anordnungsformen repräsentiert werden kann“ [Möller 1979, S. 88]. Eine Auflistung konkreter Anordnungsformen für den Fall von Perzeption wird nun vorgestellt.

8.2 Syntaktische Anordnungsformen für Perzeption

Die Zeitlogik von Allen (siehe Abschnitt 5.5) beschreibt zahlreiche Möglichkeiten, zwei Segmente in Relation zueinander zu setzen. Für die grundlegenden Überlegungen zur Einteilung dieser Relationen müssen die strukturellen Eigenschaften der Montage und zusätzlich die zeitlichen Relationen der Diegese betrachtet werden.

Bei Relationen von Segmenten auf der Layoutebene sind grundsätzlich zwei Fälle zu unterscheiden: Im ersten Fall sind die Segmente auf Montageebene direkt zusam-

menhängend montiert. Es handelt sich dabei um die zeitliche Relation „meets“ (also ohne Lücke und ohne Überschneidung) auf der Layoutebene. Im zweiten Fall sind die beiden Segmente nicht direkt aufeinander folgend montiert, sondern enthalten eine Lücke im Layout. Dies entspricht den zeitlichen Relation „before“ und „after“ auf der Layoutebene. Da die Reihenfolgen der Einstellungen bereits durch die diegetischen Zeitrelationen behandelt werden, kann im Folgenden für eine zeitliche Lücke auf der Layoutebene ganz allgemein die MPEG-7-Relation „gap“ verwendet werden.

Nach der Unterscheidung nach Layoutgesichtspunkten werden dann – gesondert für jeden der beiden Fälle – Beispiele von Dramaturgien mit den Allen’schen Zeitrelationen innerhalb der Diegese aufgezeigt. Die in Abschnitt 5.5 behandelten Zeitrelationen werden für diese Betrachtungsweise paarweise zusammengezogen, da ein zeitliches Vertauschen der Segmente bei Beibehaltung der zeitlichen Start- und Endbedingungen zu keiner anderen Montage führen würde.

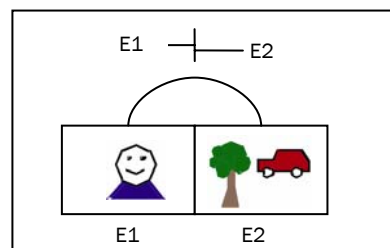
8.2.1 Layoutrelation „meets“

Sind zwei Segmente S' und S'' im Layout benachbart (also durch die Layoutrelation „meets“ verbunden) so bedeutet dies für die Layoutsicht des Dokuments eine unmittelbare Hintereinanderschaltung dieser beiden Segmente. Unabhängig davon ergeben sich hinsichtlich der diegetischen Zeit der Filmbilder folgende diegetische Relationen:

8.2.1.1 Diegetische Relation „meets (metBy)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in der ersten Einstellung den Agenten Anton. Die darauf folgende Einstellung zeigt einen Baum und ein vorbei fahrendes rotes Auto. Dem Beobachter wird der Eindruck vermittelt, dass Anton das Auto und den Baum sieht. Dieser Eindruck entsteht dadurch, dass



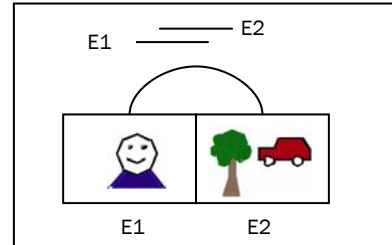
die Perspektive, aus der E2 aufgenommen worden ist, der Perspektive des Standorts von Anton in E1 entspricht. Des Weiteren entspricht die Wahrnehmung Antons perceptiven Möglichkeiten. Diese grundlegende Voraussetzung kann auch für alle folgenden Perzeptionen angenommen werden.

Der Endpunkt der diegetischen Zeit aus Einstellung E1 schließt direkt an den Startpunkt der diegetischen Zeit aus Einstellung E2 an. Es entstehen weder eine Lücke, noch eine Überlappung in der diegetischen Zeit.

8.2.1.2 Diegetische Relation „overlaps (overlappedBy)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in einer ersten Einstellung E1 den Agenten Anton. Er wartet scheinbar auf etwas. Kurz darauf wird dem Zuschauer klar, dass Anton etwas beobachtet. Plötzlich hört er (und somit auch der Beobachter) das Quietschen von Bremsen und einen lauten Knall. Antons Blick ist sehr erschrocken. Eine zweite Einstellung E2



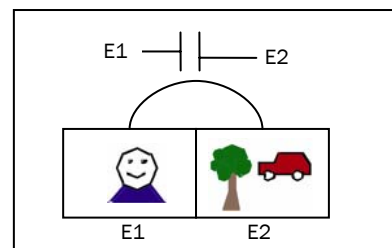
zeigt ein heranfahrendes Fahrzeug, das von der Straße abkommt und gegen einen Baum fährt. Der Fahrer des Fahrzeugs steigt unversehrt aus. In E2 sieht der Zuschauer das, was Anton bis zum Ende der repräsentierten Beobachtung in E1 gesehen hat und dann das, was auch weiterhin von Anton beobachtet, dem Zuschauer gezeigt wird.

Der Startpunkt der diegetischen Zeit aus Einstellung E2 liegt vor dem Endpunkt der diegetischen Zeit aus Einstellung E1. Man erkennt dies daran, dass Anton den Unfall beobachtet, der erst in der darauf folgenden Einstellung zu sehen ist. Es handelt sich deshalb um eine Überlappung der diegetischen Zeitintervalle.

8.2.1.3 Diegetische Relation „before (after)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in einer ersten Einstellung den wartenden Agenten Anton. Eine zweite Einstellung zeigt ein Auto, das gegen einen Baum geprallt ist. Wie und wann der Unfall passiert ist, wird nicht dargestellt.



Zwischen den beiden Einstellungen liegt offensichtlich eine diegetische Lücke. Zwischen den

Einstellungen E1 und E2 passiert etwas, das dem Zuschauer vorenthalten wird. Die dramaturgische Darstellung mit einer zeitlichen Lücke in der Diegese ist vermutlich kein normales Mittel, um eine Perception darzustellen. Vielmehr ist diese Montage dafür geeignet, den Zuschauer zu verwirren. Da hier nur die theoretischen Möglichkeiten untersucht werden, soll die Frage nach der dramaturgischen Sinnhaftigkeit einer solchen Montage nicht diskutiert werden.

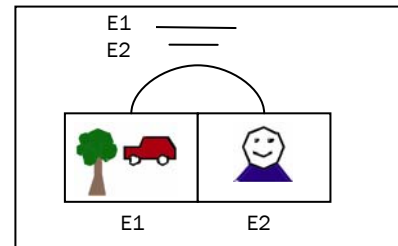
8.2.1.4 Diegetische Relation „during (contains)“

Die Zeitrelation „during“ gliedert sich in die drei Untergruppen „strictDuring“, „start“ und „finishes“, wie die folgenden Relationen zeigen.

8.2.1.5 Diegetische Relation „strictDuring (strictContains)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in der Einstellung E1 ein heranfahrendes Fahrzeug. Plötzlich hupt das Fahrzeug, da ein Reh auf die Straße springt. Das Auto kommt daraufhin von der Straße ab und prallt gegen einen Baum. Der Fahrer des Fahrzeuges steigt aus dem Fahrzeug aus und schreit um Hilfe. Eine zweite Einstellung E2 zeigt den Agenten Anton, während er den Unfall wahrnimmt. Dem Zuschauer wird vermittelt, dass Anton mindestens einen Teil von E1 sieht.

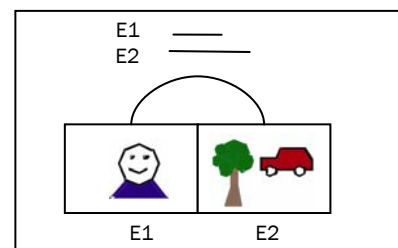


Der Startpunkt der diegetischen Zeit aus Einstellung E2 liegt nach dem Startpunkt der diegetischen Zeit aus Einstellung E1. Man erkennt dies daran, dass in E1 das Auto hupt, in E2 das Hupen jedoch nicht vorkommt. Der Endpunkt der diegetischen Zeit aus Einstellung E1 liegt nach dem Endpunkt der diegetischen Zeit aus Einstellung E2. Der Fahrer des Wagens steigt in der Einstellung E1 aus und ruft um Hilfe. Anton wird in Einstellung E2 aber nur mit einem erschrockenen Blick gezeigt. Die vermutlich darauf folgenden Hilferufe kommen in E2 nicht vor.

8.2.1.6 Diegetische Relation „starts (startedBy)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in einer ersten Einstellung E1 den Agenten Anton, der zunächst ein Hupen hört und daraufhin offensichtlich etwas beobachtet. Plötzlich hört Anton (und somit auch der Zuschauer) das Quietschen von Bremsen und einen lauten Knall. Eine zweite Einstellung E2 zeigt ein hupendes Fahrzeug, das von der Straße abkommt und gegen einen Baum fährt. Der Fahrer des Fahrzeuges steigt aus und ruft um Hilfe.



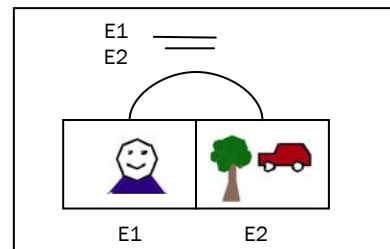
Der Startpunkt der diegetischen Zeit aus Einstellung E2 ist gemäß dem akustischen Signal des Hupens identisch mit dem Startpunkt der diegetischen Zeit aus Einstel-

lung E1. Der Endpunkt der diegetischen Zeit aus Einstellung E2 liegt nach dem Endpunkt der diegetischen Zeit aus Einstellung E1. Dies wird dadurch deutlich, dass Anton in E1 die Hilferufe aus E2 nicht mehr wahrnimmt.

8.2.1.7 Diegetische Relation „finishes (finishedBy)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in einer ersten Einstellung E1 den Agenten Anton, der zunächst wartet. Nach einigen Minuten hört Anton (und damit der Zuschauer) ein Hupen und danach das Quietschen von Bremsen und einen lauten Knall. Eine zweite Einstellung E2 zeigt ein heranfahrendes hupendes Fahrzeug, das von der Straße abkommt und gegen einen Baum fährt.

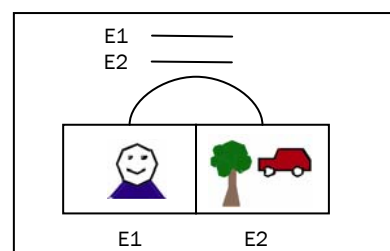


Der Startpunkt der diegetischen Zeit aus Einstellung E2 liegt nach dem Startpunkt der diegetischen Zeit aus Einstellung E1. Dies erkennt man daran, dass Anton zunächst einige Minuten wartet, bevor er das hupende Auto sehen kann, E2 aber direkt mit einem hupenden Auto beginnt. Die beiden Einstellungen enden diegetisch zur gleichen Zeit. Dies wird dadurch deutlich, dass beide Einstellungen mit dem lauten Knall des Aufpralls enden.

8.2.1.8 Diegetische Relation „equal“

Dramaturgie (Drehbuch):

Die Kamera zeigt in einer ersten Einstellung E1 den Agenten Anton, der sofort ein Hupen und danach das Quietschen von Bremsen sowie einen lauten Knall wahrnimmt. Eine zweite Einstellung E2 zeigt ein heranfahrendes hupendes Fahrzeug, das von der Straße abkommt und gegen einen Baum fährt.



Der Startpunkt der diegetischen Zeit aus Einstellung E2 ist gleich dem Startpunkt der diegetischen Zeit aus Einstellung E1. Dies erkennt man daran, dass E1, genauso wie E2, schon mit dem akustischen Signals des Hupens beginnt. Die beiden Einstellungen enden zudem diegetisch zur gleichen Zeit mit dem lauten Knall des Aufpralls.

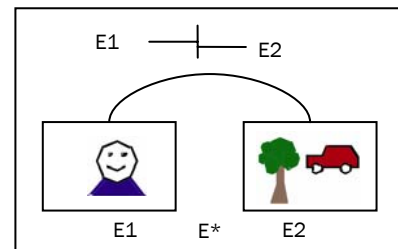
8.2.2 Layoutrelation „gap“

Anders als bei der Relation „meets“ bedeutet „gap“ hier eine layoutierte Lücke zwischen zwei Segmenten. Diese Lücke birgt die Möglichkeit, die diegetischen Relationen dramaturgisch unterschiedlich zu inszenieren.

8.2.2.1 Diegetische Relation „meets (metBy)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in der ersten Einstellung E1 den Agenten Anton. Eine zwischengeschaltete Einstellung E* zeigt dem Bobachter das Bild des wolkenverhangenen Himmels, der baldigen Regen ankündigt. Die darauf folgende Einstellung E2 zeigt einen Baum und ein vorbeifahrendes rotes Auto. Dem Beobachter wird der Eindruck vermittelt, dass Anton das Auto und den Baum sieht.

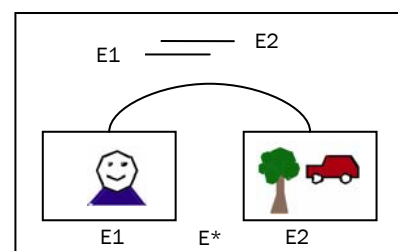


Der Endpunkt der diegetischen Zeit aus Einstellung E1 schließt direkt an den Startpunkt der diegetischen Zeit aus Einstellung E2 an. Es entsteht weder eine Lücke, noch eine Überlappung in der diegetischen Zeit. Die im Layout zwischengeschaltete Einstellung der Regenwolken hat keine Auswirkung auf die zeitlichen Zusammenhänge der Perzeption innerhalb der Diegese.

8.2.2.2 Diegetische Relation „overlaps (overlappedBy)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in einer ersten Einstellung E1 den Agenten Anton. Er wartet scheinbar auf etwas. Kurz darauf wird dem Zuschauer klar, dass Anton etwas beobachtet. Plötzlich hört er ein hupendes Auto und kurz darauf das Quietschen von Bremsen und einen lauten Knall. Eine zwischengeschaltete Einstellung E* zeigt dem Zuschauer das Bild einer Uhr, die sich wie von Wunderhand um zwei Minuten zurückstellt und dem Zuschauer einen Rücksprung in der Zeit signalisiert. Eine weitere Einstellung E2 zeigt ein Fahrzeug das von der Straße abkommt und gegen einen Baum fährt. Der Fahrer steigt aus und ruft um Hilfe.



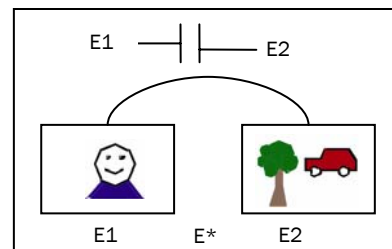
Der Startpunkt der diegetischen Zeit aus Einstellung E2 liegt vor dem Endpunkt der diegetischen Zeit aus Einstellung E1 und zugleich nach dem Startpunkt der diegeti-

schen Zeit aus Einstellung E1. Man erkennt dies daran, dass der Aufprall des Fahrzeuges sowohl in Einstellung E1 als auch nochmals in Einstellung E2 zu hören (und in E2 auch zu sehen) ist. Das Hupen des Fahrzeuges, das in Einstellung E1 zu hören ist, kommt dagegen in Einstellung E2 nicht vor. Es handelt sich deshalb um eine Überlappung der diegetischen Zeitintervalle. Zusätzlich verstärkt wird dieser Eindruck des Überlappendens durch die zwischengeschaltete Einstellung E*. Es handelt sich dabei um eine für dramaturgische Zwecke verwendete Einstellung, die keinen Einfluss auf den Verlauf der diegetischen Zeit hat.

8.2.2.3 Diegetische Relation „before (after)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in einer ersten Einstellung E1 den wartenden Agenten Anton, der in Richtung einer Straße blickt. Eine weitere Einstellung E* zeigt Anna in ihrem Auto, laut Musik hörend. Plötzlich erschrickt Anna und bremst. Die dritte Einstellung E2 zeigt das bremsende Auto aus Antons Perspektive. Das Auto fährt gegen einen Baum. Anna steigt unverseht aus dem Fahrzeug.



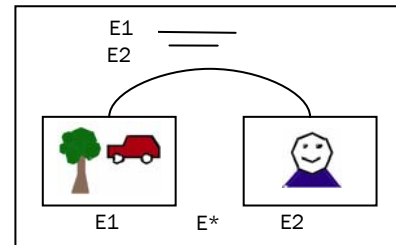
Zwischen den beiden Einstellungen E1 und E2 liegt eine diegetische Lücke. Die Einstellung E* kann nicht mehr zum Perzept von Anton gehören, da die Einstellung nicht aus der Perzeptionsperspektive von Anton gezeigt wird und zudem außerhalb seiner perzeptiven Möglichkeiten liegt. Dennoch läuft die diegetische Zeit in E* weiter. Man erkennt dies an der Tatsache, dass die Zeit in E2 erst kurz vor dem Aufprall beginnt, Anton aber in E2 noch kein Auto gesehen hat.

Die Darstellung einer strukturellen Lücke, kombiniert mit einer zeitlichen Lücke zwischen Perzept und Perzeptor in der Diegese – also einer diegetisch weiterlaufenden Zeit über alle layoutierten Einstellungen – ist ein häufiger Fall der cinematographischen Inszenierung.

8.2.2.4 Diegetische Relation „strictDuring (strictContains)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in einer ersten Einstellung E1 ein heranfahrendes Fahrzeug, in dem eine Person sitzt, die gerade telefoniert. Plötzlich hupt das Fahrzeug. Kurz darauf kommt das Auto von der Straße ab und prallt gegen einen Baum. Der Fahrer des Fahrzeuges steigt aus und ruft um Hilfe. Eine weitere Einstellung E* zeigt den vermeintlichen Telefonpartner des Fahrers aus Einstellung E1. Plötzlich reißt die Telefonverbindung ab. Eine dritte Einstellung E2 zeigt den Agenten Anton, der sichtlich erschrocken ist. Die Perspektive von Einstellung E1 entspricht dem Standort von Anton in Einstellung E2. Man kann deshalb davon ausgehen, dass Anton den Unfallhergang wahrgenommen hat.

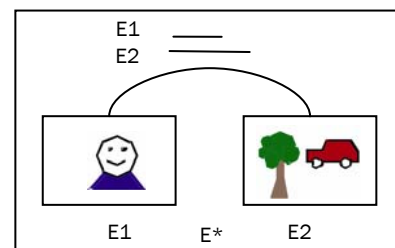


Der Startpunkt der diegetischen Zeit aus Einstellung E2 liegt nach dem Startpunkt der diegetischen Zeit aus Einstellung E1 und zugleich liegt der Endpunkt der diegetischen Zeit aus Einstellung E2 vor dem Endpunkt der diegetischen Zeit aus Einstellung E1. Man erkennt dies daran, dass Anton in Einstellung E2 zwar gezeigt wird, wie er den Unfallhergang wahrnimmt, aber weder das Hupen des Fahrzeugs noch die Hilferufe des Fahrers in Einstellung E2 vorkommen. Die eingeschobene Einstellung des Gesprächspartners des Fahrers hat keinen Einfluss auf die zeitlich diegetische Einbettung von E1 in E2.

8.2.2.5 Diegetische Relation „starts (startedBy)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in einer ersten Einstellung E1 den Agenten Anton, der ein laut hupendes Auto beobachtet. Kurz darauf hört Anton das Quietschen von Bremsen und einen lauten Knall. Eine weitere Einstellung E* zeigt eine Person, die gerade telefoniert. Plötzlich reißt die Telefonverbindung ab. Eine dritte Einstellung E2



zeigt ein hupendes Fahrzeug. Plötzlich kommt das Auto von der Straße ab und prallt gegen einen Baum. Der Fahrer des Fahrzeuges steigt aus und ruft um Hilfe.

Der Startpunkt der diegetischen Zeit aus Einstellung E2 ist identisch mit dem Startpunkt der diegetischen Zeit aus Einstellung E1. Man erkennt dies daran, dass sowohl

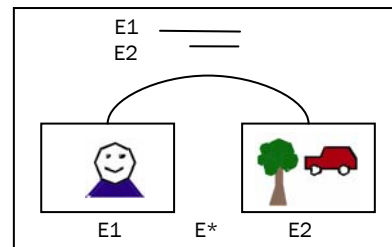
in Einstellung E1 als auch in Einstellung E2 das Hupen des Fahrzeuges zu Beginn hörbar ist. Der Endpunkt der diegetischen Zeit aus Einstellung E2 liegt nach dem Endpunkt der diegetischen Zeit aus Einstellung E1. Dies wird dadurch deutlich, dass der Fahrer des Wagens in Einstellung E2 um Hilfe ruft, dieser Hilferuf jedoch in Einstellung E1 nicht vorkommt

Die eingeschobene Einstellung E* des Telefonpartners des Fahrers hat keinen Einfluss auf die zeitliche diegetische Einbettung von E1 in E2.

8.2.2.6 Diegetische Relation „finishes (finishedBy)“

Dramaturgie (Drehbuch):

Die Kamera zeigt in einer ersten Einstellung E1 den Agenten Anton, der zunächst wartet. Plötzlich hört Anton das Quietschen von Bremsen und einen lauten Knall. Danach hört Anton einen Hilferuf. Eine weitere Einstellung E* zeigt eine Person, die gerade telefoniert. Plötzlich reißt die Telefonverbindung ab.



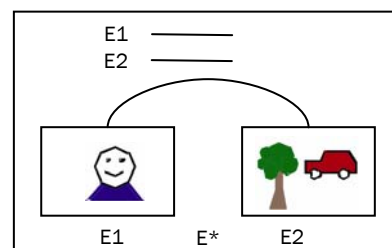
Eine dritte Einstellung E2 zeigt ein laut quietschend bremsendes Fahrzeug, das von der Straße abkommt und gegen einen Baum fährt. Der Fahrer des Fahrzeuges steigt aus dem Fahrzeug und ruft um Hilfe.

Der Startpunkt der diegetischen Zeit aus Einstellung E2 liegt nach dem Startpunkt der diegetischen Zeit aus Einstellung E1. Dies erkennt man daran, dass Anton zunächst wartet, diese Zeit sich aber nicht in der diegetischen Zeit von E2 wiederfindet. Die beiden Einstellungen enden diegetisch zur gleichen Zeit. Dies wird dadurch deutlich, dass sowohl die Einstellung E1 als auch die Einstellung E2 mit dem Hilferuf des Fahrers enden. Die eingeschobene Einstellung des Gesprächspartners des Fahrers hat keinen Einfluss auf die zeitlich diegetische Einbettung von E1 in E2.

8.2.2.7 Diegetische Relation „equal“

Dramaturgie (Drehbuch):

Die Kamera zeigt in einer ersten Einstellung E1 den Agenten Anton, der sofort ein Hupen und danach das Quietschen von Bremsen und einen lauten Knall wahrnimmt. Eine weitere Einstellung E* zeigt eine Person, die gerade telefoniert. Plötzlich reißt die Telefonverbindung ab.



Eine dritte Einstellung E2 zeigt ein heranfahrendes hupendes Fahrzeug, das von der Straße abkommt und mit einem Knall gegen einen Baum fährt.

Der Startpunkt der diegetischen Zeit aus Einstellung E2 ist gleich dem Startpunkt der diegetischen Zeit aus Einstellung E1. Dies erkennt man daran, dass E1, genauso wie E2, mit einem dem akustischen Signals des Hupens beginnt. Die beiden Einstellungen enden zudem diegetisch zur gleichen Zeit: mit dem lauten Knall des Aufpralls. Die eingeschobene Einstellung des Gesprächspartners des Fahrers hat keinen Einfluss auf die zeitliche diegetische Einbettung von E1 in E2.

Mit Erweiterung des MDS durch das in Abschnitt 7.4 vorgestellte und in MPEG-7 formalisierte perzeptive Syntagma, lassen sich nun konkrete Perzeptionssituationen in MPEG-7 modellieren. Dies geschieht im folgenden Kapitel für Beispiele aus zwei Spielfilmen mit vielen Perzeptionssituationen.

9 Beispiele für die dramaturgische Umsetzung von Perzeption mit MPEG-7

Die in diesem Kapitel vorgestellten Beispiele zeigen Perzeptionssituationen im Spielfilm, die im Anschluss an ihre Vorstellung mit dem in Abschnitt 8.1 entwickelten Metadatenschema in MPEG-7 beschrieben werden.

Als empirisches Material für die ersten beiden Beispiele dient der Film „Stakeout“ von John Badham. Die Beispiele drei und vier stammen aus dem Spielfilm „Fahrenheit 451“ von François Truffaut. Beide Filme beinhalten Szenen, bei denen Beobachtung im Vordergrund steht, was ein reichhaltiges Angebot an Perzeptionssituationen liefert.

9.1 Beispiel 1

Ein Beispiel für Perzeption liefert eine Szene in „Stakeout“ in der drei Polizisten den Vorgehensplan für eine Observation diskutieren. Interessant dabei sind die Einstellungen E12 und E13 der Szene (s. Abbildung 29). Ein Polizist blättert in E12 in den Ermittlungsunterlagen. In E13 zeigt die Kamera diese Unterlagen.



E12

E13

Abbildung 29: Perzeptionseinstellungen (E12, E13) aus dem Spielfilm „Stakeout“.

Die Kameraposition in E13 lässt auf die Position des rechten Polizisten in E12 schließen. Es handelt sich hierbei um Kamerakonstellation K.1 (gemäß Abschnitt 7.4). Das Segment, bestehend aus E12 und E13, kann daher als perzeptiv intern fokalisiert klassifiziert werden.

E12 ist Bestandteil des Perzeptorsegmentes S_x und E13 des Perzeptsegmentes S_y . Der rechte Polizist in S_x kann als intradiegetischer Pezeptor erkannt und damit als ein Moving-Region-Segment S_{mr} deklariert werden. Die Blickrichtung entspricht der Kameraposition in E12. Somit kann das Gesamtsegment S als perzeptiv intern fokalisiert klassifiziert werden. E12 und E13 stehen in einem räumlichen und zeitlichen Zusammenhang und sind daher szenisch zusammenhängend.

Diese Perzeptionssituation kann nun mit MPEG-7-Mitteln in Verbindung mit dem Erweiterungsschema wie folgt modelliert werden:

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video id="S">
        <CreationInformation>
          <Creation>
            <Title>Drei Polizisten diskutieren den Vorgehensplan
              für eine Observation.</Title>
          </Creation>
        </CreationInformation>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT12M15S</MediaDuration>
        </MediaTime>
        <TemporalDecomposition>
          <VideoSegment id="Sx">
            <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Szene"/>
            <CreationInformation>
              <Creation>
                <Title>Zwei Polizisten sehen in die Ermittlungsakte.</Title>
              </Creation>
            </CreationInformation>
            <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets"
              target="#Sy"/>
            <MediaTime>
              <MediaTimePoint>T00:06:30</MediaTimePoint>
              <MediaDuration>PT0M10S</MediaDuration>
            </MediaTime>
            <SpatioTemporalDecomposition>
              <MovingRegion id="Smr">
                <CreationInformation>
                  <Creation>
                    <Title>Rechter Polizist als Moving Region</Title>
                  </Creation>
                </CreationInformation>
                <SpatioTemporalLocator>
                  <FigureTrajectory type="polygon">
                    <MediaTime>
                      <MediaTimePoint>T00:06:30</MediaTimePoint>
                      <MediaDuration>PT0M10S</MediaDuration>
                    </MediaTime>
                    <!-- Beschreibung eines Polygons -->
                  </FigureTrajectory>
                </SpatioTemporalLocator>
              </MovingRegion>
            </SpatioTemporalDecomposition>
          </VideoSegment>
          <VideoSegment id="Sy" >
            <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Szene"/>
            <CreationInformation>
              <Creation>
                <Title>Die Kamera zeigt die Ermittlungsakte.</Title>
              </Creation>
            </CreationInformation>
            <MediaTime>
              <MediaTimePoint>T00:06:41</MediaTimePoint>
              <MediaDuration>PT0M12S</MediaDuration>
            </MediaTime>
          </VideoSegment>
        </TemporalDecomposition>
      </Video>
    </MultimediaContent>
  </Description>
```

```

<Description xsi:type="SemanticDescriptionType">
  <Semantics>
    <Label>
      <Name>Drei Polizisten diskutieren den Vorgehensplan für eine Observation.</Name>
    </Label>
    <!-- ***** -->
    <SemanticBase xsi:type="EventType" id="EV_perceiving">
      <Label>
        <Name>Rechter Polizist blickt in die Ermittlungsakte.</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
        target="#Sx"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
        target="#l_perceiving"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
        target="#t_perceiving"/>
    </SemanticBase>
    <!-- ***** -->
    <SemanticBase xsi:type="SemanticPlaceType" id="l_perceiving">
      <Label>
        <Name>Polizeibüro</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:overlaps"
        target="#l_percept"/>
    </SemanticBase>
    <!-- ***** -->
    <SemanticBase xsi:type="SemanticTimeType" id="t_perceiving">
      <Label>
        <Name>21.2.2001</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets"
        target="#t_percept"/>
    </SemanticBase>
    <!-- ***** -->
    <SemanticBase xsi:type="EventType" id="EV_percept">
      <Label>
        <Name>Ermittlungsakte</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
        target="#Sy"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
        target="#l_percept"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
        target="#t_percept"/>
    </SemanticBase>
    <!-- ***** -->
    <SemanticBase xsi:type="SemanticPlaceType" id="l_percept">
      <Label>
        <Name>Polizeibüro</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:overlaps"
        target="#l_perceiving"/>
    </SemanticBase>
    <!-- ***** -->
    <SemanticBase xsi:type="SemanticTimeType" id="t_percept">
      <Label>
        <Name>21.2.2001</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metby"
        target="#t_perceiving"/>
    </SemanticBase>
    <!-- ***** -->
    <SemanticBase xsi:type="AgentObjectType" id="AO_perceptor">
      <Label>
        <Name>Polizist</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
        target="#Smr"/>

```



```

<Relation type="urn:mpeg7_extension:perception:cameraAxisRelationK1"
target="#EV_percept" />
<Agent xsi:type="PersonType">
  <Name>
    <GivenName>Bill</GivenName>
  </Name>
</Agent>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="EventType" id="EV_perception">
  <Label>
    <Name>Polizist sieht in die Ermittlungsakte.</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
target="#AO_perceptor" />
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:stimulusOf"
target="#EV_percept" />
</SemanticBase>
</Semantics>
</Description>
</Mpeg7>

```

Das Perzeptsegment Sx besteht aus der zehn Sekunden langen Einstellung E12, Sy aus der 12 Sekunden langen Einstellung E13. Neben der Kamerakonstellation K1 besteht zudem eine zeitliche Folge dieser beiden Segmente, sowohl in der Diegese als auch in der Layoutsicht. Da Sx und Sy im gleichen Raum spielen, wird für die räumliche Relation beider Segmente der Attributwert `overlaps` angegeben.

9.2 Beispiel 2

Ein komplexeres Beispiel für Perzeption liefern die Einstellungen E38 bis E42 (s. Abbildung 30). Chris beobachtet ein heranfahrendes Fahrzeug.

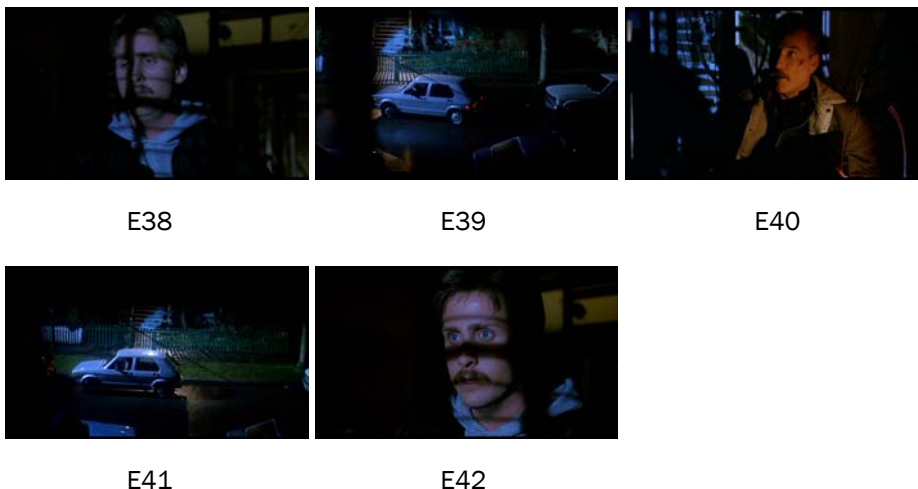


Abbildung 30: Perzeptionseinstellungen (E38-E42) aus dem Spielfilm „Stakeout“.

Der zeitliche Ablauf in der Diegese ist direkt aufeinander folgend, also ohne Lücke. Die zeitlichen Verhältnisse auf der Layoutebene gestalten sich wie folgt:

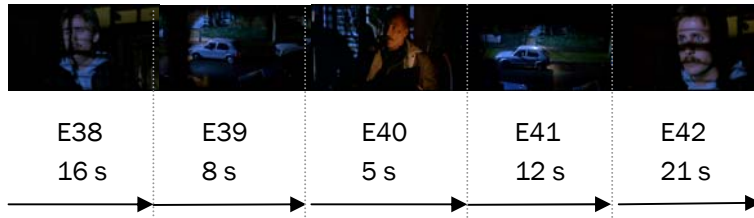


Abbildung 31: Zeitliche Verhältnisse auf der Layoutebene.

In E38 kann ein Beobachter einen intradiegetischen Perzeptor P konzeptionalisieren. Damit gehört E38 zum Perzeptorsegment S_x auf Einstellungsebene. E39 zeigt das Perzept von P. Die Kameraposition lässt auf den Standpunkt des Perzeptors aus E38 schließen, was der Kameraposition K.1 (gemäß Abschnitt 7.4) entspricht. E39 stellt also das Perzept des Perzeptors in E38 dar und gehört damit zum Perzeptsegment S_y . Interessant wird nun die dramaturgische Zusammenstellung durch die Einstellung E40, die Bill zeigt. Die Position der Kamera entspricht nicht der des Perzeptors. E40 ist daher weder dem Segment S_x noch dem Segment S_y zuzuschreiben. Dem Beobachter wird nicht gezeigt, ob Chris während des Ablaufs von E40 weiter perzipiert oder nicht. Laut Definition muss das Fortsetzen der Perception angenommen werden. Dies führt dazu, dass E41 dem Perzeptsegment S_y und E42 dem Perzeptorsegment S_x zugeordnet werden kann. Es ergibt sich also folgende Zusammensetzung der Segmente:

S_x {E38, E42}; S_y {E39, E41}.

Für das Gesamtsegment S ergibt sich die Repräsentation der Perceptionssituation mit zeitlicher Lücke. Durch die unterstellte räumliche Nähe des Perzeptors P kann der Beobachter B die zeitliche Lücke, die durch E40 entsteht, interpolieren. Somit kann das Gesamtsegment S als eine visuelle Erfahrung klassifiziert werden. Es bleibt anzumerken, dass E40 durch den zusammenhängenden Raum mit E38 und E42 bereits szenisch integriert ist. Daraus ergibt sich folgende Darstellung mit MPEG-7:

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video id="S">
        <CreationInformation>
          <Creation>
            <Title>Chris beobachtet ein heranfahrendes Fahrzeug.
              Bill sitzt mit im Zimmer.</Title>
          </Creation>
        </CreationInformation>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

```

    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT02M10S</MediaDuration>
  </MediaTime>
  <TemporalDecomposition>
    <VideoSegment id="Sx">
      <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Sequenz"/>
      <CreationInformation>
        <Creation>
          <Title>Chris sieht aus dem Fenster.</Title>
        </Creation>
      </CreationInformation>
      <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets" target="#Sy"/>
      <TemporalMask>
        <SubInterval>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT00M16S</MediaDuration>
        </SubInterval>
        <SubInterval>
          <MediaTimePoint>T00:00:42</MediaTimePoint>
          <MediaDuration>PT00M21S</MediaDuration>
        </SubInterval>
      </TemporalMask>
      <SpatioTemporalDecomposition>
        <MovingRegion id="Smr">
          <CreationInformation>
            <Creation>
              <Title>Chris als Moving Region</Title>
            </Creation>
          </CreationInformation>
          <SpatioTemporalLocator>
            <FigureTrajectory type="polygon">
              <MediaTime>
                <MediaTimePoint>T00:00:00</MediaTimePoint>
                <MediaDuration>PT1M03S</MediaDuration>
              </MediaTime>
              <!-- Beschreibung eines Polygons -->
            </FigureTrajectory>
          </SpatioTemporalLocator>
        </MovingRegion>
      </SpatioTemporalDecomposition>
    </VideoSegment>
    <VideoSegment id="Sy">
      <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Sequenz"/>
      <CreationInformation>
        <Creation>
          <Title>Die Kamera zeigt das Auto.</Title>
        </Creation>
      </CreationInformation>
      <TemporalMask>
        <SubInterval>
          <MediaTimePoint>T00:00:17</MediaTimePoint>
          <MediaDuration>PT00M8S</MediaDuration>
        </SubInterval>
        <SubInterval>
          <MediaTimePoint>T00:00:30</MediaTimePoint>
          <MediaDuration>PT00M12S</MediaDuration>
        </SubInterval>
      </TemporalMask>
    </VideoSegment>
  </TemporalDecomposition>
</Video>
</MultimediaContent>
</Description>
<Description xsi:type="SemanticDescriptionType">
  <Semantics>
    <Label>
      <Name>Chris beobachtet ein herannahendes Fahrzeug.</Name>
    </Label>
  </Semantics>
</Description>

```

```

<!-- ***** -->
<SemanticBase xsi:type="EventType" id="EV_perceiving">
  <Label>
    <Name>Chris blickt aus dem Fenster.</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
  target="#Sx"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
  target="#l_perceiving"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
  target="#t_perceiving"/>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="SemanticPlaceType" id="l_perceiving">
  <Label>
    <Name>Wohnung, 1. Stock</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:disjoint"
  target="#l_percept"/>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="SemanticTimeType" id="t_perceiving">
  <Label>
    <Name>01.12.1997</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets"
  target="#t_percept"/>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="EventType" id="EV_percept">
  <Label>
    <Name>Heranfahrendes Auto</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
  target="#Sy"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
  target="#l_percept"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
  target="#t_percept"/>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="SemanticPlaceType" id="l_percept">
  <Label>
    <Name>Hauptstraße</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:disjoint"
  target="#l_perceiving"/>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="SemanticTimeType" id="t_percept">
  <Label>
    <Name>01.12.1997</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metby"
  target="#t_perceiving"/>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="AgentObjectType" id="AO_perceptor">
  <Label>
    <Name>Chris</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
  target="#Smr"/>
  <Relation type="urn:mpeg:mpeg7:extension:perception:cameraAxisRelationK1"
  target="#EV_percept"/>
  <Agent xsi:type="PersonType">
    <Name>
      <GivenName>Chris</GivenName>
    </Name>
  </Agent>
</SemanticBase>

```

```

    </Agent>
  </SemanticBase>
  <!-- ***** -->
  <SemanticBase xsi:type="EventType" id="EV_perception">
    <Label>
      <Name>Chris sieht ein herannahendes Auto.</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent "
      target="#AO_perceptor" />
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:stimulusOf "
      target="#EV_percept" />
  </SemanticBase>
</Semantics>
</Description>
</Mpeg7>

```

Die Besonderheit bei diesem Beispiel besteht darin, dass sowohl das Segment Sx als auch das Segment Sy zeitliche Lücken enthalten. Deshalb werden zur Definition der Segmente die im MDS definierten temporalen Masken verwendet, die es erlauben, zeitliche Subintervalle anzugeben, die das Segment kennzeichnen. Die Modellierung der Perzeptionsrelation erfolgt dann analog zum vorherigen Beispiel.

Zwei Beispiele von Perzeption im Spielfilm „Fahrenheit 451“ wurden 1979 von Möller-Naß in [Moeller 1979] syntagmatisch analysiert. Nachfolgend werden Ausschnitte seiner Analyse wiedergegeben und die beschriebenen Perzeptionssituationen im Anschluss in MPEG-7 formalisiert.

9.3 Beispiel 3

Ein erstes Beispiel für Perzeption liefert die Folge von Einstellungen E36–E40 (s. Abbildung 32), in denen der Feuerwehrmann Montag eine Entdeckung macht: In einer Deckenlampe liegt ein Buch.

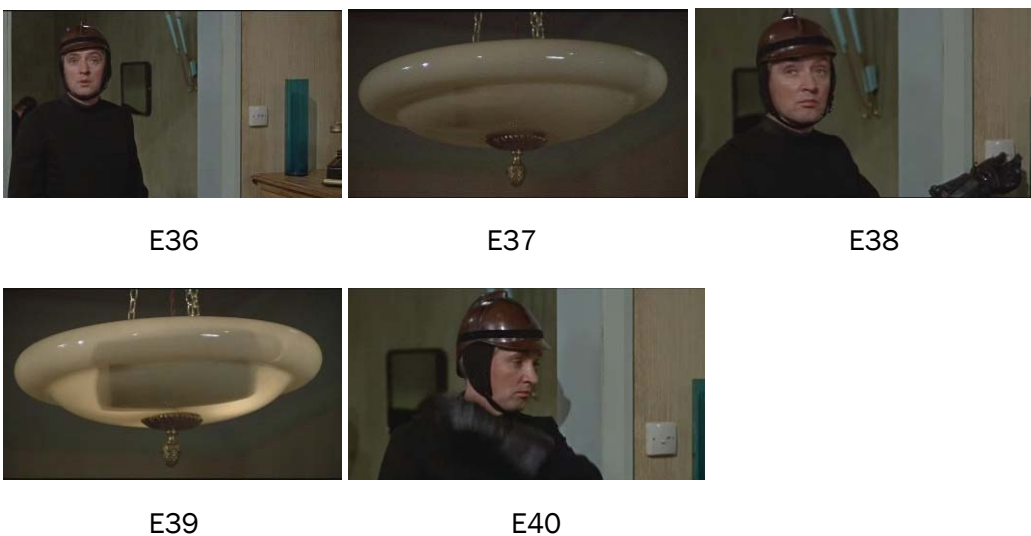


Abbildung 32: Perzeptionseinstellungen (E36–E40) aus dem Spielfilm „Fahrenheit 451“.

Möller-Naß beschreibt die Einstellungen in [Moeller 1979, S. 97] wie folgt:

36	Montag sieht (etwas).	(xS)
37	(Montag sieht) eine Deckenlampe. (Montag weiß, dass eine solche Lampe ein gutes Versteck für Bücher ist. Er weiß auch, wie man das überprüfen kann.)	(yS')
38	Montag sieht noch immer (die Lampe). Er reagiert, indem er das Licht anstellt.	(xS)
39	(Was Montag sieht:) Die Lampe, die nun hell wird. (Das Resultat seiner Aktion.) (In der Lampe ist der Schatten eines Buchs zu sehen.)	(yS')
40	Montags Entdeckung und seine Reaktion: Er winkt einen Feuerwehrmann herbei, der das Buch aus der Lampe holen soll.	(xR)

Tabelle 4: Einstellungsfolgen (36-40) aus „Fahrenheit 451“.

In E36 kann der Beobachter B einen intradiegetischen Perzeptor P konzeptionalisieren. Damit gehört E36 zum einem ersten Perzeptorsegment auf Einstellungsebene S1x. E37 zeigt das Perzept von P. Die Kameraposition lässt auf den Standpunkt des Perzeptors aus E36 schließen, was der Kameraposition K.1 (gemäß Abschnitt 7.4) entspricht. E37 stellt also das Perzept des Perzeptors in E36 dar und gehört damit zum ersten Perzeptsegment S1y.

Der Perzeptor P wendet sich in E38, nachdem er die Lampe gesehen hat, von seinem Perzept ab, um den Lichtschalter zu betätigen und perzipiert dann erneut. Dadurch entsteht eine neue visuelle Erfahrung und damit eine neue Perzeptionssituation mit einem zweiten Perzeptorsegment S2x. Die Kameraposition in E39 lässt auf den Standpunkt des Perzeptors aus E38 schließen. E39 stellt also das Perzept des Perzeptors in E38 dar und gehört damit zum Perzeptsegment S2y. Für E40 muss keine eigene Klassifikation vorgenommen werden, da hier nicht mehr perzipiert wird. Daraus ergibt sich die folgende formalisierte Darstellung mit MPEG-7:

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video id="S">
        <CreationInformation>
          <Creation>
```

```

    <Title>Der Feuerwehrmann Montag beobachtet eine Lampe und entdeckt
    darin ein Buch.</Title>
  </Creation>
</CreationInformation>
<MediaTime>
  <MediaTimePoint>T00:00:00</MediaTimePoint>
  <MediaDuration>PT0M18S</MediaDuration>
</MediaTime>
<TemporalDecomposition>
  <VideoSegment id="S1x">
    <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Szene"/>
    <CreationInformation>
      <Creation>
        <Title>Montag läuft durch den Raum und sieht etwas</Title>
      </Creation>
    </CreationInformation>
    <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets"
    target="#S1y"/>
    <MediaTime>
      <MediaTimePoint>T00:00:00</MediaTimePoint>
      <MediaDuration>PT0M10S</MediaDuration>
    </MediaTime>
    <SpatioTemporalDecomposition>
      <MovingRegion id="S1mr">
        <CreationInformation>
          <Creation>
            <Title>Montag als Moving Region.</Title>
          </Creation>
        </CreationInformation>
        <SpatioTemporalLocator>
          <FigureTrajectory type="polygon">
            <MediaTime>
              <MediaTimePoint>T00:00:00</MediaTimePoint>
              <MediaDuration>PT0M10S</MediaDuration>
            </MediaTime>
            <!-- Beschreibung eines Polygons -->
          </FigureTrajectory>
        </SpatioTemporalLocator>
      </MovingRegion>
    </SpatioTemporalDecomposition>
  </VideoSegment>
  <VideoSegment id="S1y" >
    <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Szene"/>
    <CreationInformation>
      <Creation>
        <Title>Die Kamera zeigt eine dunkle Lampe.</Title>
      </Creation>
    </CreationInformation>
    <MediaTime>
      <MediaTimePoint>T00:00:11</MediaTimePoint>
      <MediaDuration>PT0M02S</MediaDuration>
    </MediaTime>
  </VideoSegment>
  <VideoSegment id="S2x" >
    <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Szene"/>
    <CreationInformation>
      <Creation>
        <Title>Montag läuft zum Lichtschalter und blickt erneut zur Lampe.</Title>
      </Creation>
    </CreationInformation>
    <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets"
    target="#S2y"/>
    <MediaTime>
      <MediaTimePoint>T00:00:14</MediaTimePoint>
      <MediaDuration>PT0M03S</MediaDuration>
    </MediaTime>
    <SpatioTemporalDecomposition>
      <MovingRegion id="S2mr">
        <CreationInformation>

```

```

        <Creation>
          <Title>Montag als Moving Region</Title>
        </Creation>
      </CreationInformation>
    <SpatioTemporalLocator>
      <FigureTrajectory type="polygon">
        <MediaTime>
          <MediaTimePoint>T00:00:14</MediaTimePoint>
          <MediaDuration>PT0M03S</MediaDuration>
        </MediaTime>
        <!-- Beschreibung eines Polygons -->
      </FigureTrajectory>
    </SpatioTemporalLocator>
  </MovingRegion>
</SpatioTemporalDecomposition>
</VideoSegment>
<VideoSegment id="S2y" xsi:type="PerceptVideoSegmentType">
  <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Szene"/>
  <CreationInformation>
    <Creation>
      <Title>Die Kamera zeigt eine helle Lampe mit einem Buch.</Title>
    </Creation>
  </CreationInformation>
  <MediaTime>
    <MediaTimePoint>T00:00:17</MediaTimePoint>
    <MediaDuration>PT0M02S</MediaDuration>
  </MediaTime>
</VideoSegment>
</TemporalDecomposition>
</Video>
</MultimediaContent>
</Description>
<Description xsi:type="SemanticDescriptionType">
  <Semantics>
    <Label>
      <Name>Montag beobachtet eine dunkle Lampe</Name>
    </Label>
    <!-- ***** -->
    <SemanticBase xsi:type="EventType" id="EV1_perceiving">
      <Label>
        <Name>Montag macht eine Beobachtung.</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
        target="#S1x"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
        target="#l1_perceiving"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
        target="#t1_perceiving"/>
    </SemanticBase>
    <SemanticBase xsi:type="SemanticPlaceType" id="l1_perceiving">
      <Label>
        <Name>Wohnhaus, Erdgeschoß</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:overlaps"
        target="#l1_percept"/>
    </SemanticBase>
    <SemanticBase xsi:type="SemanticTimeType" id="t1_perceiving">
      <Label>
        <Name>20.04.1969</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets"
        target="#t1_percept"/>
    </SemanticBase>
    <!-- ***** -->
    <SemanticBase xsi:type="EventType" id="EV1_percept">
      <Label>
        <Name>Eine im Wohnzimmer aufgehängte Deckenlampe</Name>
      </Label>

```



```

    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
    target="#Sly"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
    target="#l1_percept"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
    target="#t1_percept"/>
  </SemanticBase>
  <!-- ***** -->
  <SemanticBase xsi:type="SemanticPlaceType" id="l1_percept">
    <Label>
      <Name>Wohnhaus, Erdgeschoß</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:overlaps"
    target="#l1_perceiving"/>
  </SemanticBase>
  <SemanticBase xsi:type="SemanticTimeType" id="t1_percept">
    <Label>
      <Name>20.04.1969</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metby"
    target="#t1_perceiving"/>
  </SemanticBase>
  <!-- ***** -->
  <SemanticBase xsi:type="AgentObjectType" id="A01_perceptor">
    <Label>
      <Name>Montag</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
    target="#S1mr"/>
    <Relation type="urn:mpeg7_extension:perception:cameraAxisRelationK1"
    target="#EV1_percept"/>
    <Agent xsi:type="PersonType">
      <Name>
        <GivenName>Montag</GivenName>
      </Name>
    </Agent>
  </SemanticBase>
  <!-- ***** -->
  <SemanticBase xsi:type="EventType" id="EV1_perception">
    <Label>
      <Name>Montag beobachtet eine Deckenlampe.</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
    target="#A01_perceptor"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:stimulusOf"
    target="#EV1_percept"/>
  </SemanticBase>
  <!-- ***** -->
</Semantics>
<Semantics>
  <Label>
    <Name>Montag beobachtet eine helle Lampe.</Name>
  </Label>
  <!-- ***** -->
  <SemanticBase xsi:type="EventType" id="EV2_perceiving">
    <Label>
      <Name>Montag macht eine Beobachtung</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
    target="#S2x"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
    target="#l2_perceiving"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
    target="#t2_perceiving"/>
  </SemanticBase>
  <!-- ***** -->
  <SemanticBase xsi:type="SemanticPlaceType" id="l2_perceiving">
    <Label>
      <Name>Wohnhaus, Erdgeschoß</Name>
    </Label>

```

```

    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:overlaps"
      target="#l2_percept"/>
  </SemanticBase>
  <SemanticBase xsi:type="SemanticTimeType" id="t2_perceiving">
    <Label>
      <Name>20.04.1969</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets"
      target="#t2_percept"/>
  </SemanticBase>
  <!-- ***** -->
  <SemanticBase xsi:type="EventType" id="EV2_percept">
    <Label>
      <Name>Eine erleuchtete Deckenlampe mit einem darin
        befindlichen Buch.</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
      target="#S2y"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
      target="#l2_percept"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
      target="#t2_percept"/>
  </SemanticBase>
  <!-- ***** -->
  <SemanticBase xsi:type="SemanticPlaceType" id="l2_percept">
    <Label>
      <Name>Wohnhaus, Erdgeschoß</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:overlaps"
      target="#l2_perceiving"/>
  </SemanticBase>
  <!-- ***** -->
  <SemanticBase xsi:type="SemanticTimeType" id="t2_percept">
    <Label>
      <Name>20.04.1969</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metby"
      target="#t2_perceiving"/>
  </SemanticBase>
  <!-- ***** -->
  <SemanticBase xsi:type="AgentObjectType" id="AO2_perceptor">
    <Label>
      <Name>Montag</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
      target="#S2mr"/>
    <Relation type="urn:mpeg7_extension:perception:cameraAxisRelationK1"
      target="#EV2_percept"/>
    <Agent xsi:type="PersonType">
      <Name> <GivenName>Montag</GivenName></Name>
    </Agent>
  </SemanticBase>
  <!-- ***** -->
  <SemanticBase xsi:type="EventType" id="EV2_perception">
    <Label>
      <Name>Montag beobachtet erleuchtete eine Deckenlampe in der sich
        ein Buch befindet.</Name>
    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
      target="#AO2_perceptor"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:stimulusOf"
      target="#EV2_percept"/>
  </SemanticBase>
  <!-- ***** -->
</Semantics>
</Description>
</Mpeg7>

```

Dieses Beispiel weist die Besonderheit auf, dass aufgrund von zwei Perzeptionssituationen auch zwei Perzeptor- und zwei Perzeptionssegmente existieren. Die Modellierung dieser beiden Perzeptionsrelationen erfolgt dann analog zu den vorherigen beiden Beispielen.

9.4 Beispiel 4

Ein weiteres Beispiel für Perzeption liefert die Folge von Einstellungen E306–E309 (s. Abbildung 33) aus dem Film „Fahrenheit 451“, in denen Fabian den Feuerwehrmann Montag zusammen mit Clarissa beobachtet.



Abbildung 33: Perzeptionseinstellungen (E306–E309) aus dem Spielfilm „Fahrenheit 451“.

Möller-Naß beschreibt die Einstellungen in [Moeller 1979, S. 104] wie folgt:

306	Montag schaut sich um, nimmt Clarissa bei der Hand. Fahrt und Schwenk mit Montag und Clarissa, die die Treppe auf der anderen Seite der Brücke wieder hinuntergehen. Sie nähern sich einem Restaurant.	
307	Fabian auf der anderen Straßenseite. Solange er geht, schwenkt die Kamera mit ihm. Als er die beiden bemerkt, bleibt er stehen und schaut ihnen nach.	(xS)
308	Montag und Clarissa gehen in das Restaurant. (Von hinten gesehen.)	(yS')

309	(wie 307, Fortsetzung) Fabian schaut den beiden noch immer nach. Dann geht er weiter, die Kamera schwenkt erst mit, bleibt dann stehen. Fabian geht nach links aus dem Bild	(xS)
-----	--	------

Tabelle 5: Einstellungsfolgen (306-309) aus „Fahrenheit 451“.

In E307 kann ein Beobachter B einen intradiegetischen Perzeptor P konzeptionalisieren. Damit gehört E307 zum Perzeptorsegment S_x auf Einstellungsebene. E308 zeigt das Perzept von P. Die Kameraposition lässt auf den Standpunkt des Perzeptors aus E307 schließen. E308 stellt also das Perzept des Perzeptors in E307 dar und gehört damit zum Perzeptsegment S_y.

E309 zeigt die Fortsetzung des Perzeptionsvorgangs von E307, was eine Hinzunahme von E309 zum Perzeptorsegment S_x nötig macht.

Für E306 muss keine eigene Klassifikation vorgenommen werden, da diese Einstellung noch nicht als Perzept gelten kann. Daraus ergibt sich folgende formalisierte Darstellung mit MPEG-7:

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video id="S">
        <CreationInformation>
          <Creation>
            <Title>Montag läuft mit Clarissa eine Treppe hinunter. Die beiden gehen in ein Restaurant und werden dabei von Fabian beobachtet.</Title>
          </Creation>
        </CreationInformation>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT1M11S</MediaDuration>
        </MediaTime>
        <TemporalDecomposition>
          <VideoSegment id="Sx" >
            <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Planszene"/>
            <CreationInformation>
              <Creation>
                <Title>Fabian beobachtet etwas.</Title>
              </Creation>
            </CreationInformation>
            <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets" target="#Sy"/>
            <TemporalMask>
              <SubInterval>
                <MediaTimePoint>T00:01:02</MediaTimePoint>
                <MediaDuration>PT00M03S</MediaDuration>
              </SubInterval>
              <SubInterval>
                <MediaTimePoint>T00:01:09</MediaTimePoint>
                <MediaDuration>PT00M03S</MediaDuration>
              </SubInterval>
            </TemporalMask>
          </VideoSegment>
        </TemporalDecomposition>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

```

    <SpatioTemporalDecomposition>
      <MovingRegion id="Smr">
        <CreationInformation>
          <Creation>
            <Title>Fabian als Moving Region</Title>
          </Creation>
        </CreationInformation>
      <SpatioTemporalLocator>
        <FigureTrajectory type="polygon">
          <MediaTime>
            <MediaTimePoint>T00:01:02</MediaTimePoint>
            <MediaDuration>PT0M9S</MediaDuration>
          </MediaTime>
          <!-- Beschreibung eines Polygons -->
        </FigureTrajectory>
      </SpatioTemporalLocator>
    </MovingRegion>
  </SpatioTemporalDecomposition>
</VideoSegment>
<VideoSegment id="Sy">
  <StructuralUnit href="urn:mpeg:mpeg7:cs:StructuralUnitCS:Planszene"/>
  <CreationInformation>
    <Creation>
      <Title>Die Kamera zeigt Montag und Clarissa beim Eintreten in ein
      Restaurant.</Title>
    </Creation>
  </CreationInformation>
  <MediaTime>
    <MediaTimePoint>T00:01:06</MediaTimePoint>
    <MediaDuration>PT0M03S</MediaDuration>
  </MediaTime>
</VideoSegment>
</TemporalDecomposition>
</Video>
</MultimediaContent>
</Description>
<Description xsi:type="SemanticDescriptionType">
  <Semantics>
    <Label>
      <Name>Montag und Clarissa gehen in ein Restaurant und werden dabei von
      Fabian beobachtet.</Name>
    </Label>
    <!-- ***** -->
    <SemanticBase xsi:type="EventType" id="EV_perceiving">
      <Label>
        <Name>Fabian blickt</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
      target="#Sx"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
      target="#l_perceiving"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
      target="#t_perceiving"/>
    </SemanticBase>
    <!-- ***** -->
    <SemanticBase xsi:type="SemanticPlaceType" id="l_perceiving">
      <Label>
        <Name>vor dem Restaurant</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:overlaps"
      target="#l_percept"/>
    </SemanticBase>
    <SemanticBase xsi:type="SemanticTimeType" id="t_perceiving">
      <Label>
        <Name>23.05.1969</Name>
      </Label>
      <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets"
      target="#t_percept"/>
    </SemanticBase>
  </Semantics>

```

```

<!-- ***** -->
<SemanticBase xsi:type="EventType" id="EV_percept">
  <Label>
    <Name>Montag und Clarissa sind auf dem Weg in das Restaurant.</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
  target="#Sy"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
  target="#l_percept"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
  target="#t_percept"/>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="SemanticPlaceType" id="l_percept">
  <Label>
    <Name>vor dem Restaurant</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:overlaps"
  target="#l_perceiving"/>
</SemanticBase>
<SemanticBase xsi:type="SemanticTimeType" id="t_percept">
  <Label>
    <Name>23.05.1969</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metby"
  target="#t_perceiving"/>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="AgentObjectType" id="AO_perceptor">
  <Label>
    <Name>Fabian</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
  target="#Smr"/>
  <Relation type="urn:mpeg7_extension:perception:cameraAxisRelationK1"
  target="#EV_percept"/>
  <Agent xsi:type="PersonType">
    <Name>
      <GivenName>Fabian</GivenName>
    </Name>
  </Agent>
</SemanticBase>
<!-- ***** -->
<SemanticBase xsi:type="EventType" id="EV_perception">
  <Label>
    <Name>Fabian beobachtet Montag und Clarissa beim Eintreten in das
    Restaurant.</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
  target="#AO_perceptor"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:stimulusOf"
  target="#EV_percept"/>
</SemanticBase>
</Semantics>
</Description>
</Mpeg7>

```

Dieses Beispiel weist die Besonderheit auf, dass das Perzeptorsegment durch eine zeitliche Maske als nicht zusammenhängend klassifiziert werden kann. Dies ist möglich, weil in E309 der Sehvorgang von E307 fortgesetzt wird. Daraus folgt die Klassifikation einer einzigen Perzeptionssituation und somit die Modellierung eines einzigen Perzeptorsegmentes Sx

10 Schlussbetrachtung

Nach der Einführung des Begriffs des „Dokuments“ zu Beginn dieser Arbeit wurden darauf aufbauend verschiedene Ansätze der Annotation von Dokumenten durch Metadaten vorgestellt. Es konnte gezeigt werden, dass die verschiedenen Strukturen von Dokumenten entscheidenden Einfluss darauf haben, wie und wie gut diese Metadaten an Dokumente und ihre Teile gebunden werden können.

Darauf aufbauend wurden unterschiedliche Typen von Metadaten diskutiert. Es wurde beschrieben, dass bibliothekarische Metadaten das Auffinden von Dokumenten als Ganzes erleichtern können. Allerdings ist es z. B. mit dem genormten Metadaten-satz „Dublin Core“ nicht möglich, Strukturen von Dokumenten mit differenzierten Metadaten zu versehen.

Voraussetzung für solch eine differenzierte Metadatenannotation ist das Vorhandensein von strukturierten Dokumenten. Vor allem im Bereich der Textdokumente sind durch die Metasprache XML die technischen Voraussetzungen für die Annotation mit Metadaten für Teildokumente gegeben. Auch Filmdokumente können strukturiert sein, bieten jedoch unter Umständen nur eingeschränkte Möglichkeiten, diese Strukturen in sich selbst zu tragen, da Filmdokumente im Allgemeinen häufig als Binärdokumente vorliegen und somit deren logische Strukturen von extern annotiert werden müssen.

Dem Aufbau der Arbeit entsprechend wurde nach der Vorstellung der Grundlagen von XML zunächst die Regelsprache DTD dahingehend untersucht, inwieweit ihre sprachlichen Mittel für die Beschreibung von Nicht-Text-Strukturen ausreichen, um Metadaten für cinematographische Dokumente zu beschreiben. Es ist offensichtlich, dass DTDs nicht den benötigten Umfang an Beschreibungswerkzeugen bereitstellen. Darauf aufbauend wurde die ausdrucksstärkere Regelsprache XML-Schema (XSD) eingeführt und die Analyse für die Beschreibung von Nicht-Text-Strukturen auf diese Sprache angewandt. Es zeigte sich, dass XSD zwar wesentlich umfangreichere Strukturierungsmöglichkeiten für Dokumente bietet, diese jedoch nicht ausreichend auf binäre Inhalte anwendbar sind.

Der in XSD definierte Metadatenstandard MPEG-7 nutzt die Mächtigkeit von XSD, um einen neuen, speziell für Multimediadokumente entwickelten Standard zur Metadatenannotation zu beschreiben. Mit den Mitteln von XSD wird ein Metadaten-schema definiert, mit dem sich Multimedia-Dokumente klassifizieren und deren Eigenschaften in Instanzen formalisiert beschreiben lassen.

Zentral ist dabei die Definition von Deskriptoren, die die Möglichkeit bieten, jedes noch so kleine Detail einer Bildsequenz zu beschreiben. Zudem ermöglicht MPEG-7

die zeitliche und räumliche Zerlegung von Segmenten in Subsegmente. Dadurch lassen sich Relationen zwischen Segmenten mit Eigenschaften beschreiben, wodurch ein Beziehungsnetzwerk zwischen diesen Segmenten aufgebaut werden kann.

Im Unterschied zu standardisierten Anwendungen von XML im Bereich der Textdokumente (z.B. DocBook) bietet MPEG-7 keine Definition von speziellen cinematographischen Dokumentstrukturen. Für den inhaltlichen Zugriff auf cinematographische Dokumente sind solche Strukturen allerdings unverzichtbar. Dieser Mangel muss medienspezifisch behoben werden. Für den Fall von Videodokumenten (im Sinne von MPEG-7) wurden zuerst die frühen Arbeiten zu Filmgrammatiken beschrieben. Aufbauend auf den formalisierten Grammatiken für basale narrative Syntagmen durch Schmidt und Strauch [Schmidt, Strauch 2002] und Schmidt [Schmidt 2004] wurden dann neue, logische Strukturen in Filmdokumenten herausgearbeitet, die im klassischen Spielfilm unterhalb der Einstellungsebene für das Verständnis des Beobachters wichtig sind.

Kapitel 7 ging der zentralen Fragestellung dieser Arbeit nach: Gibt es Syntagmen, die „oberhalb“ eines räumlichen Zusammenhangs Segmente als zusammengehörig klassifizieren können?

Als Basis für Syntagmen, die eine solche Aggregationsleistung bieten, dienen die Arbeiten von Branigan [Branigan 1984, Branigan 1992]. Es zeigte sich, dass die Ebene der sogenannten Fokalisierung eine solche Aggregation ermöglicht. Dies wurde hier speziell dadurch erreicht, dass wenigstens einem der in einer Einstellung repräsentierten Objekte perzeptive Fähigkeiten zugeschrieben wurde.

Existiert in einer Einstellung ein Subsegment, das diegetisch mit perzeptiven Fähigkeiten ausgestattet werden kann, so kann der Beobachter des Dokumentes eine Aggregationsleistung vollbringen, die unabhängig von räumlichen Zusammenhangsbedingungen funktioniert.

Es wurde gezeigt, dass die filmische Repräsentation von Perzeption in der Syntagmatik von Videodokumenten ein entscheidendes Mittel für die Aggregation von Einstellungen ist. Dies geschieht durch die Verknüpfung eines Teissegmentes, das einen Perzeptor repräsentiert, mit einer ganzen Einstellung, die ein Perzept repräsentiert, so dass einzelne Segmente unterhalb der Einstellungsebene mit ganzen Einstellungen verknüpft werden.

Die letzten zwei Kapitel dieser Arbeit enthalten die formalisierte Aufbereitung der neu gewonnenen perzeptiven Syntagmen. In Kapitel 8 wurde das perzeptiv fokalisierte Syntagma mit dem MDS beschrieben und in Kapitel 9 anhand verschiedener Beispiele filmischer Perzeption getestet.

11 Literatur

- [Allen 1983] J. F. Allen: Maintaining knowledge about temporal intervals. In: Communications of the ACM, Vol. 26/1983, S. 832–843.
- [Allen, Hayes 1985] J. F. Allen, P. J. Hayes: A commonsense theory of time. In: Proceedings of AAAI-85, 1985, S. 528–531.
- [Barwise, Perry 1983] J. Barwise, J. Perry: Situations and Attitudes. Cambridge 1983.
- [Bordwell 1986] D. Bordwell: Narration in the fiction film. London (Wisconsin) 1986.
- [Branigan 1984] E. Branigan: Point of View in the Cinema. A Theory of Narration and Subjectivity in Classical Film. Berlin/New York 1984.
- [Branigan 1992] E. Branigan: Narrative comprehension and Film. London/New York 1992.
- [Bray et al. 2006] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler: Extensible Markup Language (XML) 1.0 (Fourth Edition). W3C Recommendation 16 August 2006.
- [Buckland 1995] W. Buckland: The Film Spectator. From Sign to Mind. Amsterdam 1995.
- [Chatman 1992] S. Chatman: What Novels Can Do That Films Can't (and Vice Versa). In: Film Theory and Criticism: introductory readings, Fourth Edition, 1992, S. 403–419.
- [Davis 1990] E. Davis: Representations of Commonsense Knowledge. San Mateo/California 1990.
- [DCMI 2006] DCMI Usage Board: DCMI Metadata Terms. 2006.
URL: <http://dublincore.org/documents/2006/12/18/dcmi-terms/>.
Stand: 20.7.2007.
- [Devlin 1993] K. Devlin: Infos und Infone. Die mathematische Struktur der Informationen. Basel/Boston/Berlin 1993.
- [Erdmann 2001] M. Erdmann: Ontologien zur konzeptuellen Modellierung der Semantik von XML. Dissertation, Karlsruhe 2001.
- [Genette 1994] G. Genette: Die Erzählung. München 1994.
- [Grimm 1996] P. Grimm: Filmnarratologie. Eine Einführung in die Praxis der Interpretation am Beispiel des Werbespots. München 1996.

- [**Guha 1995**] R. V. Guha: Towards a Theory of Meta Content. Apple Technical Report Nr. 169. Cupertino 1995.
- [**Hedler 2002**] M. Hedler: Evaluierung von Topic Maps für einen Wissenschaftsverlag. Diplomarbeit, Stuttgart 2002.
- [**Hickethier 1993**] K. Hickethier: Film- und Fernsehanalyse. Stuttgart 1993.
- [**Hunter 1999**] J. Hunter: A Comparison of Schemas for Video Metadata Representation. In: Proceeding of the eighth international conference on World Wide Web, 1999, S. 1431–1451.
- [**ISO/IEC 15938**] ISO/IEC: Information Technology – Multimedia Content Description Interface. MPEG-7. ISO/IEC 15938/2003.
- [**ISO/IEC 15938-5**] ISO/IEC: Information Technology – Multimedia Content Description Interface; Part5: Multimedia description schemes. MPEG-7. ISO/IEC 15938-5/2003.
- [**ISO/IEC 8601**] ISO/IEC: Data elements and interchange formats – Information Exchange – Representation of dates and times. ISO/IEC 8601/1988.
- [**ISO/IEC 8612**] ISO/IEC: Open Document Architecture Consortium. The Open Document Architecture. ISO/IEC 8612/1995.
- [**Lanser 1981**] S. Lanser: The Narrative Act. Point of View in Prose Fiction. Princeton 1981.
- [**Lee 1997**] T. B. Lee: Metadata Architecture. 1997.
URL: <http://www.w3.org/DesignIssues/Metadata>.
Stand: 20.7.2007.
- [**Lee 1998**] T. B. Lee: Why RDF model is different from the XML model. 1998.
URL: <http://www.w3.org/DesignIssues/RDF-XML.html>.
Stand: 20.7.2007.
- [**Martinez 2002**] J. M. Martinez: Coding of moving pictures and audio. MPEG-7 Overview. 2002.
URL: <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>.
Stand: 20.7.2007.
- [**Metz 1964**] C. Metz: Le cinéma: langue ou langage? In: Communications, Vol. 4/1964, S. 52–90.
- [**Metz 1965**] C. Metz: Une étape dans la réflexion sur le cinéma. In: Critique, Vol. 214/1965, S. 227–248.

- [**Metz 1966**] C. Metz: La grande syntagmatique du film narratif. In: Communications, Vol. 8/1966, S. 120–124.
- [**Metz 1972**] C. Metz: Semiologie des Films. München 1972.
- [**Möller 1979**] K.-D. Möller: Diagrammatische Syntagmen und einfache Formen. In: Untersuchungen zur Syntax des Films I, 1979, S. 70–117.
- [**Möller-Naß 1988**] K.-D. Möller-Naß: Filmsprache – eine kritische Theoriegeschichte. Münster 1988.
- [**Obasanjo 2003**] D. Obasanjo: W3C XML Schema-Entwurfsmuster – Vermeiden von Komplexität. 2003.
URL: <http://www.microsoft.com/germany/msdn/library/data/xml/W3CXMLSchemaEntwurfsmusterVermeidenVonKomplexitaet.aspx?mfr=true>.
Stand: 20.7.2007.
- [**Pemberton et al. 2002**] P. Pemberton et al.: XHTML 1.0. The Extensible HyperText Markup Language (Second Edition). W3C Recommendation 26 January 2000, 2002.
- [**Russel, Norvig 1995**] S. Russel, P. Norvig: Artificial Intelligence, a modern approach. New Jersey 1995.
- [**Salembier, Smith 2001**] P. Salembier, J. R. Smith: MPEG-7 Multimedia Description Schemes. In: IEEE Transaction on circuits and systems for video technology, Vol. 11, No. 6/2001, S. 748-759.
- [**Schmidt 1999**] K.-H. Schmidt: Wissensmedien für kognitive Agenten. Bonn 1999.
- [**Schmidt 2004**] K.-H. Schmidt: Zur chronologischen Syntagmatik von Bewegtbilddaten (II): Polyspatiale Alternanz. In: Kodikas/Code, Vol. 27/2004, S. 95–123.
- [**Schmidt 2007**] K.-H. Schmidt: Zur chronologischen Syntagmatik von Bewegtbilddaten (III): Das Deskriptive Syntagma. Erscheint in: Kodikas/Code, Vol. 29/2008.
- [**Schmidt, Strauch 2002**] K.-H. Schmidt, T. Strauch: Zur Syntagmatik von visuellen Bewegtbilddaten. Eine semiologische Reklassifikation der Metz'schen Syntagmatik. In: Kodikas/Code, Vol. 25/2002, S. 64–94.
- [**Strohmeier 2002**] M. Strohmeier: XML-Strukturaquisition. In: Informatik Spektrum, Vol. 8/2002, S. 12–17.
- [**Thompson et al. 2004**] H. S. Thompson, D. Beech, M. Maloney, N. Mendelson: XML Schema Part 1: Structures. W3C Recommendation 28 October 2004.

[van der Vlist 2003] E. van der Vlist: XML Schema. Köln 2003.

[van Beek et al. 2001] P. van Beek et al.: Information Technology – Multimedia Content Description Interface – Part 5: Multimedia Description Schemes. ISO Working Draft ISO/IEC JTC 1/SC 29/WG 11/N3966. Singapore 2001.

[Wulf 1999] H. J. Wulf: Darstellen und Mitteilen. Elemente der Pragmasemiotik des Films. Tübingen 1999.

[Wilhelm, Heckmann 1996] R. Wilhelm, R. Heckmann: Grundlagen der Dokumentenverarbeitung. Bonn 1996.