Ein integriertes Hypertext- und Information-Retrieval-System für digitale Bibliotheken

Norbert Fuhr
Universität Dortmund

1 Einführung

Digitale Bibliotheken bieten die Möglichkeit, effizient und ortsunabhängig auf große Dokumentbestände zugreifen zu können. Da ein Benutzer aber in der Regel nicht die eindeutige Identifikationnummer des gewünschten Dokumentes kennt (etwa in der Form eines URN oder einer ISBN), müssen geeignete Such- und Navigationsmöglichkeiten bereitgestellt werden, damit ein Zugriff über Dokumentattribute oder eine inhaltsorientierte Suche ermöglicht wird. Herkömmliche Datenbank-Management- und Information-Retrieval-Systeme sind hierfür jedoch wenig geeignet: Erstere, weil sie keine adäquate Funktionalität zur Behandlung vager Anfragen bereitstellen, und letztere, weil sie die reichhaltigen Informationsstrukturen digitaler Bibliotheken nicht unterstützen.

In diesem Beitrag wird eine Rahmenkonzeption für die Such- und Navigationsfunktionalität von Informationssystemen für digitale Bibliotheken vorgestellt. Ausgehend von der Beschreibung der zu verwaltenden Informationsstrukturen und einer Grobklassifikation der Suchaktivitäten wird dann die Funktionalität ausführlich beschrieben. Abschließend wird die Architektur eines entsprechenden Systems kurz skizziert.

2 Informationsstrukturen

Durch unterschiedlich hohe Abstraktion gibt es in digitalen Bibliotheken Informationsstrukturen auf folgenden Ebenen:

Schema: Auf dieser Ebene werden die möglichen Attribute von Dokumenten bzw. deren Metadaten modelliert. Ferner wird die Struktur der Datensätze beschrieben. Bekannte Schemata sind die bibliographischen Datenformate wie z.B. MAB, PICA und die verschiedenen Varianten von MARC. Neben der Festlegeung der Attributmenge wird auch die Struktur der Datensätze

im Schema spezifiziert. Dublin Core war ursprünglich nur eine Menge von Attributen (für WWW-Seiten), doch gibt es jetzt verstärkt Bestrebungen nach einer stärkeren Strukturierung ([Weibel 95], [Weibel & Hakala 98]).

Attributwerte: Um seine Anfrage zu formulieren, benötigt ein Benutzer häufig Einblick in die für ein Attribut vorhandene Wertemenge (z.B.: gibt es überhaupt einen Autor mit diesem Namen?). Daneben sind die Attributwerte selbst häufig strukturiert (etwa Klassifikationen oder Thesauri), und der Benutzer möchte in dieser Struktur navigieren bzw. durch Bezug auf die Struktur eine Menge von möglichen Werten spezifizieren (etwa einen Deskriptor mitsamt all seinen Unterbegriffen).

Metadaten: Hierunter werden im folgenden die dokumentspezifische Metadaten verstanden. Im wesentlichen sind dies die Werte für die bibliographischen Attribute wie Autor, Titel, Erscheinungsjahr und Quelle, ferner häufig eine Kurzfassung (Abstract) oder auch die Referenzen (wie sie etwa in Zitationsdatenbanken zu finden sind).

aggregierte Dokumente: Hierunter fallen z.B. Sammelbände, Tagungsreihen und Zeitschriften. Da zunächst die Metadaten interessieren, sind aggregierte Dokumente eigentlich ein Spezialfall von Metadaten.

Volltexte: Die eigentlichen Dokumente besitzen eine reichhaltige Struktur, wobei den Benutzer primär die logische Struktur interessiert, weniger die Layout-Struktur. Die logische Struktur eines Dokumentes ist in der Regel hierarchisch (z.B. Kapitel, Abschnitte, Paragraphen), mit zusätzlichen Verweisstrukturen wie etwa Inhaltsverzeichnis und Index sowie Querverweisen.

Abbildung 1 gibt einen Überblick über diese Informationsstrukturen und die zugehörigen Navigations- und Suchmöglichkeiten.

Die Navigation muss sowohl innerhalb der einzelnen Ebenen als auch zwischen den Ebenen möglich sein. Bei einem komplexen Schema wie z.B. MARC sollte es möglich sein, zwischen ähnlichen Attributen zu navigieren. Von einem Attribut (z.B. persönlicher Herausgeber) möchte man zu den zugehörigen Attributwerten übergehen, evtl. in den Werten navigieren. Hat man einen Wert (z.B. eine Person) ausgewählt, möchte man die Metadaten von Dokumenten mit diesem Wert sehen (welche Dokumente wurden von dieser Person herausgegeben?). Auf der Ebene der Metadaten kann zwischen den Metadaten verschiedener Dokumente navigiert werden (welche Beiträge enthält der ausgewählte Sammelband, welche anderen Dokumente werden dort zitiert?). Schließlich will man die eigentlichen Dokumente sehen und in deren logischer Struktur navigieren. Auch ein Aufsteigen in höhere Ebenen muss möglich sein, also von einem Dokument zu den zugehörigen Metadaten, von dem Wert eines Metadaten-Attributs zur Menge der Attributwerte,

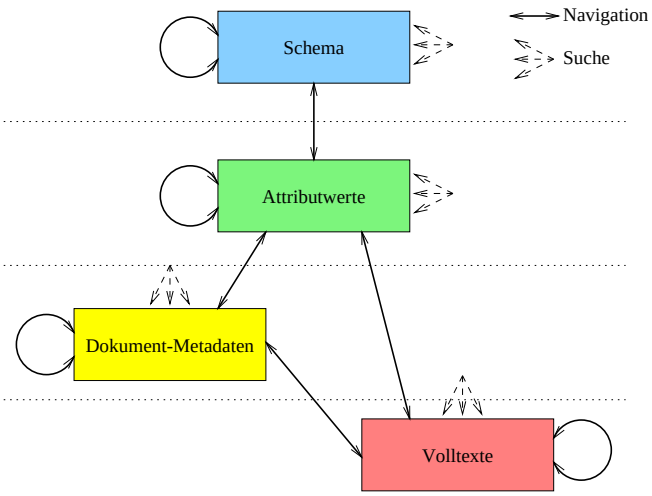

Abbildung 1: Navigation und Suche in Informationsstrukturen

und von dort zum Schema. Diese Navigation auf verschiedenen Ebenen wird in [Agosti et al. 91] als „Multi-Level-Hypertext“ bezeichnet.

Neben der Navigation muss auch die Suche auf den unterschiedlichen Ebenen möglich sein: Auf der Schemaebene nach einem passenden Attribut, auf der Werteebenen nach geeigneten Werten, auf der Metadatenebene nach Dokumenten mit bestimmten Attributwerten, und schließlich auf der Dokumentebene nach Dokumentstellen, die z.B. bestimmte Formulierungen beinhalten.

3 Suchaktivitäten

Während wir im vorigen Abschnitt Aspekte der Navigation und Suche aus der Sicht der Informationsstrukturen betrachtet haben, wollen wir hier kurz auf die benutzerorientierte Sicht eingehen. In [Bates 89] werden für die Literatursuche folgende Ebenen von Suchaktivitäten unterschieden:

Elementare Aktionen sind identifizierbare Gedanken oder Aktionen, die Teil einer Informationssuche sind. Diese entsprechen typischerweise den Kommandos, die von einem Datenbank-, Retrieval- oder Hypertext-System zur Verfügung gestellt werden, also z.B. Hinzufügen eines Terms oder einer Bedingung zu einer Anfrage oder Verfolgen eines Verweises.

Taktiken umfassen mehrere elementare Aktionen, die eine Suche vorantreiben, z.B. die Verbreiterung oder die Einengung einer Anfrage.

Strategeme umfassen meist mehrere Taktiken, um auf einem bestimmten Wege zum Ziel zu kommen, z.B. eine themenorientierte Suche.

Strategien sind Pläne, die Aktionen, Taktiken und Strategeme beinhalten, um eine Informationssuche vollständig durchzuführen. Eine einfache Strategie zur Literatursuche über ein neues Thema wäre etwa eine themenorientierte Suche im ersten Schritt, wonach man mit den in den relevanten Dokumenten gefundenen Namen eine Autorensuche anschließt.

Ein Grundproblem heutiger Informationssysteme besteht darin, daß die Unterstützung des Benutzers bei der Informationssuche sich meist auf die elementaren Aktionen beschränkt. Daher sollten zukünftige Systeme danach streben, auch Aktivitäten der höheren Ebenen in geeigneter Weise zu unterstützen. Im folgenden geben wir einige Beispiele für Strategeme und Taktiken, die ein digitales Bibliothekssystem berücksichtigen muss.

Die *inhaltsorientierte Suche* sollte in verschiedenen Varianten unterstützt werden:

- Die *inhaltsorientierte Navigation* folgt der Struktur eines Klassifikationsschemas oder Thesaurus', wobei insbesondere auch zwischen Attributwerten und Metadaten hin- und hergesprungen werden kann.

- Bei der *Freitext-Suche* in Abstracts wird nach bestimmten Formulierungen im Text von Abstracts (auf der Ebene der Metadaten) gesucht. Dies ist die typische Funktion klassischer Information-Retrieval-Systeme.

- Die *Suche nach ähnlichen Dokumenten* soll es ermöglichen, zu einem bereits bekannten oder im Laufe der Suche gefundenen relevanten Dokument ähnliche zu finden — ohne dass der Nutzer von sich aus die relevanten Begriffe aus dem Dokument heraussuchen muss, um sie anschließend als Suchfrage wieder einzutippen,

- Die *Volltextsuche* soll es ermöglichen, direkt die relevanten Dokumentteile zu lokalisieren (statt z.B. eine Dissertation im Umfang von 200 Seiten als Ganzes nachzuweisen),

- Bei der *strukturorientierten Volltextsuche* spezifiziert der Benutzer zusätzlich strukturelle Bedingungen, um z.B. in mathematischen Texten nach dem Beweis eines bestimmten Sachverhaltes zu suchen.

Die *Autorensuche* soll es ermöglichen, nach Dokumenten eines Autors oder einer Forschungsgruppe zu suchen.

Die *Zitations-basierte Navigation* muss die Navigation in zwei Richtungen ermöglichen, und zwar einerseits rückwärts zu referenzierten Publikationen eines gegebenen Dokumentes, und andererseits vorwärts zu denjenigen Werken, die das vorliegende Dokument zitieren.

Die *Navigation in Sammelwerken* ermöglicht das Bewegen innerhalb der Struktur einer Zeitschrift oder den Tagungsbänden einer Konferenzreihe, wobei insbesondere auch der Übergang von einem einzelnen Beitrag (als Ergebnis einer Suche) auf den umgebenden Kontext möglich sein muss.

4 Suchfunktionalität

Damit ein System die oben beschriebenen Suchaktivitäten angemessen unterstützen kann, muss eine geeignete Suchfunktionalität bereitgestellt werden. Diese lässt sich grob in drei Bereiche untergliedern:

- Die Navigation unterstützt das Browsen in den vielfältigen Strukturen digitaler Bibliotheken.
- Die Suche in Metadaten entspricht dem klassischen Retrieval.
- Die Volltextsuche ermöglicht es, in umfangreichen, komplex strukturierten Dokumenten gezielt suchen zu können.

Nachfolgend beschreiben wir diese Bereiche detailliert.

4.1 Navigation

Gemäß den in Abschnitt 2 beschriebenen Informationsstrukturen sollte das System ermöglichen, in und zwischen Schemas, Attributwerten, dokumentbezogenen Metadaten und Volltexten zu navigieren. Die theoretische Grundlage bildet dabei das Konzept des Multi-Level-Hypertext, das auch dem Benutzer in dieser Form angeboten werden sollte. Natürlich müssen auch entsprechende Navigationshilfen wie z.B. lokale und globale Übersichten angeboten werden, die dieses Konzept entsprechend berücksichtigen.

Eine wichtige Erweiterung gegenüber reinen Hypertext-Systemen bildet die notwendige Integration mit der eigentlichen Suche. Hierfür wird in [Bates 89] das Konzept des "berrypicking" vorgeschlagen: Bei der Navigation werden dabei potentielle Suchbegriffe — im wesentlichen Attributwerte — aufgesammelt und automatisch die Anfrage entsprechend erweitert. Dieses Sammeln geschieht nicht nur bei der Navigation innerhalb der Attributwerte, sondern auch beim Browsen in den Dokumenten, wo das System aus den vom Benutzer als relevant markierten Dokumenten automatisch wichtige Begriffe extrahiert; diese Technik ist als automatische Frageerweiterung bei Relevance Feedback bekannt ([Salton & Buckley 90]). Wenn der Benutzer dann beim nächstenmal zu den Metadaten oder den Volltexten navigiert, wird implizit mit der derart modifizierten Anfrage gesucht und dem Benutzer das entsprechende Suchergebnis vorgelegt.

4.2 Suche in Metadaten

Die Suche in den Metadaten muss zunächst einmal die Datenstruktur ununterstützen, in der die Metadaten abgelegt sind. Hierfür setzt sich in letzter Zeit XML ([Connolly 97]) als neuer Standard durch (siehe z.B. das Projekt Alexandria Digital Library [Frew et al. 98]). Zwar sind Metadaten in der Regel nicht sehr komplex strukturiert, doch muss ein System zumindest Schachtelungen von Tupeln erlauben, um z.B. im nachfolgenden Datensatz die Autoren korrekt zu den jeweiligen Institutionen zuzuordnen:

> `<author><name>` Michael Kifer `<affil>` SUNY at Stony Brook `<author> <name>` V.S. Subrahmanian `<affil>` University of Maryland `</author>`.

Da hauptsächlich in Texten gesucht wird, müssen hierfür flexible Textsuchoperatoren bereitgestellt werden. Neben der Suche nach Wörtern mit gleicher Grundform (*computer / computers*) bzw. Stammform (*computer / computing*) müssen auch Teile von Komposita (*Differentialgleichungssystem*) und sowie Nominalphrasen (*System von Differentialgleichungen*) suchbar sein.

Darüber hinaus werden oft auch noch fachspezifische Textsuchoperatoren benötigt, um etwa im nachfolgenden Beispiel nach technischen Messgrößen und chemischen Formeln suchen zu können:

> In metal/oxide/SnTe tunnel junctions (where the oxide is Al_2O_3 or SiO_2 and the metal is lead or aluminium) on BaF_2 or NaCl substrates the tunnel current I(U) and its derivatives I'(U) and I"(U) were measured at 4.2 K.

Neben Text als solchem beinhalten Metadaten in der Regel auch (Zeichenketten)-Werte bestimmter Datentypen. Vor allem sind hier Taxonomien zu nennen, wie etwa Klassifikationsschemata und Thesauri. Hierfür müssen Operatoren bereitgestellt werden, um automatisch Ober- oder Unterbegriffe bzw. verwandte Begriffe in die Suche einbeziehen zu können. Für viele Datentypen müssen zusätzlich geeignete vage Suchprädikate angeboten werden: Bei Personennamen müssen neben unterschiedlichen Schreibweisen (etwa aufgrund von Transkription, z.B. *Tschebischeff / Chebychev*) auch unterschiedliche Gepflogenheiten bei der Abkürzung von Vornamen berücksichtigt werden (z.B. *F. J. Smith / F. Jack Smith / Francis J. Smith*). Ein anderes Beispiel sind Datumsangaben, die der Benutzer häufig nur näherungsweise angeben kann (z.B. *pubdate* $\approx$ *05/95*).

4.3 Volltextsuche

Bei der Suche in Volltexten sollte ein System einerseits die Suche nach relevanten Dokumentteilen unterstützen, andererseits auch eine strukturorientierte Suche (etwa nach dem Inhalt bestimmter XML-Felder) ermöglichen.

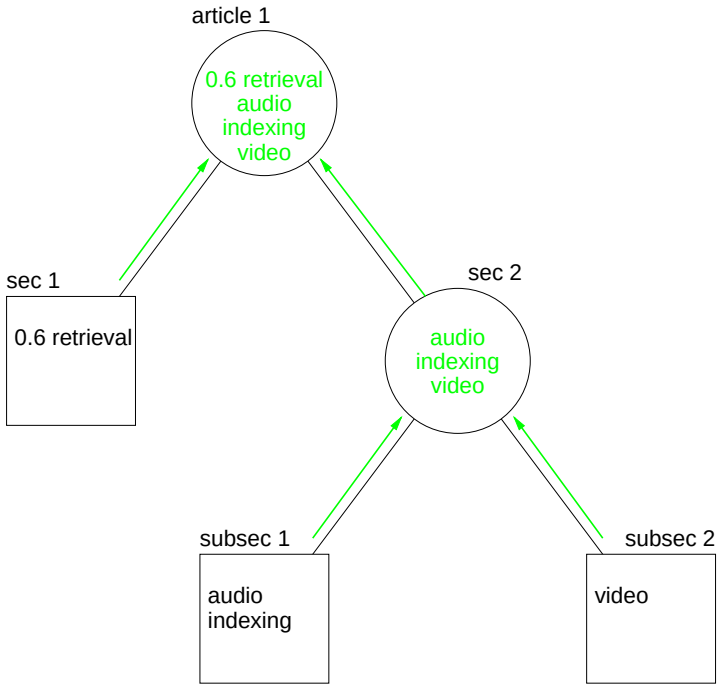

Abbildung 2: Suche nach relevanten Dokumentteilen mit Augmentierung

Dokumente sind üblicherweise hierarchisch strukturiert (wenn man einmal von zusätzlichen Querverweisen absieht). Sucht ein Benutzer nach relevanten Dokumentteilen zu seiner Anfrage, so sollte das System sinnvollerweise entsprechende Teilbäume der Dokumentstruktur nachweisen. Abbildung 2 illustriert diese Vorgehensweise an einem Beispiel (indexierte Begriffe in schwarz): Sucht der Benutzer nach den Begriffen „audio“ und „video“, so erfüllt kein einzelner Dokumentknoten diese Bedingung. Lässt man aber auch Teilbäume als Antworten zu, so wäre der beim Knoten sec2 beginnende Teilbaum die passende Antwort. Intern wird diese Vorgehensweise durch das Prinzip der Augmentierung realisiert, bei dem die Indexierung von Knoten in Richtung Wurzel propagiert wird (in der Abbildung grün bzw. grau). Das System sucht dann jeweils nach dem kleinsten Teilbaum, der die Anfrage erfüllt.

Bei der strukturorientierten Suche wird nach dem Inhalt bestimmter Felder bzw. Feldkombinationen gesucht, z.B. um in einem mathematischen Text Definitionen oder Beweise zu bestimmten Konzepten zu suchen. In Erweiterung zur relevanzorientierten Volltextsuche wird also zusätzlich der Typ der einzelnen Dokumentknoten berücksichtigt. Ausgangsbasis sind dabei Texte in SGML oder XML. Anfragen beziehen sich dann auf die Schachtelung sowie den Inhalt bestimmter

Tags. Prinzipiell sind hier beliebig komplexe Anfragesprachen möglich — z.B. um nach Dokumenten zu suchen, bei denen mindestens zwei Wörter einmal in der Einleitung und einmal in einer Kapitelüberschrift vorkommen. Allerdings werden solche komplexen Anfragen in der Regel kaum benötigt, zudem ist deren Prozessierung äußerst ineffizient. Es gilt hier also, einen guten Kompromiss zwischen Effizienz und Ausdrucksstärke zu finden. Eine solche Anfragesprache wird in [Meuss 98] vorgestellt, wo die Bestimmung der Antwortmenge nur linearen Aufwand in Abhängigkeit von der Größe der Anfrage erfordert.

Abbildung 3 gibt den entsprechenden Teil der Syntax einer solchen Anfragesprache wieder: Eine Strukturbedingung ist eine *path-expression*, die aus einer Folge von *path-element*s besteht. Ein *path-element* ist meist eine als *field* benannte Feldbedingung, bestehend aus einer Bedingung an den Tag-Namen (*tag-cond*) und möglicherweise einer Wertebedingung, (bestehend aus *predicate* und *value*). Z.B. sucht die Anfrage `book chapter section title [= 'conclusions']` nach der angegebenen Schachtelung von Tags im Dokumenttyp `book`, wobei zusätzlich im Tag `title` das Wort `'conclusions'` vorkommen soll. Die *tag-cond* kann Maskierungszeichen enthalten, um ein einzelnes oder eine Folge beliebiger Tags zuzulassen. Ein *path-element* kann auch eine *sequence* sein, die es erlaubt, das gemeinsame Vorkommen verschiedener *path-expression*s mit und ohne Berücksichtigung der Reihenfolge zu spezifizieren. Mit der Anfrage `* author (name [= 'Smith'], affil [= 'MIT'])` wird z.B. nach einem Autor-Feld (an beliebiger Stelle in der Struktur) gesucht, wobei der Name `'Smith'` sein soll und die zugehörige Affiliation `'MIT'`.

path-expression	::=	path-element *
path-element	::=	field | sequence
field	::=	tag-cond ([predicate value])?
tag-cond	::=	'*' | '+' | '.' | XML-tag
sequence	::=	'('path-expression (';' path-expression)*')' |
		'('path-expression (',' path-expression)*')'

Abbildung 3: Anfrage-Syntax für strukturierte Dokumente

5 Zusammenfassung und Ausblick

In deisem Beitrag wurde die Funktionalität eines integrierten Hypertext- und Information-Retrieval-Systems für digitale Bibliotheken skizziert. Das System soll die in digitalen Bibliotheken auftretenden Informationsstrukturen modellieren können und typische Suchaktivitäten durch die Bereitstellung flexibler Such- und Navigationsoperatoren adäquat unterstützen können.

Es ist beabsichtigt, dieses System im Rahmen der beantragten GlobalInfo-Sonderfördermaßnahme CARMEN (Content Analysis, Retrieval and Metadata: Effective Networking) zu realisieren. Dabei soll ein erweiterbares Basissystem für Metadaten und Volltexte in digitalen Bibliotheken entwickelt werden. Dieses System wird ergänzt um eine Gatherer- und eine Extractor- Komponente. Erstere dient zum Aufsammeln von Dokumenten und Metadaten aus Repositories, während letztere die Extraktion von Metadaten aus unterschiedlichen Dokumentformaten übernehmen soll. Die Gesamtfunktionalität ist dabei sehr ähnlich zum Harvest-System ([Bowman et al. 95]), allerdings sind die behandelten Dokumentstrukturen komplexer und die Suchfunktionalität wesentlich erweitert.

Literatur

Agosti, M.; Colotti, R.; Gradenigo, G. (1991). A two-level hypertext retrieval model for legal data. In: Bookstein, A.; Chiaramella, Y.; Salton, G.; Raghavan, V. (Hrsg.): *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, S. 316–325. ACM, New York.

Bates, M. (1989). The design browsing and berrypicking techniques for the online search interface. *Online review 13(5)*, S. 407–424.

Bowman, C.; Danzig, P.; Hardy, D.; Manber, U.; Schwartz, M. (1995). The Harvest Information Discovery and Access System. *Computer Networks and ISDN Systems 28*, S. 119–125.

Connolly, D. (Hrsg.) (1997). *XML: Principles, Tools, and Techniques*, Band 2 von *World Wide Web Journal*. O'Reilly, Sebastopol, California.

Frew, J.; Freeston, M.; Freitas, N.; Hill, L.; Janee, G.; Lovette, K.; Nideffer, R.; Smith, T.; Zheng, Q. (1998). The Alexandria Digital Library Architecture. In: Nikolaou, C.; Stephanidis, C. (Hrsg.): *Research and Advanced Technology for Digital Libraries*, S. 61–74. Springer, Berlin et al.

Meuss, H. (1998). Indexed Tree Matching with Complete Answer Representations. In: *Proceedings of the Workshop on Principles of Digital Document Processing 1998*. http://www.cis.uni-muenchen.de/people/Meuss/newtreerevis.ps.gz.

Salton, G.; Buckley, C. (1990). Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science 41(4)*, S. 288–297.

Weibel, S.; Hakala, J. (1998). DC-5: The Helsinki Metadata Workshop; A Report on the Workshop and Subsequent Developments. *D-lib Magazine 4(2).* http://www.dlib.org/dlib/february98/02weibel.html.

Weibel, S. (1995). Metadata: The Foundations of Resource Description. *D-Lib Magazine 1(July).* http://www.dlib.org/dlib/July95/07weibel.html.