



TECHNISCHE UNIVERSITÄT ILMENAU

Fakultät für Informatik und Automatisierung

Institut für Praktische Informatik

Fachgebiet Telematik/Rechnernetze

# Zeitmodelle für Multimedia-Streams

**Hauptseminar kooperatives Multimedia-Streaming**

Sebastian Häfelbarth

Betreuer: Dipl.-Inf. Thorsten Strufe

Sommersemester 2004

## **Inhaltsverzeichnis**

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Begriffserklärungen</b>	<b>3</b>
<b>3</b>	<b>Zeit in Kommunikationsmodellen</b>	<b>4</b>
3.1	Warum beschäftigen wir uns mit der Zeit? . . . . .	4
3.2	Die Rolle der Zeit in multimedialen Streams . . . . .	4
<b>4</b>	<b>Zeitmodelle</b>	<b>5</b>
4.1	Was ist ein Zeitmodell? . . . . .	5
4.2	Klassen von Zeitmodellen . . . . .	5
4.3	Zeitmodelle kurz vorgestellt . . . . .	6
4.3.1	Time Line Modell . . . . .	6
4.3.2	Temporal Point Net . . . . .	6
4.3.3	Timed Petri Net . . . . .	7
4.3.4	Path expressions . . . . .	7
4.3.5	Multimedia Hypermedia Expert Group (MHEG) . . . . .	8
4.3.6	Fazit . . . . .	8
4.4	Ilmenauer Zeitmodell nach Süßkraut [2003] . . . . .	9
4.4.1	Bestandteile des Zeitmodells . . . . .	9
4.4.2	Prozessoren und Rechnerknoten . . . . .	11
4.4.3	Medienobjekte und Stromobjekte . . . . .	12
4.4.4	Fünf Erscheinungsformen der Zeit . . . . .	14
<b>5</b>	<b>Fazit</b>	<b>18</b>
	<b>Literatur</b>	<b>19</b>

## 1 Einleitung

Dieses Dokument beschäftigt sich mit Zeitmodellen für Multimedia-Streams. Es entstand in der Hauptseminar-Reihe kooperatives Multimedia-Streaming im Fachgebiet Telematik/Rechnernetze an der TU Ilmenau.

Es wird erklärt, warum man sich bei multimedialem Streaming mit der Zeit beschäftigen muss, was Zeitmodelle sind und welche es gibt.

Am Ende wird ein Zeitmodell vorgestellt, dass im Rahmen der Diplomarbeit von Süßkraut [2003] an der TU Ilmenau entstanden ist.

## 2 Begriffserklärungen

Zuerst sollen einige Begriffe dieser Arbeit geklärt werden.

Ein *multimediales Dokument* entsteht durch die Verschmelzung von mehreren medialen Dokumenten. So besteht das multimediale Dokument Film z.B. aus den medialen Dokumenten Video und Audio.

Die Verarbeitung multimedialer Dokumente geschieht durch *multimediale Anwendungen*. Diese lassen sich allgemein als gerichteter, kreisfreier Graph darstellen.

Die Knoten dieses Graphen repräsentieren die *Prozessoren*, die die multimedialen Dokumente verarbeiten. Ein Prozessor ist eine Softwarekomponente der multimedialen Anwendung. Auf einem Rechnerknoten können somit auch mehrere Prozessoren laufen. Prozessoren lassen sich in *Quellen*, *Senken* und *Filter* unterteilen. Quellen sind Prozessoren ohne eingehende Kanten und erzeugen somit multimediale Dokumente. Eine Quelle kann z.B. ein Mikrophon oder eine Kamera, aber auch eine Datei auf einer Festplatte sein. Senken sind Prozessoren ohne ausgehende Kanten. Sie konsumieren multimediale Dokumente. Dies kann z.B. die Wiedergabe eines Filmes oder auch die Ablage eines Videos auf einer Festplatte sein. Filter sind Prozessoren, die ein multimediales Dokument manipulieren. Sie besitzen eingehende und ausgehende Kanten und können Datenströme erzeugen und konsumieren. Dies geschieht z.B. wenn ein Untertitel in ein Video eingefügt wird. Weitere Beispiele für Filter sind Equalizer und Encoder.

Die Kanten zwischen den Prozessoren werden als *logische Übertragungskanäle* bezeichnet.

Ein multimediales Dokument bewegt sich innerhalb einer multimedialen Anwendung von der Quelle zur Senke auf einem festgelegten Weg über die Übertragungskanäle und bildet damit den *multimedialen Datenstrom*.

### 3 Zeit in Kommunikationsmodellen

#### 3.1 Warum beschäftigen wir uns mit der Zeit?

Multimediale Anwendungen, ob nun Teleteaching, Telemedizin, Videokonferenzen oder Live-Übertragungen auf Handys haben in den letzten Jahren an Bedeutung gewonnen. Dabei werden große multimediale Datenmengen über Kommunikationskanäle übertragen, die für diese teilweise nicht ausreichen.

Die meisten dieser Kommunikationskanäle sind für Echtzeitanwendungen nicht ausgelegt, da sie kein Quality of Service anbieten. Dadurch kann es bei Überlastung eines Kommunikationskanals dazu kommen, dass Teile der Übertragung verloren gehen.

Des Weiteren kann es, gerade bei den neuen drahtlosen Anwendungsszenarien mit Handy oder Laptop dazu kommen, dass Übertragungsfehler auftreten. Diese müssen erkannt und berichtigt werden.

Bei fehlenden bzw. fehlerhaften Datenpaketen stellt sich nun die Frage, ob es sich zeitlich lohnt, ein Paket erneut anzufragen, oder ob es dann ohnehin zu spät ankommen würde, so dass es dann nur verworfen wird. Diese Frage soll das Zeitmodell beantworten.

#### 3.2 Die Rolle der Zeit in multimedialen Streams

Zeit tritt in vier verschiedenen Rollen in multimedialen Datenströmen in Erscheinung.

Die *zeitliche Ordnung* beschreibt die Reihenfolge der Medienobjekte in einem multimedialen Dokument. Zwei Medienobjekte stehen in der *bevor* Relation, wenn das erste Medienobjekt zeitlich vor dem zweiten auftritt.

Die *Rechtzeitigkeit* ist für multimediale Echtzeitanwendungen besonders wichtig. Kommt ein Stromobjekt zu zeitig bei der Senke an, so muss es zwischengespeichert werden. Wenn es dagegen zu spät ankommt, so ist es für die echtzeitgebundene Anwendung meist nutzlos. In beiden Fällen liegt ein zeitlicher Fehler vor.

Meist sind die medialen Datenströme innerhalb eines multimedialen Datenstromes zeitlich voneinander abhängig. Die *Synchronität* beschreibt diese Abhängigkeit. So sind z.B. Ton und Bild eines Filmes voneinander abhängig und müssen synchron abgespielt werden.

Die vierte Rolle ist die *Periodizität*. Meist treten Medienobjekte in medialen Dokumenten periodisch auf. Bei einem Film wird z.B. alle 1/25 Sekunde ein Bild dargestellt.

Für kooperatives Multimedia-Streaming sind die zeitliche Ordnung sowie die Rechtzeitigkeit in besonderem Maße wichtig.

## 4 Zeitmodelle

### 4.1 Was ist ein Zeitmodell?

In [Süßkraut 2003] wird das Zeitmodell folgendermaßen kurz beschrieben: “Das Zeitmodell soll dazu beitragen, eine möglichst ressourcenschonende Fehlertoleranz zu implementieren.”

Ein Zeitmodell enthält Objekte, Ereignisse, Zeitpunkte der Ereignisse, zeitliche Beziehungen der Zeitpunkte, Zeitintervalle und Zeitspannen. Objekte sind z.B. die Medienobjekte, aber auch die Prozessoren. Ein Ereignis ist z.B. das Eintreffen eines Medienobjektes an einem Prozessor. Wichtig für die Fehlersemantik ist der Zeitpunkt, an dem die Ereignisse statt finden. Die zeitlichen Beziehungen der Zeitpunkte regeln die Reihenfolge der Medienobjekte. Anstelle oder als Ergänzung der Zeitpunkte können auch Zeitintervalle und Zeitspannen verwendet werden.

“Aufgabe eines Zeitmodells im Rahmen einer adaptiven Fehlersemantik für verteilte multimediale Anwendungssysteme ist es, zeitbezogene Kontextinformationen bereit zu stellen, die eine effiziente und ökonomische Reaktion auf Kommunikationsfehler ermöglichen.” [Kühnhauser und Süßkraut 2003]

### 4.2 Klassen von Zeitmodellen

Zeitmodelle lassen sich aufgrund ihrer atomaren Zeiteinheit in drei Klassen einteilen: zeitpunktbasierte, intervallbasierte und hybride Modelle.

In *zeitpunktbasierten Modellen* ist der Zeitpunkt die atomare Einheit. Es gibt maximal drei binäre, elementare Relationen in denen die Zeitpunkte stehen können:

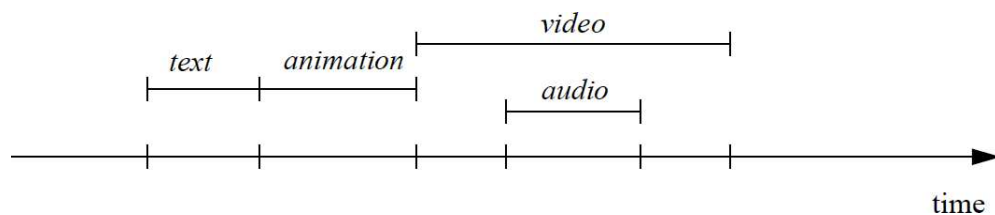
$<$ ,  $=$ ,  $>$ . Aus ihnen lassen sich  $2^3 = 8$  disjunkte, binäre Relationen bilden. Ein Beispiel ist  $\leq$ , das sich aus  $<$  und  $=$  konstruieren lässt. Zeitpunktbasierte Modelle sind z.B. das Time Line Modell und das Temporal Point Net.

*Intervallbasierte Modelle* können bis zu 10 weitere binäre, elementare Relationen als atomare Einheiten beinhalten. Es handelt sich dabei um Relationen wie z.B. treffen, überlappen und während. Beispiele für diese Klasse sind Path expressions und das MHEG Zeitmodell.

*Hybride Modelle* sind eine Kombination aus zeitpunkt- und intervallbasierten Modellen. Die atomaren Einheiten sind also Zeitpunkte und Intervalle. Das Timed Petri-Net ist z.B. ein hybrides Modell.

### 4.3 Zeitmodelle kurz vorgestellt

#### 4.3.1 Time Line Modell

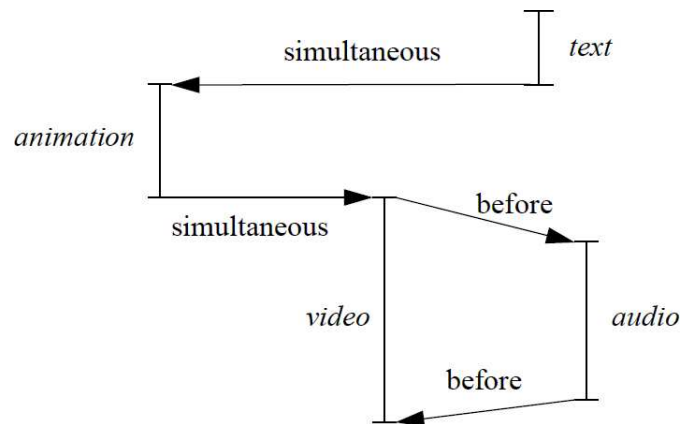


**Abb. 1:** *Time Line Modell (Wahl und Rothermel [1994])*

Das Time Line Modell ist zeitpunktbasiert. Alle Ereignisse sind entlang einer Zeitachse total geordnet (Abbildung 1). Das bedeutet, dass nur die drei elementaren Relationen  $<$ ,  $=$  und  $>$  möglich sind.

#### 4.3.2 Temporal Point Net

Ebenfalls zeitpunktbasiert ist das Temporal Point Net (Abbildung 2). Im Gegensatz zum Time Line Modell befinden sich die Zeitpunkte allerdings nicht entlang einer Zeitachse, sondern in einer Ebene. Dadurch existieren nicht nur die drei elementaren Relationen, sondern auch die daraus konstruierten, disjunkten Relationen, wodurch dieses Zeitmodell eine höhere Flexibilität erreicht.



**Abb. 2:** *Temporal Point Net* (Wahl und Rothermel [1994])

#### 4.3.3 Timed Petri Net

Das Timed Petri Net (Abbildung 3) ist ein hybrides Zeitmodell. Die Plätze repräsentieren die Intervalle und die Transitionen die Relationen. Die Petri-Netze müssen konfliktfrei sein. Dieses Modell ist laut Süßkraut [2003] genau so mächtig wie das Temporal Point Net. Die Modellierung mittels Petri-Netzen ist allerdings aufwändiger als bei den beiden vorangegangenen Modellen.

#### 4.3.4 Path expressions

Ein intervallbasiertes Zeitmodell ist Path expressions. Laut Wahl und Rothermel [1994] besitzt dieses Zeitmodell drei Relationen: *sequence*, *parallel-first* und *parallel-last* (Abbildung 4).

Zwei Intervalle stehen genau dann in der Relation *sequence*, wenn sie nacheinander verarbeitet werden, also wenn das zweite erst beginnt, nachdem das erste beendet ist. *Parallel-first* und *parallel-last* modellieren die synchrone Verarbeitung zweier Intervalle. Bei *parallel-first* endet die Verarbeitung, sobald eines der beiden Intervalle beendet ist, während sie bei *parallel-last* so lange andauert, bis beide beendet sind.

Laut Bertino und Ferrari [1998] gibt es drei weitere Relationen: *Selective* verarbeitet eines von zwei Intervallen, wobei die Bedingung, welches gewählt wird, außerhalb des Modells liegt. *Repetition* wiederholt ein Intervall. *Concurrency* spielt ein Intervall mehrfach gleichzeitig ab.

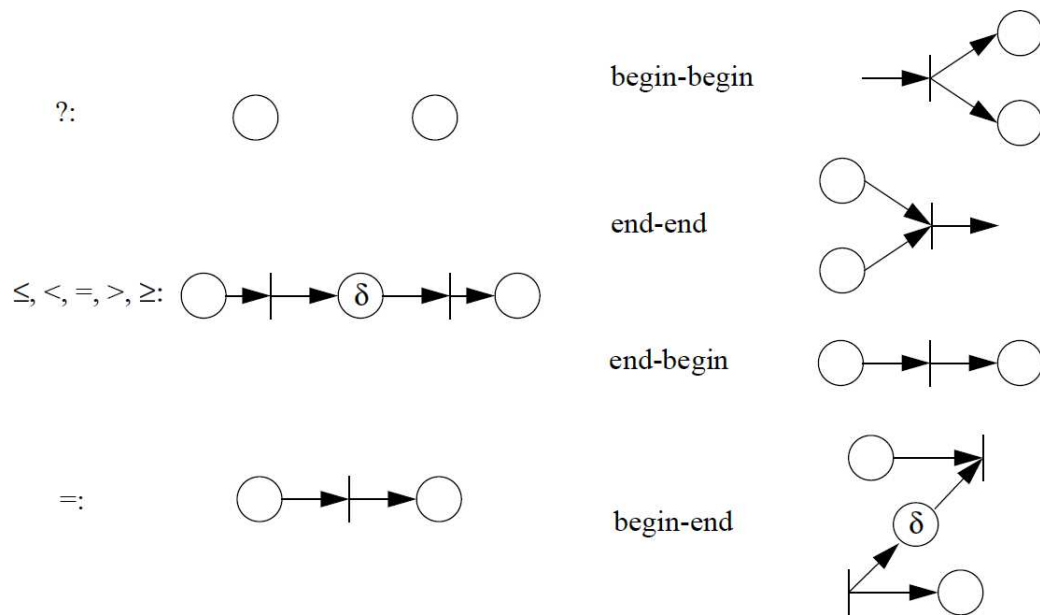


Abb. 3: *Timed Petri Net (Wahl und Rothermel [1994])*

#### 4.3.5 Multimedia Hypermedia Expert Group (MHEG)

Auch das MHEG Zeitmodell (Abbildung 5) ist intervallbasiert. Es stammt von der Multimedia Hypermedia Expert Group und definiert die beiden Relationen *sequence* und *parallel*. *Sequence* hat die gleiche Funktion, wie bei Path expressions. *Parallel* ist eine Relation zwischen zwei Intervallen, die parallel ablaufen.

#### 4.3.6 Fazit

In [Wahl und Rothermel 1994] werden die Zeitmodelle anhand ihrer mathematischen Ausdruckskraft beurteilt, wodurch die Autoren zu dem Schluß kommen, dass intervallbasierte Zeitmodelle besser geeignet sind, als zeitpunktbasierte.

Süßkraut [2003] stellt Fragen anhand der Rolle der Zeit auf und untersucht wie gut die Zeitmodelle diese Fragen beantworten können. Er schlußfolgert, dass zeitpunktbasierte Modelle besser geeignet sind und hat ein eigenes konstruiert. Dieses wird im folgenden Abschnitt vorgestellt.



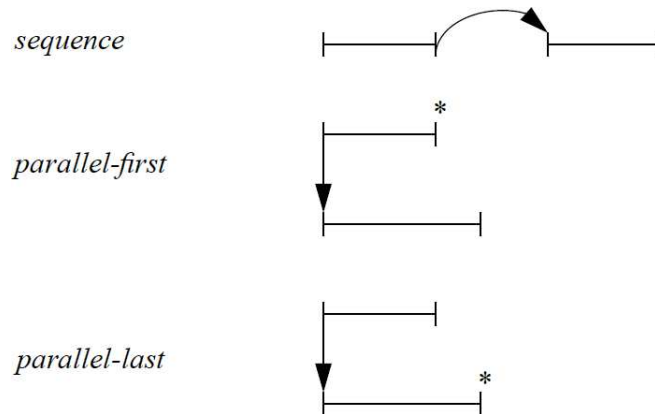


Abb. 4: Path expressions (Wahl und Rothermel [1994])

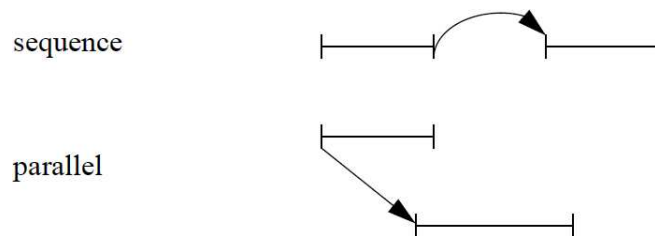


Abb. 5: MHEG Zeitmodell (Wahl und Rothermel [1994])

#### 4.4 Ilmenauer Zeitmodell nach Süßkraut [2003]

##### 4.4.1 Bestandteile des Zeitmodells

Das Zeitmodell  $Z$  ist ein 22-Tupel:

$$Z = (D, M, DS, S, B, P, R, T, v, a, dokument, strom, mo, quelle, senke, eze, t_M, o_E, o_A, t_S, t_G, t_L)$$

Es besteht aus Mengen und Abbildungen. Die Mengen modellieren die Umgebung des Zeitmodells:

- $D$  ist die Menge der medialen Dokumente.
- $M$  ist die Menge der Medienobjekte.
- $DS$  ist die Menge der medialen Datenströme.
- $S$  ist die Menge der Stromobjekte.
- $B$  ist die Menge der Beobachtungspunkte an einem Prozessor.

- $P$  ist die Menge der Prozessoren der multimedialen Anwendung.
- $R$  ist die Menge der Rechnerknoten auf denen die multimediale Anwendung ausgeführt wird.
- $T = \mathbb{R}$  ist die Zeit. Sie wird durch die Menge der reellen Zahlen modelliert. Negative Zeiten sind für relative Angaben zulässig.

$v \in \mathbb{R}$  ist die einzige Variable. Sie ist der Geschwindigkeitsfaktor und beschreibt das Verhältnis der Verarbeitungsgeschwindigkeit des Datenstroms zu der durch das transportierte mediale Dokument definierten Geschwindigkeit.

Die übrigen Tupелеlemente sind Abbildungen, die vor allem die zeitlichen Beziehungen zwischen den oben aufgeführten Mengen beschreiben:

- $a : P \rightarrow R$  bildet jeden Prozessor auf denjenigen Rechnerknoten ab, auf dem er ausgeführt wird.
- $dokument : M \rightarrow D$  bildet jedes Medienobjekt auf sein mediales Dokument ab.
- $strom : S \rightarrow DS$  bildet jedes Stromobjekt auf seinen medialen Datenstrom ab.
- $mo : S \rightarrow M$  bildet jedes Stromobjekt auf das von ihm transportierte Medienobjekt ab.
- $quelle : DS \rightarrow P$  bildet jeden medialen Datenstrom auf seinen erzeugenden Prozessor (Quelle) ab.
- $senke : DS \rightarrow P$  bildet jeden medialen Datenstrom auf seinen konsumierenden Prozessor (Senke) ab.
- $eze : DS \rightarrow T$  ordnet jedem medialen Datenstrom eine Ende-zu-Ende-Verbindungszeit zu.
- $t_M : M \rightarrow T$  weist jedem Medienobjekt einen Medienzeitstempel ähnlich dem Zeitstempel im RTP-Kopf zu.
- $o_E : P \times M \rightarrow T$  beschreibt die zeitliche Ordnung, in der ein Prozessor Medienobjekte erwartet.

- $o_A : P \times M \rightarrow T$  beschreibt die zeitliche Ordnung, in der ein Prozessor Medienobjekte ausgibt.
- $t_S : P \times B \times S \rightarrow T$  weist jedem Stromobjekt den Zeitpunkt zu, zu dem es einen Beobachtungspunkt an einem Prozessor passiert hat.
- $t_G : DS \rightarrow T$  bildet jeden medialen Datenstrom auf seine stromglobale logische Uhr ab.
- $t_L : R \rightarrow T$  sind die lokalen physikalischen Uhren der einzelnen Rechnerknoten.

In den folgenden Abschnitten werden die Bestandteile definiert und an Beispielen erklärt.

#### 4.4.2 Prozessoren und Rechnerknoten

**Definition 4.1** Eine verteilte multimediale Anwendung besteht aus mindestens zwei Softwarekomponenten, im Folgenden Prozessoren genannt. Prozessoren erzeugen, verändern und konsumieren Medienobjekte.

Jeder Prozessor verfügt über eine beliebige Anzahl von Ein- und Ausgängen, über welche die Medienobjekte ein- bzw. von ihm ausgegeben werden. Ein Prozessor besitzt jedoch zumindest einen Ein- oder Ausgang. Jeder Ausgang eines Prozessors ist mit mindestens einem Eingang eines anderen Prozessors durch einen logischen Übertragungskanal verbunden. Jeder Eingang ist mit genau einem Ausgang eines anderen Prozessors verbunden.

Die Menge  $P$  ist die Gesamtheit aller Prozessoren einer verteilten multimedialen Anwendung.

Ein erzeugender Prozessor ist z.B. eine Videoquelle, die von einer Kamera aufgenommene Videobilder auf ihrem Ausgang ausgibt. Eine Datenbanksenke, die ein Video auf einer Festplatte speichert, ist ein konsumierender Prozessor. Ein Stereo-zu-Mono-Mixer ist ein verändernder Prozessor. Es gibt aber auch Prozessoren, die

verändern und konsumieren. Wird z.B. ein Untertitel in ein Video eingebunden, so wird das Untertitelobjekt konsumiert und das Videoobjekt verändert.

**Definition 4.2** Eine multimediale Anwendung wird durch einen gerichteten Graphen  $G = (V, E)$  modelliert. Wobei  $V = P$  die Menge der Prozessoren und  $E \subseteq P \times P$  die Menge der logischen Übertragungskanäle zwischen den Ein- und Ausgängen der Prozessoren ist. Der Graph  $G$  ist kreisfrei.

**Definition 4.3** Die Menge  $R$  ist die Menge aller Rechner auf denen Prozessoren einer verteilten multimedialen Anwendung ausgeführt werden.

Rechnerknoten können ebenso Datenbankserver mit mehreren Prozessoren, wie auch gewöhnliche Arbeitsplatzrechner, Handys, PDAs oder digitale Videorecorder sein.

**Definition 4.4** Die Funktion  $a : P \rightarrow R$  bildet jeden Prozessor einer verteilten multimedialen Anwendung eindeutig auf denjenigen Rechner ab, auf dem dieser Prozessor ausgeführt wird.

#### 4.4.3 Medienobjekte und Stromobjekte

**Definition 4.5** Die Menge  $M$  ist die Menge aller Medienobjekte, die in einer verteilten multimedialen Anwendung verarbeitet werden. Jedes Medienobjekt ist Bestandteil genau eines medialen Dokumentes. Innerhalb eines medialen Dokumentes sind alle enthaltenen Medienobjekte über die Zeit geordnet.

Die Menge  $D$  ist die Menge der medialen Dokumente, welche eine verteilte multimediale Anwendung verarbeitet.

Die Funktion  $dokument : M \rightarrow D$  bildet jedes Medienobjekt eindeutig auf sein mediales Dokument ab.

Alle Medienobjekte eines medialen Dokumentes sind vom gleichen Medientyp, wie Audio, Video oder Text. Ein Textobjekt kann beispielsweise aus einer Textzeile eines Untertiteltextes bestehen. Unter Videoobjekten werden im Allgemeinen die Einzelbilder eines Videofilmes verstanden.

**Definition 4.6** Jedes Medienobjekt wird durch ein Stromobjekt transportiert.  $S$  ist die Menge der Stromobjekte einer verteilten multimedialen Anwendung.

Die Funktion  $mo : S \rightarrow M$  ordnet jedem Stromobjekt das von ihm transportierte Medienobjekt eindeutig zu.

**Definition 4.7** Ein medialer Datenstrom transportiert ein mediales Dokument durch eine verteilte multimediale Anwendung.  $DS$  ist die Menge der medialen Datenströme einer verteilten multimedialen Anwendung.

Jedem Stromobjekt wird mittels der Funktion  $strom : S \rightarrow DS$  sein medialer Datenstrom zugeordnet.

Ein Medienobjekt kann durchaus mehrfach in der Anwendung verarbeitet werden. Beispielsweise, wenn ein Videodokument wiederholt wiedergegeben wird. Es wird aber in jedem Fall vom selben medialen Datenstrom transportiert.

**Definition 4.8** Alle Medienobjekte eines von einem medialen Datenstrom  $d \in DS$  transportierten Dokumentes werden an dem Prozessor  $quelle(d)$  erzeugt. Sie alle passieren die gleichen Prozessoren auf dem gleichen Weg im Graphen über die Prozessoren, bis sie schließlich von dem Prozessor  $senke(d)$  konsumiert werden.

Das Stromobjekt  $s \in S$ , welches das vom Prozessor  $q = quelle(strom(s))$  erzeugte Medienobjekt  $mo(s)$  transportiert, wird ebenfalls am Ausgang von  $q$  erzeugt. Die Zuordnung zwischen dem Medienobjekt und dem Stromobjekt  $s$  bleibt bis zu dessen Senke  $senke(strom(s))$  erhalten.

**Definition 4.9** Die Funktion  $eze : DS \rightarrow T$  ordnet jedem Datenstrom seine Ende-zu-Ende-Verbindungszeit zu. Das ist die Zeit, in der ein Medienobjekt zwischen dem Ausgang seines Erzeugers und dem Eingang seines Konsumenten verarbeitet werden muss. Wird ein Medienobjekt schneller verarbeitet, so ist es zu früh und muss zwischengespeichert werden. Wird es langsamer verarbeitet, ist es zu spät und stellt einen Zeitfehler dar.

#### 4.4.4 Fünf Erscheinungsformen der Zeit

**Definition 4.10** Die Medienzeit definiert die zeitliche Ordnung und den zeitlichen Abstand der Medienobjekte eines medialen Dokumentes. Sie wird vom Erzeuger definiert.

Die Medienzeitfunktion  $t_M : M \rightarrow T$  weist jedem Medienobjekt einen Zeitpunkt zu.

Die zeitliche Ordnung der Medienobjekte wird durch die Ordnung der Medienzeitstempel auf der Zeitachse definiert:

**Definition 4.11** Vorausgesetzt  $m, m' \in M$  und es gilt:

$$\text{dokument}(m) = \text{dokument}(m')$$

dann gilt auch:

$$m < m' \Leftrightarrow t_M(m) < t_M(m')$$

Der Medienabstand, also der zeitliche Abstand zwischen zwei Medienobjekten in einem medialen Dokument, ist wie folgt definiert:

**Definition 4.12** Vorausgesetzt  $m, m' \in M$  und es gilt:

$$\text{dokument}(m) = \text{dokument}(m')$$

dann gilt auch:

$$\text{medienabstand}(m, m') = t_M(m) - t_M(m')$$

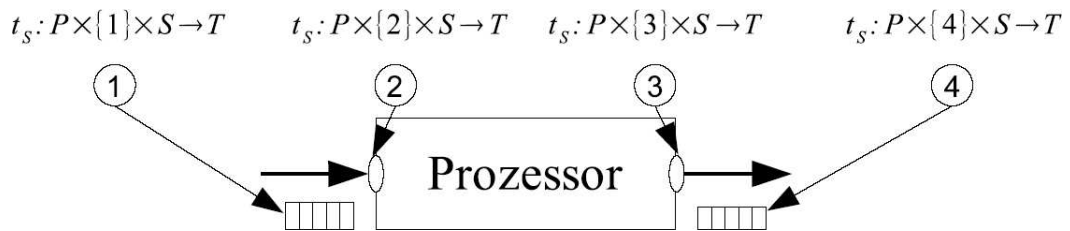
Prozessoren, welche unter Echtzeitbedingungen arbeiten, erwarten Medienobjekte in einer definierten Reihenfolge und in einem definierten zeitlichen Abstand am Eingang. Analog geben sie die Medienobjekte in einer definierten Reihenfolge und einem definierten zeitlichen Abstand zueinander aus.

Sind zwei Prozessoren  $p, p' \in P$  durch einen Übertragungskanal  $(p, p')$  verbunden, so erwartet Prozessor  $p'$  an seinem Eingang die Medienobjekte in genau der gleichen Reihenfolge und dem gleichen Abstand in der bzw. in dem sie Prozessor  $p$  ausgibt. Zwischen dem Eingang und dem Ausgang eines Prozessors muss die Reihenfolge und der Abstand der Medienobjekte nicht erhalten bleiben. Dies ist z.B. bei der MPEG-Codierung der Fall.

**Definition 4.13** Jeder Prozessor definiert für seine Ein- und Ausgänge eine Ein- bzw. Ausgangsordnung über die Medienobjekte eines medialen Dokumentes. Diese Ordnungen werden als Medienordnung bezeichnet.

Die Funktion der Eingangsordnung  $o_E : P \times M \rightarrow T$  weist jedem Medienobjekt und jedem Prozessor einen Zeitstempel zu, der die Reihenfolge und den Abstand des Medienobjekts zu den anderen Medienobjekten des medialen Dokumentes, die bzw. den der Prozessor an seinem Eingang erwartet, definiert. Dieser Zeitstempel ist relativ zum Medienzeitstempel des Medienobjekts. Kann ein Medienobjekt einen Prozessor nicht erreichen, so ist die Eingangsordnung für dieses Tupel aus Prozessor und Medienobjekt nicht definiert.

Die Funktion der Ausgangsordnung  $o_A : P \times M \rightarrow T$  ist analog für den Ausgang eines jeden Prozessors definiert.



**Abb. 6:** Strömungszeitstempel an den vier Beobachtungspunkten eines Prozessors

Wie in Abbildung 6 zu erkennen, befinden sich in den Ein- und Ausgängen der Prozessoren die Puffer der Flusskontrolle. Sie gehören weder zum Prozessor noch zu den Übertragungskanälen. Deshalb reicht es nicht aus, einfach am Eingang und am Ausgang eines Prozessors die Zeit abzulesen.

**Definition 4.14** Die Menge  $B = \{1, 2, 3, 4\}$  modelliert vier Beobachtungspunkte an einem Prozessor  $p \in P$  für ein bestimmtes Stromobjekt  $s \in S$ . Beobachtungspunkt  $b = 1$  ist der Ausgang des Übertragungskanals vor dem Eingangspuffer der Flusskontrolle an dem Eingang, an welchem  $s$  bei  $p$  eintritt. Beobachtungspunkt  $b = 2$  bezeichnet den Übergang vom Eingangspuffer der Flusskontrolle in den Prozessor selbst. Analog dazu sind die Beobachtungspunkte  $b = 3$  und  $b = 4$  für den Ausgangspuffer definiert.



**Definition 4.15** Die Strömungszeit markiert das Passieren eines Beobachtungspunktes an einem Prozessor durch ein Stromobjekt.

Die Funktion  $t_S : P \times B \times S \rightarrow T$  weist jedem Stromobjekt den Zeitpunkt zu, zu dem es einen Beobachtungspunkt an einem Prozessor passiert hat. Dieser Zeitpunkt ist undefiniert, wenn das Stromobjekt diesen Beobachtungspunkt und Prozessor noch nicht passiert hat bzw. nie passieren wird.

Da die Strömungszeitstempel von unterschiedlichen Beobachtungspunkten und Prozessoren für ein Stromobjekt vergleichbar sein müssen, benötigt man eine globale Uhr:

**Definition 4.16** Die stromglobale logische Uhr  $t_G : DS \rightarrow T$  bildet jeden medialen Datenstrom auf eine für alle Prozessoren, die von diesem Datenstrom transportierte Medienobjekte verarbeiten, sichtbare logische Uhr ab. Die Uhrzeiten der einzelnen stromglobalen Uhren wachsen streng monoton.

**Definition 4.17** Die Funktion  $t_L : R \rightarrow T$  bildet jeden Rechnerknoten auf seine lokale Uhr ab.

Jede lokale Rechneruhr für sich betrachtet wächst streng monoton.

**Definition 4.18** Der Faktor der Verarbeitungsgeschwindigkeit  $v \in \mathbb{R}$  ist definiert als das Verhältnis der Verarbeitungsgeschwindigkeit eines medialen Datenstroms  $v_{ds}$  (mit  $ds \in DS$ , als besagter medialer Datenstrom) zur Verarbeitungsgeschwindigkeit des von  $ds$  transportierten medialen Dokumentes  $v_d$  (mit  $d \in D$  mediales Dokument und  $\forall s \in S$  mit  $strom(s) = ds$  muss gelten:  $dokument(mo(s)) = d$ ):

$$v = \frac{v_{ds}}{v_d}$$

Ist die Geschwindigkeit  $v_d$  eines Filmes z.B. 25 Bilder/s und er wird mit  $v_{ds} = 50$  Bildern/s wiedergegeben, so ist die Geschwindigkeit  $v = 2$ .

## 5 Fazit

Zeitmodelle sind ein wichtiges Instrument für das multimediale Streaming. Sie helfen der Fehlersemantik besser zu reagieren und dadurch Ressourcen zu sparen, indem nur noch die fehlenden oder fehlerhaften Pakete nachgefragt werden, die noch rechtzeitig ankommen.

Es existieren eine Reihe von Zeitmodellen, die aber alle nicht sehr ausführlich sind. Süßkraut [2003] hat ein zeitpunktbasiertes Zeitmodell entwickelt, das sehr umfangreich definiert ist. Es mißt die Zeit für jedes Stromobjekt an jedem Prozessor an vier verschiedenen Stellen und kann so sehr genaue Ergebnisse liefern. In [Süßkraut 2003] wurde das Zeitmodell praktisch getestet und die Ergebnisse zeigen, dass die theoretisch erwartete bessere Ausnutzung der Ressourcen auch in der Praxis auftrat. Das Modell ist damit eine gute Grundlage für die Fehlerbehandlung bei kooperativem Multimedia-Streaming.

## Literatur

**Bertino und Ferrari 1998** BERTINO, Elisa ; FERRARI, Elena: Temporal Synchronization Models for Multimedia Data. In: *IEEE Transactions on knowledge and data engineering*, July/August 1998

**Kühnhauser und Süßkraut 2003** KÜHNHAUSER, Winfried ; SÜSSKRAUT, Martin: *Zur Rolle der Zeit in verteilten multimedialen Systemen*. TU Ilmenau, 2003

**Süßkraut 2003** SÜSSKRAUT, Martin: *Die Rolle der Zeit in Fehlersemantiken für Kommunikationsmodelle multimedialer Anwendungen*, TU Ilmenau, Diplomarbeit, 2003

**Wahl und Rothermel 1994** WAHL, Thomas ; ROTHERMEL, Kurt: *Representing Time in Multimedia-Systems*. University of Stuttgart, 1994