

# Prefetching für verteilte, hypermediale Lernanwendungen unter Berücksichtigung der Lerndauer

Jan Tamm

Jan.Tamm@uni-rostock.de

**Abstract:** Der Einsatz von Prefetchingverfahren für verteilte, hypermediale Lernanwendungen ermöglicht es, zeitliche Verzögerungen während der Navigation des Lernenden durch das Hypermedium nachhaltig zu reduzieren. In diesem Papier wird ein Prefetchingansatz dargestellt, mit welchem gezielt die Lerndauer minimiert werden kann. Darauf basierend können die Lerninhalte innerhalb eines performanceorientierten Entwurfs segmentiert und sequenziert werden. Es werden für den Entwurf von Hypermedien verteilter Lernanwendungen ein Modell und ein darauf basierendes Optimierungsproblem formuliert, Ergebnisse theoretischer Untersuchungen des Problems dargestellt und die Unterstützung von Entwurfsentscheidungen diskutiert.

## 1 Einleitung

Prefetchingverfahren werden dazu eingesetzt, vom Nutzer wahrgenommene Verzögerungen während der Navigation durch verteilte Hypermedien zu reduzieren. Nach empirischen Studien zu einzelnen Verfahren ist eine Halbierung der Verzögerungszeiten gegenüber dem Einsatz klassischer, das heißt passiver, Cacheverfahren möglich [Dav04], [BKK04], [Ang03]. Während der Dauer der Nutzung von Lernobjekten ist es sinnvoll, vom Nutzer möglicherweise zukünftig verwendete Objekte spekulativ in den lokalen Speicher (Cache) einer Präsentationskomponente (zum Beispiel Webbrowser) im Voraus zu laden. Im Voraus laden heißt, dass Lernobjekte bevor vom Nutzer angefragt über ein algorithmisches Entscheidungsverfahren angefordert werden.

Seit Mitte der 90er Jahre sind eine Vielzahl von Prefetchingansätzen für webbasierte Anwendungen entstanden. Sie sind aber nicht oder nur teilweise auf die speziellen Erfordernisse verteilter hypermedialer Lernanwendungen zugeschnitten. Mit dem Entwurf von Hypermedien für Lernanwendungen sind die Lerninhalte zu segmentieren und zu sequenzieren. Die Entwurfsentscheidungen basieren vorrangig auf der didaktischen Konzeption der Anwendung, den angenommenen Lernstilen und zu erwartenden Lernstrategien der Lernenden. Sie werden unter Berücksichtigung zu erwartender Lerngeschwindigkeit und angestrebter Dauer des zu unterstützenden Lernprozesses getroffen ([NHHM<sup>+</sup>04], S.100ff.). Letztendlich werden die Leistung und das Anwendungspotential einer Lernanwendung auch an der Lerndauer gemessen - Zeit, die vom Lernenden benötigt wird, die spezifizierten Lernziele zu erreichen. Die Lerndauer wird häufig nachhaltig negativ von Verzögerungen beim Zugriff auf Lerninhalte in verteilten Lernanwendungen beeinflusst ([Röl03],

S. 323). Verzögerungen unmittelbar folgend auf Navigationsentscheidungen der Lernenden wirken störend, erzwingen Lernpausen, geben künstlich einen Lernrhythmus vor und können die Lerndauer ungewollt verlängern. Je nach Lernszenario und daraus resultierenden Rahmenbedingungen ist es notwendig, Prefetchingverfahren einzusetzen, welche eine angemessene Lerndauer gewährleisten können. Es sind Segmentierungs- und Sequenzierungsentscheidungen des Hypermediums in Hinblick auf die mögliche Leistung eines Prefetchingverfahrens zu überdenken. Dafür fehlen Entwurfswerkzeuge, die einen performanceorientierten Entwurf ermöglichen.

## 2 Problemstellung

Gesucht ist ein Prefetchingverfahren, welches unter Berücksichtigung der Struktur eines Hypermediums, das heißt abhängig von der Segmentierung und Sequenzierung der Lerninhalte, eine minimale Lerndauer bei angenommener Nutzzeit je Segment garantieren kann.

Es wird angenommen, dass alle Segmente des Hypermediums zu beliebigen Zeitpunkten im lokalen Cache der Präsentationkomponente repliziert werden können, dass dessen Inhalte bzw. dessen Beschreibung über den Zeitablauf nicht variabel ist, dass der lokale Cache beliebig groß ist und dass die Größe der Segmente sowie die Bandbreite der Netzverbindung zwischen lokalem Cache und entfernten Ressourcen bekannt und konstant sind. Für jedes Segment kann eine Ladezeit angegeben werden, die notwendig ist, um es vollständig in den lokalen Cache zu laden. Die Annahmen beeinflussen die Praxisrelevanz der Problemstellung nur in sehr geringem Maße.

## 3 Navigationsbaum

Segmentierungs- und Sequenzierungsentscheidungen werden in Form eines Navigationsbaums  $B$  beschrieben. Der Baum besteht aus der Knotenmenge  $A$  mit  $n$  Knoten (den Segmenten des Hypermediums) und gerichteten Kanten (den navigationsbezogenen, die Segmente verknüpfenden Links). Für jeden Knoten  $a_i \in A$  existiert ein Tripel  $a_i \in A : a_i = \{u_i, l_i, o_p\}$ , wobei  $u_i$  die Nutzzeit und  $l_i$  die Ladezeit beschreibt. Es ist eine Zeitdauer von  $l_i$  Zeiteinheiten notwendig, um das Objekt  $o_p$  in den lokalen Cache vollständig zu laden, bevor es genutzt werden kann. Während der Nutzung von  $o_p \in a_i$  steht eine Dauer von  $u_i$  Zeiteinheiten zur Verfügung, andere Objekte als  $o_p$  in den lokalen Cache im Voraus zu laden. Die Objekte der Knoten entsprechen einem Segment des Hypermediums, welches ein oder mehrere Lernobjekte zusammenfasst, die alle gleichzeitig dem Lernenden zu präsentieren sind und von der Dauer  $u_i$  genutzt werden. Jeder Weg des Navigationsbaums vom Wurzelknoten zu einem Blatt beschreibt genau ein Navigationsweg entsprechend einem Lernweg oder auch Navigationsmuster. Identische Objekte, welche das selbe Segment des Mediums beschreiben, können in unterschiedlichen Knoten mehrfach auftreten. Solche Knoten werden zueinander redundante Knoten genannt und in Mengen, sogenannte Knotengruppen  $G_j$ , zusammen gefasst. Nur die Nutzzeit der zueinander redundanten Knoten

kann sich je nach Weg im Baum unterscheiden. Zueinander redundante Knoten beschreiben die Verwendung des selben Objekts auf unterschiedlichen Lernwegen. Ein Navigationsbaum mit  $n$  solchen knotendisjunkten, jeweils ein Element starken Knotengruppen ist ein spezieller Baum. Er wird nicht redundanter Navigationsbaum genannt.

Für jeden Knoten eines jeden Weges in  $B$  ist zu entscheiden, wie die Nutzzeit zum Laden im Voraus von Objekten anderer Knoten zu verwenden ist. Für jeden Knoten  $a_i \in A$  wird ein Vektor mit Entscheidungsvariablen  $x_i \in X$  angenommen. Jeder Vektor der Entscheidungsmatrix besteht aus genau so vielen Werten  $x_{ij}$  wie Knotengruppen in  $A$  existieren. Es ist über die Werte der  $x_{ij} \in X$  zu entscheiden. Sie beschreiben jeweils genau eine Prefetchingentscheidung: die Dauer des Ladens eines Objekts der Knotengruppe  $G_j$  während der Nutzung von  $a_i$ .

Es gelten folgende Bedingungen, aus welchen das Modell als ganzzahliges Programm formuliert werden kann.

Es kann nicht mehr Zeitdauer für Prefetching aufgewendet werden, als die Nutzzeit des Knotens beträgt.

$$\forall a_i \in A : u_i \geq \sum_{\forall G_j \in B} x_{ij} \quad (1)$$

Ein Knoten kann nicht länger im Voraus geladen werden, als seine Ladezeit beträgt. ( $Weg(B, a_1, a_i)$  liefert alle Knoten des Baums  $B$  auf dem Weg vom Wurzelknoten  $a_1$  zu  $a_i$ .)

$$\forall a_i \in A : \ell_i \geq \left( \sum_{\forall a_h \in Weg(B, a_1, a_i)} x_{hj} \right), \quad a_i \in G_j \quad (2)$$

Die Nutzzeit eines Knotens kann nicht dazu verwendet werden, Objekte anderer Knoten der zueinander redundante Knoten  $G_j$  bzw. sich selbst zu laden.

$$\forall x_{ij} \in X : x_{ij} = 0, \quad \text{falls } a_i \in G_j \quad (3)$$

Objekte der Knoten können nicht zu beliebig kleinen Teilen im Voraus geladen werden. Daraus ergibt sich die Ganzzahligkeit der Modellparameter  $u, \ell$  und Entscheidungsvariablen  $x$ .

$$\forall a_i \in A : u_i, \ell_i \in Z_0^+ \quad (4)$$

$$\forall x_{ij} \in X : x_{ij} \in Z_0^+ \quad (5)$$

Mit dieser Modellformulierung wird impliziert, dass Objekte entweder während der Nutzung anderer Objekte im Voraus zu laden sind oder ein Laden nach deren Anforderung durch den Nutzer erfolgt.

Zur einfacheren Formulierung des Optimierungsziels wird zusätzlich die Restladezeit  $r_i$  eines jeden Knotens betrachtet. Sie beschreibt die Zeitdauer, die noch notwendig ist, das

Objekt eines Knotens nach deren Anforderung durch den Nutzer vollständig in den lokalen Cache zu laden. Sie ist abhängig von vorangegangenen Prefetchingentscheidungen und stellt die zu erwartende zeitliche Verzögerung dar:

$$r_i = \ell_i - \left( \sum_{\forall a_h \in \text{Weg}(B, a_1, a_i)} x_{hj} \right), \quad a_i \in G_j \quad (6)$$

Es ergibt sich folgende Zielstellung, wobei  $\sigma$  eine Folge von Knoten eines Weges von der Wurzel zu einem Blatt des Baums, das heißt genau ein Navigationsmuster bzw. Lernweg, beschreibt:

$$\min \Rightarrow \max_{\forall \sigma \in B} \left[ \sum_{\forall a_i \in \sigma} (r_i + u_i) \right] \quad (7)$$

Gesucht ist die Menge von Prefetchingentscheidungen  $X$  bzw. der dazugehörige Optimalwert, so dass die höchstmögliche Lerndauer für beliebige durch den Baum beschriebene Lernwege minimal ist.

#### 4 Optimierungproblem *MiniMaxLerndauer*

Entsprechend der Modellformulierung der Zielstellung kann das Optimierungsproblem *MiniMaxLerndauer* formuliert werden.

Optimierungsproblem *MiniMaxLerndauer* (*MML*)

Gegeben ist ein Navigationsbaum  $B$ , inklusive Nutz-, Ladezeiten und Knotengruppen.

Gesucht ist eine Entscheidungsmatrix  $X$ , unter den oben formulierten Nebenbedingungen (1) bis (6), die dem Optimalwert der Zielstellung (7) entspricht.

Die gelieferte Entscheidungsmatrix enthält für den Zeitraum der Nutzung eines jeden Objekts Prefetchingentscheidungen. Setzt ein Prefetchingverfahren die Entscheidungen während der Laufzeit der Lernanwendung um, kann die dem Optimalwert entsprechende Lerndauer garantiert werden. Damit kann ein das Problem lösendes, algorithmisches Verfahren dazu eingesetzt werden, Entwurfsentscheidungen in Hinblick auf die angestrebte maximal zu erwartende Lerndauer während des Betriebs einer verteilten hypermedialen Lernanwendung zu bewerten.

Untersuchungen zur Lösung von *MML* ergeben einige für den praktischen Einsatz interessante Untersuchungsergebnisse:<sup>1</sup>

<sup>1</sup>Auf die Darstellung der Korrektheit der einzelnen Ergebnisse wird hier allein aus Platzgründen verzichtet.

1. *MML* ist für redundante Navigationsbäume *NP*-hart.

Es kann gezeigt werden, dass *3DM* polynomial reduzierbar auf das zu *MML* korrespondierende Entscheidungsproblem ist, wobei *3DM* für das *NP*-vollständige Entscheidungsproblem 3-Dimensionales-Matching steht.<sup>2</sup>

2. Auch die Approximation von *MML* bei einer Güte des Approximats besser als das Optimum zuzüglich einer beliebig großen Konstante ist *NP*-hart.

Es kann eine polynomiale Reduzierbarkeit von *3DM* auf das entsprechende Approximationsproblem gezeigt werden.

3. Für nicht redundante Navigationsbäume kann ein effizientes Lösungsverfahren  $MML_{opt}(B)$  angegeben werden.<sup>3</sup>

Es ist ein effizientes Verfahren bekannt, welches jeden Teilbaum des Navigationsbaums, inklusive des Baums selbst, in seine Teilbäume zerlegt und für jeden dieser Teilbäume kann effizient eine optimale Lösung erzeugt und die optimalen Teillösungen effizient zu einer optimalen Gesamtlösung zusammengeführt werden.

4. Für redundante und nicht redundante Navigationsbäume kann ein effizientes Approximationsverfahren  $MML_{app}(B)$  angegeben werden, welches eine Lösung liefert, dessen Wert nicht schlechter als die Summe aus Optimalwert und Fehler  $f \leq (\delta(n-1))^2$  ist, wobei  $\delta$  die Dauer einer Zeiteinheit angibt.

Unter der Annahme reellwertiger Entscheidungsvariablen der Entscheidungsmatrix  $Y$  statt  $X$  kann *MML* als lineares Programm (LP) formuliert werden. Dessen optimale Lösung kann mit standardisierten Verfahren effizient, das heißt mit polynomial beschränktem Aufwand, berechnet werden. Ein Abrunden der reelwertigen Entscheidungsvariablen der LP-Lösung auf den nächst ganzzahligen Wert führt zu einer approximativen Lösung, die den Nebenbedingungen von *MML* genügt. Der Rundungsfehler übersteigt nicht das angegebene  $f$ .<sup>4</sup>

5. Beliebig „schlechte“ Entscheidungsverfahren, die den Modellbedingungen genügen, können nie eine Lerndauer schlechter als das 2-fache des Bestmöglichen liefern.

Die Summe aus Nutz- und Ladezeiten aller Knoten einer beliebigen Sequenz des Baums kann entsprechend der Modellformulierung nicht überschritten werden.

6. Es sind eine Vielzahl von heuristischen Regeln zum Generieren von Segmentierungs- und Sequenzierungsentscheidungen entwickelt worden.

Das Ergebnis der optimierenden Verfahren, Prefetchingentscheidungen und der Optimalwert, sind nicht ausreichend, um Handlungsempfehlungen für den Entwurf direkt ableiten zu können. Deshalb sind weitere Verfahren notwendig, die Empfehlungen zur Segmentierung und Sequenzierung aus der mindestens möglichen Leistung eines Prefetchingverfahrens ableiten.

---

<sup>2</sup>Vgl. zu  $3DM \in NP$ -vollständig [Ata99], S.28-3

<sup>3</sup>Der Rechenaufwand wächst asymptotisch mit  $O(n^3)$ .

<sup>4</sup>Das Ergebnis steht nicht im Widerspruch zu 2.

## 5 Unterstützung von Entwurfsentscheidungen

Die beiden effizient berechenbaren Verfahren  $MM\text{Lapp}(B)$  und  $MM\text{Lopt}(B)$  sind verwendbar, um die Performance einer hypermedialen Lernanwendung unter Prefetching-Aspekten zu beurteilen. Die Verfahren liefern zu erwartende Lerndauern für jeden Lernweg und die zeitliche Verzögerung während der Navigation durch das Hypermedium unter Annahme des Einsatzes eines Prefetchingverfahrens, welches entsprechend  $MM\text{Lapp}(B)$  bzw.  $MM\text{Lopt}(B)$  Lernobjekte des Hypermediums im Voraus anfordert. Mit dem Ergebnis können Entwurfsentscheidungen am spezifizierten Antwortzeitverhalten, insbesondere den festgelegten Lerndauern für einzelne Lerninhalte, gemessen werden. Gegebenfalls sind Segmentierungs- und Sequenzierungsentscheidungen zu überarbeiten.

Die Größe bzw. die in den Segmenten enthaltenen Lernobjekte beeinflussen Lade- und Nutzzeiten. Die Reihenfolge der Segmente, festgelegt mit Sequenzierungsentscheidungen, beeinflusst die verfügbare Nutzzeit, in welcher dieses im Voraus geladen werden kann. In folgender Abbildung ist ein Ausschnitt des iterativen Entwurfsprozesses mit den Aktivitäten Segmentierung, Sequenzierung sowie Abschätzen der zu erwartenden Performance, basierend auf den Optimierungsverfahren, dargestellt.

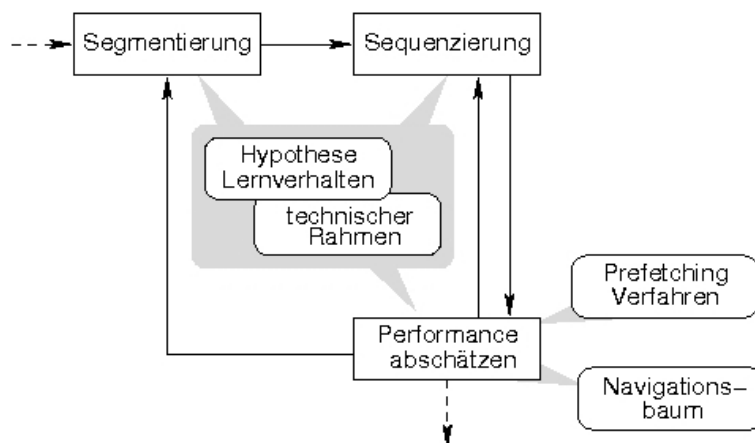


Abbildung 1: Ausschnitt der Aktivitäten eines performanceorientierten Entwurfs unter Berücksichtigung von Prefetching. Je nach verwendetem Prefetchingverfahren sind bestimmte Verzögerungszeiten zu erwarten. Ergebnis ist ein Navigationsbaum, mit welchem unter Einsatz des Verfahrens ein bestimmtes Antwortzeitverhalten garantiert werden kann.

Angenommene Nutz- und Ladezeiten können adaptiv gestaltet werden, denn je nach zu erwartendem Lernstil, Lernstrategie und Lernziel der Lernenden sind unterschiedliche Nutzzeiten anzunehmen [Alo04]. Ladezeiten sind abhängig von einer variablen Bandbreite der Verbindung zwischen lokaler Präsentationskomponente bzw. Cache und entfernten Ressourcen. Die Parameter des Modells, das heißt Nutz- und Ladezeit der einzelnen Knoten, können dynamisch über den Zeitablauf angepasst werden. Es sind personalisierte Navigationsbäume für Lernende, Lerngruppen, aber auch Lehrende genauso möglich wie

Bäume, die an die technischen Rahmenbedingungen angepasst sind. Mit solchen angepassten Bäumen lässt sich der Entwurf adaptiver Lernanwendungen genauso unterstützen wie der Betrieb von adaptiven Lernanwendungen, in welchen im Zeitablauf dynamische Anpassungen notwendig sind.

## 6 Angrenzende Arbeiten

*Khan* und *Tao* weisen für webbasierte hypermediale Anwendungen darauf hin, dass Entwurfsentscheidungen bezüglich der Eigenschaften des Hypermediums unter Einsatz von Prefetching gezielt zu unterstützen sind [KT03]. Sie untersuchen die Leistung von Prefetchingverfahren in Abhängigkeit von der Struktur des Hypermediums. *Bulterman* beschreibt Probleme beim performanceorientierten Entwurf von verteilten, hypermedial strukturierten Präsentationen [Bul04]. Er verweist auf die Notwendigkeit des Einsatzes spezieller Entwurfswerkzeuge, die die Segmentierung und Sequenzierung von Präsentationssequenzen unterstützen. Er beschreibt ein Werkzeug, welches in Abhängigkeit von der Bandbreite der Netzverbindung für den Zugriff auf verteilte Ressourcen zu erwartende Verzögerungszeiten visualisiert. Prefetching spielt jedoch in seinem Ansatz keine Rolle. Diese Ergebnisse stimmen auch mit den Untersuchungen von *Lang* über die besondere Notwendigkeit von Werkzeugen für den performanceorientierten Entwurf von verteilten Hypermedien überein [Lan04]. *Lang* fordert die Entwicklung solcher Werkzeuge zur Unterstützung von adhoc-Entscheidungen im Entwurfsprozess von Hypermedien.

In der „Prefetching-Literatur“ werden nahezu ausschließlich anwendungsunabhängige, größtenteils webbasierte Prefetchingverfahren dargestellt. Mit diesen Verfahren werden besondere Anforderungen nur aus rein technischer Sicht berücksichtigt. Die gute Leistungsfähigkeit der Verfahren, ohne spezielle Kenntnisse über Nutzergruppen und Anwendungseigenschaften, wird immer wieder herausgestellt und spricht für die Güte der verwendeten stochastischen Verfahren zur Vorhersage des Navigationsverhaltens, ist aber auch umstritten, da mit Prefetching eine Vielzahl negativer Seiteneffekte hinsichtlich der Belastung von Netzwerkressourcen verknüpft wird ([CK02], [Dav02], S. 65ff., [VYK<sup>+</sup>02]). In den letzten Jahren zeichnet sich aber ein neuer Trend hin zu anwendungsabhängigen Verfahren ab. E-learning spielte aber bis jetzt keine Rolle.

Die Gesamtdauer einer Session, hier die Lerndauer, war bisher für Prefetching von geringer Bedeutung. Sie wurde nur in soweit berücksichtigt, dass die Reduktion von Wartezeiten auch dazu führt, dass die Dauer einer Session und damit die Lerndauer verkürzt wird. *Khan* und *Tao* verweisen zwar bereits in [KT01] schon darauf, dass Segmentierungs- und Sequenzierungsentscheidungen im Entwurf von Hypermedien direkten Einfluss auf die Dauer einer Session haben, ihre Erkenntnisse bezüglich dessen fanden aber keinen Eingang in die Konstruktion von Prefetchingverfahren, die explizit die Sessiondauer bei bekannten Nutz- und Ladezeiten der Segmente eines Hypermediums berücksichtigen.

## 7 Diskussion der Ergebnisse

Es ist möglich, Prefetchingverfahren so zu konstruieren, dass das Antwortzeitverhalten verteilter hypermedialer Lernanwendungen schon im Entwurf sinnvoll abgeschätzt werden kann. Es reicht dafür aus, das Hypermedium, mögliche Nutzmuster sowie zu erwartende Nutzzeiten zu kennen. Stochastische Vorhersagemodelle über das Navigationsverhalten sind im Gegensatz zu herkömmlichen Prefetchingverfahren nicht notwendig.<sup>5</sup> Diese ließen sich aber mit dem hier vorgestellten Ansatz verknüpfen und ermöglichen dann eine Leistungssteigerung über das garantierte Antwortzeitverhalten hinaus, führen aber nicht dazu, dass ein bestimmtes Antwortzeiten garantiert werden kann.

In dem hier vorgestellten Ansatz kann die Lerndauer explizit über das Optimierungsziel berücksichtigt werden. Andere Zielstellungen, die zum Beispiel auf einzelne ausgewählte Verzögerungszeiten oder die Gesamtheit aller Verzögerungszeiten des Baums abzielen, erscheinen sinnvoll, wurden auch untersucht und führen zu ähnlichen Ergebnissen. Für die Entwicklung von Lernanwendungen ist jedoch von herausragender Bedeutung, die Dauer des rechnergestützten Lernprozesses abzuschätzen sowie Wartezeiten im Verhältnis zu den Nutzzeiten der Lernobjekte zu bewerten.

Letztendlich muss es das Ziel sein, Werkzeuge zu konstruieren, die einen iterativen, häufig von adhoc-Entscheidungen geprägten, aber zielgerichteten Entwurf von verteilter Hypermedien in Lernanwendungen unterstützen. Es ist möglich, solche rechnergestützten Werkzeuge zu entwickeln. Erst wenn sie verfügbar sind, stellt Prefetching eine ernstzunehmende Option im Entwurf verteilter Lernanwendungen dar. Es können mit Prefetching Lernszenarien sinnvoll rechnerunterstützt werden, die von schlechten technischen Rahmenbedingungen geprägt sind. Bisher ungenutzte Netzwerkressourcen können anwendungsabhängig, zugunsten eines besseren Antwortzeitverhaltens genutzt werden. Von Lernenden wird häufig in Lernszenarien erwartet, dass sie Rechnernetzverbindungen mitfinanzieren. Contentanbieter sind dementsprechend verpflichtet, die von Lernenden zur Verfügung gestellten Ressourcen besonders effektiv zu nutzen.

## Literatur

- [Alo04] Hesham Alomyan. Individual Differences: Implications for Web-based Learning Design. *International Education Journal, Educational Research Conference 2003 Special Issue*, 4(4):188–195, 2004.
- [Ang03] Michael Angermann. Differences in Cost and Benefit of Prefetching in Circuit-Switched and Packet-Switched Networks. In *10th International Conference on Telecommunications IEEE ICT'2003*, Tahiti, Papeete, 2003.
- [Ata99] Mikhail J. Atallah. *Algorithms and Theory of Computation Handbook*. CRC Press LLC, London, 1999.

---

<sup>5</sup>Vgl. zu stochastischen Vorhersagemodellen für Prefetchingverfahren [Dav04]



- [BKK04] Christos Bouras, Agisilaos Konidaris und Dionysios Kostoulas. Predictive Prefetching on the Web and Its Potential Impact in the Wide Area. *World Wide Web archive*, 7(2):11–30, 2004.
- [Bul04] Dick C.A. Bulterman. Engineering Aspects of Web Hypermedia: Examples and Lessons from the GRiNS Editor. In *Proceedings of the ACM Hypertext 2004*, Santa Cruz, 2004.
- [CK02] Edith Cohen und Haim Kaplan. Caching Documents with Variable Sizes and Fetching Costs: An LP-Based Approach. *Algorithmica*, 32(3):459–466, 2002.
- [Dav02] Brian D. Davidson. *The Design and Evaluation of Web Prefetching and Caching Techniques*. Dissertation, Graduate School New Brunswick RutgersState, University of New Jersey, New Jersey, 2002.
- [Dav04] Brian Davison. Learning Web Request Patterns. In A. Poulouvassilis and M. Levene (Hrsg.): *Web Dynamics*, Seiten 435–460, Boston, 2004. Springer.
- [KT01] Javed I. Khan und Quingping Tao. Partial Prefetch for Faster Surfing in Composite Hypermedia. In *3rd USENIX Symposium on Internet Technologies and Systems (USITS 2001)*, Seiten 13–24, San Francisco, 2001.
- [KT03] Javed I. Khan und Quingping Tao. Exploiting Webspace Organization for Accelerating Web Prefetching. In *Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI'03)*, Seiten 579–583, Halifax, 2003.
- [Lan04] Michael Lang. A Critical Review of Challenges in Hypermedia Systems Development. *Information Systems Development Advances in Theory, Practice and Education 13th International Conference on Information Systems Development, ISD'2004 Proceedings*, 7(1):51–69, 2004.
- [NHHM<sup>+</sup>04] H.M. Niegemann, S. Hessel, D. Hochscheid-Mauel, K. Aslanski, M. Deichmann und G. Kreuzberger. *E-Learning Kompendium*. Springer, Berlin, 2004.
- [Röl03] Franz J. Röhl. *Pädagogik der Navigation: Selbstgesteuertes Lernen durch Neue Medien*. Kopaed, München, 2003.
- [VYK<sup>+</sup>02] A. Venkataramani, P. Yalagandula, R. Kokku, S. Sharif und M. Dahlin. The potential costs and benefits of long term prefetching for content distribution. *Elsevier Computer Communications*, 25(4):267–375, 2002.