

# **Unterstützung beim Finden und Durchführen von Suchstrategien in Digitalen Bibliotheken**

**Vom Fachbereich Ingenieurwissenschaften der Universität  
Duisburg-Essen zur Erlangung des akademischen Grades eines  
Doktors der Ingenieurwissenschaften (Dr.-Ing.) genehmigte  
Dissertation**

von  
**Dipl.-Inform. Sascha Kriewel**  
aus Essen

Referent: Prof. Dr.-Ing. Norbert Fuhr  
Korreferent: Prof. Dr. Dagobert Soergel  
Tag der mündlichen Prüfung: 25. März 2010



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Struktur der Arbeit . . . . .	3
<b>I. Informationssuche und Suchunterstützung</b>	<b>5</b>
<b>2. Informationssuche</b>	<b>7</b>
2.1. Benutzerverhalten . . . . .	7
2.2. Suchaufgaben . . . . .	10
2.3. Suchstrategien . . . . .	13
2.3.1. Einordnungen und Klassifikationen . . . . .	14
2.3.2. Konkrete Strategien . . . . .	21
2.3.3. Konkrete Suchtaktiken . . . . .	25
2.4. Zusammenfassung . . . . .	28
<b>3. Benutzerunterstützung</b>	<b>31</b>
3.1. Suchsituationen . . . . .	31
3.1.1. Benutzerparameter einer Suchsituation . . . . .	33
3.1.2. Generische Situationen . . . . .	35
3.1.3. Aktionen und Übergänge . . . . .	35
3.2. Strategische Unterstützung . . . . .	37
3.2.1. Notwendigkeit . . . . .	37
3.2.2. Existierende Systeme . . . . .	40
3.2.3. Studien . . . . .	44
3.3. Scaffolding . . . . .	48
3.3.1. Scaffolding als Lehrmethode . . . . .	48
3.3.2. Software-realisiertes Scaffolding . . . . .	50
3.3.3. Unterstützung bei Suchstrategien mit Scaffolding . . . . .	51
3.4. Zusammenfassung . . . . .	51
<b>4. Suchlogs und Suchhistorien</b>	<b>53</b>
4.1. Suchhistorien . . . . .	53
4.2. Suchpfade und Pfadanalyse . . . . .	56
4.3. Zusammenfassung . . . . .	59

<b>II. Unterstützung beim Finden und Durchführen von Suchstrategien</b>	<b>61</b>
<b>5. Architektur</b>	<b>63</b>
5.1. DAFFODIL . . . . .	63
5.1.1. Werkzeuge und Strategeme . . . . .	64
5.1.2. Agentenarchitektur . . . . .	70
5.2. Aufgabenabhängige Perspektiven . . . . .	70
5.3. Strategieframework . . . . .	71
5.3.1. Modulare Suchstrategien . . . . .	72
5.3.2. Makros . . . . .	73
5.4. Loggingframework . . . . .	73
5.5. Visualisierung von Suchpfaden . . . . .	78
5.6. Zusammenfassung . . . . .	80
<b>6. Strategisches Vorschlagssystem</b>	<b>81</b>
6.1. Vorschläge durch fallbasiertes Schließen . . . . .	81
6.1.1. Auswahl der Vorschläge . . . . .	83
6.1.2. Berechnung der Ähnlichkeiten und Auswahl der Vorschläge . . . . .	86
6.1.3. Umsetzung für DAFFODIL . . . . .	88
6.1.4. Umsetzung für Websuche . . . . .	93
6.1.5. Vorschläge basierend auf Suchergebnissen . . . . .	98
6.2. Vorschläge auf Basis von Suchpfaden . . . . .	98
6.2.1. Vorgehensweise . . . . .	100
6.2.2. Umsetzung für DAFFODIL . . . . .	102
6.2.3. Nutzung von Ergebnislogs in situationsbasierten Vorschlägen . . . . .	104
6.3. Zusammenfassung . . . . .	105
<b>7. Unterstützende Nutzerführung</b>	<b>107</b>
7.1. Automatische Ausführung einfacher Suchstrategeme . . . . .	107
7.1.1. Suche nach einem bekannten Objekt (1-1) . . . . .	108
7.1.2. Suche nach einer einzelnen Information (1-2) . . . . .	109
7.1.3. Suche nach einer Liste von Dokumenten mit definierten Eigenschaften (2-1) . . . . .	110
7.1.4. Suche nach einer Liste ähnlicher Dokumente (2-2) . . . . .	110
7.2. Nutzerführung durch Scaffolding . . . . .	112
7.2.1. Aufgabe . . . . .	113
7.2.2. Umsetzung des Scaffolding-Konzepts . . . . .	114
7.2.3. Implementierung . . . . .	116
7.2.4. Evaluation . . . . .	120
7.3. Zusammenfassung . . . . .	122

<b>III. Evaluation</b>	<b>123</b>
<b>8. Evaluation</b>	<b>125</b>
8.1. Strategisches Vorschlagssystem: Experiment 1 . . . . .	125
8.1.1. Versuchsaufbau . . . . .	126
8.1.2. Ergebnisse . . . . .	127
8.2. Strategisches Vorschlagssystem: Experiment 2 . . . . .	130
8.2.1. Versuchsteilnehmer . . . . .	131
8.2.2. Versuchsaufbau . . . . .	132
8.2.3. Auswertung . . . . .	134
8.2.4. Ergebnisse . . . . .	134
8.3. Logauswertungen . . . . .	138
8.4. Zusammenfassung . . . . .	144
<b>9. Zusammenfassung und Ausblick</b>	<b>147</b>
9.1. Ausblick . . . . .	149
<b>IV. Anhänge und Verzeichnisse</b>	<b>153</b>
<b>A. DTDs</b>	<b>155</b>
<b>Literaturverzeichnis</b>	<b>172</b>
<b>Abbildungsverzeichnis</b>	<b>173</b>
<b>Tabellenverzeichnis</b>	<b>175</b>



# Kapitel 1.

## Einleitung

Während noch bis in die 80er Jahre zumeist Suchexperten oder Informationsmittler die Hauptnutzer von Information-Retrieval- (IRS) oder Digital-Library-Systemen (DLS) waren [MARKEY 2007a], sind es heutzutage in der Mehrzahl Endnutzer. In direkter Interaktion mit dem System und den verfügbaren Informationsobjekten versuchen Endnutzer die Problemsituation zu lösen, die zu ihrer Suche geführt hat [XIE und COOL 2009]. Moderne IRS oder DLS bieten diesen oft zahlreiche Such- und Browserfunktionen an, sowie zusätzliche Funktionalitäten zum erfolgreicherem Suchen von und Arbeiten mit Information. Doch auch mit der Suchdomäne und dem Suchwerkzeug vertrauten Endnutzern fehlt oft das strategische oder prozedurale Wissen, um die angebotenen Werkzeuge effizient einzusetzen und zu einer erfolgreichen Suchstrategie zur Befriedigung ihres Informationsbedürfnisses zu verknüpfen.

Marti A. Hearst [HEARST 2009] hebt hervor, dass bei der Evaluation der Benutzbarkeit von Suchschnittstellen das Augenmerk insbesondere auf drei Aspekte gerichtet ist: Effektivität, Effizienz und Zufriedenheit bei der Benutzung. Dabei wird allerdings meist nur die Effizienz und Effektivität des Systems und nicht des Benutzers betrachtet. Der effektive und effiziente Einsatz von Informationssuchtechnologie durch den Endnutzer bleibt eine Herausforderung [BHAVNANI et al. 2001b, DRABENSTOTT 2003, WILDEMUTH 2004, MARKEY 2007b] und führt zu geringer Zufriedenheit mit den benutzten Systemen.

Ein effizienter Umgang zeichnet sich durch die situationsangemessene Verwendung der dargebotenen Werkzeuge und den Einsatz von Suchstrategien aus, die den kognitiven und zeitlichen Aufwand des Benutzers gering halten. Mögliche Maße für die Effizienz sind die Bearbeitungszeit für eine Aufgabe (*Task Completion Time*), die Zahl benötigter Suchaktivitäten oder die Zahl relevanter Dokumente, die der Benutzer betrachtet hat, im Verhältnis zur Zahl aller vom Benutzer im Laufe der Aufgabenbearbeitung inspizierten Dokumente. Effektiv arbeitet der Endbenutzer, wenn er die vorhandenen Mittel des Suchsystems zielführend nutzen kann, um sein Suchproblem vollständig zu lösen. Ein mögliches Effektivitätsmaß ist die *Task Completion Rate*, deren genaue Definition abhängig ist von der Art der Suchaufgabe. Die *Task Completion Rate* kann beispielsweise die Anzahl gefundener relevanter Dokumente oder den Anteil gelöster Teilaufgaben messen [AULA und NORDHAUSEN 2006].

---

Das Phänomen des Mangels an strategischem Wissen beim Einsatz eines Systems ist nicht allein bei Suchsystemen zu beobachten, sondern tritt allgemein bei komplexer Software auf. So schildern Bhavnani und John [BHAVNANI und JOHN 2000], dass auch im Umgang mit Computersystemen erfahrenen Benutzern oft der Sprung zu einer wirklich effizienten Nutzung komplexer Anwendungen nicht gelingt. Wissen über die Aufgabe und Wissen über die vorhandenen Werkzeuge genügt nicht, sondern Benutzer müssen auch effiziente Strategien erlernen.

Zahlreiche Untersuchungen haben gezeigt, dass gerade unerfahrene Benutzer ineffiziente, oft zielkonträre und damit ineffektive Suchtaktiken oder Suchstrategien bei ihrer Suche nach Information benutzen [MARCHIONINI 1989] – und auch angesichts offenkundigen Scheiterns meist nicht die Art effektiver Strategien entwickeln, wie sie von Suchexperten eingesetzt werden [BRAJNIK et al. 1996]. Zugleich zeigt sich, dass von Suchsystemen angebotene fortgeschrittene Werkzeuge oft nicht eingesetzt werden, weil das Wissen fehlt, wie und wann diese sinnvoll zu benutzen wären.

Während alle IRS und DLS Benutzern die Ausführung einfacher Suchaktionen und oft auch terminologische Hilfestellungen anbieten, hat Marcia Bates [BATES 1990] drei höhere Stufen der Abstraktion für Suchfunktionalitäten identifiziert: Taktiken, Strategeme und Strategien. Obwohl viel Energie investiert wurde, um Benutzer bei der Ausführung von Suchaktionen, Taktiken oder Strategemen zu unterstützen (z. B. im Rahmen des DAFFODIL-Projektes [KLAS 2007, SCHAEFER 2008]), fehlt es an der Unterstützung auf der höchsten Abstraktionsebene [BRAJNIK et al. 1996, BRAJNIK et al. 2002, HSIEH-YEE 1993]. Endnutzer von Informationssystemen benötigen Hilfe, die angemessenste oder nützlichste Aktion in einer spezifischen Situation zu wählen, um zu einem zielführenden Suchplan oder einer Suchstrategie für ihr Informationsproblem zu gelangen.

Abhängig vom Grad der Systemunterstützung könnte ein Suchsystem etwa nützliche Strategien für die Bearbeitung einer bestimmten Aufgabe – entweder auf Nutzeraufforderung oder pro-aktiv – oder sogar fertige Strategien für bekannte, einfache Suchprobleme automatisiert oder mit geringerer Nutzerintervention ausführen. Dabei sollte das System dem Benutzer auch ein Gefühl für die strategischen Aspekte der Informationssuche vermitteln, um ihm die Möglichkeit an die Hand zu geben, das nötige prozedurale Wissen zu erlangen, um eigenständig Teilziele und -aufgaben für ein Suchproblem zu strukturieren und eine Suche strategisch planen zu können. Es ist also notwendig, dass das System die Gründe für strategische Vorschläge erklärt und aufzeigt, warum eine Strategie in einer Situation effektiv oder effizient wäre, und wie der Benutzer diese Vorschläge nutzen kann, um selbst bessere Suchstrategien zu entwickeln.

Für bekannte Aufgaben wäre es nützlich, wenn ein System fertige Strategien bereitstellt, die erfahrene Benutzer für ihre eigenen Zwecke adaptieren kön-



nen. Unerfahrene Suchneulinge können von softwarebasierten Scaffoldingtechniken profitieren, die sie für bekannte Problemstellungen durch eine Suche führen und deren Hilfestellungen mit wachsender Erfahrung des Nutzers nach und nach zurückgeführt werden.

Aber auch im Falle komplexer, eher explorativer Suchaufgaben, die sich nicht vorab planen lassen, können Benutzer im Laufe der Suche an einen Punkt gelangen, von dem aus sie ihr Suchziel auf sich allein gestellt nicht oder nur ineffizient und unbefriedigend erreichen können. Hier würden sie von einem Suchsystem profitieren, das diese Situation erkennen und ihnen Strategien vorschlagen kann, um aus dieser heraus zu gelangen.

Dabei stellen sich eine Reihe von Fragen, denen in dieser Arbeit nachgegangen werden soll:

1. Kann ein DL-System unerfahrene Benutzer bei der Bildung einer umfassenden Suchstrategie für ihr Informationsproblem unterstützen?
2. Ist es für wohldefinierte Suchaufgaben möglich, Software-Scaffolding einzusetzen, um Benutzer zu unterstützen und ihnen dabei das Erlernen eines strategischen Umgangs mit den vorhandenen Suchwerkzeugen zu ermöglichen?
3. Lassen sich aus den erfolgreichen Suchaktionen von erfahrenen Benutzern situationsspezifische Vorschläge lernen, mit denen auch unerfahrene Benutzer zielführend Taktiken und Strategeme einsetzen können

## **1.1. Struktur der Arbeit**

Im ersten Teil dieser Arbeit werden zunächst in Kapitel 2 das Problem der Informationssuche aus Benutzersicht diskutiert und verschiedene Modelle untersucht, die den Prozess und das Verhalten von Benutzern bei der Suche beschreiben; ebenso werden Suchaufgaben betrachtet und es findet eine Auseinandersetzung mit existierenden Arbeiten zu Suchstrategien statt. Kapitel 3 behandelt Benutzerunterstützung aus verschiedenen Betrachtungswinkeln; es werden Suchsituationen untersucht, existierende Arbeiten zur strategischen Unterstützung bei der Informationssuche, sowie das Software-Scaffolding als eine Methode zur Unterstützung bei der Erlernung komplexer Aufgaben beschrieben. Kapitel 4 schließt den ersten Teil ab und betrachtet unter anderem die Rolle von Suchhistorien für die Strategiebildung. Allgemeine Suchlogs werden ebenfalls angesprochen, und wie diese für den Vorschlag weiterer Suchschritte genutzt werden können.

Im zweiten Teil werden zwei Hauptszenarien zur Unterstützung von Benutzern bei der Findung und Durchführung von Suchstrategien entwickelt. In Kapitel 5 wird dabei zunächst auf die notwendige Architektur eingegangen,

## 1.1. STRUKTUR DER ARBEIT

---

die im Rahmen der DAFFODIL-Software implementiert wurde. Kapitel 6 stellt eine Methode für strategische Vorschläge basierend auf der Technik des fallbasierten Schließens vor. Dabei wird auch behandelt, wie die Suchhistorie oder der Suchpfad, der zu einer Situation geführt hat, zur Unterstützung genutzt werden kann. Kapitel 7 schildert die automatische Umsetzung einfacher Strategien und eine Methode zur geführten Benutzerunterstützung durch aufgabenabhängiges Software-Scaffolding.

Die Evaluationen der vorgestellten Unterstützungsszenarien in Bezug auf die eingangs aufgeführten Fragestellungen werden in Kapitel 8 vorgestellt und diskutiert, bevor ein Ausblick die Arbeit beschließt.

**Teil I.**

**Informationssuche und  
Suchunterstützung**



# Kapitel 2.

## Informationssuche

### 2.1. Benutzerverhalten

Um Benutzern angemessene strategische Unterstützung während ihrer Informationssuche anbieten zu können, ist es wichtig, zunächst zu betrachten, wie Benutzer suchen, welches typische Verhalten sie dabei zeigen und welche Stadien sie bzw. ihre Informationssuche durchlaufen. Hierzu sind im Information Retrieval (IR) und im Bereich der Informations- und Bibliothekswissenschaften eine Reihe von Modellen entwickelt worden.

Bei der Suche konstruieren sich die Informationssuchenden ihr eigenes kognitives Modell der Welt, in dem sie das während der Suche gewonnene Wissen mit ihrem existierenden Wissen verknüpfen. Die Informationssuche selbst ist dabei ein hoch komplexer Vorgang, der aus zahlreichen Schritten besteht und sowohl Aufgabenwissen als auch Systemwissen und Suchexpertise erfordert [MARCHIONINI 1995]. Aufgabenwissen wird auch oft als Domänenwissen, Suchexpertise als prozedurales Wissen bezeichnet.

Als eines der wichtigsten frühen Modelle ist Ellis' Verhaltensmodell der Informationssuche anzusehen [ELLIS 1989]. Er studierte das Suchverhalten von Sozialwissenschaftlern und erarbeitete ein aus sechs Aktivitäten bestehendes Modell: *Starting*, *Chaining*, *Browsing*, *Differentiating*, *Monitoring* und *Extracting*. Das Modell wurde später von Ellis und Kollegen [ELLIS et al. 1993] um die Aktivitäten *Verifying* und *Ending*, sowie von Meho und Tibbo [MEHO und TIBBO 2003] um *Accessing*, *Networking* und *Information Managing* erweitert.

Kuhlthau entwickelte und verifizierte basierend auf einer Reihe von Studien mit Bibliotheksbenutzern ein Prozessmodell der Informationssuche, das gegenüber Ellis' Modell Gefühle, Gedanken und Aktionen der Suchenden gleichermaßen berücksichtigt [KUHLT HAU 1991]. Sie beschreibt den Suchprozess als Abfolge von sechs Phasen, wobei jede Phase geprägt ist von typischen Gefühlen der Suchenden, die von Verwirrung über Enttäuschung bis hin zu Optimismus reichen können.

In der Phase der *Initiation* erkennt der Informationssuchende sein Informationsbedürfnis, versucht seine Aufgabe zu verstehen und sie mit vorhandenem

## 2.1. BENUTZERVERHALTEN

---

Wissen und früheren Erfahrungen in Beziehung zu setzen. Die Auswahl des spezifischen Suchthemas schließt sich in der Phase der *Selection* an, in der Möglichkeiten erwogen und bewertet werden. Vor der eigentlichen Formulierung des Suchproblems steht dann die Phase der *Exploration*. In dieser werden die vorhandenen Informationsquellen explorativ untersucht, um einen Fokus zu gewinnen und zu einer geeigneten Möglichkeit zu gelangen, das eigentliche Informationsbedürfnis auszudrücken.

Mit den gewonnenen Einsichten kann der Informationssuchende in der Phase der *Formulation* einen Fokus für seine Suche bilden, und aus der gesichteten Information zentrale Ideen identifizieren und auswählen. Während der Phase der *Collection* findet dann die eigentliche Suche nach und das Sammeln von Informationen statt, die in der abschließenden Phase der *Presentation* aufbereitet und synthetisiert werden.

Neben den sechs beschriebenen Phasen ist ein zweiter wichtiger Bestandteil des Modells die Idee der „Zonen der Intervention“. Zonen der Intervention sind Bereiche im Suchprozess in denen ein Informationssuchender seine Suche erfolgreicher mit Unterstützung durchführen kann:

The zone of intervention is that area in which an information user can do with advice and assistance what he or she cannot do alone or can do only with great difficulty. [KUHLETHAU 2002]

Dabei ist ein Indiz für eine solche Zone der Intervention eine erhöhte Unsicherheit des Suchenden, die innerhalb des Suchprozesses mit dem Übergang von der Phase der *Selection* zur Phase der *Exploration* einhergeht. Interventionen lassen sich nach Kuhlthau in insgesamt fünf unterschiedliche Zonen unterteilen. Innerhalb der ersten Zone kann der Suchende sein Problem noch alleine lösen, doch die weiteren Zonen sind durch die steigende Involvierung eines Hilfestellenden in den Suchprozess charakterisiert. Dabei sieht Kuhlthau den menschlichen Bibliothekar als Hilfestellenden, doch die Idee kann leicht auf eine automatisierte Unterstützung mit einem Software-System in dieser Rolle übertragen werden.

- In Zone 2 (*single source*) ist der Hilfestellende ein *Locator*, der Referenzinformationen bereitstellt.
- In Zone 3 (*group of sources*) ist der Hilfestellende ein *Identifier*, der potentiell nützliche Ressourcen identifiziert.
- In Zone 4 (*sequence of sources*) agiert der Hilfestellende als *Advisor* und führt den Suchenden durch die nützlichen Ressourcen.
- In Zone 5 (*process intervention*) führt der Hilfestellende als *Counsellor* durch den gesamten Prozess der Suche.

Belkin [BELKIN 1993] beschreibt Informationssuche als „Interaktion mit Texten“. Zahlreiche Studien mit Suchenden [BELKIN et al. 1990, ELLIS 1989,

KUHLTHAU 1991] haben gezeigt, dass diese im Laufe einer Informationssuche eine Reihe von Interaktionsarten einsetzen und dabei zwischen diesen wechseln. Belkin identifiziert vier Facetten oder Dimensionen zur Beschreibung des Benutzerverhaltens, die von Belkin, Marchetti und Cool [BELKIN et al. 1993] genauer beschrieben werden: Methode, Ziel, Modus und Ressourcen. Diese vier Facetten können jeweils zwei unterschiedliche Werte annehmen und beliebig kombiniert werden, um so einen Raum von insgesamt sechzehn verschiedenen sogenannten Informationssuchstrategien oder Suchverhalten zu definieren (siehe Abschnitt 2.3). Informationssuchverhalten wird dann charakterisiert als Bewegung von einer Informationssuchstrategie zur nächsten innerhalb einer Suchepisode. Benutzer legen dabei ein Problemlösungsverhalten an den Tag und setzen je nach Ziel, Wissensstand und Absicht unterschiedliche Interaktionsarten ein.

Problemlösungsverhalten ist ein Untersuchungsfeld der Kognitionswissenschaften und wird dort als Verhalten beschrieben, das auf die Lösung eines spezifischen Problems ausgerichtet ist, und dabei sowohl die Planung von Lösungsschritten als auch die Auswahl aus den möglichen Schritten umfasst [SOLSO et al. 2005]. Ein Problemlösungsmodell der Informationssuche wird von Wilson [WILSON 1999] beschrieben und besteht aus vier aufeinanderfolgenden Phasen, zwischen denen jeweils eine Rückkopplung existiert. In der ersten Phase erfolgt die Identifizierung des Problems. Durch fortschreitende Auflösung von Unsicherheit beim Suchenden folgen darauf die Problemdefinition, die Problemresolution und die Lösung. Die Auflösung der Unsicherheit ist dabei ein wichtiger Punkt in Wilsons Modell. Sie kann durch Benutzung anderer Modelle, etwa Ellis' oder Kuhlthaus näher beschrieben werden.

Marchioninis bereits früher definiertes Modell der Informationssuche [MARCHIONINI 1995] zeigt Parallelen zum menschlichen Problemlösungsverhalten und Wilsons Modell. Marchionini beschreibt das Benutzerverhalten als eine Kette von Teilprozessen, an deren Anfang das Erkennen und Akzeptieren des Informationsproblems steht (Wilsons *problem identification*). Der sich anschließende Prozess des Definierens und Verstehens des Problems (Wilsons *problem definition*) zieht sich durch die gesamte Informationssuche. Durch die Definition des Problems repräsentiert der Suchende seine Aufgabe und entwickelt eine Strategie zu ihrer Lösung. In weiteren Teilprozessen folgt die Auswahl der Quellen, die Formulierung von Anfragen, das Untersuchen der Ergebnisse, das Extrahieren von Informationen und schließlich die Reflektion des Gefundenen (Wilsons *solution statement*) und die Beendigung der Suche. Dabei beschreibt das Modell kein geradliniges Verhalten, sondern jeder Teilprozess erlaubt das Rückspringen zu früheren Teilprozessen. So existiert beispielsweise eine hohe Wahrscheinlichkeit, dass der Extraktion von Information eine neue Auswahl der Quellen und eine neue Anfrage folgt. Verglichen mit Wilsons Modell bietet Marchionini also insbesondere eine feinere Behandlung der Phase der *problem resolution*.

## 2.2. SUCHAUFGABEN

---

Wichtig ist beim Problemlösungsansatz zur Beschreibung von Suchverhalten die klare Trennung zwischen Problemformulierung oder -definition und Problemresolution und -lösung. Durch die Formulierung wird der Lösungsraum definiert und die Anforderungen der Aufgabe aufgestellt. Durch diesen Schritt wird ein Teil der Unsicherheit und Verwirrung beseitigt, die der Suchende zu Beginn der Bearbeitung verspürt [KUHLLTHAU 1991]. In Kuhlthaus Prozessmodell gehören die Phasen der *Initiation*, *Selection*, *Exploration* und *Formulation* zur Problemformulierung. *Collection* und *Presentation* zur Problemlösung.

Brajnik [BRAJNIK 1999] beschreibt Informationssuche als „Exploratives Lernen“ und kritisiert die bestehenden Modelle als schlecht geeignet, um das Design und die Evaluation von Informationssystemen zu unterstützen. Er unterscheidet zwei Abstraktionsebenen: jene der Präsentation, auf der die direkte Interaktion des Benutzers mit dem Suchsystem stattfindet, und jene der Navigation, auf der Benutzer mit Informationsobjekten interagieren. Dabei hält Brajnik die Ebene der Navigation für die entscheidende. In der Navigationsmetapher erklärt er den Suchprozess des Benutzers als Exploration des Suchraums mit dem Ziel, etwas über diesen zu erfahren (z. B. seine Struktur, ob er bestimmte Informationsobjekte enthält, welcher Art diese sind, oder spezifische Informationen aus diesen Objekten). Folglich sei die Informationssuche als exploratives Lernen zu verstehen.

## 2.2. Suchaufgaben

Ein wichtiger Aspekt bei der Betrachtung des Suchverhaltens von Benutzern ist die Suchaufgabe. Es herrscht weitgehende Übereinstimmung darin, dass das Verhalten bei der Informationssuche von der Suchaufgabe des Benutzers und den durch diese induzierten typischen Problemen abhängt [BYSTRÖM und JÄRVELIN 1995]. Die Suchaufgabe beeinflusst die Wahl der Strategie [RAMIREZ et al. 2002] und die Art möglicher strategischer Unterstützung. Dabei lassen sich sowohl die objektiven Aufgaben als auch subjektiv wahrgenommene Aufgaben der Informationssuchenden untersuchen, wie von Järvelin und Ingwersen als zwei von neun Dimensionen zur Beschreibung von Informationssuche vorgeschlagen [JÄRVELIN und INGWERSEN 2004]. In der Informationssuche sind die wahrgenommenen Aufgaben ein wichtiger Faktor, da sie die Basis für die Durchführung der Aufgabe und die Wahl der Aktivitäten zur Befriedigung des interpretierten Informationsbedürfnisses durch den Suchenden bilden [JÄRVELIN und WILSON 2003].

Byström und Järvelin konzentrieren sich in ihrer Kategorisierung von Suchaufgaben auf die Beziehungen zwischen Aufgabenart und der Art der für die Aufgabe benötigten Information [BYSTRÖM und JÄRVELIN 1995]. Dabei betrachten sie Information unter einem Problemlösungsansatz und kategorisieren sie nach Domäneninformation (bekannte Fakten), Probleminformation (Ei-



genschaften des Problems) und Problemlösungsinformation. Aufgaben werden in dem Modell anhand ihrer Komplexität kategorisiert, wobei eine ein-dimensionale Repräsentation der subjektiven Aufgabenkomplexität gewählt wurde. Diese stützt sich auf die Strukturiertheit der Aufgabe, bzw. auf die Vorabbestimmbarkeit von Ergebnis, Suchprozess oder Anforderungen.

- *Automatic information processing tasks* sind die einfachste Art von Informationsaufgaben, bei denen das Ergebnis der Aufgabe vollkommen bestimmbar ist. Sie erfordern im Allgemeinen keine Informationssuche.
- *Normal information processing tasks* sind fast vollkommen bestimmbar, erfordern jedoch fallabhängig zusätzliche (zu suchende) Information zur vollständigen Klärung.
- *Normal decision tasks* sind wohl strukturierte Aufgaben, bei denen fallabhängige Information gesucht und zur Entscheidungsfindung herangezogen werden muss.
- Bei *known, genuine decision tasks* sind Art und Struktur des Ergebnisses bekannt, aber es existieren keine festen Herangehensweisen zur Bearbeitung der Aufgabe. Der Prozess zur Entscheidungsfindung und die benötigten Informationen sind weitestgehend unbestimmt.
- *Genuine decision tasks* sind Aufgaben mit völlig neuen, unerwarteten Anforderungen. Sie sind unstrukturiert und die Art des Ergebnis ist genauso wenig vorab bestimmbar, wie der Prozess oder die Informationsanforderungen.

In Untersuchungen fanden Byström und Järvelin dabei heraus, dass mit wachsender Komplexität der Aufgabe die Komplexität der benötigten Information, die Anzahl benutzter Informationsquellen und der Bedarf an Domäneninformation und Problemlösungsinformation zunimmt. Gleichzeitig nimmt der Erfolg des Suchenden bei der Erfüllung der Aufgabe mit steigender Komplexität ab. Es sind insbesondere Aufgaben der letzten drei Komplexitätsklassen, bei denen Benutzer Unterstützung bei der Strategiefindung benötigen.

Xie [XIE 1997] untersuchte die Beziehung zwischen den Zielen von Suchenden und ihrem Informationssuchverhalten. Dabei klassifizierte sie die Ziele nach langfristigen Ziel, treibendem Suchziel, aktuellem Suchziel und interaktiver Absicht. Das aktuelle Suchziel beschreibt dabei, welche Art von Suchergebnisse der Suchende zu erzielen beabsichtigt und entspricht den Suchaufgaben anderer Klassifikationen. Sie identifiziert als Kategorien die Suche nach einem spezifischen Objekt, nach einer spezifischen Information, nach einer Gruppe von Objekten mit gemeinsamen Eigenschaften, das Suchen, um auf dem neuesten Stand zu bleiben, sowie sonstige Suchaufgaben.

Shneiderman [SHNEIDERMAN und PLAISANT 2009] beschreibt vier unstrukturierte Suchaufgaben. Die spezifische Faktensuche umfasst Xies Suche nach

## 2.2. SUCHAUFGABEN

---

einem spezifischen Objekt oder einer spezifischen Information; die erweiterte Faktensuche entspricht der Suche nach einer Gruppe von Objekten mit gemeinsamen Eigenschaften. Daneben identifiziert Shneiderman das offene Browsing (zur Beantwortung einer offen formulierten Frage) und die Erforschung der Verfügbarkeit von Information.

Basierend auf früheren Klassifikationen und den Erkenntnissen einer Feldstudie unterteilen Kellar, Watters und Sheppard [KELLAR et al. 2006] Informationsaufgaben für das World Wide Web in drei Kategorien, die durch das Interaktionsziel bestimmt werden: *Information Maintenance*, *Information Exchange* und *Information Seeking*. Unter das Ziel der Informationssuche fallen nach dieser Klassifikation drei Arten von Suchaufgaben:

- *Fact finding* umfasst solche Aufgaben, bei denen Benutzer nach spezifischen Fakten oder Informationsstücken suchen. Solche Aufgaben sind typischerweise einfach und kurz und lassen sich in einer einzelnen Suchsitzung bearbeiten.
- *Info gathering* umfasst das Sammeln von Information aus meist mehreren unterschiedlichen Quellen. Diese Aufgaben können sich über mehrere Suchsitzungen und sogar über mehrere Tage erstrecken. Anders als beim *fact finding* ist bei diesen Aufgaben nicht immer klar, ob die Aufgaben vollständig bearbeitet wurden, da keine spezifische Antwort existiert. Diese Art Aufgaben sind es, die am stärksten von Suchstrategien und folglich durch automatisierte Unterstützung profitieren können.
- *Browsing* umfasst entdeckende Aufgaben, bei denen eine Suche ohne ein festes Ziel begonnen wird. Dabei gefundene Informationen sind glückliche (*serendipitous*) und oft unerwartete Zufallsfunde.

Anders als einige frühere Klassifikationsversuche sehen Kellar und Kollegen das *Monitoring* (mit dem Ziel auf dem neuesten Stand zu bleiben) nicht als eine separate Aufgabenkategorie, sondern als eine Aktivität, die in allen Aufgaben der Informationssuche auftreten kann.

Li und Belkin [LI und BELKIN 2008] geben einen ausführlichen Überblick über verschiedene Klassifikationssysteme zur Einordnung von Arbeitsaufgaben, Informationsaufgaben und Suchaufgaben. Da die meisten der von ihnen untersuchten Systeme sich nur auf bestimmte Teilaspekte von Aufgaben konzentrieren, schlagen sie eine umfassende auf Facetten aufbauende Klassifikation vor. Generische Facetten von Aufgaben sind die Quelle, der Akteur, die Zeit (Häufigkeit, Länge und Stadium der Aufgabe), das erwartete Resultat oder Produkt der Aufgabe, der Prozess und die Ziele. Daneben werden als Attribute objektive und subjektive Komplexität der Aufgabe, Schwierigkeit, Wichtigkeit, Dringlichkeit und das Aufgaben- und Prozesswissen des Akteurs betrachtet.

Eine Taxonomie von IR-Aufgaben mit dem expliziten Ziel, diese zur Auswahl von Suchstrategien zu nutzen, schlagen Bhavnani, Drabenstott und Radev vor

[BHAVNANI et al. 2001b]. Sie verwenden zwei Dimensionen zur Klassifikation: was der Suchende weiß (verschwommene oder unvollständige Information) und was er von der Suche als Ergebnis erwartet (Fakten, Beispiele oder ausführliche und umfassende Information).

### 2.3. Suchstrategien

Wie bereits Harter in „Search Strategies and Heuristics“ [HARTER 1986] anmerkte, existiert in der Information-Retrieval-Literatur keine einheitliche Definition des Begriffs „Strategie“ oder „Suchstrategie“. Einige Autoren bezeichnen schon die Auswahl von Termen, die in ein Suchsystem eingegeben werden, als Strategie. Die Informationssuchstrategien, die sich aus den vier Dimensionen nach Belkin, Marchettio und Cool [BELKIN et al. 1993] ergeben, wurden bereits angesprochen und befinden sich dagegen auf einer deutlich höheren Ebene der Abstraktion.

Hier soll der Begriff Strategie im Sinne von Marcia Bates [BATES 1979b] und Harter benutzt werden, um die Strategie von Taktiken abzugrenzen:

*A search strategy is an overall plan or approach for a search problem, while a search tactic or heuristic is a move made to advance that strategy.* [HARTER 1986]

Eine Strategie kann also die gesamte Herangehensweise umfassen, um ein bestimmtes Suchziel zu erreichen. Innerhalb einer solchen Strategie kommen dann zahlreiche niedrigstufigere Aktionen zum Einsatz, wobei meist Heuristiken genutzt werden, um über ihren Einsatz zu entscheiden.

Ein ähnliches, wenn nicht äquivalentes Konzept wird von Xie [XIE 2000] als Plan beschrieben. Im *planned model* des Informationssuchverhaltens sind die Aktionen eines Benutzers Teil eines zielorientierten Plans, der den Benutzer von einer Ausgangssituation zu einem gewünschten Ziel bringen soll. Dabei setzt ein Plan sogenannte „Landmarks“, oder Subziele auf dem Weg zu einem komplexeren Ziel. Innerhalb eines Plans kommen dann Informationssuchstrategien zum Einsatz, wobei sie den Begriff der Suchstrategie im Sinne von Belkin und Kollegen benutzt.

Ramirez und Kollegen [RAMIREZ et al. 2002] betrachten die Faktoren, die bei Informationssuchenden zum Auswahl einer Strategie führen. Sie unterscheiden Faktoren aus fünf Klassen:

- Faktoren der Kommunikation sind entweder inhärente Eigenschaften des Suchenden oder reflektieren ein Muster der Informationssuche, das geprägt ist durch den Hintergrund und die Geschichte des Suchenden. Sie bestimmen stark die Auswahl einer initialen Strategie, einer bestimmten Sequenz von Strategien oder einer Technologie. So wird ein

## 2.3. SUCHSTRATEGIEN

---

Suchender, der Erfahrung mit einer bestimmten Strategie hat, diese bevorzugt wählen, auch wenn andere Strategien möglicherweise die bessere Wahl wären.

- Faktoren der Situation und des Kontextes stellen die äußeren Einflüsse auf das Informationsverhalten dar. Die Faktoren umfassen etwa die verfügbare Zeit für die Suche, Anzahl der in Frage kommenden Informationsquellen und andere Kontextaspekte. Einige Kontextfaktoren können bestimmte initiale Strategien prädestinieren oder sogar vorschreiben.
- Zielfaktoren beeinflussen die Wahl der Strategie zum einen dadurch, dass einige Strategien ungeeignet oder besonders geeignet sind, um das Suchziel zu erreichen; zum anderen spielt aber auch eine Rolle, ob es sich um lang- oder kurzfristige Ziele handelt.
- Zu den Informationsfaktoren zählen die Eigenschaften der gewünschten Information, darunter Verfügbarkeit, Wichtigkeit oder Qualität, und die Definiertheit des Informationsbedürfnisses.
- Die Faktoren der Technologie sind von den verfügbaren Werkzeugen und Möglichkeiten der Informationssysteme abhängig, die zur Suche bereitstehen. Die verfügbare Technologie kann bestimmte Strategien ausschließen oder erschweren.

### 2.3.1. Einordnungen und Klassifikationen

Verschiedene Versuche wurden unternommen, um identifizierte Suchstrategien zu klassifizieren und zu ordnen. Hier sollen vier Systematiken präsentiert und verglichen werden.

#### Suchaktivitäten nach Bates

Bates [BATES 1990] abstrahiert Benutzeraktivitäten während einer Suche mit Hilfe von vier Ebenen, die sich in ihrer Spezifität wie auch im Umfang ihrer Möglichkeiten unterscheiden. Die kleinste, atomare Aktivität wird als *Move* (Schritt) definiert. Auf diesen bauen dann Taktiken auf, die aus mehreren einzelnen Schritten bestehen und den Zweck haben, die Suche voranzubringen. Harter [HARTER 1986] bezeichnet das gleiche Konzept als Heuristik. Über den Taktiken angesiedelt liegen sogenannte Strategeme. Bates entlehnte die Begriffe für Suchaktivitäten dem militärischen Sprachgebrauch, wo ein Strategem eine Kriegsliste ist. In Bates' Klassifikation beschreiben Strategeme Aktivitäten, die sich einzelner Moves und Taktiken bedienen, um die Struktur einer bestimmten Suchdomäne auszunutzen (also eine auf die Domäne angepasste Vorgehensweise oder List). Eine Strategie schließlich setzt eine komplette Informationssuche zu einem existierenden Informationsbedürfnis um.

Wie angedeutet setzt sich in dieser Klassifikation eine Aktivität der höheren Ebenen jeweils aus einer oder mehreren Aktivitäten der feineren Ebenen zusammen. Dabei kommt es auf die situative Anwendbarkeit und die vom Suchenden zu treffenden Entscheidungen an, ob etwa aus einer Kombination von Taktiken und Moves tatsächlich ein Strategem wird. Es soll im folgenden kurz beispielhaft darauf eingegangen werden, was unter den einzelnen Ebenen der Suchaktivität genau zu verstehen ist. In den Tabellen 2.1 und 2.2 wird eine Übersicht über die von Bates aus der Praxis gesammelten [BATES 1979a, BATES 1979b, BATES 1989] Suchtaktiken bzw. Strategeme gegeben.

Ein **Move** stellt den Grundbaustein der Suchaktivitäten dar. Hierunter fallen alle minimalen Schritte während einer Informationssuche. Dabei kann ein solcher Move sowohl auf der rein mentalen Ebene (als Gedanke oder Entscheidung) als auch auf der physikalischen Ebene (als Aktion) stattfinden. So wäre etwa die Eingabe eines neuen Suchbegriffs oder die Ersetzung eines booleschen AND durch ein OR ein Move.

Als **Taktik** versteht man das Zusammenspiel mehrerer Schritte zu einem bestimmten Zweck (die Suche voranzubringen). Taktiken sollen entweder eine problematische Situation auflösen oder vermeiden, wobei problematische Situationen auftreten, wenn das Ergebnis einer Suche nicht die Erwartungen des Suchenden trifft. Bates beschrieb zunächst vier Kategorien von Taktiken (*Monitoring tactics* zur Überwachung und effizienten Gestaltung der Suche, *File structure tactics* zur Ausnutzung der Struktur der Informationsquelle, *Search formulation tactics* zur Verbesserung der Anfrageformulierung und *Term tactics* für die Auswahl der benutzten Suchbegriffe). Diese ergänzte sie später [BATES 1979a] um die sogenannten *Idea tactics*.

**Strategeme** sind komplexer in ihrem Aufbau als Taktiken und können mehrere Einzeltaktiken und -schritte umfassen, die hintereinander ausgeführt werden, um einen klar definierten und strukturierten Suchraum mit Hilfe einer auf diesen abgestimmten Methode effizient zu durchsuchen. Dabei kann ein Suchraum etwa ein Zitationsnetz, ein Klassifikationssystem oder ein Verzeichnis von Zeitschriften sein. Für jeden Suchraum gibt es eigene, geeignete Strategeme.

Die Beschreibung einer **Strategie** bleibt eher vage und bezeichnet einen vollständigen Plan für die Durchführung einer Informationssuche unter Einsatz von Schritten, Taktiken und Strategemen.

### Suchverhalten nach Marchionini

Marchionini [MARCHIONINI 1995] unterscheidet ähnlich wie Bates zwischen vier Granularitätsstufen, auf denen man das Informationssuchverhalten beschreiben und unterstützen kann. Er sieht in diesen Stufen teilweise Analoge

## 2.3. SUCHSTRATEGIEN

---

zur den Ebenen des GOMS-Modells (*Goals, Operators, Methods, Selection Rules*) [CARD et al. 1983, JOHN und KIERAS 1996].

Die größte Stufe der Granularität bezeichnet er als *patterns* (Muster). Diese sind meist internalisierte Verhaltensweisen, erlernt durch Wiederholung und Erfahrung. Sie sind nicht notwendigerweise geeignet, eine Aufgabe auf genaueste oder effizienteste Weise zu lösen, sondern sollen insbesondere die momentane kognitive Last des Informationssuchenden mindern. Muster sind höchst individuell und spezifisch für den einzelnen Informationssuchenden, aber sie werden auch beeinflusst durch Domäne und Suchsystem. Für ein bestimmtes Suchsystem und eine feste Domäne kann es also möglich sein, die Erfahrung eines neuen Nutzers etwa durch Scaffolding (siehe Abschnitt 3.3) derart zu beeinflussen, dass bestimmte, als effektiv bekannte Muster verinnerlicht werden.

Die nächst feinere Granularität nach Marchionini ist jene der Strategie. Diese sieht er auf ähnlicher Ebene wie bei Bates angeordnet und beschreibt sie als „those sets of ordered tactics that are consciously selected, applied, and monitored to solve an information problem“ [MARCHIONINI 1995]. Diese Strategien können sehr spezifisch sein und auf ein bestimmtes Problem abgestimmt, oder eher grob, allgemein und flexibel. Sehr grob lassen sich Strategien in sogenannte *analytical strategies* und *browsing strategies* einteilen, wobei analytische Strategien geplant und browsende Strategien eher opportunistisch und interaktiv sind. Eine feinere, aber immer noch sehr breite Einteilung von Strategien wären die *Information Seeking Strategies* nach Belkin und Kollegen [BELKIN et al. 1993], kategorisiert nach den vier identifizierten Facetten oder Dimensionen der Informationssuche (siehe den folgenden Abschnitt). Während Muster nutzerspezifisch sind, sind Strategien eher suchspezifisch.

Auf den Stufen der feinsten Granularität finden sich dann die bereits von Bates identifizierten Taktiken und *Moves* (Schritte) wieder. Taktiken sind für Marchionini diskrete intellektuelle Entscheidungen und fokussierter als Strategien. Taktiken nutzen dabei oft Inferenzen über die Suche, um zum Beispiel eine Suche weiter einzuschränken und sie weiter voranzubringen. Schritte sind letztlich feine, diskrete Aktionen, wie das Eingeben eines Suchwortes, das Bewegen eines Sliders, der Mausklick auf einen Knopf oder die Auswahl eines Objektes aus einer Auswahlliste. Schritte sind stark systemspezifisch und konzeptionell analog zu den Operatoren des GOMS-Modells.

### **Vier Dimensionen der Informationssuchstrategien nach Belkin, Marchetti und Cool**

Belkin, Marchetti und Cool [BELKIN et al. 1993, BELKIN et al. 1995, COOL und BELKIN 2002] beschreiben das Verhalten von Benutzern bei der Informationssuche mit Hilfe von *Information Seeking Strategies* (ISSs). Dabei stellten sie fest,

**Tabelle 2.1.** Beispielhafte Suchtaktiken nach Bates [BATES 1979a, BATES 1979b].

<b>Monitoring Tactics (Beobachtungstaktiken)</b>	
RECORD	Ein Übersicht der bereits verfolgten und der noch nicht abgeschlossenen Suchpfade
<b>File Structure Tactics (Strukturtaktiken)</b>	
SELECT	Aufteilung eines komplexen Suchproblems in Teilprobleme
CUT	Aus möglichen Anfragespezifikationen die auswählen, die die Ergebnismenge am stärksten einschränkt
<b>Search Formulation Tactics (Anfragetaktiken)</b>	
SPECIFY	Auswahl möglichst spezifischer Anfragebegriffe
EXHAUST	Initiale Suchanfrage unter Benutzung (fast) aller Begriffe, oder Ergänzung einer Suchanfrage um weitere Begriffe
REDUCE	Initiale Suchanfrage unter Benutzung möglichst weniger Begriffe, oder Reduzierung einer Suchanfrage um Begriffe
PARALLEL	Suchanfrage durch Synonyme oder andere „parallele“ Begriffe erweitern
PINPOINT	Suchanfrage durch Entfernung von parallelen Begriffen (und Beibehaltung der genauesten Begriffe) einengen
BLOCK	Ergebnisse aus der Suchanfrage fernhalten, die durch bestimmte Begriffe beschrieben werden
<b>Term Tactics (Termtaktiken)</b>	
SUPER	Ersetzen eines Begriffs durch allgemeineren Oberbegriff
SUB	Ersetzen eines Begriffs durch spezielleren Unterbegriff
RELATE	Benutzung verwandter Kategorien für die Begriffsauswahl
NEIGHBOR	thematisch oder systematisch benachbarte Begriffe wählen
TRACE	Begriff aus der bereits gefundenen Information wählen
VARY	Variation der Anfrage
FIX	Austausch oder Streichung von Präfix, Suffix oder Infix
REARRANGE	Umstellen der Suchbegriffe
CONTARY	Suche über einen gegenteiligen Begriff (z. B. Krankheit statt Gesundheit)
RESPELL	Ersetzen des Begriffs durch eine Schreibweisenvariante (z. B. Delfin statt Delphin)
RESPACE	Änderung der Trennung von mehrteiligen Suchbegriffen
<b>Idea Tactics (Ideentaktiken)</b>	
BRAINSTORM	Ungefilterte Ideensammlung
CONSULT	Ratschlag durch andere Personen einholen
CATCH	Unproduktives Suchverhalten bemerken und einstellen
BREAK	Suchmuster durchbrechen und ein anderes wählen
SKIP	Ein Teilproblem zunächst auslassen
STOP	Suche zunächst unterbrechen und später fortführen

## 2.3. SUCHSTRATEGIEN

---

dass sich diese Suchstrategien in relativ wenigen Dimensionen unterscheiden. Jede dieser Dimensionen oder Facetten kann im einfachsten Fall zwei verschiedene Werte annehmen, so dass sich durch Kombination der Facetten insgesamt 16 Kategorien für mögliche Informationssuchstrategien ergeben. Eine empirische Bestätigung der Klassifikation wurde durch Studium des Suchverhaltens der Benutzer eines experimentellen IR-Systems gesucht. Die dabei gefundenen Erkenntnisse flossen in das von Belkin [BELKIN 1996] vorgeschlagene Modell des Information Retrievals als Interaktion mit Information ein.

Die **Methode** der Interaktion unterscheidet zwischen Suchen und Scannen. Hier ist eine Parallele zu Marchioninis Unterteilung von Strategien in analytische (*searching*) und browsende (*scanning*) zu erkennen.

Das **Ziel** der Interaktion unterscheidet zwischen Lernen und Selektieren. Dabei ist Lernen als die Erweiterung des Wissens über die Aufgabe, das Problem, das System, seine Ressourcen oder ein Thema definiert. Das Selektieren hingegen hat die Identifizierung oder Auswahl bestimmter (relevanter) Objekte zum Ziel.

Der **Modus** unterscheidet zwischen Erkennen und Spezifizieren. Erkennen identifiziert dabei das Gesuchte anhand der Anzeige, während das Spezifizieren vom Informationssuchenden verlangt, das Gesuchte vorab zu beschreiben. Oft wird Erkennen mit der Methode des Scannens und Spezifizieren mit der Methode des Suchens assoziiert.

Die betrachtete **Ressource** schließlich können entweder die Informationsobjekte selbst sein oder Meta-Information über diese. Meta-Informationen sind dabei Ressourcen, die Struktur und Inhalt von Informationsobjekten oder Quellen beschreiben.

Belkin und Kollegen räumen selbst einige Mängel ihrer Klassifikation ein [BELKIN et al. 1993]. Die Dimensionen sind nicht umfassend und decken nicht alle Eigenschaften von Suchstrategien ab, können aber einen guten Ausgangspunkt für ihre systematische Untersuchung bieten. Zudem stellt sich die Frage der Unabhängigkeit der Dimensionen, insbesondere der Dimension der Methode und jener des Retrievalmodus. Sie argumentieren aber, dass Beispiele für alle Kombinationen gefunden werden können und keine Facette eine andere prädestiniert. Schließlich ist es auch möglich, die Facetten nicht als binär anzusehen, sondern mehr diskrete Werte zu erlauben oder sogar ein Kontinuum von Werten anzunehmen. Unter der Annahme der Unabhängigkeit der Facetten würden die orthogonalen Dimensionen dann einen vierdimensionalen Raum der Suchstrategien aufspannen.

Suchende wenden dabei typischerweise innerhalb einer Suchepisode nicht nur eine ISS an, sondern wechseln im Verlauf der Episode zwischen verschiedenen ISSs. Man kann die Klassifikation daher in der Terminologie von Bates auch als eine facetierte Klassifikation von Suchstrategemen betrachten, zwischen denen sich ein Suchender bewegt und so eine Strategie gestaltet.



### Suchstrategien, interaktive Intentionen und *Shifts* nach Xie

Während bei systemorientierten Ansätzen oder analytischen Suchstrategien meist von einem gleichbleibenden Suchbedürfnis ausgegangen wird, zeigen interaktive oder opportunistische Vorgehensweisen bei der Suche oft Verschiebungen oder *Shifts* in den Benutzerintentionen und den verwendeten Suchstrategien. Xie untersuchte diese Verschiebungen in einer Studie [XIE 2000] und stellte fest, dass Benutzer typischerweise ihr Suchverhalten variieren und mehrere unterschiedliche Muster innerhalb einer Suchepisode aufweisen.

Dabei charakterisiert sie Suchstrategien (im Sinne von Belkin und Kollegen) als Mittel, um interaktive Intentionen zu erreichen. Sie identifiziert Methode und Ressource als die zwei Dimensionen von Strategien, wobei acht mögliche Methoden und sechs Ressourcenarten ausgemacht werden. Die Methoden sind *acquiring, comparing, consulting, scanning, searching, selecting, tracking* und *trial-and-error*. Als Ressourcen werden Mensch, multiple Datenbanken, einzelne Informationsquelle, Informationsobjekt, Teil eines Informationsobjekts (oder spezifische Information) und Meta-Information unterschieden.

Die interaktiven Intentionen sind davon getrennt zu sehen und werden in acht Arten unterteilt: Identifizieren, Lernen, Finden, Zugreifen, Lokalisieren, Bewerten, Protokollieren, Erwerben [XIE 1997].

Aus der Beobachtung von Benutzern verallgemeinerte Xie dann vier Arten intentionaler *Shifts* und drei Arten von *Shifts* in der Informationssuchstrategie, aus denen Vorschläge für die Entwicklung von IR-Systemen zur Unterstützung derartiger Vorgehensweisen abgeleitet wurden.

- Die sogenannten geplanten Verschiebungen (*planned shifts*) interaktiver Intentionen sind Teil einer größeren Suchstrategie des Benutzers. Es wird empfohlen, die häufigsten derartigen intentionalen *Shifts* funktionsell durch das Suchsystem zu unterstützen und den Benutzern durch das System Anleitung bei der Durchführung ihrer geplanten *Shifts* zu geben.
- Opportunistische Verschiebungen (*opportunistic shifts*) treten dann auf, wenn der Benutzer eine Situation während einer Suchepisode erkennt, in der eine anderer Strategie als die bislang verfolgte vielversprechender ist. Ein System sollte den Benutzern dabei helfen, solche Situationen zu erkennen, nützliche opportunistische *Shifts* vorschlagen, die zur aktuellen Situation passen, und den Benutzer wieder zu seiner ursprünglichen Intention zurück geleiten, wenn der opportunistische *Shift* abgeschlossen ist.
- Unterstützte Verschiebungen (*assisted shifts*) repräsentieren Probleme während einer Suche, die von einem IR-System erkannt werden müssen und für die situationsspezifische Unterstützung bereitgestellt werden sollte in Gestalt von Vorschlägen für intentionale *Shifts*.

### 2.3. SUCHSTRATEGIEN

---

**Tabelle 2.2.** Beispielhafte Suchstrategeme nach Bates [BATES 1989].

---

---

Footnote chasing	Der Suchende folgt den Fußnoten oder Referenzen aus einem oder mehreren relevanten Dokumenten. Diese benutzt er, um durch <i>Known-Item-Instantiation</i> weitere relevante Dokumente zu erhalten und dann die Technik gegebenenfalls iterativ anwenden. Der Blick ist dabei in die Vergangenheit gerichtet.
Citation searching	Der Suchende versucht ausgehend von einem als relevant identifizierten Dokument in Zitationsdatenbanken andere Dokumente zu finden, die dieses zitieren. Der Blick ist dabei ausgehend vom ersten Dokument in die Zukunft gerichtet. Wie beim <i>Footnote chasing</i> wird <i>Known-Item-Instantiation</i> benutzt, um die neu identifizierten Dokumente zu erhalten.
Journal run	Kennt der Suchende ein zentrales Journal oder hat es anhand von bereits gefundenen Dokumenten identifiziert, so kann er durch diese Technik weitere relevante Dokumente finden, indem er ein Zeitschriftenverzeichnis nach weiteren Artikeln aus diesem Journal durchsucht. Insbesondere bei Themenausgaben ist dies ein vielversprechendes Strategem.
Area scanning	Der Suchende betrachtet das Umfeld eines bereits als relevant erkannten Artikels. Das könnte physikalischer Natur sein und sich etwa auf das Bücherregal einer Bibliothek beziehen, aber auch andere Organisationsstrukturen können so ausgenutzt werden, etwa Klassifikationssysteme (zur Suche nach Dokumenten mit dem gleichen Klassifikationscode) oder Konferenzproceedings (zur Suche nach Veröffentlichungen aus der gleichen Session).
Subject Search	Der Suchende nutzt Deskriptoren und andere thematische Begriffe aus einem kontrollierten Vokabular, die zu einem relevanten Dokument gehören, um weitere relevante Dokumente zu finden.
Author searching	Der Suchende hat den Autoren eines relevanten Dokumentes oder einen zentralen Autoren eines Gebietes identifiziert und sucht nach weiteren Dokumenten dieses Autors oder seiner Koautoren, um zusätzliche relevante Dokumente zu erhalten.

---

---

- Alternative Verschiebungen (*alternative shifts*) der interaktiven Intentionen treten auf, wenn Benutzer ihre ursprünglichen Intentionen nicht verwirklichen können und daher neue interaktive Intentionen aufgreifen, um ihr Ziel zu erreichen. Die alternativen Verschiebungen der Suchstrategie wechseln stattdessen das Suchverhalten unter Beibehaltung der Intention.
- Wenn Benutzer ihre ursprünglichen Intentionen mit Hilfe der eingesetzten Suchstrategien nur teilweise verwirklichen konnten, kommt es zu ergänzenden Verschiebungen (*supplemented shift*) der Strategie, um auch die anderen Teile der Intention zu erreichen. Verbesserte Verschiebungen (*improved shifts*) der Strategie hingegen modifizieren die ursprüngliche Strategie, um sie besser an die Intention anzupassen.

Einige der von Xie beschriebenen *Shifts* ähneln dabei den von Bates identifizierten *idea tactics*. So basiert eine opportunistische Verschiebung auf einer Anwendung der Taktik BREAK.

### 2.3.2. Konkrete Strategien

Chen und Dhar [CHEN und DHAR 1991] identifizierten zwei Strategien, die besonders häufig von Endnutzern eingesetzt werden, aber dabei als schwache und oft ineffiziente Strategien gelten müssen. In der so genannten *Trial-and-Error*-Strategie wird ohne Kenntnisse des im System benutzten Vokabulars durch Ausprobieren von Suchbegriffen versucht, sich einem Ergebnis zu nähern. Die *Screen-Browsing*-Strategie beschäftigt sich übermäßig lange mit einem angezeigten Ergebnis in dem Bemühen, durch langes Studium Hinweise für die weitere Suche zu finden. Effektiver waren hingegen Strategien, die Wissen über die Domäne, das Vokabular oder Klassifikationsschema oder das System voraussetzen:

- Mit Hilfe von Zitationen und Referenzen kann man durch die Strategie der *Known-Item-Instantiation* zu neuer nützlicher Information in Form von Autoren, Titeln oder Suchbegriffen gelangen. Dies lässt sich dann iterativ anwenden. Dies entspricht den Strategemen *citation searching* und *footnote chasing* bei Bates [BATES 1989].
- Über die Benutzung der unterschiedlichen Suchfelder und Suchoptionen eines Systems kann ausgenutzt werden, dass sich manche Suchoptionen besser für bestimmte Informationsprobleme eignen.
- Über einen Thesaurus und ein Klassifikationsschema lassen sich Suchbegriffe und Kategorien finden, mit denen sich eine effektive Suche gestalten lässt. Hier finden sich Bates' Strategeme *subject searching* und *area scanning* wieder.

## 2.3. SUCHSTRATEGIEN

---

Früh schon wurden andere generelle Suchstrategien identifiziert, die sich in der Praxis als effektiv erwiesen haben. Harter [HARTER 1986] beschreibt eine Reihe solcher grundlegender Strategien: *briefsearch*, *building blocks*, *successive facets*, *pairwise facets*, *multiple briefsearch*, *citation pearl growing*, *interactive scanning*, *implied concepts*, *cited publication*, *cited author*, *co-cited authors*, *non-subject searching*, *fact searching*.

Obwohl sich Harter an Bates' Terminologie orientiert, sind einige der Strategien dabei eher als Strategeme einzuordnen und können als Bestandteile einer größeren Suchstrategie angesehen werden, z. B. *cited author* oder *briefsearch*. Eine *briefsearch* ist dabei als das einfachste Strategem anzusehen und umfasst nach Wagers [WAGERS 1989] folgende Schritte: Auswahl der Datenquelle, Formulierung einer einfachen Anfrage aus wenigen durch Konjunktion und Disjunktion verknüpften Konzepten und das Analysieren der kompletten angezeigten Resultatmenge. Für einfache Anfragen (vor allem im Kontext von Websuchmaschinen, wo Suchbedürfnisse oft einfach und durch die erste Resultatseite befriedigbar sind) ist dies häufig ausreichend, für komplexere Suchaufgaben jedoch sind die Resultatmengen meist zu umfangreich für eine komplette Analyse und die kurze Suche wird als Teilstrategem einer ausgefilterten Strategie benutzt.

Marchionini greift Harters Strategien wieder auf und ordnete sie den analytischen Strategien zu [MARCHIONINI 1995]. Er weist darauf hin, dass in der Praxis oft eine Kombination der Strategien zielführender ist als eine einzelne Strategie. Daneben identifiziert er vier Arten von Browsingstrategien: *Scannen*, *Observieren*, *Navigieren* und *Beobachten*. Marchionini kategorisiert sie mit Hilfe von fünf Dimensionen (Grad der externen und internen Definiertheit, Organisation der Information, Interaktivität der Strategie und kognitiver Aufwand). Damit bietet er eine kognitivere Einteilung von browsenden Strategien als etwa Cove und Walsh, die Browsingstrategien in systematisches, opportunistisches und freies (*casual*) Browsen ordnen [COVE und WALSH 1988].

Marcia Bates beschrieb mit dem „*Berrypicking*“ Model ein neues Modell für das Informationssuchverhalten [BATES 1989]. Die Suchen vieler Benutzer würden durch dieses Modell besser charakterisiert, als durch das bis dahin vorherrschende klassische Modell. Im Kern aber ist Berrypicking eine bestimmte Suchstrategie. Bei diesem Vorgehen setzt der Suchende eine Reihe verschiedener Suchstrategeme ein, und sammelt kontinuierlich neue Informationen und Dokumente, die dabei gefunden werden (dieses stückweise Sammeln gibt dem *Berrypicking* seinen Namen). Jedes neue Stück Information, das dabei gefunden wird, nutzt der Suchende als Quelle neuer Ideen und Richtungen für seine Suche. Dabei kann sich, muss sich aber nicht, das Informationsbedürfnis selbst ändern.

Allgemeine und effiziente Vorgehensweisen zur Informationssuche beschreiben Bhavnani, Drabenstott und Athota [BHAVNANI et al. 2001a] und geben dabei Beispiele für ihre Umsetzung in verschiedenen Arten von Suchsystemen.

men (Bibliothekskataloge, Abstracting- und Indexing-Dienste oder Websuchmaschinen). Die Vorgehensweisen zur Ausnutzen der Regularitäten im Informationsraum entsprechen den Strategemen von Bates. Dabei unterscheiden Bhavnani und Kollegen noch einmal nach emergenten Regularitäten (von einem Dokument referenzierte Autoren, das Dokument zitierende Dokumente, andere Dokumente des gleichen Autoren) und aufgesetzten Regularitäten (relevante Kollektionen, Indexierungsschemata, iterative Verfeinerung einer Begriffssuche, Suche über mehrere Kollektionen und „Erraten“ basierend auf Namenskonventionen).

### **Beschreibung von Strategien:**

Im Folgenden sollen konkrete Strategien beschrieben werden, die von professionellen Informationssuchenden eingesetzt werden.

- Die Strategie der *Building Blocks* ist nach Harter [HARTER 1986] eine der am weitesten verbreiteten und nützlichsten Suchstrategien. Sie identifiziert zunächst die wichtigen Konzepte oder Facetten eines Informationsproblems, sowie die logischen Beziehungen zwischen den Konzepten. Für jede Facette werden dann Suchbegriffe ermittelt, die diese beschreiben oder repräsentieren, mit Hilfe von Thesauri, Klassifikationen, Domänenwissen oder durch kurze Suchen (*briefsearches*). Ebenso wird ermittelt, welche Suchfelder eines Formulars für das jeweilige Konzept am geeignetsten sind. Die einzelnen Aspekte einer Facette werden durch Disjunktion zusammengeführt, die verschiedenen Facetten durch Konjunktion. Dabei sollten bereits bei der Planung Alternativen bedacht werden, um interaktiv die Suche basierend auf den tatsächlichen Ergebnissen geeignet zu modifizieren.
- Erzielt ein bestimmtes Konzept der Suche sehr wenige Treffer und führt eine Strategie der *Building Blocks* zu leeren oder sehr kleinen Ergebnismengen oder ist ein Teilkonzept sehr schwammig und mehrdeutig, so dass viele irrelevante Treffer erzielt werden, so empfiehlt Harter eine Strategie der *Successive Facets* oder *Successive Fractions* [MEADOW und COCHRANE 1981]. Dabei wird bei den *Successive Facets* entweder zunächst mit dem spezifischsten Konzept bzw. dem mit den wenigsten zu erwartenden Ergebnissen begonnen. Nur bei Bedarf werden weitere Konzepte nach und nach hinzugenommen bis das Ergebnis handhabbar geworden ist. Dies spart Zeit gegenüber der vollständig geplanten Strategie der *Building Blocks* und ist von höherer Interaktivität gekennzeichnet. Bei der verwandten Strategie der *Successive Fractions* werden auch Dokumentarten, Publikationsjahr oder Sprachen für die Einschränkung genutzt.
- Das *Pearl Growing* ist weniger algorithmisch als die vorangegangenen Strategien und erfordert mehr Interaktion zwischen System und Benut-

## 2.3. SUCHSTRATEGIEN

---

zer. Die Suche erfordert zunächst die Identifikation eines einzelnen relevanten Dokumentes. Dies kann dem Benutzer bekannt und mittels *Known Item Instantiation* gefunden worden sein, oder muss zunächst z. B. durch eine kurze Suche (*briefsearch*) ermittelt werden. Hat der Suchende ein solches Dokument gefunden, wird es benutzt, um weitere Dokumente zu finden. Dabei werden Deskriptoren, Identifikatoren, Klassifikationsbegriffe, Titelsterme oder häufige Wörter aus dem Text des relevanten Dokumentes genutzt, um Anfragen zu formulieren und weitere relevante Dokumente zu finden. Dies kann dann iteriert werden, indem die neuen Dokumenten wiederum als Quelle genutzt werden. In einer Variante oder als Komplement der Strategie, von Harter separat als Strategie der *Cited Publication* beschrieben [HARTER 1986] aber eigentlich als Teilstrategie anzusehen, werden stattdessen oder zusätzlich die Referenzen der anfänglichen „Perle“ genutzt, um sie wachsen zu lassen (hier kommt also das Strategem des *Footnote chasing* nach Bates zum Einsatz). Gleichfalls lassen sich auch Zitationen nutzen, wenn ein Zitationsindex zur Verfügung steht.

- Das *Interactive Scanning* wurde von Hawkins und Wagner vorgeschlagen [HAWKINS und WAGNER 1982] und wird als Strategie oft von Novizen benutzt. Während die bisherigen Strategien nach Marchionini als analytische Strategien bezeichnet werden können, ähnelt diese browsenden Strategien. Wie Harter ausführt, ist es eine der zeitaufwändigsten und iterativsten Strategien zur Informationssuche. Die Strategie beginnt mit einer sehr breiten Suchanfrage, die einen möglichst großen Teil der relevanten Dokumente umfasst. Aus der Ergebnismenge wird dann durch Extraktion und Analyse ein besseres Bild des Suchfelds gewonnen. Zu häufige und wenig bedeutsame Begriffe können ausgeschlossen und genauere Begriffe hinzugenommen werden. Dies wird solange iteriert, wobei in jedem Schritt relevante Dokumente identifiziert und gespeichert werden, bis ein zufriedenstellendes Ergebnis erzielt wurde. Diese Strategie kann durch geeignete Visualisierungen der Dokumentenmenge unterstützt werden, etwa durch *Map Displays* [LIN 1997].
- Eine Strategie des *Closed-Loop Relevance Clustering* wird von Vigil [VIGIL 1983] beschrieben und bedient sich des NOT-Operators, um sukzessive ungewünschte oder redundante Dokumente auszuschließen. Dabei wird die Ursprungsanfrage modifiziert, um neue Dokumente zu erhalten, und anschließend die Differenz mit den bereits eingegrenzten Dokumenten ermittelt. Dies wird solange fortgesetzt, bis eine neue Anfrage keine oder nur noch sehr wenige relevante Dokument erzielt.
- Das *Scanning* ist die grundlegende der browsenden Strategien, die eine Auswahlliste mit einem mehr oder minder wohldefinierten Bild eines Suchproblems im Geiste des Suchenden abgleicht. Dabei kann zwischen dem linearen und dem selektiven Scanning unterschieden werden, wo-

bei ersteres sequentiell eine Liste durchgeht und letzteres eine inhärente Ordnung der Liste ausnutzt, um den Suchraum zu unterteilen. Lineares Scanning ist als Taktik besonders geeignet, wenn die Objekte im Suchraum präzise und schnell identifizierbare Eigenschaften besitzen und der Suchraum relativ klein ist. Bates' Strategem des *Area Scanning* könnte als Beispiel für eine Scanning-Strategie angesehen werden.

- Beim *Navigieren* als browsender Strategie schränkt die Umgebung das Browsen durch vorgegebene Wege ein, und Nutzer wählen unter den möglichen Wegen an jeder Verzweigung. Dabei hat der Informationssuchende ein Ziel im Auge, das aber nicht notwendigerweise am Ende des Weges zu finden ist. Tatsächlich kann der „Weg das Ziel“ sein und sich die gesuchte Information entlang des navigierten Weges finden lassen. Bates' *Berrypicking* kann als navigierende Browsingstrategie angesehen werden. Dabei weist Bates allerdings selbst darauf hin, dass *Berrypicking* und Browsing als Suchaktivität nicht identisch sind, sondern Browsing zusammen mit anderen Techniken innerhalb einer *Berrypicking*-Episode eingesetzt werden kann.

### 2.3.3. Konkrete Suchtaktiken

Insbesondere zur interaktiven Ausgestaltung von Suchstrategien wichtig sind die dabei benutzten Taktiken [BATES 1979a, BATES 1979b, MARCHIONINI 1995], Heuristiken [HARTER und PETERS 1985, HARTER 1986] oder *Moves* [FIDEL 1985]. Im folgenden werde ich den Begriff Taktik für eine solche, die Suche voranbringende Aktivität verwenden. Bates stellte früh eine Taxonomie solcher Taktiken zusammen, die in Tabelle 2.1 dargestellt sind. Dabei unterscheiden sich die von ihr identifizierten Taktiken stark in ihrer Spezifität. Während einige sehr abstrakt bleiben, sind andere sehr konkret und spezifisch. Eine empirische Validierung der Klassifikation fand allerdings nicht statt.

Fidel [FIDEL 1985] stellt eine Sammlung von Taktiken zur Suche in Online-Suchsystemen zusammen, die bei einer Studie von sieben erfahrenen Recherchenden über die Dauer von etwa 90 natürlichen Suchen beobachtet wurden. Dabei unterscheidet sie zwischen operationalen und konzeptionellen Taktiken. Operationale Taktiken nutzen unterschiedliche Suchfelder und Möglichkeiten eines Suchsystems, während konzeptionelle Taktiken die Bedeutung des Gesuchten durch die Auswahl der Suchbegriffe variieren. Die Einordnung der Taktiken in die Kategorien ist zum Teil durchaus angreifbar. So beschreiben *Narrow 1* (Tab. 2.4) und *Weight 2* (Tab. 2.3) fast identische Taktiken in den beiden Kategorien. Daneben unterscheiden sich die Taktiken in ihrer Zielsetzung, wobei drei Ziele ausgemacht wurden: eine zu kleine oder leere Ergebnismenge zu vergrößern, eine zu große Ergebnismenge einzuschränken, und eine thematisch nicht das Informationsbedürfnis treffende Ergebnismenge zu korrigieren.

### 2.3. SUCHSTRATEGIEN

---

Für die meisten der Taktiken nach Fidel lassen sich leicht Korrespondenzen zu Bates' Taktiken finden. Einige sind direkte Umsetzungen für die Onlinesuche, andere konkrete Beispiele eher abstrakter Taktiken nach Bates'. Eine Einordnung von Fidels Taktiken in Bates' Taxonomie findet sich in den Tabellen 2.3 und 2.4. Beide Klassifikationssysteme wurden zur empirischen Untersuchung von Suchmustern genutzt, etwa durch Wildemuth und Moore zur Evaluation der Sucheffektivität von Endnutzern [WILDEMUTH und MOORE 1995].

Eine detaillierte Taxonomie von über hundert aus der Literatur extrahierter Suchtaktiken stellten Harter und Rogers-Peters [HARTER und PETERS 1985] vor. Dabei sind diese spezifischer auf die zur Zeit der Veröffentlichung verbreiteten bibliographischen IR-Systeme mit boolescher Anfragesprache ausgelegt als Bates' allgemeine Taktiken. Taktiken werden hier in sechs Kategorien unterteilt:

- Taktiken der „philosophischen Einstellung“ und Herangehensweise des Suchenden sind allgemeine Taktiken, die beispielsweise Flexibilität, Interaktivität, Skeptizismus gegenüber den Ergebnissen des Suchsystems (etwa bei leeren Ergebnismengen), Bereitschaft zur Analyse und zur Nutzung von Zwischenergebnissen, oder das Vertrautmachen mit den Möglichkeiten des Suchsystems fordern. Insgesamt 13 Taktiken werden unter dieser Kategorie zusammengefasst, die teilweise mit den Ideentaktiken von Bates korrespondieren.
- Taktiken, welche die Sprache der Problembeschreibung zum Gegenstand haben, entsprechen den Termtaktiken von Bates. Die 26 Taktiken sind weiter unterteilt in die generelle Entwicklung der Suchbegriffe, das Arbeiten mit Akronymen und Abkürzungen, mit Schreibweise- und Nutzungsvarianten, mit zusammengesetzten Begriffen und mit Klassifikationscodes.
- Die Gruppe der auf Eintrags- und Dateistruktur basierenden Taktiken umfasst 12 Taktiken und ist schon von der Namensgebung an die *file structure tactics* bei Bates angelehnt. Die Taktiken basieren auf Wissen über die Organisation der Information innerhalb der Datenquellen, den verfügbaren Suchfeldern und den Einschränkungen der Systeme.
- Formulierungs- und Reformulierungstaktiken sind in zwei Gruppen unterteilt. Taktiken zur Konzeptformulierung behandeln z. B. die Benutzung des NOT, die Arbeit mit Facetten oder die Warnung vor der Benutzung sehr breiter Konzepte bei der Suche. Reformulierungstaktiken befassen sich mit der interaktiven Anpassung der Anfrage. Zusammen sind 18 Taktiken in dieser Kategorie. Einige sind direkt aus den Ideentaktiken von Bates übernommen, andere sind Anfragetaktiken.
- Taktiken zur Verbesserung von Recall und Precision beziehen sich entweder auf die Spezifität der Konzeptbeschreibungen einer Anfrage oder



**Tabelle 2.3.** Fidels operationale Suchtaktiken in Bates' Taxonomie.

<b>Operationale Taktiken zur Einschränkung der Ergebnismenge</b>		
Weight 1	Deskriptor auf Hauptdeskriptor einschränken	PINPOINT
Weight 2	Freitext-Suche durch Angabe eines breiten Deskriptor einschränken	EXHAUST
Weight 3	Freitext-Suche auf bestimmte Felder einschränken	CUT
Weight 4	Abstand zwischen Freitext-Termen festlegen	REARRANGE
Weight 5	Auf bestimmte Dokumentarten einschränken	CUT
Negate	Unerwünschte Elemente durch AND NOT ausschließen	BLOCK
Limit 1	Dokumente auf eine bestimmte Sprache einschränken	CUT
Limit 2	Dokumente auf eine bestimmte Zeitspanne einschränken	CUT
Limit 3	Dokumente auf Teil der Datenquellen einschränken	CUT
Limit 4	Herkunft der Dokumente auf solche Quellen (z. B. Zeitschriften) beschränken, die einen bestimmten Begriff im Titel haben	CUT
Cut	Nur einen Teil der Ergebnismenge berücksichtigen	—
<b>Operationale Taktiken zur Vergrößerung einer Ergebnismenge</b>		
Add 1	Synonyme und Schreibweisenvarianten hinzufügen	PARALLEL
Add 2	Deskriptoren zur Freitext-Suche benutzen	VARY
Add 3	Deskriptoren aus bereits gefundenen relevanten Dokumenten nutzen	TRACE
Add 4	Begriffe mit vielen Treffern aus Index der Datenquelle hinzufügen	PARALLEL
Include	Alle spezifischeren Deskriptoren zu einem Deskriptor hinzunehmen	PARALLEL
Cancel	Gemachte Einschränkungen aufheben	REDUCE
<b>Operationale Taktiken zur Verbesserung der Genauigkeit</b>		
Refine	Bessere Deskriptoren finden	SPECIFY

auf die Vollständigkeit einer Suchformulierung. Die Spezifität kann entweder innerhalb einzelner Suchfacetten verändert werden oder es können Facetten entfernt oder hinzugenommen werden. Die Vollständigkeit der Formulierung kann durch Einschränkung oder Verbreiterung verändert werden. Die meisten der insgesamt 20 Taktiken entsprechen Anfrage- oder Termtaktiken.

- Letztlich werden noch 12 Taktiken zur Reduzierung der Suchkosten oder zur Verbesserung der Sucheffizienz aufgeführt.

Die von Harter und Rogers-Peters vorgestellten Taktiken sind insgesamt deutlich anwendungsnäher als Bates' Taktiken, dadurch jedoch teilweise auch eng an eine bestimmte Art von Suchsystem gebunden. Zahlreiche Taktiken lassen sich nicht direkt auf volltextbasierte Suchsysteme mit Ergebnisranking und eingeschränkter Unterstützung boolescher Algebra übertragen.

## 2.4. Zusammenfassung

In diesem Kapitel wurden die Grundlagen des Suchverhaltens behandelt. Dabei wurden einige Modelle vorgestellt, um das Suchverhalten von Benutzern zu beschreiben. Die Bedeutung unterschiedlicher Suchaufgaben für die Ausgestaltung und ihre Bedeutung für eine eventuelle Unterstützung der Suche wurden besprochen. Suchstrategien beschreiben ein zielgerichtetes Benutzerverhalten und sind nach Bates die höchste Ebene der Suchaktivität. Sie umfassen alle Taktiken und Strategeme, die zur Lösung einer bestimmten Suchaufgabe notwendig sind. Eine Reihe bekannter und erfolgreicher Strategien aus der Bibliothekswissenschaft wurden kurz beschrieben.

Suchtaktiken sind ein zentraler Baustein von interaktiven Suchstrategien. Der Suchende muss nach jeder Aktivität seine Situation neu evaluieren und die vielversprechendste Suchtaktik zur Verfolgung seiner Strategie auswählen oder seine Strategie ändern. Dies bietet den Ansatz, um Suchende durch geeignete und situativ angemessene Vorschläge zu unterstützen. Drei Taxonomien für Suchtaktiken wurden vorgestellt, von denen Bates' die allgemeinste ist und sich am flexibelsten auf beliebige Suchsysteme übertragen lässt. Daher werde ich diese Taxonomie in Kapitel 6 nutzen, um mögliche Vorschläge für ein automatisches Unterstützungssystem einzuordnen.

---

**Tabelle 2.4.** Fidels konzeptionelle Suchtaktiken in Bates' Taxonomie.

---

<b>Konzeptionelle Taktiken zur Vergrößerung einer Ergebnismenge</b>		
Expand 1	Breiteren Deskriptor wählen	SUPER
Expand 2	Suchbegriffe gruppieren, um die abgedeckte Bedeutung zu erweitern	PARALLEL
Expand 3	Deskriptor mit äquivalentem Rollenindikator gruppieren	VARY
Expand 4	Teile der Anfrage nur über Rollenindikator spezifizieren	VARY
Expand 5	Breitere Konzepte zu spezifischem Ergebnis hinzunehmen	PARALLEL
Exclude	Zu weit verbreitete Konzepte aus der Formulierung entfernen	BLOCK
<b>Konzeptionelle Taktiken zur Einschränkung einer Ergebnismenge</b>		
Intersect 1	Ursprungsanfrage mit Anfrage zu anderem Teilkonzept schneiden	EXHAUST
Intersect 2	Ergebnis durch Rollenindikator einschränken	VARY
Narrow 1	Eine Anfrage unter Nutzung von Deskriptoren durch Freitext-Begriffe einschränken	EXHAUST
Narrow 2	Deskriptor durch Rolle qualifizieren	VARY
<b>Konzeptionelle Taktiken zur Verbesserung der Genauigkeit</b>		
Probe 1	Eine Anfrage zum Erkunden der Indexbegriffe erstellen	SCAFFOLD
Probe 2	Unterschiedliche Verteilung von Suchbegriffen in einzelnen Datenquellen für Begriffsauswahl nutzen	SURVEY

---

## 2.4. ZUSAMMENFASSUNG

---

# Kapitel 3.

## Benutzerunterstützung

### 3.1. Suchsituationen

Um Benutzern optimale Unterstützung für die Planung und Durchführung ihrer Informationssuche oder bei der Entwicklung einer geeigneten Strategie zur Befriedigung ihres Informationsbedürfnis geben zu können, ist es notwendig, die aktuelle Situation des Benutzers zu betrachten. Eine Suchsituation ist ein spezifischer Moment oder eine spezifische Zeitspanne während der Suche aus Sicht des Systems oder aus der Wahrnehmung des Suchenden. Suchsituationen wurden in einer Reihe früherer Arbeiten definiert.

Pharo [PHARO 2002, PHARO 2004] präsentiert das *Search Situation Transition Schema* als ein Modell für Informationssuchverhalten. Dabei definiert er Situationen für die Web-Informationssuche als eine Periode der Interaktion mit „echten Daten“ oder Dokumenten, während Suchübergänge oder Transitionen Perioden der Interaktion mit Metadaten sind. Die Nützlichkeit dieses methodischen Werkzeugs für die Analyse von Informationssuchverhalten wird im Kontext von Websuchmaschinen gezeigt.

Suchsituationen und -übergänge sind nach Pharo durch eine Reihe gemeinsamer Attribute beschrieben. Von den hier genannten Charakteristika sind nur Relevanzurteile und Relevanzgrad typisch für Suchsituationen, während alle anderen Attribute in beiden Fällen zur Beschreibung genutzt werden:

- Aktionen während der Situation,
- die akkumulierten Ergebnisse,
- die akkumulierte Mühe (*effort*) des Benutzers,
- der Informationsraum oder die aktuelle Suchdomäne, wie sie der Benutzer wahrnimmt,
- die Zeit,
- verbleibende Informationsbedürfnisse,
- verfügbare Ressourcenarten,

### 3.1. SUCHSITUATIONEN

---

- technische Probleme,
- Relevanzurteile und Relevanzgrad.

Bei Brajnik et al. [BRAJNIK et al. 2002] ist eine Situation ein bestimmter Zustand (*state*) einer Suchsitzung, der auf eine Serie von Aktionen (entweder des Benutzers oder des Systems) folgt. Sie unterscheiden zwei distinkte Arten von Situation: kritische Situationen (*critical situations*), in denen es keinerlei Suchfortschritte gibt, und verbesserbare Situationen (*enhanceable situations*), die nicht kritisch sind, in denen ein Benutzer aber durch Systemunterstützung Nutzen ziehen könnte. Eine Situation wird charakterisiert durch

- die Anfrage,
- die gefundenen Dokumente,
- die gelesenen Dokumente,
- die als relevant oder nützlich klassifizierten Dokumente,
- und die letzten Benutzer- oder Systemaktionen.

Pharo betrachtet sowohl Suchübergänge als auch Suchsituationen als Zeitspannen, die sowohl Benutzer- als auch Systemaktionen umfassen können. Der Unterschied zwischen beiden liegt allein im Objekt der Interaktion (Metadaten oder Daten). Brajnik und Kollegen hingegen sehen eine Situation als einen Zeitpunkt an. Diese Situation kann dann durch Aktionen verändert werden, welche als Übergänge zwischen zwei Suchsituationen anzusehen sind.

Ebenfalls als Zeitpunkte in einer Informationssuche werden Suchsituationen von Iris Xie [XIE 2000, XIE 2007] behandelt. Sie erweitert eine frühere Klassifikation von Schutz und Luckmann, die Routinesituationen und problematische Situationen identifizierten. In einer Routinesituation kann die Situation durch den Suchenden ausreichend durch Anwendung erworbenen Wissens erfasst und mit dem Plan (d. h. der Strategie) für die Suche abgeglichen werden. In einer problematischen Situation hingegen fehlen dem Suchenden wichtige Aspekte für die volle Erfassung der Situation. Eine Anpassung und Neubewertung der benutzten Taktiken und Aktivitäten ist erforderlich. Als Folge einer problematischen Situation muss es zu einer Neubestimmung der Suchstrategie kommen. Xie führt nun zusätzlich eine dritte Situationsart ein, die sie „störende Situation“ nennt. Störende Situationen ergeben sich im Kontext des interaktiven IR, wobei der Benutzer des Systems nicht auf ein Problem stößt, aber von seinem ursprünglichen Suchplan bzw. seinem ursprünglichen Suchziel abgelenkt wird.

In dieser Arbeit soll der Begriff einer Suchsituation in folgendem Sinne verwendet werden: es ist ein bestimmter Punkt in einer Suche, der durch Aktionen des Benutzers und des Systems erreicht wurde. Eine Situation setzt sich aus dem aktuellen Zustand der Suche und der Historie der aktuellen Suchsitzung zusammen (zu Suchhistorien siehe Abschnitt 4.1). Ignoriert man die

Suchhistorie, kann man auch den aktuellen Zustand der Suche isoliert als eine vereinfachte Situation betrachten (insbesondere gilt dies für den Beginn einer Suchsitzung, zu dem noch keine Suchhistorie existiert).

Ein Zustand wird durch System- und Benutzerparameter charakterisiert. Für ein komplexes Suchsystem mit unterschiedlichen Werkzeugen oder Suchmöglichkeiten könnte ein Zustand folgende Systemparameter besitzen, die aufgrund ihrer Natur relativ einfach durch Logging erhoben werden können:

- das zuletzt benutzte Such- oder Browsing-Werkzeug, sowie die Domäne auf der das Werkzeug operiert (z. B. Metadaten, Journale, Thesaurusbegriffe, Autoren, usw.),
- die benutzten Informationsquellen,
- die Suchbegriffe der Anfrage oder das gewählte Objekt im Falle des Browsens,
- die Anzahl benutzter Suchbegriffe,
- die gefundenen Objekte (z. B. Dokumente),
- die Anzahl gefundener Objekte pro Informationsquelle,
- die benötigte Zeit für die Ausführung der Aktion (pro Quelle),
- ggf. die Retrievalgewichte der gefundenen Objekte,
- die klassifizierten Objekte (betrachtet und bewertet, wobei die Bewertung durch implizites Feedback erfolgen kann, d. h. etwa durch Speichern oder Nicht-Speichern des Objekts).

### 3.1.1. Benutzerparameter einer Suchsituation

Da zu Beginn einer Suchsitzung, d. h. vor der Durchführung von Benutzeraktionen, viele der Parameter einer Suchsituation für alle Benutzer des Systems identisch sind, kann gerade hier die Betrachtung von Nutzerparametern zusätzliche Einblicke bringen, die für strategische Vorschläge genutzt werden können.

Benutzerparameter stellen eine größere Herausforderung dar und ihre Erhebung, z. B. die Erhebung der „unerfüllten Informationsbedürfnisse“ (nach Pharo) ist ein offenes Problem, trotz ihrer Bedeutung für die Definition einer Suchsituation und insbesondere für die Bereitstellung von strategischer Hilfe, die auf die Bedürfnisse individueller Benutzer angepasst ist. Einige Parameter über den Benutzer und sein Informationsbedürfnis könnten durch Betrachtung des Umfelds der Suchsituation bzw. unter Einbeziehung früherer Suchhistorien des Benutzers zumindest angenähert werden.

So stellen zum Beispiel die akkumulierten gespeicherten Dokumente (sei es lokal, in einem gemeinsamen Dokumentenrepositorium oder in Form von

### 3.1. SUCHSITUATIONEN

---

Browser-Lesezeichen) oder frühere Suchsituationen eine Möglichkeit dar, langfristige Interessen des Benutzers zu beschreiben. Die akkumulierten Dokumente der aktuellen Suchhistorie können eventuell Aufschluss über ein aktuelles Informationsbedürfnis geben. Die Zeit und die Anzahl der Schritte, die ein Benutzer bisher für eine Sitzung oder eine Anfrage aufgewandt hat, lassen Rückschlüsse über seine Hartnäckigkeit und möglicherweise über seinen bereits erreichten Grad der Frustration mit dem System oder der aktuellen Suche zu.

Ist der Benutzer aber ein neuer Benutzer des Systems oder hat er bisher das System für gänzlich andere Zwecke genutzt, greifen auch diese Methoden zu kurz.

Nur sehr schwierig zu erfassende, aber potentiell wichtige Aspekte, die eine Situation bzw. einen Zustand aus Nutzersicht beschreiben, sind:

- das Ziel des Benutzers (z. B. das Anfertigen eines Reviews für einen Artikel, das Schreiben einer umfangreichen Arbeit, das Sammeln von Quellen für eine Bibliographie),
- das Stadium des Benutzers innerhalb seiner Arbeitsaufgabe (z. B. Problemdefinition, Fokussieren, Sammeln, Abschluss, Verifizierung),
- das Verständnis des Benutzers von seiner Aufgabe (z. B. verschwommen oder klar); dies ist nach Kuhlthau ([KUHLETHAU 1991, KUHLETHAU 1993]) nicht unabhängig vom Aufgabenstadium des Benutzers und hängt mit dem Grad der Unsicherheit zusammen, die der Benutzer für die Aufgabe verspürt,
- die Komplexität der Suchaufgabe (z. B. Faktensuche, Wiederfinden eines bekannten Dokuments, Lösung eines Problems, Suche zum Stützen oder Treffen einer Entscheidung, Suche zur Gewinnung eines thematischen Überblicks, zieloffene Erkundung),
- die Anzahl der Ergebnisse, die der Benutzer zur Befriedigung seines Informationsbedürfnisses erwartet,
- das Wissen des Benutzers über die Suchdomäne und die Suchbegriffe,
- die Zeit, die der Benutzer willens und in der Lage ist, in die Bearbeitung der Suchaufgabe zu investieren (dies hängt mit Motivation, Geduld und Hartnäckigkeit des Benutzers zusammen [PHARO 2002]).

Kann man diese Aspekte etwa durch Befragung des Benutzers während oder vor seiner Suche erheben, oder durch das System abschätzen, so könnten sie als zusätzliche Parameter zur Beschreibung der Situation herangezogen werden und strategische Hilfestellung unterstützen. Beispielsweise ist eine langwierige Suche kein unbedingter Indikator für eine problematische Situation des Benutzers, wenn dieser zur Erreichung eines Suchziels bereit ist oder erwartet, erhebliche Zeit aufzuwenden.



### 3.1.2. Generische Situationen

Für die Nutzung von Suchsituationen zur Unterstützung von Benutzern beim Finden und Durchführen von Strategien ist ein gewisser Grad der Abstraktion von den konkreten Werten der Zustandsparameter notwendig.

In einer spezifischen Situation können mehrere generische Situationen (oder Typen von Situationen) gelten, z. B. können unterschiedliche Teilmengen der Situationsparameter auf unterschiedliche Situationstypen hindeuten, die als Grundlage für strategische Hilfestellung dienen können. Je nach Ähnlichkeit der aktuellen Situation zu vorgegebenen generischen Situationen können dann verschiedene Lösungen angeboten werden.

Ein Beispiel für eine solche generische Situation wären Situationen, in denen die letzte Aktion eine Suchaktion war, die keine Resultate zurückgeliefert hat. Diese generische Situation kann ggf. durch andere Parameter des Zustands oder durch die Suchhistorie, die zu ihr geführt hat, noch näher aufgeschlüsselt werden, da es eine Reihe von Gründen geben kann, die zu einer derartigen Situation führen können:

- Überspezifikation der Anfrage,
- Auswahl der falschen Informationsquelle(n),
- technische Probleme des Systems,
- ein ungeduldiger Benutzer bricht die Suche ab, bevor sie Resultate liefern konnte.

[TIPS 1999] nennt eine Reihe weiterer häufiger generischer Situationen, die auf eine Vielzahl verschiedener Suchsysteme übertragbar sind, darunter: zu viele Resultate bei wenigen spezifizierten Suchbegriffen; durchweg niedrige Retrievalgewichte bei allen Suchergebnissen; identische oder fast identische Retrievalgewichte bei allen Suchergebnissen; keine Benutzung von Strategemen (siehe Abschnitt 2.3) während der bisherigen Suchsituation.

### 3.1.3. Aktionen und Übergänge

Aktionen des Benutzers oder des Softwaresystems können den Zustand des Systems ändern und damit potentiell zu einer neuen Situation führen. Dabei lassen sich die Aktionen in verschiedene Kategorien einordnen, wie in Tabelle 3.1 zu sehen. Hilfestellung kann entweder nur nach bestimmten Arten von Aktionen gegeben werden, bei denen davon auszugehen ist, dass sie die Situation des Benutzers wesentlich verändert haben (z. B. das Durchführen einer Suche, siehe hierzu Abschnitt 6.1), oder nach jeder Benutzeraktion (siehe hierzu Abschnitt 6.2).

Dabei führen bestimmte Aktionen besonders häufig zu Situationen, in denen Unterstützung durch das System erforderlich ist. Während nach einer Studie

### 3.1. SUCHSITUATIONEN

---

**Tabelle 3.1.** Ordnungssystem für Benutzer- und Systemaktionen.

---

---

**Aktionen, die eine Anfrage modifizieren**

- Hinzufügen, Entfernen, Ersetzen von Begriffen
- Hinzufügen, Entfernen, Ersetzen eines booleschen Operators
- Neue Anfrage (Ersetzen der kompletten Anfrage)

**Aktionen zur Suche oder zum Browsen**

- Absenden einer Anfrage
- Navigation einer Hierarchie
- Verfolgen eines Hyperlinks
- Benutzung eines Thesaurus

**Aktionen zur Manipulation eines Ergebnisses**

- Blättern in einem Ergebnis
- Sortieren des Ergebnisses
- Extrahieren von Begriffen aus Dokumenten
- Filtern eines Ergebnisses

**Aktionen zur Manipulation von Objekten**

- Betrachten eines Dokuments, Scrollen innerhalb eines Dokuments
- Speichern oder Löschen eines Dokuments
- Verschieben eines Dokuments
- Annotieren eines Dokuments

**Hilfsaktionen**

- Benutzung des Hilfesystems
- Lesen von Online-Dokumentation
- Bitten anderer Benutzer um Hilfe
- Anfrage nach Systemvorschlägen

**Verlaufsaktionen**

- Abspeichern von Anfragen
- Wiederausführen von Anfragen
- Abspeichern oder Rückholen eines Systemzustands
- Betrachten des Suchverlaufs
- Planen nächster Schritte

**Systemaktionen**

- Nicht durch den Benutzer angestoßene Systemvorgänge
- 
- 

von Xie und Cool [XIE und COOL 2009] Browsing nur zu 4,2% aller Situationen führte, in denen vom Benutzer Hilfe gesucht wurde, gehen problematischen Situationen am häufigsten das Modifizieren (41,5%) oder Erstellen einer neuen Anfrage (18%) voran. Ähnliche Ergebnisse zeigt eine Studie von Jansen, der untersucht hat, wann im Suchprozess und nach welchen vorangegangenen Aktionen Benutzer Hilfestellung suchen und wann sie diese implementieren [JANSEN 2005]. Gerade hier kann also ein System ansetzen, das Benutzer dann unterstützen soll, wenn diese die Unterstützung am meisten benötigen.

## 3.2. Strategische Unterstützung

Zahlreiche Studien haben gezeigt, dass viele Benutzer mit grundlegenden Befehlskenntnissen trotz Erfahrung nicht von sich aus in der Lage sind, eine effiziente Nutzung komplexer Computeranwendungen zu entwickeln [BHAVNANI und JOHN 2000]. Die Autoren erklären dies damit, dass Wissen über die Aufgabe und Wissen über die Werkzeuge nicht ausreicht, um ein effizienter Nutzer komplexer Systeme zu werden. Stattdessen müsse ein Benutzer auch Strategien in der Ebene zwischen Aufgabe und Werkzeuge erwerben.

Fields und Kollegen [FIELDS et al. 2004] untersuchten die Wichtigkeit der unterschiedlichen Arten von Wissen, die notwendig für das Finden von Informationen sind, und stellten dabei fest, dass insbesondere das Suchwissen, über das beispielsweise Bibliothekare und andere Suchexperten verfügen, den „normalen“ Benutzern von Informationssystemen fehlt. Dies verhindere, dass Digitale Bibliotheken heute tatsächlich eine reiche Informationsressource mit Zugriff für alle Nutzer darstellen.

Dabei gibt es zahlreiche Belege, dass die existierende Hilfe von Benutzern in problematischen Situationen oft nicht genutzt wird, da sie vielfach zu systemorientiert ist, zu sehr auf die Benutzung von Funktionalitäten des Programms und zu wenig auf die Unterstützung in der konkreten Situation und auf Interaktion mit dem Benutzer ausgelegt ist [XIE und COOL 2009]. Es ist daher wichtig, das Augenmerk auf die Probleme zu legen, die gerade Anfänger und wenig erfahrene Nutzer bei ihrer Informationssuche haben und auf die Strategien, die sie dabei zur Lösung einsetzen.

### 3.2.1. Notwendigkeit

Wie Fenichel [FENICHEL 1980] bereits darlegte, liegen die Probleme, die unerfahrene Endbenutzer haben, meist nicht in den Fähigkeiten des Systems, sondern in den von ihnen benutzten Suchstrategien. Sie wissen nicht, wie sie die passenden Suchstrategien unter den verschiedenen Umständen anzuwenden haben. Dadurch beschränkt sich ein substantieller Teil der Benutzer auf einfache Suchen und nutzt nur wenige der Möglichkeiten des Suchsystems.

Unerfahrene Benutzer haben oft Probleme bei der Suche mit Informationssystemen, da sie nur ein schwaches Verständnis von effektiven Suchstrategien besitzen [BRAJNIK et al. 1996]. Sie sind kaum in der Lage selbstständig Suchstrategien zu planen und durchzuführen [STELMASZEWSKA und BLANDFORD 2002]. In Studien zeigt sich, dass Novizen meist nur einfachste Strategien einsetzen: sie benutzen in ihren Anfragen nur Begriffe, die ihnen gerade einfallen, ohne zuvor zu planen, zusätzliche Quellen zu Rate zu ziehen oder durch Interaktion mit den Suchquellen einen besseren Eindruck über gute Begriffe zu erlangen [FENICHEL 1980, SULLIVAN et al. 1990, HSIEH-YEE 1993, CHEN

### 3.2. STRATEGISCHE UNTERSTÜTZUNG

---

und DHAR 1991]. Fields, Keith und Blandford [FIELDS et al. 2004] stellten fest, dass unerfahrene Benutzer typischerweise Schwierigkeiten haben, Suchbegriffe oder Phrasen zu finden, um die Essenz ihres Informationsbedürfnisses auf eine Weise zu beschreiben, die ein gutes Ergebnis erzielt. Suchexperten hingegen wissen um die Notwendigkeit, gute Suchbegriffe zu ermitteln, und verfügen über Strategien, ihr Domänenwissen dahingehend zu erweitern.

Anfragemodifikationen und -verfeinerungen werden von Novizen nur selten taktisch zielführend eingesetzt. Ihnen steht nur eine sehr begrenzte Menge an Aktionen oder Taktiken zur Verfügung. Einschränkungen der Suche über zusätzliche Felder werden auch da, wo diese Möglichkeiten vorhanden sind, nicht vorgenommen [BRAJNIK et al. 1996, KRIEWEL und FUHR 2010]. Gefundene Dokumente werden nicht ausgiebig genutzt, wodurch unerfahrene Benutzer eine wichtige Möglichkeit zur Verbesserung ihrer Suche ignorieren. Selbst wenn Benutzer die gefundenen Dokumente untersuchen, übersehen sie oft, dass sie relevante Dokumente nutzen können, um so weitere Begriffe für die Reformulierung ihrer Anfrage zu gewinnen [SULLIVAN et al. 1990, MARCHIONINI 1989]. Außerdem neigen viele Nutzer dazu, eine einzelne Suchstrategie zu wählen und an dieser für eine gesamte Suche oder sogar für alle Suchen, die sie durchführen, festzuhalten [MANGLANO et al. 1998], selbst wenn diese nicht für die aktuelle Suchsituation (das Informationsbedürfnis, die Suchaufgabe) geeignet ist. Dazu geben unerfahrene Benutzer bei Problemen eine ungewohnte Strategie schnell auf und kehren zu gewohnten Vorgehensweisen zurück [TWIDALE et al. 1995] – auch wenn diese sich zuvor als eindeutig ineffektiv erwiesen haben, beenden die Suche ohne zufriedenstellende Ergebnisse gefunden zu haben oder führen bei kritischen Situationen unsinnige oder unnötige Aktionen aus [MARCHIONINI 1989].

Diese Beobachtungen bestätigten sich auch in weiteren Studien und Experimenten mit Benutzern [SPINK et al. 2001, KLAS et al. 2004b]. Brajnik schließt daraus, dass Benutzer nicht nur terminologisches Wissen und Wissen über die Anfragesprache benötigen, sondern dass zur erfolgreichen Unterstützung von Suchenden auch strategische Hilfe notwendig ist, die das Portfolio der dem Benutzer zur Verfügung stehenden Strategien erweitert [BRAJNIK 1999].

Schon früher merkte Marchionini an, dass die praktischste und vielversprechendste Möglichkeit zur Verbesserung von Informationssuchsystemen auf der Ebene der Benutzerschnittstelle die bessere Unterstützung von Taktiken sei [MARCHIONINI 1995]. Dabei ließen sich etwa in Studien identifizierte Herangehensweisen (Muster, Strategien und Taktiken) von professionellen Informationssuchenden in Systeme einbauen, welche die Ausführung dieser optimieren. Über die bloße Ermöglichung oder bessere Ausführung von Taktiken (oder auch Strategemen nach Bates) hinaus benötigen Nichtexperten jedoch auch Anleitung darin, wann sie diese zielführend einsetzen können, da ihnen die Heuristiken fehlen, die erfahrenen Suchenden zur Verfügung stehen. Von Marchionini durchgeführte Studien [MARCHIONINI 1989] zeigten, wie schwie-

rig analytische Strategien für Suchnovizen zu erlernen und anzuwenden sind. Bei der Evaluation von Suchschnittstellen, die sich auf die Unterstützung von analytischen Strategien beschränkten und präzise Hilfe bei deren Ausführung gaben, konnten Endnutzer zwar erfolgreich suchen, aber die Suchen waren unnötig komplex und oft kognitiv belastend [BORGMAN et al. 1989]. Auch vertrauen Suchnovizen vielfach übermäßig den Vorgaben und Voreinstellungen des Systems. Obwohl analytische Strategien oft effizienter und effektiver sein können als informelle, sind sie selbst bei längerer Benutzung von Suchsystemen für Endbenutzer schwierig zu erwerben. Ohne Unterstützung neigen Endbenutzer zur Anwendung naiver Strategien. Bei allen außer den einfachsten und schematischsten Suchproblemen ist es notwendig, abduktives Schließen einzusetzen.

Daraus leitet sich der Bedarf für Systeme ab, die informelle, interaktive und opportunistische Informationssuchstrategien für Endbenutzer unterstützen. Darüber hinaus sollte aber auch das Wissen darüber vermittelt werden, welche Grenzen bestimmte Taktiken oder Strategien besitzen. Insbesondere müssen Informationssuchende beachten, welche der zahlreichen in der Literatur identifizierten Heuristiken an den verschiedenen Stellen einer Informationssuche einzusetzen seien. Hier kann ein System ansetzen, um Suchende zu unterstützen.

Bhavnani und Kollegen [BHAVNANI et al. 2003] heben die Bedeutung von prozeduralem Suchwissen zur Unterteilung einer Suchaufgabe in geordnete Teilaufgaben oder Teilziele hervor. Experten entwickeln solche Teilziele zur Organisation der Suche basierend auf ihrem Verständnis der Informationsquellen, des Suchsystems und der verfügbaren Werkzeuge. Da dieses prozedurale Wissen aber von gebräuchlichen Suchwerkzeugen nicht vermittelt wird und durch den Benutzer nicht leicht zu inferieren ist, müssen Suchnovizen solche Teilziele durch Ausprobieren finden. Dieser Vorgang ist sowohl zeitaufwendig, als auch fehleranfällig. Ziel eines unterstützenden Suchsystems sollte daher sein, Suchprozeduren oder -strategien explizit zu machen.

Nach Belkin [BELKIN 2000] ist es wichtig, den Benutzer in das IR-System zu integrieren, etwa indem das System eine Mittlerrolle zwischen Benutzern und Texten einnimmt. Eine wichtige Aufgabe des Systems besteht darin, die Charakteristika von Benutzern zu verstehen, die problematische Situation des Benutzers zu erkennen, und basierend darauf, passende Texte zu identifizieren, mit denen der Benutzer in Interaktion treten kann, um seine problematische Situation zu lösen.

Koenemann und Belkin [KOENEMANN und BELKIN 1996] verglichen drei Information-Retrieval-Systeme mit unterschiedlichem Angebot an Interaktion. Dabei wurde eine Studie mit 64 Studierenden der Rutgers University durchgeführt, bei der unter anderem die Frage untersucht werden sollte, wie unterschiedliche Stufen der Interaktivität das Informationssuchverhalten der Benutzer beeinflussen, insbesondere gemessen an der Anzahl der Anfragen

## 3.2. STRATEGISCHE UNTERSTÜTZUNG

---

und der Art der Anfrageformulierung. Die Suchzeiten unterschieden sich zwar nicht signifikant, aber mit steigenden Möglichkeiten zur Benutzerinteraktion erwies sich die Arbeit der Suchenden mit dem System als effizienter und einfacher, bei gleichbleibender oder steigender Effektivität.

In einer Meta-Analyse der Ergebnisse von Endnutzerstudien aus 25 Jahren der Forschung betrachtet Markey interventionsfreie Studien über nutzerinitiierte Suchen und kommt zu dem Schluss, dass die Suchen von Endnutzern keine Ähnlichkeit zu jenen von Suchexperten aufweisen [MARKEY 2007a]. Die meisten Studien zeigen auf, dass Nutzer selten fortgeschrittene Suchmöglichkeiten über die Phrasensuche hinaus bei ihren Suchen einsetzen. Rudner stellte dazu fest, „not only is the typical end-user doing a poor job of searching, they are not taking advantage of the available tools“ [RUDNER 2000].

Markey schlägt IR-Systeme vor, die Nutzern verdeckt die Suchmöglichkeiten des Systems und gute Suchstrategien nahebringen, während diese Nutzer das System für ihre normalen Aktivitäten nutzen [MARKEY 2007b]. Dabei seien die drei Stufen der Suchaktivität, wie sie Bates vorgeschlagen hat (siehe Abschnitt 2.3), ein guter Ansatzpunkt, um die Probleme von Nutzern zu identifizieren und ihnen auf ihre Bedürfnisse abgestimmte Unterstützung zu geben. Ohne Unterstützung neigen Endbenutzer dazu, schwache Suchbegriffe für ihre Suchen zu wählen, was durch die niedrige Präzision des Suchergebnisses zur Folge hat, dass fortgeschrittene Systemwerkzeuge zur Ausnutzung des Ergebnis nicht zum Einsatz kommen. Fehlendes strategisches Wissen führt zu einer hohen Zahl an Fehlern, unstrukturiertem Wechseln zwischen Suchbegriffen, Wiederholungen und schnellen Abbrüchen von Strategien.

Es ist also für die meisten Endbenutzer ein Problem, strategische Möglichkeiten in ihrer Suche zu erkennen, eine Suche geeignet zu planen und zu gestalten, auf problematische Situationen geeignet zu reagieren und die angebotenen Optionen des Suchsystems sinnvoll einzusetzen. Sie sind sich der fortgeschrittenen Möglichkeiten funktionsreicher IR-Systeme oft nicht bewusst, oder wissen nicht, wann oder wie sie diese effektiv einsetzen können. Suchsysteme, die automatisierte strategische Hilfestellung anbieten, können durch Vorschlag dieser fortgeschrittenen Möglichkeiten und Optionen die Sucheffektivität von Endbenutzern steigern [JANSEN 2005].

### 3.2.2. Existierende Systeme

Ein frühes System zur Unterstützung von Endbenutzern bei der Informationssuche mit einem Online-Suchsystem wurde von Meadows, Hewett und Aversa beschrieben und evaluiert [MEADOW et al. 1982a]. IIDA (*Individualized Instruction for Data Access*) bestand aus zwei separaten Komponenten: einem Einweisungsmodus, der Such- und Systemwissen mit Hilfe von *Computer Aided Instruction* vermitteln sollte, sowie einen Unterstützungsmodus, der bei der tatsächlichen Suche eines Benutzers zum Einsatz kam. Bei der Umsetzung

beschränkte man sich auf eine sehr spezifische, eingeschränkte Zielgruppe als mögliche Benutzer des Systems: computervorgebildete Wissenschaftler und Ingenieure, die über technisches Vokabular, aber kein gezieltes Vorwissen über das Suchsystem verfügen. Dabei wurde angenommen, dass die Zielgruppe das System nur ein- bis zehnmal pro Jahr und nur für einfache Aufgaben nutzen würden. Umfangreiche Suchen mit hoher Abdeckung relevanter Literatur wurden explizit außen vorgelassen.

In der Evaluation [MEADOW et al. 1982b] ergab sich, dass Nutzer des Unterstützungssystems mit den Ergebnissen ihrer Suchen ähnlich zufrieden waren wie Nutzer, die ihre Suche durch Experten hatten ausführen lassen. Auch zeigten Benutzer, die durch den Einweisungsmodus in die Suche mit dem System eingeführt wurden, eine vergleichbare Leistung zu Benutzern, die eine herkömmliche Einführung erhalten hatten.

Ein Versuch zur Umsetzung von spezifisch strategischen Vorschlägen stammt von Smith et al. [SMITH 1989]. Sie setzten dabei domänenspezifische, wissensbasierte Suchtaktiken als Vorschläge ein, die durch die Taktiken und Heuristiken von Bates und Harter (siehe Abschnitt 2.3) und durch eine empirische Studie der Interaktion von Suchenden mit Suchmittlern beeinflusst wurden. Die Vorschläge wurden für EP-X, ein wissensbasiertes Expertensystem für die Online-Suche [KRAWCZAK et al. 1987], modelliert. Eine Umsetzung fand in Form automatischer Reformulierung von Anfragen statt. Die Benutzer des Systems erfuhren nur marginale Verbesserungen der Precision in ihren Anfragen, brauchten aber signifikant weniger Anfragen für ihre Aufgabe [GAUCH und SMITH 1991, GAUCH und SMITH 1993].

Brajnik und andere [BRAJNIK et al. 2002] beschreiben ein prototypisches strategisches Hilfesystem, das auf kollaborativem Coaching basiert und versucht, Benutzer durch Vorschläge und Hinweise in sogenannten kritischen oder verbesserbaren Situationen während der Informationssuche zu unterstützen. Dabei stand die Unterstützung bei der Anwendungen strategischer Optionen im Vordergrund. Das System arbeitet mit einer handgefertigten Wissensbasis aus 94 Produktionsregeln basierend auf den von Bates [BATES 1979b, BATES 1990] vorgeschlagenen Taktiken, sowie einigen Strategemen. Das strategische Hilfemodul wurde in FIRE integriert, ein Benutzerinterface für ein boolesches IR-System [BRAJNIK et al. 1996]. Ein Benutzerexperiment mit sechs Teilnehmern wurde durchgeführt, das tendenziell die generelle Nützlichkeit strategischer Vorschläge bestätigte. Weitergehende, umfangreichere Experimente wurden entweder nicht durchgeführt oder nicht veröffentlicht.

Anders als in EP-X agiert das Hilfemodul dabei als ein Problemlösungs-Partner des Benutzers; der Benutzer muss im Dialog mit dem System die Aufgaben bearbeiten. Das System soll dem Benutzer nicht nur vermitteln, welche Aktionen er durchführen kann, sondern auch wie diese arbeiten und warum sie vom System in dieser Situation ausgewählt wurden. Dadurch soll der Benutzer ein Verständnis dafür entwickeln, wie in bestimmten Situationen vor-

### 3.2. STRATEGISCHE UNTERSTÜTZUNG

---

gegangen wird und welche Aktionen zur Lösung von Problemen eingesetzt werden können. Dieses Wissen soll dann vom Benutzer auf andere Suchen übertragen werden können. Das System zeigt aber kein größeres Bild der Suche auf und schlägt auch keinen Suchplan über den nächsten Schritt hinaus vor.

Treten bei der Suche Probleme auf oder sind die Ergebnisse der Suche nicht erwartungskonform, so werden zum weiteren Vorgehen Hilfestellungen etwa durch Vorschläge zur Termmanipulation, Wortstammreduktion oder zur Benutzung des Thesaurus gegeben. Das System bietet darüber hinaus in verbesserbaren Situationen autonom ohne ausdrücklichen Benutzerwunsch Hilfestellungen an. Nach dem Konzept des kollaborativen Coaching bleibt die Kontrolle über die Suche dabei beim Benutzer. Dieser gibt das Problem vor, entscheidet aktiv über angebotene Lösungsschritte und kontrolliert den Lösungsweg. Durch diese Zusammenarbeit von System und Benutzer soll erreicht werden, dass die aktuelle Suchstrategie verbessert wird und sich durch die vorgeschlagenen Optionen des Systems bei dem Benutzer ein Grundverständnis für die Durchführung einer strategischen Suche bildet.

Die Entscheidung, welche Vorschläge für die aktuelle Situation sinnvoll sein können, erfolgt durch manuell aufgestellte Regeln. Eine Regel setzt sich aus fünf Komponenten zusammen [BRAJNIK et al. 2002]:

**Name:** ein eindeutiger Name, der der Regel gegeben wurde,

**Kommentar:** eine Menschen verständliche Beschreibung der Regel,

**Hypothese:** die Bedingung für die Anwendung der Regel,

**Aktion:** eine auszuführende Aktion, wenn die Regelbedingung erfüllt ist,

**Aktivierungsebene:** dieser Wert legt fest, für welche Benutzergruppen die Regel geeignet ist.

Der Entscheidungsprozess, welche Regeln in der aktuellen Situation angewandt werden, hängt von verschiedenen Faktoren ab: den bisherigen Aktionen des Benutzers, dem aktuellen Zustand des Systems bezogen auf die Suche und den bereits vom System gemachten Vorschlägen. Beispiele für solche Regeln und deren terminologischen Grundlagen werden von Armellini und anderen [ARMELLINI et al. 2000] beschrieben. Das Projekt FIRE wird nicht weiter entwickelt, aber Teile der Arbeit sind als *Information Retrieval Assistant* (IRA) in das „Tools for Innovative Publishing in Science Project“ [TIPS 1999] eingeflossen.

Belkin et al. [BELKIN et al. 1993] schlagen ein System mit gemischter Initiative vor, das spezifische Prozeduren bereit stellt, um Benutzer während Informationssuchepisoden zu unterstützen, die sich im Laufe der Interaktion mit dem Benutzer ändern und verzweigen. Das System basiert auf den von Belkin beschriebenen Informationssuchverhalten oder -strategien (ISS). Im Verlaufe einer Suchepisode wechseln Benutzer mehrfach zwischen einzelnen ISS.



Mit MERIT wurde schließlich ein System präsentiert [BELKIN et al. 1995], das auf diesen Konzepten basiert. Das System benutzt Interaktionsskripte, die mit Hilfe von Techniken des fallbasierten Schließens gewählt werden, um Benutzer durch eine Informationssuchepisode zu führen. Der Suchprozess wird als konversationsartige Interaktion zwischen System und Benutzer präsentiert.

Der Benutzer formuliert zunächst sein Problem, wobei er zu Beginn der Dialogsitzung entweder aus vorgegebenen Skripten wählen kann oder unterstützt durch den Dialogmanager (CADI) einen Fall aus der Fallbasis benutzt. Im Weiteren werden abhängig vom konkret gewählten Fall Anfrageformulare angeboten, die vom Benutzer modifiziert und gefüllt werden können. Bei der Auswahl der Anfragekonzepte erfährt der Suchende dabei Unterstützung durch ein „*Knowledge Explorer*“ genanntes Modul, das konzeptuelles Domänenwissen berücksichtigt und terminologische Vorschläge zur Erweiterung der Anfrage machen kann. Dabei basieren die Vorschläge auf positiver und negativer Assoziation zwischen Konzepten, sowie Generalisierungs- und Spezialisierungsbeziehungen.

Die komplette Benutzerführung basiert auf dem Konversationsmodell. Ein Benutzer kann über den CADI einen beschrifteten Dialogpfad speichern und damit in die Fallbasis übernehmen. Die Dialoge selbst greifen für die Interaktion zwischen System und Suchenden auf die ISS zurück und nutzen Skripte, die mit spezifischen Regionen des ISS-Raums assoziiert sind.

Bhavnani et al. [BHAVNANI et al. 2003, BHAVNANI et al. 2006] präsentieren die Idee eines *Strategy Hubs*. Ein Strategy Hub bietet domänenspezifische Suchprozeduren oder Strategien für die Websuche an. Suchmaschinen haben Probleme, Benutzer bei der Suche nach umfassender Information zu unterstützen, da in vielen Domänen keine einzelne Webseite existiert, die sämtliche und vollständige Informationen zu einer Fragestellung anbietet. Daher ist es für Benutzer wichtig, zusätzlich zu Systemwissen und Domänenwissen auch prozedurales Suchwissen zu erlangen, damit sie in der Lage sind, ihre Suche z. B. in Form von geordneten Teilzielen zu strukturieren und zu organisieren. Es besteht der Bedarf, dieses prozedurale Wissen in einem Suchsystem explizit zu machen, um dem Benutzer zu ermöglichen, ein umfassendes Verständnis eines Themengebietes zu erlangen.

Studien zeigten, dass Suchexperten über dieses prozedurale Suchwissen verfügen, und etwa in der Lage sind, mit Hilfe von Suchstrategien ihr Informationsproblem effizient zu unterteilen. Dieses Expertenwissen ist aber für Nichtexperten im Gegensatz zu Systemwissen allein durch die Benutzung herkömmlicher Suchmaschinen schwer zu erlangen. In Zusammenarbeit mit Domänenexperten identifizierten Bhavnani und Kollegen daher eine Reihe kritischer Suchprozeduren (im Bereich der Melanomforschung), und stellten sie mit Hilfe eines Strategy Hubs Nicht-Expertenutzern zur Verfügung. Ihnen gelang es zu zeigen, dass diese Suchprozeduren auf andere Gesundheitsdomänen ebenso wie für neue Domänen generalisierbar und übertragbar sind. Ei-

## 3.2. STRATEGISCHE UNTERSTÜTZUNG

---

ne Pilotstudie, die Strategy Hubs mit konventionellen Suchsystemen verglich, zeigte auf, dass Benutzer durch Strategy Hubs effektiver bei der Bearbeitung komplexer und umfassender Problemstellungen sind, und dabei weniger Seiten aufsuchen, die nicht zielführend sind.

Stelmaszewska, Blandford und Buchanan [STELMASZEWSKA et al. 2005] stellen ein Benutzerinterface für Digitale Bibliotheken vor, das durch Adaption auf die Fähigkeiten des Benutzers das Informationssuchverhalten beeinflussen soll. Dazu wurden zwei Arten von sogenannten Tips in das Interface eines existierenden Digitalen Bibliothekssystem integriert. Anfängertips stellen unflexible, nicht-adaptive Hilfestellungen für Suchnovizen dar, die anhand detaillierter Erklärungen Suchunterstützung geben sollen. Anfragetips sind angepasste Systemreaktionen auf Nutzeranfragen und machen Vorschläge mit Beispielen in Suchsituationen, bei den entweder zu viele oder keine Resultate gefunden wurden. Dabei wurden die Anfragetips simplistisch umgesetzt, indem eine kleine Menge an Voraussetzungen überprüft wurde. Für insgesamt 10 unterschiedliche Situationen, die sich durch Kombination dieser Voraussetzungen ergeben, wurden angepasste Tips gegeben.

Eine formative Evaluation mit insgesamt sechzehn Teilnehmern wurde durchgeführt, von denen je acht mit bzw. ohne Tips arbeiteten. Die Studie setzte ein videobasiertes Beobachtungsprotokoll und ein abschließendes Interview zur Datenerhebung ein und zeigte, dass die unterstützten Benutzer des Systems eine ausgefeiltere Suchstrategie einsetzten als jene, die keine Tips erhielten, ausgewiesen durch einen höheren Anteil von umformulierten gegenüber neuen Anfragen. Der Einsatz von Vorschlägen führte nach Schilderung der Autoren zu einer qualitativ beobachteten Verhaltensveränderung, allerdings wurden keine Unterschiede im Sucherfolg der beiden Gruppen festgestellt.

### 3.2.3. Studien

Ausführliche Studien des Nutzens von automatisierter, strategischer Unterstützung gibt es kaum. Die meisten im vorangegangenen Abschnitt vorgestellten Systeme blieben im prototypischen Stadium. Zwei der wenigen Untersuchungen mit einer nennenswerten Anzahl von Benutzern stammen von Jansen [JANSEN 2005] bzw. Jansen und McNeese [JANSEN und MCNEESE 2005]. Sie führten qualitative und quantitative empirische Untersuchungen eines automatisierten Unterstützungssystems für die Informationssuche durch. Dabei standen zwei Fragen im Vordergrund:

1. Steigert automatisierte Unterstützung die Suchleistung der Benutzer?
2. Wann wünschen und implementieren Benutzer Systemintervention während ihres Suchprozesses?

### Studie 1: Jansen/McNeese 2005

Jansen und McNeese verglichen hierzu zwei IR-Systeme, die bis auf die automatische, clientseitige Suchunterstützung eines der beiden Systeme identisch waren. Insgesamt 40 Studierende aus Veranstaltungen zur Informationswissenschaft und -technologie wurden rekrutiert, und führten auf beiden Systemen jeweils eine 15-minütige Suche zu einer vorgegebenen Aufgabe durch, bei der sie so viele relevante Dokumente wie möglich finden sollten. Die benutzte Kollektion bestand aus ca. 550.000 TREC-Dokumenten und es wurden für die Aufgaben sechs zufällige TREC-Topics gewählt, zu denen 904 identifizierte, relevante Dokumente in der Kollektion zu finden waren.

Für die Untersuchung wurde die Nutzerinteraktion mit dem System mit Hilfe eines Transaktionslogs mitgeschnitten. Zusätzlich wurde eine Videoaufnahme der Suche angefertigt und die Benutzer aufgefordert, ihre Arbeit zu kommentieren (*think aloud protocol*). Die Interaktion zwischen Benutzer und System wurde dann von Hand kodiert, um Muster erkennen zu können.

Das System bot Unterstützung basierend auf der zuletzt ausgeführten Benutzeraktion an:

- bei der Ausführung einer Anfrage
  - **Anfrage-Reformulierung**, wenn mehr als 30 oder weniger als 10 Ergebnisse zurückgeliefert wurden (es wurden dann AND, PHRASE und OR als Operationen vorgeschlagen)
  - **Anfrage-Verfeinerung**, durch Vorschlag von Synonymen zu den aktuellen Suchbegriffen aus einem Thesaurus
  - **Ähnliche Anfragen** aus der Anfragehistorie basierend auf gemeinsamen Anfragetermen
  - **Rechtschreibkorrektur**
- nach einer impliziten Feedback-Aktion (z. B. Lesezeichen setzen, Abspeichern oder Drucken)
  - **Terme** aus dem „relevanten“ Dokument **extrahieren**

Drei Hypothesen wurden zur Frage der Verbesserung der Suchleistung untersucht. Dabei wurden 10 der Teilnehmer für die Untersuchung außen vorgelassen, da sie nie die automatische Unterstützung betrachteten oder umsetzten.

**Hypothese 1a.** Es gab eine signifikante Steigerung der Suchleistung mit dem System, das automatisierte Unterstützung während der Informationssuche anbietet, gegenüber dem System ohne eine solche, gemessen an der Anzahl relevanter Dokumente, die der Benutzer selektiert. Allerdings musste festgestellt werden, dass 50% der Teilnehmer keine Verbesserung erfuhren, und teilweise sogar schlechtere Ergebnisse mit der Unterstützung erzielten.

### 3.2. STRATEGISCHE UNTERSTÜTZUNG

---

**Hypothese 1b.** Es gab keine signifikante Steigerung der *Precision* mit dem System, das automatisierte Unterstützung während der Informationssuche anbietet, gegenüber dem System ohne eine solche, gemessen an der Anzahl relevanter TREC-Dokumente, die der Benutzer in einer einzelnen Anfrage findet.

**Hypothese 1c.** Gemessen an der Anzahl von Benutzern pro Sitzung gefundenen relevanten TREC-Dokumenten gab es ebenfalls keine signifikante Steigerung der *Precision* mit dem System, das automatisierte Unterstützung während der Informationssuche anbietet, gegenüber dem System ohne eine solche. Allerdings erzielten 63% der Teilnehmer mit der automatischen Unterstützung bessere Ergebnisse als ohne.

**Weitere Ergebnisse.** Von insgesamt 322 möglichen Unterstützungen wurden von den Teilnehmern 91 Unterstützungen betrachtet (das entspricht einem Anteil von 28%). Dabei benutzten einige Teilnehmer die Unterstützung nie. Bei abschließenden Befragungen stellten sich zwei Hauptgründe hierfür heraus: eine Gruppe von Benutzern war Hilfestellung grundsätzlich abgeneigt, zog es daher vor, alleine zu arbeiten, und suchte nie von sich aus Hilfe bei ihrer Arbeit; eine zweite Gruppe bemerkte die angebotene Unterstützung nicht, da sie nicht sichtbar genug war. Wurde die Unterstützung betrachtet, so wurde sie allerdings meist (in 71% der Fälle) auch von den Benutzern umgesetzt. Dabei nehmen die Autoren an, dass diese Zahl eine untere Schwelle darstellt, da sie einen Lerneffekt unterstellen.

**Hypothese 2.** Bezüglich der Frage des von Suchenden gewünschten Zeitpunkts der Systemintervention wurden Muster im Suchprozess untersucht. Dazu wurden die kodierten Interaktionslogs der Benutzer herangezogen und die Häufigkeit von 1er- und 2er-Übergängen zwischen Arten von Benutzeraktionen (z. B. Anfrage ausführen, Resultate betrachten, Dokument betrachten, navigieren, Unterstützung ansehen, Unterstützung implementieren) ermittelt. Dabei stellte sich heraus, dass es keine direkte Beziehung zwischen dem Erfolg bei der Suche und der Neigung der Benutzer gibt, Unterstützung bei der Suche anzufordern. Benutzer forderten nicht häufiger Unterstützung an, wenn sie wenige relevante Dokumente fanden, und nicht seltener, wenn sie viele relevante Dokumente fanden.

Bei den beobachteten Mustern zeigte sich, dass Unterstützung meist direkt nach dem Betrachten der Resultatliste angefordert wurde (45%), in jeweils 24,3% der Fälle direkt nach dem Eingeben der Anfrage (noch vor Betrachten der Resultate) bzw. nach dem Betrachten eines möglicherweise nicht relevanten Dokuments. Bei der Auswertung von 2er-Übergängen war das Muster Anfrage → Resultat betrachten → Unterstützung betrachten (ohne zwischendurch einzelne Dokumente genauer zu untersuchen) am häufigsten zu beobachten.

Insgesamt kommen Jansen und McNeese zum Schluss, dass automatisierte Unterstützung aus Nutzersicht hilfreich und nützlich sein kann (1a), aber personalisiert und besser auf die individuellen Unterschiede und Situationen abgestimmt werden muss, da 50% der Benutzer keine Verbesserung ihrer Leistung sahen. Problematisch ist die relativ kurze Dauer der Systembenutzung, die möglicherweise nicht ausreichend ist, um eine wirkliche Verbesserung durch Suchunterstützung zu erzielen.

Über alle Teilnehmer führte die Unterstützung nicht zu besseren Anfragen, allerdings konnte eine Mehrzahl der Suchenden ihre Anfragen mit Hilfe der Suchunterstützung durchaus signifikant verbessern. Hier wird spekuliert, dass die Unterschiede zwischen einzelnen Benutzern größer sind als der Effekt der Suchunterstützung, und dass automatische Suchunterstützung zwar bei der Formulierung präziserer Anfragen helfen kann, aber dazu besser auf die individuellen Benutzerbedürfnisse abgestimmt werden muss. Für die Präsentation der Unterstützung muss ein Kompromiss gefunden werden zwischen der Sichtbarkeit der Unterstützung und der Gefahr, manche Benutzer zu irritieren, die automatische Unterstützung für sich grundsätzlich ablehnen.

### **Studie 2: Jansen 2005**

In einer zweiten Studie [JANSEN 2005] wurden 30 Versuchspersonen (Studenten im zweiten Semester) herangezogen, die mit einem speziell vorbereitetem Suchsystem arbeiten sollten, das automatische Suchunterstützung anbot, wo immer diese verfügbar war.

Die Versuchsteilnehmer sollten dabei Suchen zu einem aus zwei zufällig ausgewählten TREC-Topics durchführen und hatten für ihre Suchen 5 Minuten Zeit. Die Teilnehmer wurden auf die Verfügbarkeit automatischer Unterstützung hingewiesen und es wurde ihnen erklärt, wie sie diese aufrufen können. Dabei wurden sowohl Videoprotokolle als auch Transaktionslogs angefertigt. Diese wurden kombiniert, um ein handkodiertes Suchprotokoll mit einer Granularität von 10 Sekunden pro Interaktionsschritt zu erstellen.

Insgesamt wurde drei Fragen nachgegangen, die für die Implementierung von strategischer Unterstützung bei der Informationssuche relevant sind. Zunächst wurde untersucht, wie oft Benutzer Hilfe suchten und wie oft sie diese gegebenenfalls in ihrer Suche umsetzten. Dabei stellte sich heraus, dass alle Benutzer die Unterstützung mindestens einmal während ihrer Suche anforderten (womöglich nur aus Neugier), aber nur 54% aller 141 angebotenen Unterstützungen wurden tatsächlich von den Versuchsteilnehmern aufgerufen. Von diesen aufgerufenen Angeboten wurden allerdings 82% dann auch von den Teilnehmern umgesetzt. Jansen schließt daraus, dass Suchende zwar möglicherweise dazu tendieren, zunächst selbstständig und ohne Hilfestellung zu suchen, dass sie aber die angebotene Unterstützung, die von ihnen betrachtet wurde, als hilfreich wahrgenommen haben.

Bei der Untersuchung des Zeitpunktes der Unterstützungsanforderung überwog die Hilfesuche zu Beginn des Suchprozesses gegenüber der späteren Phase deutlich. Hier spielten womöglich jedoch der Neuigkeitsfaktor zu Beginn und die relativ kurze Zeit, die für die Aufgabe zur Verfügung stand, eine Rolle. Zusätzlich wurde daher untersucht, wo im Suchprozess Benutzer sich um Hilfe bemüht haben. Dabei suchten sie in 57% der Fälle direkt nach der Durchführung einer Suche und dem Betrachten der Ergebnisse Hilfe, in 25% der Fälle nach der Navigation oder Manipulation der Ergebnisliste. Die Umsetzung des Ratschlags erfolgte meist gleich im Anschluss an den Aufruf der Unterstützung.

Für die Implementierung von strategischen Unterstützungssystemen zeigten die Resultate der Studie, dass Angebote der Unterstützung direkt nach der Aktionsfolge „Anfrage stellen“ → „Resultat betrachten“ am ehesten vom Benutzer angenommen werden. Des weiteren sollten die Resultate der Suche, bzw. Metadaten aus den Ergebnisdokumenten, sowie die Interaktion des Benutzers mit den Ergebnissen genutzt werden, um situationsspezifische, personalisierte und zielgerichtete Hilfestellung zu bieten.

### 3.3. Scaffolding

Zum Abschluss dieses Kapitels soll auf die Möglichkeit eingegangen werden, wie Scaffolding zur Benutzerunterstützung bei der Erlernung und Durchführung von Suchstrategien zum Einsatz kommen kann.

#### 3.3.1. Scaffolding als Lehrmethode

Das *Scaffolding* (englisch für „Gerüst“) ist eine Lehrmethode, die den Lernprozess durch Hilfestellungen begleiten soll, welche den Lernenden befähigen, eine neuartige Aufgabe erfolgreich auszuführen oder ein gewünschtes Ziel zu erreichen. Dabei soll der Lernende zunehmend von der Hilfestellung durch den Lehrenden befreit werden, d. h. die Unterstützung wird nach und nach abgebaut (*Fading*), so wie ein physikalisches Gerüst am Ende von Bauarbeiten, wenn das Errichtete ohne Gerüst stabil ist. Die Hilfestellungen sollen derart gestaltet sein, dass der Lernende dadurch in die Lage versetzt wird, Herausforderungen gerade außerhalb seines eigenen Fähigkeitsstandes (der *Zone of Proximal Development*) zu meistern, und so die Methoden und Fähigkeiten zur selbstständigen Bewältigung nach und nach erwirbt (*learning through activity* [GUZDIAL 1994]).

Scaffolding ist unter anderem charakterisiert durch:

- eine Auswahl und Aufbereitung der Inhalte und Problemstellungen, so dass diese für den Lernenden interessant und für spätere eigenen Auf-

gaben relevant sind; dies kann etwa dadurch erreicht werden, dass *Scaffolding* bei Aufgaben eingesetzt wird, die der Lernende ohnehin zu bewältigen sucht,

- eine Problematik, die zwar über den Fähigkeitsstand des Lernenden hinausgeht, um ihm den Erwerb neuer Fähigkeiten zu ermöglichen, aber mit den Kenntnissen des Lernenden erfassbar und (mit den gegebenen Hilfestellungen) bewältigbar ist,
- ganzheitliches Lernen, bei dem die Aufgabe als Ganzes erlernt werden soll; so wird die Notwendigkeit passiven Wissens und Ableitens durch den Lernenden reduziert und die erlernten Fähigkeiten (prozedurales Wissen) werden leichter verinnerlicht,
- eine direkte Verfügbarkeit von Hilfestellungen, um regelmäßige Erfolge des Lernenden zu ermöglichen,
- Interventionen, wo diese nötig sind, um einen sinnvollen Lernprozess im Gang zu halten und bei Ablenkungen (entsprechend Xies störenden Situationen) wieder zum eigentlichen Lehrziel zurückzuführen,
- modellhafte Vorgabe eines günstigen Prozesses, Verhaltens oder Vorgehens, durch die ein Lehrender eine Lösung demonstriert; Scaffolding wird manchmal mit der Beziehung zwischen Handwerksmeister und Lehrling verglichen.

Die Hilfestellungen des Scaffolding sollen dabei stets ein klares Lehrziel haben, eine natürliche Struktur besitzen, die den Lernenden zum Ziel seiner Aufgabe führt, und dabei die Erkenntnisse während des Lernprozesses berücksichtigen. Die externe Unterstützung nimmt in dem Maße ab, wie der Lernende die benutzten Vorgehensweise annimmt.

Dieses *Fading* ist ein kritischer Bestandteil des Konzeptes [GUZDIAL 1994]. Ist die Methode erfolgreich, so erwirbt der Lernende die Fähigkeit, seine Ziele ohne Scaffolding zu erreichen. Damit er das Vorgehen auch ohne Hilfestellung üben kann, muss diese zwangsläufig zurückgenommen werden. Dabei sollte das Scaffolding auf die Fortschritte des einzelnen Lernenden angepasst, und nur graduell entfernt werden. Schnellere Lernerfolge resultieren in einer stärkeren Reduktion der Gerüstbestandteile.

### **Vygotskys Zone of Proximal Development**

Vygotsky [VYGOTSKY 1978] unterteilt den Fähigkeitsbereich von Lernenden in drei Zonen. Das *Actual Development Level* beschreibt die tatsächlichen Fertigkeiten des Lernenden, auf die dieser zurückgreifen kann. In dieser Zone finden sich Herausforderungen, welche ohne fremde Hilfe gemeistert werden können. Das *Potential Development Level* beinhaltet all die Aufgaben, welche ein

### 3.3. SCAFFOLDING

---

Lernenden nicht selbstständig lösen kann. Die Grenze zwischen diesen beiden Zonen bildet die *Zone of Proximal Development*. Hier sind eben jene Probleme angesiedelt, die ein Lernender zwar nicht alleine zu lösen vermag, zu deren Lösung er aber durch Unterstützung eines Hilfestellenden befähigt werden kann. Dieses Konzept findet sich auch in Kuhlthaus Zonen der Intervention wieder (siehe Abschnitt 2.1).

Meistert ein Lernender die Herausforderungen in der *Zone of Proximal Development* und verinnerlicht das dabei verwendete prozedurale Wissen, so wächst sein *Actual Development Level* und eine neue *Zone of Proximal Development* bildet sich aus. Die zunächst notwendigen Hilfestellungen können dabei nach und nach abgebaut werden.

#### 3.3.2. Software-realisiertes Scaffolding

Wenn das Erlernen von Prozessen in einer Software-Umgebung stattfindet, sollte durch die Umgebung software-realisiertes oder software-basiertes Scaffolding angeboten werden. Hier werden die Mechanismen der Lehrmethode durch die Entwickler der Software anstelle eines Lehrenden vorgegeben, wobei das Ziel des Scaffolding das gleiche bleibt: die Befähigung des Lernenden, eine Aufgabe außerhalb seines direkten Fähigkeitsstands zu bearbeiten und das Erlernen prozeduralen Wissens zu ermöglichen.

Guzdial [GUZDIAL 1994] definiert zu diesem Zweck Scaffolding in den Begrifflichkeiten der Software-Entwicklung. Er unterscheidet außerdem zwischen adaptierbarem Scaffolding (das durch den Benutzer gesteuert und reduziert werden kann) und adaptivem Scaffolding (das sich selbstständig anpasst und zurücknimmt). Adaptives Scaffolding sei dabei vorzuziehen, aber erheblich schwieriger in der Umsetzung.

Software-realisiertes Scaffolding ist beispielsweise mit Erfolg umgesetzt worden in der Programmierumgebung „Emile“ [GUZDIAL 1994]. Der Prozess der Aufgabe wurde dabei strukturiert und kommuniziert, bei der Durchführung des Prozesses Hilfestellungen gegeben und der Lernende durch die bereitgestellten Werkzeuge dazu befähigt und angehalten, sich sein eigenes Vorgehen explizit zu machen (explizites Beschreiben des Ziels, des Plans und der Ergebnisse). Im „IR Game“ [SORMUNEN et al. 1998] wird software-basiertes Scaffolding für die Unterrichtung von Informationssuchkompetenz eingesetzt.

Quintana et al. [QUINTANA et al. 2000, QUINTANA et al. 2002] untersuchten Evaluationskriterien und Richtlinien für das Scaffolding in lernerzentrischen Werkzeugen und Softwareumgebungen auf der Basis von „Symphony“, einer integrierten Werkzeugumgebung mit Scaffolding. Dabei stellten sie als wichtige Maßstäbe für die Beurteilung des Erfolgs von software-basiertem Scaffolding die initiale Zugänglichkeit des Werkzeugs, die Effizienz und Genauigkeit



der Aufgabenausführung durch die Lernenden, sowie die Befähigung zur eigenständigen Arbeit und Reflektion über die gelernten Abläufe heraus.

Eine weitere Umsetzung erfolgte durch Quintana und Zhang [QUINTANA und ZHANG 2004a, QUINTANA und ZHANG 2004b] in Gestalt des digitalen „IdeaKeepers“, einem Software-Werkzeug für die Informationsarbeit in Digitalen Bibliotheken. IdeaKeeper unterstützt bei der Planung einer Nachforschung, beim Suchen, Analysieren und Interpretieren von Information, sowie bei der Synthese der Information zu einer Argumentation. Ein zentrales unterstützendes Element ist dabei der Aktivitäts-*Sidebar*, der die Informationsarbeit strukturiert und einen Überblick über bereits unternommene Aktivitäten gibt.

### 3.3.3. Unterstützung bei Suchstrategien mit Scaffolding

Software-realisiertes Scaffolding kann einen Ansatz bieten, um mit den strategischen und taktischen Aspekten der Informationssuche unvertrauten Endnutzern die Möglichkeit zu geben, erfolgreich zu suchen und während der Bearbeitungen ihrer Suchaufgaben das prozedurale Wissen zu erlangen, das sie für zukünftige Aufgaben befähigt, diese ohne Hilfe von Scaffolding auszuführen.

Nach Guzdial [GUZDIAL 1994] lassen sich grob zwei Ebenen von Scaffolding unterscheiden. Auf der Makroebene wird der Lernende befähigt, strategische Entscheidungen zu treffen und einen Plan für seine Aktivität zu entwerfen und durchzuführen. Auf der Mikroebene hingegen werden Lernende bei den taktischen Entscheidungen unterstützt, die in komplexen Aufgaben notwendig sind. Beide Ebenen sind notwendig für eine erfolgreiche Suche und sollten beim Einsatz von Scaffolding für die Informationssuche zum Einsatz kommen. Halttunen beschreibt den erfolgreichen Einsatz software-basierten Scaffoldings in klassischen Unterrichtsszenarien bei der Vermittlung von Informationssuchkompetenz [HALTTUNEN 1993, HALTTUNEN 1995].

In Kapitel 7.2 wird die Implementierung eines solches Scaffolding-Konzepts vorgestellt.

## 3.4. Zusammenfassung

Suchsituationen beschreiben den aktuellen Stand der Suche eines Benutzers. Abhängig von den Parametern, die bei der Beschreibung einer Situation benutzt werden, geben sie einen mehr oder weniger guten Einblick in das Stadium des Suchenden im Suchprozess und erlauben, Suchunterstützung zielgenau auf den Benutzer und seine Situation abzustimmen.

Dabei herrscht große Übereinstimmung in der Literatur, dass Endbenutzer von Suchsystemen strategischer Unterstützung in Form von Ratschlägen insbesondere in problematischen, aber auch in nur verbesserbaren Situationen

### 3.4. ZUSAMMENFASSUNG

---

benötigen. Eine Reihe prototypischer Systeme sind zu diesem Zweck entwickelt, aber nur selten evaluiert worden, darunter FIRE oder MERIT. Obwohl die Grundidee der Unterstützung vielversprechend ist, zeigte die Studie von Jansen und McNeese, dass es notwendig ist, Unterstützung individuell auf den Benutzer und seine Situation angepasst anzubieten, um größtmöglichen Nutzen zu erzielen.

Das Konzept des Scaffolding aus der Erziehungswissenschaft kann einen Ansatzpunkt bieten, strategisches oder prozedurales Wissen in einem Suchsystem explizit zu machen, und so Endnutzern erlauben, Suchstrategien jenseits ihres normalen Fähigkeitshorizontes einzusetzen. Durch individuell angepasstes *Fading* werden die Hilfestellungen dabei nach und nach bei eintretendem Lernerfolg zurückgenommen.

## Kapitel 4.

# Suchlogs und Suchhistorien

Dieses Kapitel soll einen Überblick über die Bedeutung der Interaktionshistorien von Benutzern für die strategische Informationssuche geben. Dabei sind zwei Verwendungen zu unterscheiden. Zum einen kann die Aufzeichnung und Visualisierung von Suchhistorien Benutzer bei der Erstellung, Kontrolle und Bewertung ihrer Suchstrategien unterstützen. Zum anderen erlauben die Suchhistorien anderer Benutzer und die Gewinnung von Suchpfaden die Unterstützung von Suchenden bei ähnlichen Aufgaben.

### 4.1. Suchhistorien

Wie eingangs beschrieben, kann Informationssuche unter kognitiven Gesichtspunkten als Problemlösungsaktivität gesehen werden. Dabei spielen beim Problemlösungsverhalten externe Repräsentationen des Problems eine wichtige Rolle, wie auch Planung und Evaluation der Aktivitäten. Suchhistorien können den Informationssuchenden dabei unterstützen [KOMLODI 2000]. Suchhistorien sind nach Komlodi Aufzeichnungen der Interaktionen zwischen Benutzern und Suchsystem während der Informationssuche. Sie enthalten Schritte der Benutzer, Systemantworten, gefundene Dokumente und andere Informationen wie z. B. Annotationen von Benutzern in zeitlicher Abfolge [KOMLODI 2002].

Marchionini weist auf die Analogie von Suchhistorien zu den Erfahrungen der realen Welt hin. Wenn ein Suchender systematisch Bücher in einem Regal untersucht, kann er ein Gefühl dafür gewinnen, wie weit fortgeschritten er bereits ist. Ähnlich kann man in einem Buch zurückblättern, um zu resümieren, was man bereits gelesen hat, oder das aktuell Gelesene neu in den Kontext des bereits Rezipierten einzuordnen. Im gleichen Sinne seien Suchsysteme zu sehen, die Benutzern Suchhistorien bereitstellen, über die sie einen Einblick in den Fortschritt ihrer Suche erhalten und früher getätigte Aktionen erneut ausführen können [MARCHIONINI 1995]. Suchhistorien erlauben, die aktuelle Suchsituation in einen Kontext einzuordnen.

Shneiderman schlägt zur Unterstützung kreativer Arbeit mit Information als Teil einer informationsreichen Benutzerschnittstelle das Aufzeichnen der In-

## 4.1. SUCHHISTORIEN

---

teraktionshistorie von Benutzern vor, um diesen zu erlauben, die Historie später einzusehen und nachzuvollziehen [SHNEIDERMAN 1999].

Auch Hearst [HEARST 1999] betont die Wichtigkeit von Verlaufsinformationen in Benutzerschnittstellen von Informationssystemen. Das Interface muss Suchenden einen Überblick über bereits unternommene Schritte erlauben und so die Möglichkeit bieten, kurz- oder langfristige Suchstrategien früherer Suchen einzusehen. Über die Darstellung der getätigten Aktionen kann ein Benutzer des Informationssystems seine Suchstrategie überprüfen und an Änderungen seiner Ziele anpassen. Obwohl Nutzerstudien die Notwendigkeit von Suchhistorien in IR- und Informationsvisualisierungs-Systemen aufgezeigt hätten, würden sie von verfügbaren Systemen nur schlecht unterstützt.

Plaisant und andere [PLAISANT et al. 1999] heben die Wichtigkeit von Aktivitätshistorien bei kollaborativen Lernsimulationen hervor und postulieren, dass Historien auch bei Aktivitäten wie der Informationssuche in Digitalen Bibliotheken sowohl Lernende wie auch Experten unterstützen können. Historien erlauben es, im Nachhinein die durchgeführten Schritte zu begutachten, und geben so die Möglichkeit zu einem besseren Verständnis über Erfolg und Misserfolg bestimmter Strategien, zur Konsultation mit Außenstehenden, zur Reflexion und Annotation der Aktivitäten und zum Erkennen von wiederkehrenden, erfolgreichen Mustern, die als Strategeme verinnerlicht werden können.

Wexelblat und Maes [WEXELBLAT 1999, WEXELBLAT und MAES 1999] beschreiben ein Framework zum Einsatz von Interaktionshistorien in Benutzerschnittstellen zur Unterstützung sozialer Navigation. Dabei definieren sie Interaktionshistorien als bestehend aus Sequenzen von Aktionen, Beziehungen zwischen Elementen, mit denen Benutzer interagiert haben, und den dadurch entstandenen Strukturen. Sequenzen sind zeitabhängige Aktionen, die eine zielgerichtete Variation darstellen. Insgesamt identifizieren Wexelblat und Maes sechs Eigenschaften von Interaktionshistorien.

**Kognitive Nähe** zwischen Benutzern und dem Raum der Interaktionshistorien: Proxemische Räume sind für den Benutzer transparent und die benutzten Zeichen und Strukturen leicht verständlich, distemische Räume sind für Benutzer opak und Zeichen werden übersehen, weil den Benutzern das Wissen fehlt, sie in eigene Erfahrungswerte zu übersetzen.

**Initiative** der Historienbildung: Die meisten Historien sind passiv und werden automatisch durch ein System aufgezeichnet, aber auch aktive Mitarbeit des Benutzers bei der Gestaltung der Historie ist möglich.

**Änderungstempo und -art** der Interaktionshistorie: Zum einen können Historien bezüglich der Granularität und „Abtastrate“ der aufgezeichneten Aktivitäten unterschieden werden, zum anderen darin, wie sich neue Informationen auf die bereits angesammelten auswirken (z. B. könnten

bestimmte Aktivitäten aggregiert werden und sind dann nicht mehr als einzelne Einträge der Historie verfügbar).

**Grad der Durchdringung** der Benutzerschnittstelle und der Informationsobjekte: Die Interaktionshistorie kann untrennbarer Teil einzelner Informationsobjekte sein oder aber völlig separat davon.

**Bezug** der Historie auf persönliche oder soziale Interaktionen: Eine persönliche Historie bietet den Einblick in die vom Benutzer selbst bei der Interaktion mit einem Suchsystem getätigten Aktionen, während soziale Historien neue Möglichkeiten der Navigation und sozialen Nutzung eröffnen (siehe den nächsten Abschnitt über „Suchpfade“).

**Informationsart**, die durch die Historie erfasst wird: Dabei ist wichtig, was getan wurde, womit (mit welchen Objekten), wer etwas getan hat (bei einer persönlichen Historie sind die beteiligten Akteure der Benutzer und das System), warum es getan wurde und wie.

Daneben haben eine Reihe von Experimenten und Studien die Bedeutung und Verwendung von Historien im Kontext des World Wide Web untersucht, so etwa [GREENBERG und COCKBURN 1999], [HIGHTOWER et al. 1998] oder [OLLERENSHAW 2002]. Diese aber haben ihren Schwerpunkt in der Untersuchung der Navigationshistorien in Hypertexten, und lassen Such- und Interaktionshistorien zumeist außen vor.

Anita Komlodi untersuchte die Bedeutung von Such- oder Interaktionshistorien bei der Entwicklung kognitiver Modelle in der Informationssuche von Juristen und Rechtsbibliothekaren [KOMLODI 2002]. Dabei zeigte sie die Wichtigkeit externer Repräsentationen zur Unterstützung bei der Planung, Durchführung und Evaluation von Suchstrategien auf: Interaktionshistorien machen den Fortschritt und die bereits gefundene Information bewusst, sie helfen bei der Planung und Beobachtung der Suche, und erleichtern darüber hinaus die Konsultation mit anderen über den Verlauf der Suche. Wo keine automatisierten Interaktionshistorien vorhanden sind, griffen die Suchenden in der durchgeführten Studie oft auf papierbasierte oder anderweitig von Hand erstellte Verlaufsprotokolle zurück.

Zusammen mit Soergel beschreibt Komlodi ein theoretisches Gerüst für Interaktionshistorien, das drei Eigenschaften umfasst [KOMLODI und SOERGEL 2002]: die unterstützte Zeitspanne (eine Sitzung, sitzungsübergreifend, über die gesamte Lebenszeit oder als Planungswerkzeug für die Zukunft), die unterstützte Aufgabenspanne (nur Suchaufgaben, Such- und Informationsaufgaben oder alle Aufgaben) und den Kontext. Basierend auf diesem Gerüst identifizieren sie sechs Hauptaufgaben für Suchhistorien:

- als Unterstützung des Gedächtnisses (*memory support*), um einen Überblick darüber zu schaffen, was bereits gesucht wurde,

## 4.2. SUCHPFADE UND PFADANALYSE

---

- als Unterstützung bei der Systembenutzung (*search system use*), um festzuhalten, wie etwas erreicht wurde, so dass man diese Strategie oder das Werkzeug für eine weitere Aufgabe erneut nutzen kann,
- als Unterstützung bei der Informationssuche, durch Möglichkeiten zur strategischen Planung, zur Kontrolle und Evaluation des Suchplans,
- zur Unterstützung bei der Nutzung der Information,
- zur Integration mit anderen Aufgaben,
- zur Integration mit anderen Personen, um z. B. Information über die Suche und ihren Verlauf zwischen Kollegen zu transferieren.

Als Konsequenz einer Studie, während der ein System für Suchhistorien in Zusammenarbeit mit späteren Anwendern entwickelt wurde, stellte Komlodi in [KOMLODI 2004] fest, dass Interaktionshistorien insbesondere bei der Planung und Evaluation von Aufgaben, bei der Integration von Aufgaben, sowie bei der Wiederherstellung des Suchkontextes bei unterbrochenen und wieder aufgenommenen Aufgaben nützlich sind. Sie untersuchte, wie Suchende komplexe Aufgaben organisieren und verwalten, und welche Informationen und Funktionalitäten ein Werkzeug für Suchhistorien zur Unterstützung dabei anbieten sollte.

Die empirischen Erkenntnisse der Feldstudien und Experimente mit Implementierungen von Suchhistorien führten schließlich zu einer Reihe von Richtlinien für die Umsetzung von Suchhistorien als Teil einer Benutzerschnittstelle in IR-Systemen [KOMLODI et al. 2007].

### 4.2. Suchpfade und Pfadanalyse

Die Abfolge der Suchsituationen und der Aktivitäten während einer Suchsituation lassen sich auch als Pfad ansehen, entlang dem ein Suchender auf dem Weg zu seinem Suchziel navigiert [LALMAS und RUTHVEN 1999]. Jede Suche lässt sich so basierend auf dem Suchverlauf oder der Suchhistorie durch einen konkreten Suchpfad aus Situationen und Übergängen beschreiben. Insbesondere bei Webinformationssystemen sind diese Pfade untersucht und zum Information Mining genutzt worden [CATLEDGE und PITKOW 1995, CHEN et al. 1996, COOLEY et al. 1997]. Dabei ist die Analogie zu Pfaden bei der Navigation im WWW deutlicher durch die Hyperlinkstruktur des Webs. Situationen sind dabei Webseiten und Übergänge der Aufruf eines Hyperlinks.

Doch Pfade lassen sich auch für komplexe Informationssysteme mit einer Vielzahl von Werkzeugen beschreiben. Nach Lalmas und Ruthven repräsentiert eine Situation einen Schritt in der Interaktion des Benutzers mit dem System. Dabei stützt eine Situation jeweils ein oder mehrere Infone (Fakten) z. B. über die Aktivität des Benutzers in der Situation. Übergänge sind hier der Wechsel

von einer Situation in eine neue, etwa durch das Aufrufen eines Hyperlinks oder das Reformulieren einer Anfrage. Jeder Übergang überführt das System in eine neue Situation.

Nils Pharo modelliert das Informationsverhalten von Benutzern mit Hilfe eines solchen Pfadmodells, das er *Search Situation Transition (SST) method schema* nennt [PHARO 2002, PHARO 2004]. Als Teil des Modells beschreibt er Informationssuche als einen Prozess, bei dem sich Perioden der Interaktion mit Metadaten (die Übergänge) mit Perioden der Interaktion mit echten Daten (die Situationen) abwechseln. Dieser Prozess wird durch Faktoren aus vier externen Kategorien beeinflusst: der Arbeitsaufgabe, der Suchaufgabe, der sozialen und organisatorischen Umgebung und dem Suchenden.

Der Suchprozess oder Suchpfad selbst beginnt stets mit einem Übergang, d. h. einer Interaktion mit Metadaten. An einen Übergang schließt sich entweder direkt ein weiterer Übergang oder eine Situation an, und eine Situation wird entweder von einem Übergang gefolgt, oder beendet den Suchprozess. Dabei sind Suchsituationen nach Pharo in Abschnitt 3.1 beschrieben.

Suchübergänge haben nach Pharo [PHARO 2004] im Wesentlichen die gleichen Eigenschaften wie Situationen, allerdings interagiert der Benutzer in ihnen nur mit Metadaten, weshalb Relevanzurteile nicht zutreffend sind. Sowohl während Suchübergängen als auch während Suchsituationen finden Aktionen des Benutzers statt. Pharo sieht in den Suchübergängen ein Analogon zu den Informationssuchstrategien nach Belkin [BELKIN et al. 1993], die auch auf Metadaten ausgeführt werden.

Chalmers und andere [CHALMERS et al. 1998] stellen ein Pfadmodell für die Informationssuche vor, das auf Benutzungshistorien anstelle inhaltlicher Analyse aufbaut. Chalmers leitet das Modell aus der urbanen Planungstheorie ab [CHALMERS 2003]. Das Pfadmodell behandelt Informationsobjekte wie räumliche Gegenstände, Historien der Informationsnutzung wie die Pfade von Menschen durch eine Stadt und häufige Pfade als emergente Nutzungsmuster. Dabei wird nicht die Bedeutung in den Gemeinsamkeiten der Suchpfade hinterfragt, sondern es sollen nur statistische Zusammenhänge in den Sammlungen von Pfaden genutzt werden, die von Benutzern hinterlassen werden.

Der Pfad selbst wird von Chalmers als eine zeitlich geordnete Historie der Benutzung von Informationsobjekten durch eine Person definiert [CHALMERS 1999]. Er setzt sich aus einer Reihen von Symbolen zusammen, die Seitenaufrufe in einem Webbrowser, vom Nutzer eingegebene Worte, verwendete Dateien, Dokumente oder Programme, Personen, oder andere Informationsobjekte oder Werkzeugnutzungen repräsentieren können. Äquivalent sind zwei Pfade, wenn sie die gleichen Autoren und die gleichen Symbole in identischer Reihenfolge beinhalten. Zusätzlich werden verschiedene Ähnlichkeitsmaße vorgeschlagen, um Aktivitäten mit eigenen Aktivitäten aus der Interaktionshistorie zu vergleichen, oder mit den Aktivitäten anderer Benutzer.

## 4.2. SUCHPFADE UND PFADANALYSE

---

Das Pfadmodell wurde in einem prototypischen Recommender-System umgesetzt, wo jeder neue Pfadeintrag eines Benutzers als impliziter Wunsch nach Empfehlungen behandelt wird [CHALMERS 2001, CHALMERS 2003]. In regelmäßigen Abständen wird das letzte Stück des Suchpfads des Benutzers ermittelt und das Vorkommen der Symbole im Suchpfad entweder in früheren Suchpfaden des Benutzers oder in jenen anderer Benutzer gesucht. Die Kontexte innerhalb eines Fensters um die Fundstellen herum werden gesammelt, wiederum in Form der innerhalb des Fensters vorkommenden Symbole. Schließlich werden jene Symbole, die häufig in früheren Kontexten aber nicht im aktuellen Pfad des Benutzers vorkommen, als Vorschlag ausgegeben.

Wexelblat und Maes [WEXELBLAT 1999, WEXELBLAT und MAES 1999] entwickelten eine Reihe von Werkzeugen unter dem Sammelnamen „Footprints“, die Interaktionshistorien einsetzen, um häufig beschriftete Pfade zur Unterstützung bei komplexen Suchaufgaben zu benutzen (z. B. der Zusammenstellung von möglichen Alternativen für einen Autokauf).

Zur Gewinnung von häufigen Pfaden oder Vorschlägen aus den aufgezeichneten Historien von Suchsystemnutzern können verschiedene Methoden aus dem Bereich des Web Mining zum Einsatz kommen. Mannila, Toivonen und Verkamo [MANNILA et al. 1997] beschreiben einen regelgenerierenden Ansatz zur Sequenzanalyse, um häufige Episoden in Ereignissequenzen zu finden. Pitkow und Pirolli [PITKOW und PIROLLO 1999] setzen Markov-Modelle  $k$ -ten Grades ein, um aus Navigationspfaden längste sich wiederholende Muster zur Vorhersage von Navigationsverhalten zu gewinnen. Chen, Park und Yu schlagen einen Algorithmus zur Bestimmung häufiger Traversionsmuster aus einer Menge längster „Vorwärtsreferenzen“ (um Rückwärtsschritte bereinigte WWW-Navigationshistorien) vor, der auf der Bestimmung von Assoziationsregeln basiert [CHEN et al. 1996].

Dabei ist es wichtig, die Benutzung aller Werkzeuge eines komplexen IR-Systems mit einzubeziehen. Die reine Analyse von Suchlogs als Teilbereich der Transaktionsloganalyse kann typische Interaktionen zwischen Benutzer und System aufdecken. Aber von Bates' vier Ebenen der Interaktion adressiert die Suchloganalyse nur die niedrigsten zwei der Schritte und Taktiken [JANSEN 2008]. Kritisch ist die Loganalyse zur Gewinnung von Strategien zu sehen, da sie nicht die zugrundeliegenden Suchbedürfnisse erfassen kann. Auch ist es schwierig, die Zufriedenheit von Benutzern mit den Ergebnissen ihrer Aktivitäten zu messen. Es ist daher notwendig, zusätzliche Informationen zu benutzen, um erfolgreiche Suchpfade zu identifizieren, etwa die als Ergebnis von Aktivitäten gespeicherten Dokumente.



### 4.3. Zusammenfassung

In diesem Kapitel wurden Arbeiten vorgestellt, welche die Wichtigkeit von Such- und Interaktionshistorien in der Benutzerschnittstelle für die Planung und die strategische Durchführungen von Suchaufgaben aufzeigen. Darüber hinaus können Suchhistorien in der Form von Suchpfaden Grundlage für die Analyse von Strategien und für Interaktionsvorschläge bilden. Wenn sich zeigen lässt, dass sich über eine große Zahl erfolgreicher Suchsitzungen klare Muster in den Suchpfaden der Benutzer ergeben, so könnten diese „Trampelpfade“ genutzt werden, um als strategische Vorschläge für unerfahrene Suchende zu dienen.



## **Teil II.**

# **Unterstützung beim Finden und Durchführen von Suchstrategien**



# Kapitel 5.

## Architektur

Im Folgenden wird zunächst die Software-Architektur beschrieben, die die Grundlage der umgesetzten Unterstützungsmöglichkeiten zum Finden und Durchführen von Suchstrategien bildet, und aufgezeigt, wo Beiträge dieser Arbeit in der Weiterentwicklung und Anpassung der Software lagen.

### 5.1. DAFFODIL

Um Benutzer bei einer komplexen Suche umfassend zu unterstützen, ist ein reines Suchwerkzeug, das die Eingabe von Anfragen erlaubt und zu diesen die Ergebnisse als Liste anzeigt, nicht ausreichend. Bates fordert die Unterstützung weit verbreiteter Suchstrategeme in der Benutzerschnittstelle moderner IR-Systeme [BATES 1990]. Marchionini macht deutlich, dass grundlegend zwischen analytischen Suchstrategien und opportunistischen Browsing-Strategien zu unterscheiden ist. Ein IR-System sollte Benutzer nicht darin einschränken, welche Strategien sie zur Bewältigung ihrer Informationsaufgaben einsetzen wollen, sondern beide Strategien unterstützen [MARCHIONINI 1995].

Es ist daher sinnvoll, die Unterstützung zur Findung und Durchführung von Suchstrategien in ein System einzubetten, das eine weite Spanne von Suchaktivitäten ermöglicht. DAFFODIL ist ein solches Suchsystem, das den Zugang zu verteilten, heterogenen digitalen Bibliotheken ermöglicht. Das System besteht aus einem Netz von Agenten (Diensten), die mittels CORBA interagieren, sowie einem Frontend, das dem Benutzer zur strategischen Unterstützung eine Vielzahl von Funktionen zum Suchen, Browsen, Erkunden und Verwalten von Informationsobjekten bietet [GÖVERT et al. 2000, KRIEWEL et al. 2004]. Bates' Strategeme werden durch Werkzeuge unterstützt, welche die direkte Ausführung dieser hochrangigen Suchaktivitäten erlauben [FUHR et al. 2002]. Durch die enge Kopplung der Strategeme innerhalb der werkzeuggestützten Oberfläche unterstützt DAFFODIL bereits auf vielfältige Weise den Benutzer in seiner Suche [KLAS 2007, SCHAEFER 2008].

Aus der Sicht dieser Arbeit ist insbesondere die große Anzahl möglicher Aktionen sowohl auf der Ebene von einfachen Moves, als auch auf der Ebene

von Taktiken oder Strategemen von Bedeutung, da diese sich zu komplexen Suchplänen kombinieren lassen. Die Agentenarchitektur erlaubt die leichte Erweiterbarkeit um neue Dienste. So wurde das in Abschnitt 6.1.3 beschriebene Vorschlagsmodul in Form zweier neuer Agenten in DAFFODIL integriert.

Zunächst sollen hier jedoch kurz die wesentlichen Werkzeuge in der Benutzeroberfläche des Systems und deren verfügbare Nutzeraktionen vorgestellt werden, da diese die Bausteine für mögliche Suchstrategien darstellen.

### 5.1.1. Werkzeuge und Strategeme

Wie von Klas [KLAS 2007] beschrieben, kombiniert DAFFODIL in seiner Oberfläche Such- und Browsingansätze zur Informationssuche. Dadurch sind Benutzer frei in der Wahl ihrer Strategie und können auch wie von Marchionini gefordert, analytische und browsende Strategien kombinieren – aber sie erhalten zunächst keine Unterstützung bei der Bildung einer solchen Strategie.

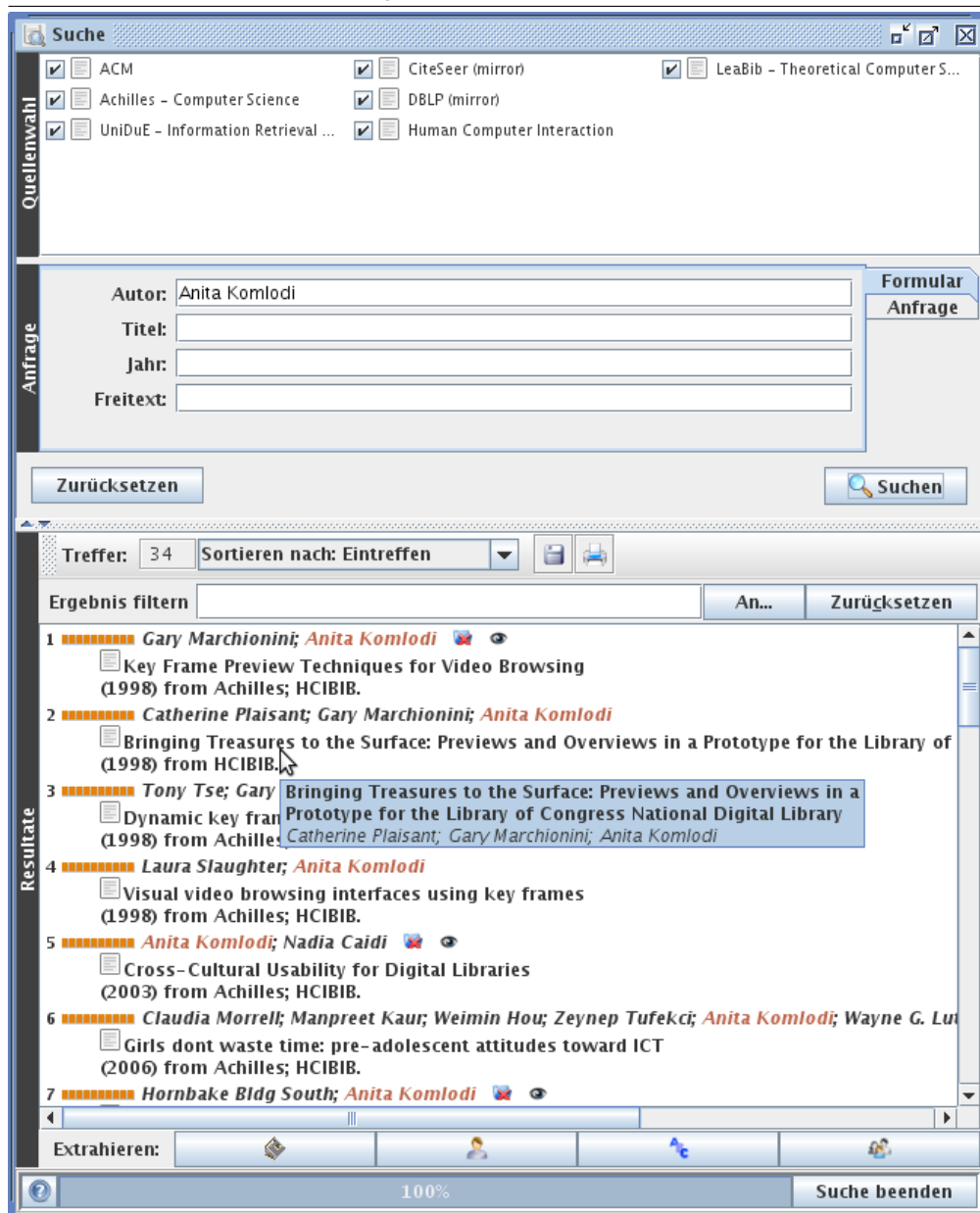
#### Suchwerkzeug

Das Suchwerkzeug (Abb. 5.1) ist das zentrale Werkzeug des Systems, das die Suche in einer Reihe digitaler Bibliotheken über ein in Suchfelder unterteiltes Suchformular erlaubt. Das Formular erlaubt die Suche in Titel, Freitext, Jahren und Autoren. Dabei werden auch boolesche Anfragen, sowie Phrasensuche unterstützt. Die Resultate aus den Datenquellen werden zusammengeführt und in einer Rangliste angezeigt.

Zur Arbeit mit den Resultaten bestehen für einen Benutzer eine Reihe Möglichkeiten:

- Das *Sortieren* der kompletten Ergebnisliste nach unterschiedlichen Kriterien (Relevanz, Autoren, Titel, Jahre) erlaubt neue Sichten auf das Ergebnis und kann z. B. zentrale Autoren aufzeigen oder eine leichtere Übersicht über aktuelle Ergebnisse geben.
- Das *Filtern* der Ergebnisliste nach Begriffen, Datenquellen oder anderen Aspekten ermöglicht die Arbeit selbst mit großen Resultatlisten. Benutzer können durch Filtern manchmal die Notwendigkeit umgehen, eine Anfrage zur weiteren Einschränkung neu zu formulieren.
- Das ordnerbasierte *Speichern* von einzelnen Dokumenten zur besseren Organisation in einer persönlichen Handbibliothek.
- Das sitzungsbasierte *Ablegen* von einzelnen Dokumenten in einer Zwischenablage zum leichten Weiterarbeiten mit als relevant erkannten Dokumenten.
- Das *Betrachten* von Details eines einzelnen Resultatdokuments.

Abbildung 5.1. Suchwerkzeug.



## 5.1. DAFFODIL

---

- Die Gewinnung neuer Information aus den Resultaten durch *Extraktion* von Begriffen.

### **Extraktionswerkzeug**

Das Extraktionswerkzeug steht sowohl in der Resultliste als auch in der Zwischenablage zur Verfügung. Die Extraktion aus einer Resultatliste erlaubt einen schnellen Überblick über häufige Begriffe im Ergebnis und bietet sowohl eine leichtere Beurteilung des Ergebnisses in Bezug auf das zu lösende Informationsproblem, als auch Anhaltspunkte für Ansätze zur Filterung oder Modifizierung eines nur teilweise zufriedenstellenden Ergebnisses. In der Zwischenablage kann die Extraktion explizit auf bereits als relevant erkannte Dokumente angewandt werden. So lassen sich neue Suchbegriffe (Terme oder Autoren) finden, die zur Ergänzung oder als Alternative zu einer existierenden Anfrage genutzt werden können.

Mit seiner Hilfe lassen sich aus einer Dokumentliste häufige Begriffe, wichtige Autoren, Journale oder Konferenzen extrahieren. Die Darstellung in normaler Listenform in absteigender Häufigkeit wird ergänzt durch die Darstellung in Form einer sogenannten Termwolke in alphabetischer Anordnung und einer von der relativen Häufigkeit des Begriffs bestimmten variablen Schriftgröße (siehe Abb. 5.2). Trotz der inhärenten Probleme dieser Visualisierungform [HASSAN-MONTERO und HERRERO-SOLANA 2006, HEARST und ROSNER 2008] erlaubt sie zusätzliche Navigations- und Analysemöglichkeiten [KUO et al. 2007].

Die extrahierten Begriffe können direkt als Filter genutzt werden, bieten darüber hinaus aber noch weitere Möglichkeiten in den anderen Werkzeugen DAFFODILS. Durch Drag'n'Drop oder über das Kontextmenü der Begriffe lassen sich diese Werkzeug direkt aufrufen.

### **Journal- und Konferenzbrowser**

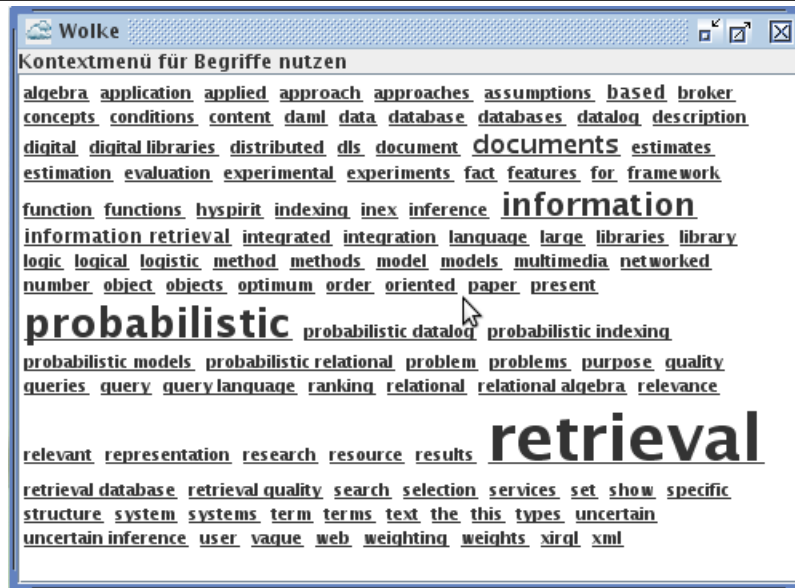
Die beiden Werkzeuge zum Browsen in Journalen und Konferenzbänden bieten eine direkte Umsetzung des Strategems *Journal Run*. Sie lassen sich beispielsweise mit einem aus relevanten Dokumenten extrahierten Journaltitel oder Konferenznamen aufrufen, und bieten dann einen browsenden Zugang zu den Dokumenten der digitalen Bibliotheken.

### **Koautorenwerkzeug**

Über das Koautorenwerkzeug wird die Möglichkeit zur Visualisierung eines Koautorennetzwerks zentraler Autoren geboten. Neben der grafischen Darstellung in Form eines Graphs bietet das Werkzeug auch eine Übersicht über



Abbildung 5.2. Extrahierte Terme.



die Koautoren einer Person als alphabetische Liste. Zu so gefundenen Autoren lassen sich dann wiederum neue Anfragen nach von diesen publizierten Dokumenten generieren. Zusammen wird so das Strategem der *Author Subject Search* ermöglicht.

### Thesaurus und Verwandte Begriffe

Ein Thesaurus erlaubt Benutzern von DAFFODIL das Nachschlagen von Begriffsdefinitionen, sowie das Finden von Ober- und Unterbegriffen oder Synonymen. Die drei letztgenannten sind Grundlage einer Reihe von gebräuchlichen Suchtaktiken.

Die Verwandten Begriffe lassen sich nicht direkt vom Benutzer aufrufen, sondern stellen vielmehr eine proaktive Unterstützungsfunktion dar [SCHAEFER 2008], die mit jeder Suche automatisch angestoßen wird. Zu den Suchbegriffen des Benutzers werden „verwandte“ Begriffe angezeigt, wobei Verwandtschaft hier als häufiges gemeinsames Auftreten in Dokumenten verstanden wird. Auch diese Begriffe stellen oft gute Ansatzpunkte für die Verfeinerung bzw. Neuformulierung einer Suche dar.

### Klassifikationsbrowser

Über eine domänenspezifische hierarchische Klassifikation wird ein weiterer Zugang zu den Dokumenten der eingebundenen digitalen Bibliotheken geboten. Einzelne Knoten der Klassifikation können benutzt werden, um entweder nach dem Klassifikationsbegriff oder wo vorhanden nach dem Klassifika-

tionscode zu suchen. Insbesondere dort, wo kontrolliertes Vokabular für die Indexierung der Dokumente eingesetzt wird, lassen sich mit Hilfe der Klassifikationsbegriffe oft Dokumente finden, die mit dem persönlichen Vokabular des Suchenden sonst nicht auffindbar gewesen wären. Das Strategem des *Area Scan* wird so möglich.

### **Suchhistorie**

Eine automatisch aktualisierte Suchhistorie zeigt alle bisher vom Benutzer gestellten Suchanfragen zusammen mit der Zahl der gefundenen Dokumente an. Insbesondere während langer Suchsitzungen sind derartige Historien wertvoll, wie auch Komlodi in mehreren Fallstudien aufgezeigt hat [KOMLODI 2002, KOMLODI et al. 2007]. Die Suchhistorie erlaubt die Wiederausführung und Speicherung besonders erfolgreicher Anfragen, die Modifikation einer früheren Anfrage und eine Übersicht über bereits benutzte Kombinationen von Suchbegriffen.

### **Chat**

Eine eingebaute Chat-Funktion erlaubt die Kommunikation mit anderen Benutzern des Systems und den direkten Austausch von Objekten. Dies erlaubt die Umsetzung von kooperativen Suchstrategien und von Taktiken wie CONSULT, bei denen der Suchende sich durch das Einholen von Rat oder die Diskussion eines Suchbedürfnisses mit anderen aus einer problematischen Situation lösen soll.

### **Ablage und Handbibliothek**

Zur Ablage von gefundenen Dokumenten, aber auch anderer bei der Suche entdeckter oder verwendeter Informationsobjekte wie beispielsweise Autoren, Journale, Begriffe oder Suchanfragen stehen zwei Möglichkeiten zur Verfügung.

Die Handbibliothek bietet die dauerhafte Speicherung solcher Objekte in persönlichen oder geteilten, hierarchischen Ordnern, ähnlich eines lokalen Dateisystems. Gespeicherte Objekte können durch Drag'n'Drop oder über Aktionen des Kontextmenüs in anderen Werkzeugen wiederverwendet werden.

Während einer Suche ist es oft jedoch nützlich, Dokumente zunächst unstrukturiert abzulegen, um damit weiterzuarbeiten und sie nicht aus den Augen zu verlieren. Das erlaubt die Zwischenablage (auch Körbchen genannt), die als Umsetzung des metaphorischen Sammelkorbs in Bates' *Berry Picking Model* [BATES 1989] verstanden werden kann. Nicht nur die Ablage gefundener Objekte ist hier möglich, sondern zur Planung, als Notizen während einer Suche

und im Rahmen von Vorüberlegungen zu einer Suchstrategie können neue Begriffe oder Autoren angelegt werden. Die Menge der im Korbchen abgelegten Objekte kann zur weiteren Analyse verwendet werden, etwa wie oben beschrieben durch Extraktion von häufigen Begriffen.

### **Detailansicht**

Die Detailansicht schließlich stellt alle verfügbaren Informationen über ein Informationsobjekt dar, im Allgemeinen über ein Dokument. Die Darstellung erfolgt mit Hilfe von Hypertext und die dargestellten Begriffe sind entweder direkt mit Suchaktivitäten verlinkt (über einen Autoren wird eine Autorensuche angestoßen, ein Deskriptor ruft den Thesaurus auf) oder können über das Kontextmenü in vielfältiger Weise in anderen Werkzeugen eingesetzt werden (z. B. zur Filterung der Ergebnisse oder zur Ergänzung der Anfrage).

### **Zusammenfassung**

Insgesamt stellt DAFFODIL das derzeit wohl mächtigste und komplexeste Werkzeug zum Zugriff auf Digitale Bibliotheken dar. Auch wenn diese Übersicht keine vollständige Darstellung der vorhandenen Werkzeuge und insbesondere der durch DAFFODIL unterstützten Kombinationen, Taktiken und Strategeme geben kann, so wird doch deutlich, dass mit einem derartigen System umgesetzte Suchstrategien ebenfalls eine erhebliche Komplexität erlangen können.

Nicht nur benötigen Benutzer Kenntnisse über die vorhandenen Möglichkeiten des Systems und über die taktischen und strategischen Möglichkeiten innerhalb einer Suche, sie müssen zudem herausfinden, wie sich bestimmte Strategien am günstigsten mit den vorhandenen Mitteln umsetzen lassen und müssen Heuristiken entwickeln, wann welche Taktiken oder Werkzeugkombinationen zielführend einzusetzen sind. Hier kann eine Systemunterstützung ansetzen und Suchende bei der Strukturierung und Durchführung ihrer Suche helfen.

Im Rahmen dieser Arbeit wurden von den vorgestellten Werkzeugen und Möglichkeiten unter anderem umgesetzt:

- Die Visualisierung mit Hilfe von Termwolken als zusätzliche Navigations- und Analysemöglichkeit im Rahmen einer interaktiven Suchstrategie.
- Die Zwischenablage zur Unterstützung einer Suchaufgabe.
- Eine weitergehende Kopplung der einzelnen Werkzeuge mittels zusätzlicher Möglichkeiten zur Kombination von Werkzeugen zu einer Strategie.

## 5.2. AUFGABENABHÄNGIGE PERSPEKTIVEN

---

- Eine überarbeitete, interaktivere Anzeige der Ergebnisse mit der Möglichkeit des direkten Aufrufs anderer Werkzeuge (Strategeme) von allen dargestellten Begriffen.
- Die bessere Integration der Suchhistorie in den Arbeitsablauf.

### 5.1.2. Agentenarchitektur

DAFFODIL benutzt eine agentenbasierte Architektur [WOOLDRIDGE und JENNINGS 1995] zur Implementierung seiner Funktionalitäten [MÜLLER 2001]. Dabei waren insbesondere die Aspekte der Autonomie, der Inferenzbildung, der Proaktivität, der Adaptivität und der Kommunikation entscheidend für die Wahl der Architektur [FUHR et al. 2000]. Die Architektur nutzt CORBA und die Agenten kommunizieren unter Benutzung von KQML. Dabei ist ein zentraler Verzeichnisagent für die Namensauflösung zuständig.

Alle Dienste sind dabei als kooperative Agenten implementiert, welche die Kommunikation der Dienste durchführen, auf Nachrichten warten und diese versenden. Die *Payload* (Anfrage, Mitteilung oder Resultat) der Nachricht wird im XML-Format übertragen. Zu jedem Agenten existiert ein sogenannter *Request*, der die Programmlogik des Dienstes implementiert.

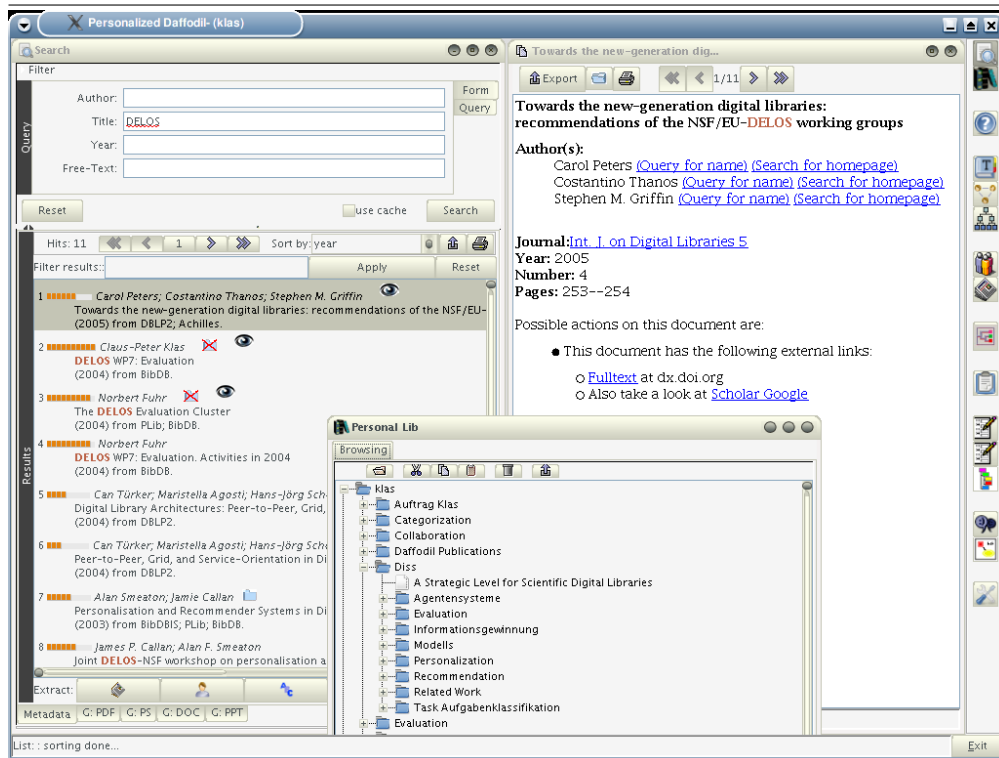
Eine genauere Beschreibung von DAFFODIL als Multiagentensystem findet sich bei Klas [KLAS 2007].

## 5.2. Aufgabenabhängige Perspektiven

Die ursprüngliche Umsetzung des DAFFODIL-Frontend geschah in Gestalt eines werkzeuggesteuerten Desktops mit frei beweglichen Klientenfenstern für einzelne Werkzeuge [KRAUSE 1997, FUHR et al. 2002]. Diese Darstellung erwies sich jedoch für Benutzer als oft verwirrend und für eine aufgabenabhängige Unterstützung als schlecht geeignet. Im Rahmen dieser Arbeit wurde daher die Benutzeroberfläche als *Tiled Panes* [TIDWELL 2005] neu gestaltet.

Dabei wird der Bildschirm in eine Reihe von senkrechten wie waagerechten *Panes* (Kacheln) unterteilt, deren Größe sich vom Benutzer anpassen lässt. Eine einzelne Kachel enthält ein oder mehrere Werkzeuge, wobei immer ein Werkzeug als aktives Werkzeug im Vordergrund der Kachel angezeigt wird. Auf die inaktiven Werkzeuge kann über Karteireiter am oberen Rand der Kachel oder mit Hilfe einer Werkzeuggesteige zugegriffen werden. Einzelne Werkzeuge haben dabei zwei mögliche Zustände: entweder sind sie in einer Kachel „angedockt“ oder als unabhängiges Fenster dargestellt. Der Benutzer kann zwischen den beiden Zuständen frei hin- und herschalten und so etwa einzelne Werkzeuge aus der Oberfläche herausziehen.

Abbildung 5.3. Ursprüngliche Umsetzung des DAFFODIL-Desktops.



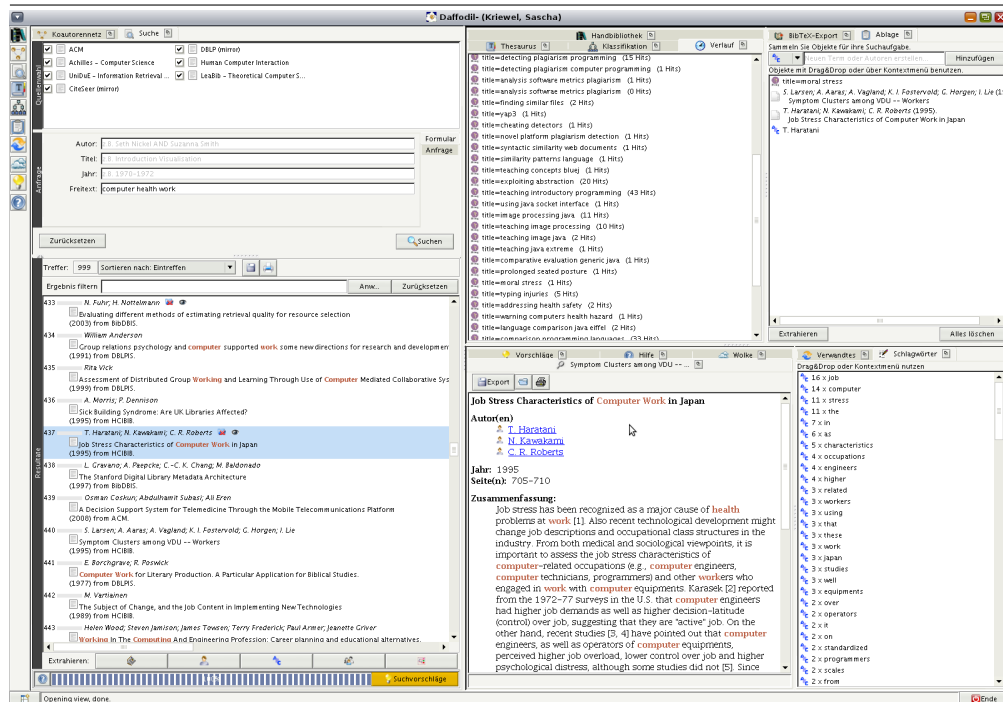
Um Aufgaben besser zu unterstützen, wurde ein Perspektivenkonzept umgesetzt. Perspektiven sind aufgaben- oder benutzerspezifische Zusammenstellungen von Werkzeugen mit vorgegebenen Platzierungen innerhalb der *Tiled Panes*. Es wurden Perspektiven für Aufgaben wie das Arbeiten mit Annotationen, das Analysieren von Suchhistorien oder die einfache Suche erstellt, ebenso wie Perspektiven für Benutzergruppen wie fortgeschrittene Suchende oder Administratoren. Auch die Möglichkeit benutzerdefinierter Perspektiven wurde gegeben. So wurde die Grundlage geschaffen für die Umsetzung aufgabenabhängiger Benutzerführung mit Hilfe von Software-Scaffolding (siehe Abschnitt 7.2). Abbildungen 5.3 und 5.4 zeigen die Oberfläche vor und nach der Umgestaltung.

### 5.3. Strategieframework

Zur Umsetzung eines strategischen Unterstützungssystems für DAFFODIL wurde ein Framework entwickelt, das modulare Strategien, sowie Makros bereitstellt, die als Bausteine für die Erstellung umfangreicherer Suchstrategien benutzt werden können [KRIEWEL et al. 2005]. Benutzerbefragungen bestätigen, dass die strategischen Module geeignet sind, grundlegende Suchstrategien für einfache Suchaufgaben umzusetzen [FRANKMÖLLE 2004]. Sie können

## 5.3. STRATEGIEFRAMEWORK

Abbildung 5.4. DAFFODIL-Desktop mit aufgabenspezifischen Perspektiven.



benutzt werden, um Teile von oder komplette Strategien für eine bekannte Problemstellung vorzudefinieren und als automatisch ausführbaren oder nutzergelenkten Suchplan zur Verfügung zu stellen.

### 5.3.1. Modulare Suchstrategien

Frankmölle implementierte eine Reihe von grundlegenden Suchmodulen für einfache, immer wiederkehrende Suchaufgaben aufbauend auf dem Strategieframework [FRANKMÖLLE 2004]. Diese stellen Bausteine dar, die im Rahmen von Vorschlägen Benutzern des Systems zur Verfügung gestellt werden können. Zwei Dimensionen von Suchaufgaben stellten sich bei der Entwicklung dieser grundlegenden Strategien als besonders wichtig heraus: die Komplexität der Aufgabe und das Problem- bzw. Domänenwissen der Benutzer [WILDEMUTH 2004, DRABENSTOTT 2003]. Mit Hilfe dieser zwei Dimensionen konnten Suchaufgaben kategorisiert und für diese theoretische Suchstrategien entwickelt werden.

Abbildung 5.5 zeigt den Aufbau beispielhafter Strategiemodule. Basierend auf dem Strategieframework wurden Strategien für die einfache Suche nach einem bekannten Dokument, die Zusammenstellung einer bekannten Dokumentliste, die Suche nach einem nur vage definierten Dokument und die Ergänzung einer existierenden Dokumentliste um ähnliche Dokumente entwickelt. Diese

werden in Kapitel 7 ausführlicher beschrieben.

### 5.3.2. Makros

Makros sind hier Methoden, die in verschiedenen Stratemodulen des Frameworks eingesetzt werden können, um die Qualität zu verbessern. Sie stellen aber für sich genommen keine eigenständige Suchstrategie dar. Ein Beispiel für ein Makro wird in Abbildung 5.6 dargestellt. Es wird eingesetzt, um zu einer gegebenen Anfrage mögliche Termmanipulationen basierend auf den von Bates beschriebenen Taktiken durchzuführen.

Allgemein kapseln Makros wichtige Basisfunktionalitäten, wie beispielsweise

- Frageerweiterung,
- Quellenauswahl oder
- die Suche nach Begriffen zur Beschreibung eines Themas.

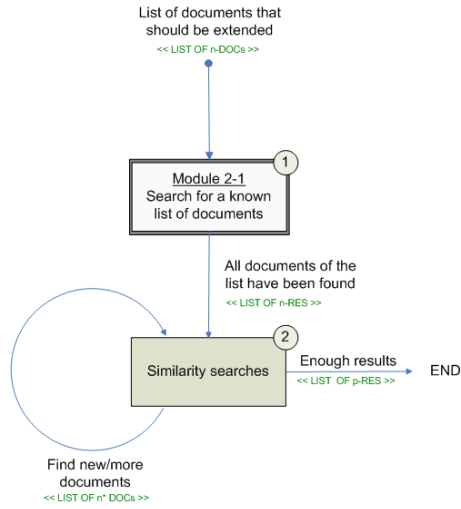
Eine ausführliche Beschreibung der umgesetzten Makros findet sich bei Frankmölle [FRANKMÖLLE 2004].

## 5.4. Loggingframework

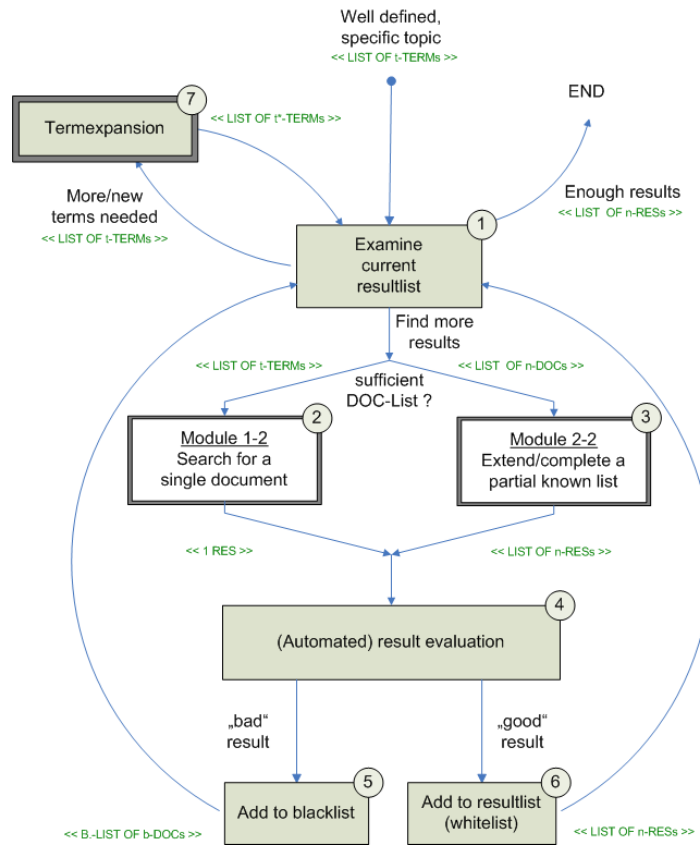
Zur effektiven Unterstützung des Benutzers ist eine genaue Analyse der Benutzersituation erforderlich. Ebenso können durch Betrachtung der Aktivitäten der Suchenden in einem System Erkenntnisse über erfolgreiche Suchstrategien oder Pfade gewonnen werden, die von diesen gewählt wurden. Schließlich erlaubt die Betrachtung der Suchhistorie einschließlich aller Aktivitäten eines Benutzers während der Sitzung diesem eine Reflexion über seine bisherige Suchstrategie oder einem außenstehenden Experten einen Einblick in bereits unternommene Anstrengungen. Zusammengenommen machen diese Erfordernisse ein umfassendes Loggingframework notwendig, das in DAFFODIL mittels Softwareagenten, einer relationalen Datenbank und XML-Nachrichten implementiert wurde [WERSECK 2006]. Abbildung 5.7 zeigt den Loggingdienst von DAFFODIL schematisch.

Die Logs des Systems umfassen alle Benutzeraktivitäten, sowie die Antworten des Systems, aber auch vom System initiierte Aktionen. Dabei wird ein XML-Logschema verwendet (ein Ausschnitt aus dem XML-Log einer Sitzung ist in Quelltext 5.4 zu sehen), das in Hinblick auf die Evaluation von digitalen Bibliotheken entwickelt wurde, aber so angelegt ist, dass es auch zur Gewinnung von strategischen Informationen genutzt werden kann [KLAS et al. 2006b, KLAS et al. 2006a]. Innerhalb des Logschemas werden die Aktivitäten der Benutzer zu Sitzungen zusammengefasst. Eine Sitzung besteht aus einer

Abbildung 5.5. Modulare Suchstrategien im Rahmen des Strategieframeworks [KRIEWEL et al. 2005].



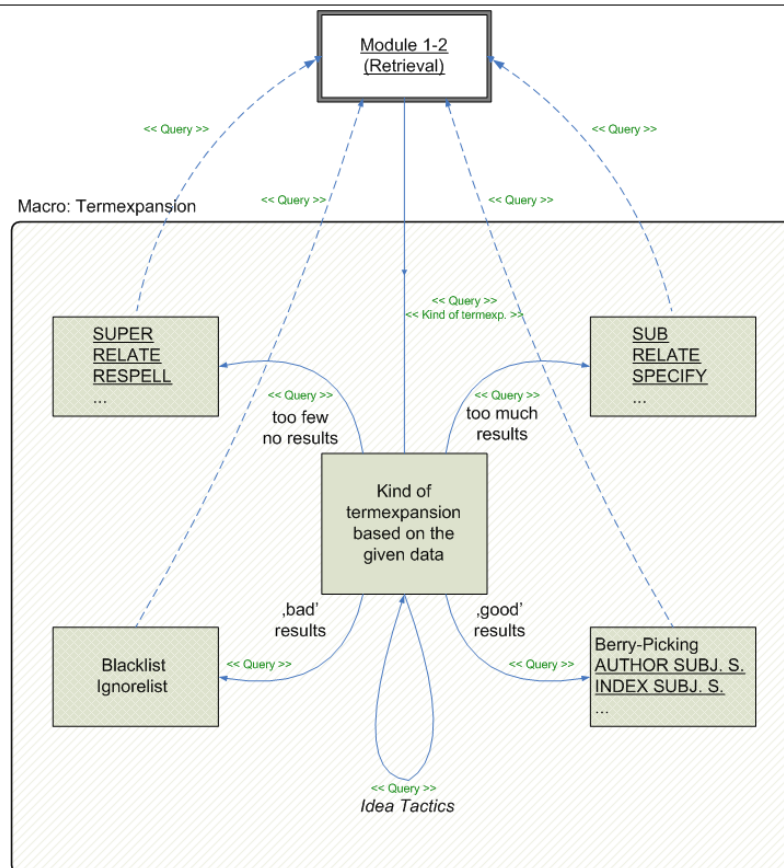
(a) Ein einfaches Strategiemodul.



(b) Ein komplexeres Strategiemodul.



Abbildung 5.6. Ein Makro des Strategieframeworks [KRIEWEL et al. 2005].

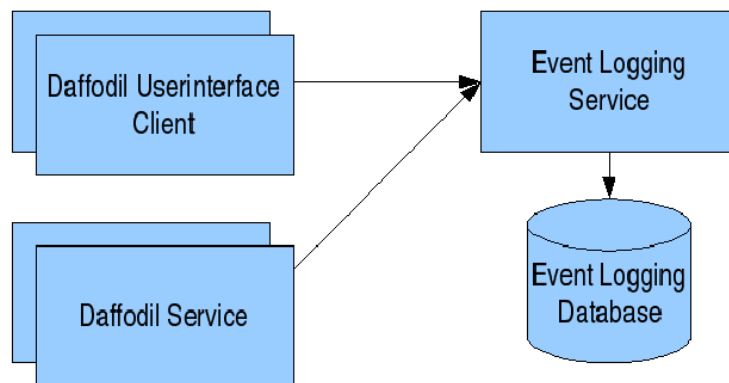


Reihe von Ereignissen in distinkten Suchkontexten. Dabei kann eine Suchsitzung mehrere Kontexte umfassen, wobei ein Kontextwechsel auf einen Wechsel in den fundamentalen Parametern einer Suchsitzung hindeutet, etwa die Bearbeitung einer neuen Aufgabe, die Hinzunahme eines zweiten Suchenden in einer kollaborativen Suche, oder Ähnliches. Obwohl es Ziel ist, solche Kontextwechsel automatisch zu erkennen, ist dies ein eigenes Forschungsfeld und wird derzeit in DAFFODIL nicht unterstützt.

Das Loggingschema unterscheidet dabei fünf Ebenen, die durch das Loggingframework in DAFFODIL allerdings nur teilweise unterstützt werden. Die Ebene des *Benutzerverhaltens* umfasst Aufgabe, soziales Umfeld, Benutzereigenschaften oder Domänenwissen und ist nur schwer systematisch zu erfassen. Auf der *konzeptuellen Ebene* werden die Aktivitäten des Benutzers in abstrakten Kategorien zusammengefasst, die etwa zur Analyse von Suchpfaden genutzt werden können. Die Ebene der *Dienste* betrachtet die angebotenen Funktionalitäten der Werkzeuge und der Agenten des Systems, während die *HCI-Ebene* der Keystroke-Ebene des GOMS-Modells [JOHN und KIERAS 1996] entspricht. Auf der Systemebene werden Ereignisse des Systemumfelds geloggt, die einen Einfluss auf die Suche des Nutzers haben können, etwa Systemlast oder Netz-

**Abbildung 5.7.** Der Loggingdienst von DAFFODIL [KLAS et al. 2006a].

---



werklatenz.

Obwohl alle Ebenen interessante Rückschlüsse auf Benutzerstrategien erlauben, wurden im Rahmen dieser Arbeit vor allem die konzeptuelle Ebene und die Ebene der Dienste betrachtet. Da die Ebene der Dienste sich an den konkreten Werkzeugen des Systems orientiert, sei hier auf die Beschreibungen in Abschnitt 5.1.1 verwiesen. Insgesamt werden auf dieser Ebene über 40 Ereignisarten erfasst [KLAS et al. 2007]. Auf der konzeptuellen Ebene werden bestimmte Arten von Ereignissen nach ihrem Typ zusammengefasst. Dabei werden folgende Ereignistypen unterschieden:

**Suchen** (*Search*) beschreibt Ereignisse, in denen der Benutzer eine explizite Anfrage formuliert, die vom System über einer Kollektion ausgewertet wird. Das umfasst z. B. Suchanfragen mit Hilfe des Suchwerkzeugs aber auch Termanfragen über den Thesaurus.

**Navigieren** (*Navigate*) bezeichnet den Vorgang der Auswahl aus einer Liste von vorgegebenen Möglichkeiten, um entweder eine neue Auswahlliste zu erlangen oder eine bestimmte Aktion auszuführen. Hierunter fällt etwa die Auswahl eines Jahrgangs einer bestimmten Zeitschrift im Journalbrowser, das Navigieren innerhalb einer hierarchischen Klassifikation oder auch die Auswahl eines Ergebnisdokuments aus einer Liste.

**Speichern** (*Store*) umfasst die Ablage, das Exportieren und Ausdrucken von Objekten zur weiteren Verwendung. Im weiteren Sinne umfasst es auch die Manipulation von Objekten innerhalb der Ablage, etwa durch Verschieben, Umbenennen und Löschen.

**Inspizieren** (*Inspect*) beschreibt Ereignisse, in denen der Benutzer die Details eines einzelnen Objektes anfordert und dadurch eine Anzeige des Systems auslöst.

**Anzeige** (*Display*) als Benutzeraktion umfasst die durch den Benutzer initiierten Änderungen der aktuellen Darstellung, etwa durch Änderung

---

**Quelltext 5.1** Beispielhafter Auszug aus den DAFFODIL-Logs.

---

```

<event beginTimeStamp="1119346127369" endTimeStamp="1119346127369"
  eventId="24.c3e82b:1049e70ade2:-7fee@ea127.0.0.1:5678">
  <search tool="Search Tool" numberOfResults="1">
    <input>
      <query>title=Information Vizualisation</query>
    </input>
  </search>
</event>
<event beginTimeStamp="1119346212614" endTimeStamp="1119346212614"
  eventId="2a.c3e82b:1049e70ade2:-7fe8@ea127.0.0.1:5678">
  <help initiator="system"
    description="spelling suggestion">
    <suggestion>
      <dlobjects>
        <dlobject type="term">
          <content>Visualization</content>
        </dlobject>
      </dlobjects>
    </suggestion>
  </help>
</event>
<event beginTimeStamp="1119346409235" endTimeStamp="1119346409235"
  eventId="31.c3e82b:1049e70ade2:-7fe1@ea127.0.0.1:5678">
  <search tool="Search Tool" numberOfResults="487">
    <input source="2a.c3e82b:1049e70ade2:-7fe8@ea127.0.0.1:5678">
      <query>title=Information Visualization</query>
    </input>
  </search>
</event>
<event beginTimeStamp="1119346512939" endTimeStamp="1119346512939"
  eventId="38.c3e82b:1049e70ade2:-7fda@ea127.0.0.1:5678">
  <filter type="sort">
    <sort attribute="query" predicate="inOrder"/>
  </filter>
</event>

```

---

der Sortierreihenfolge in einer Ergebnisliste oder dem Wechsel zwischen Darstellungsarten bei der Präsentation der Koautoren zu einem Autor (Graph oder Liste).

**Browsing** bezeichnet das Arbeiten innerhalb einer Menge von Objekten, meist Ergebnisdokumenten, entweder durch Betätigung eines Scrollbalkens, durch Vor- und Zurückspringen zwischen Teilmengen, oder durch Ausblenden von Teilmengen mit Hilfe von Filtern. Während ein Anzeige-Ereignis die Art der Darstellung modifiziert, ändert sich durch Browsing die Teilmenge dessen, was dargestellt wird.

**Annotieren** (*Annotate*) fügt einem existierenden Objekt zusätzliche Information hinzu, sei es einen Tag, eine Notiz, eine Verknüpfung oder eine Markierung.

**Erstellen** (*Author*) erzeugt ein neues Informationsobjekt oder editiert ein bestehendes.

**Hilfe** (*Help*) fordert Unterstützung von einem anderen Benutzer oder dem System an, kann aber auch in Form proaktiver Hilfe [SCHAEFER 2008] durch das System initiiert werden.

**Abbrechen** (*Cancel*) bricht eine laufende Aktivität ab.

Alle Ereignisse enthalten neben einem Zeitstempel, einer eindeutigen ID und der Zuordnung zu einer Sitzung eines Benutzers alle Eingaben des Benutzers und die vom System generierten Ergebnisse.

Auf Grundlage des Loggingframeworks und des definierten Schemas wurden eine Reihe von Analysen auf den Logdaten aus 6.374 Sitzungen durchgeführt, wobei sich zeigte, dass die Ebene der Dienste für die Zwecke der Unterstützung beim Aufbau einer Suchstrategie am geeignetsten ist [BODNARIU 2009]. Zwar lassen sich auf konzeptioneller Ebene leichter Muster erkennen, doch stellt sich aufgrund der Abstraktheit der Ereignisse die Frage, wie sich aus diesen konkrete, auf die Nutzersituation bezogene Hilfestellungen ableiten lassen.

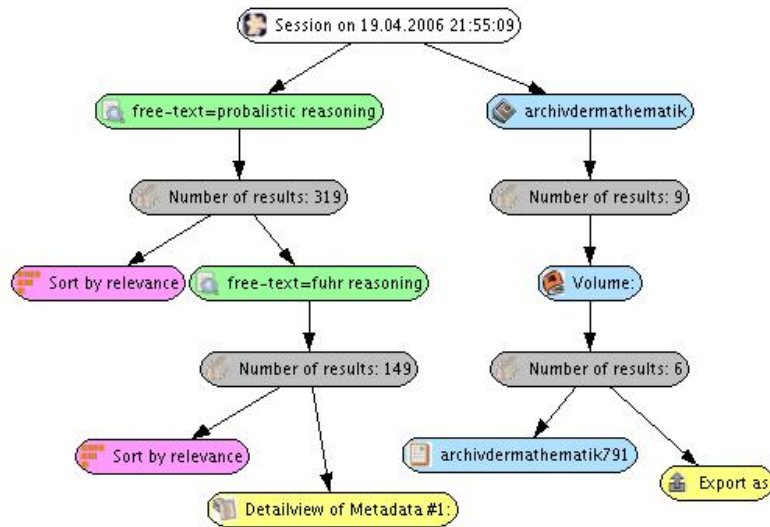
Eine ausführliche Auswertung der gesammelten Logdaten, insbesondere in Hinblick auf häufige Suchmuster bei den Benutzern des Systems, findet sich in Kapitel 8.

### 5.5. Visualisierung von Suchpfaden

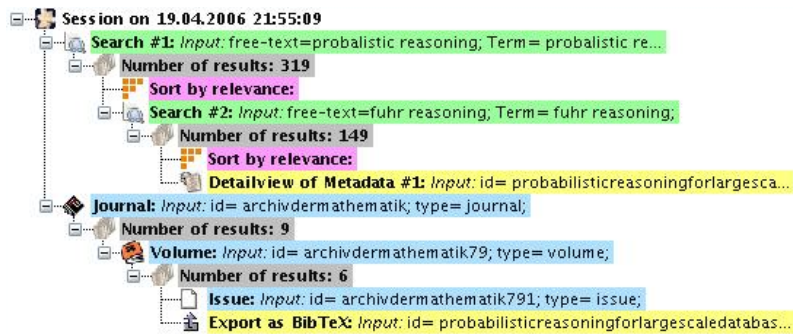
Um dem Benutzer die Möglichkeit zu geben, seine eigene Suchhistorie über eine Abfolge von Anfragen hinaus (siehe 5.1.1) zu betrachten, wurde eine Visualisierung der Suchpfade des Benutzers entwickelt [GENDERA 2006]. Diese sollte außerdem vom Benutzer hinzugezogenen Suchexperten die Möglichkeit zu geben, auf Basis der gesamten Historie der Interaktion des Benutzers mit dem System strategische Unterstützung zu leisten.

Die Visualisierung nutzt die konzeptionelle Ebene zur Gruppierung von Ereignissen durch Einfärbung, greift aber auf die Ebene der Dienste zurück, um Details der Benutzeraktionen anzuzeigen. Zwei Visualisierungsformen wurden entworfen und evaluiert, die auf der Idee einer nicht strikt linearen, sondern baumartigen Anordnung der Ereignisse basieren. Dabei haben etwa zwei Inspizieren-Ereignisse von Elementen einer Resultatliste in dem Darstellen-Ereignis, das die Anzeige dieser Liste repräsentiert, einen gemeinsamen Vorgänger. Als Darstellungsvarianten wurden eine horizontale Baumdarstellung und eine vertikale Baumdarstellung ähnlich der eines Dateibrowsers gewählt (Abbildung 5.8). In einer Evaluation wurde die Eignung der Darstellungen für eine Reihe von Anwendungsszenarien untersucht, wobei die Visualisierung

Abbildung 5.8. Visualisierungsvarianten einer Nutzersitzung.



(a) Horizontale Darstellung.



(b) Vertikale Darstellung.

entweder die aktuelle Sitzung des Benutzers oder eine bereits abgeschlossene Suchsitzung anzeigen kann:

- Suche innerhalb einer Sitzung, z. B. um festzustellen, ob eine bestimmte Aktion bereits durchgeführt wurde,
- Anzeige von Systemvorschlägen für weiterführende Aktionen,
- Einholung von strategischer Unterstützung durch Experten,
- Visuelles Erkennen von Mustern,
- Anzeige häufiger Suchpfade,
- Planen der Suche,
- Annotieren der Suchhistorie.

Dabei erwies sich die horizontale Graphvisualisierung als vorteilhafter für die meisten Aufgaben, wenn auch deutlich platzhungriger. In der Darstellung von

strategischen Vorschlägen zeigte sich allerdings, dass gerade dieser Nachteil entscheidend war [GENDERA 2006]. Durch den geringen Platz konnte den Benutzern nicht genügend Kontext und Erklärung an die Hand gegeben werden, um ihnen eine informierte Entscheidung darüber zu erlauben, welche von mehreren vorgeschlagenen Aktionen sie ausführen sollten. Für die Umsetzung eines Unterstützungssystems durch strategische Suchvorschläge wurde daher eine andere Darstellungsform gewählt (siehe 6.1.3).

### **5.6. Zusammenfassung**

In diesem Kapitel wurde die grundlegende Architektur des DAFFODIL-Systems vorgestellt, innerhalb dessen die im Rahmen dieser Arbeit entwickelten Unterstützungsmechanismen umgesetzt wurden. Darüber hinaus wurden Erweiterungen in Gestalt von aufgabenabhängigen Perspektiven, eines Strategieframeworks, eines Loggingmechanismus und einer Visualisierung für Suchhistorien vorgestellt, die zur Unterstützung des Benutzers bei der Bildung einer Suchstrategie benötigt werden oder durch den Suchenden direkt bei der strategischen Planung genutzt werden können.

## Kapitel 6.

# Strategisches Vorschlagssystem

Zur Unterstützung von Benutzern bei der Strategiebildung und beim strategischen Suchen wurden zwei Vorschlagssysteme konzipiert, implementiert und untersucht. Das erste der Systeme basiert auf der Idee, in einer problematischen oder verbesserbaren Situation des Benutzers eine Taktik, ein Strategem oder sogar eine neue Strategie vorzuschlagen, mit deren Hilfe der Benutzer seine Suche weiterbringen kann. Für das zweite System wurden die Suchpfade vergangener Sitzungen ausgenutzt, um mögliche (und häufige) nächste Schritte vorzuschlagen. Die Idee basiert auf dem in Abschnitt 4.2 vorgestellten Pfadkonzept, das Chalmers als ein Modell der Benutzeraktivität vorgeschlagen hat [CHALMERS 1999].

### 6.1. Vorschläge durch fallbasiertes Schließen

Für die Umsetzung des situationsspezifischen, strategischen Vorschlagssystems wurde die Methode des fallbasierten Schließens oder *case-based reasoning* (CBR) gewählt. Dieser Problemlösungsansatz aus der Künstlichen Intelligenz entstand in den frühen 1980ern als eine Methodik, um auf der Basis von Erfahrungswissen neue Probleme zu bearbeiten [AAMODT und PLAZA 1994]. Der Ansatz benutzt dabei zumeist *Nearest-Neighbor*-Methoden, wie sie auch in anderen Gebieten (z. B. *Data Mining*) zur Anwendung kommen.

Das Wissen des Systems liegt hier nicht in Form von Regeln vor, sondern ist in der sogenannten Fallbasis in Gestalt von Fällen gespeichert. Ein Fall ist eine parametrisierte Beschreibungen einer früheren Problemsituation zusammen mit einer Lösung. Neue Probleme lassen sich nun lösen, indem ihre Parameter mit den Fällen der Fallbasis verglichen und das ähnlichste frühere Problem ermittelt wird. Die gespeicherte Lösung dieses Problems wird dann gegebenenfalls auf die Unterschiede zwischen der aktuellen Problemsituation und der aus dem gespeicherten Fall angepasst und dann wiederverwendet. Es können auch die  $n$  ähnlichsten Fälle ermittelt und deren Lösungen in absteigender Reihenfolge der Ähnlichkeit als Alternativen zur Auswahl angeboten werden.

Das CBR hat gegenüber z. B. regelbasierten Ansätzen eine Reihe von Vorteilen [MAIN et al. 2001]:

## 6.1. VORSCHLÄGE DURCH FALLBASIERTES SCHLIESSEN

---

- Neues Wissen lässt sich leicht akquirieren, wenn man einmal die notwendigen Parameter einer Problemsituation festgelegt hat. Es ist nicht notwendig, neues Wissen zu zerlegen und aufwändig zu modellieren.
- Die Fallbasis ist im Gegensatz zu einer Regelbasis relativ einfach zu warten. Das Hinzufügen bzw. Löschen alter Fälle führt normalerweise nicht zu unvorhergesehenen Seiteneffekten.
- Problemlösungen müssen nicht algorithmisch erarbeitet, sondern es können als tauglich erwiesene Lösungen wiederverwendet werden.
- Durch den Rückgriff auf frühere Fälle ist es möglich, das Ergebnis nachzuvollziehen und ist daher von Benutzern leichter zu akzeptieren, als das oft komplexe Zusammenspiel zahlreicher Regeln.

Das Grundprinzip des fallbasierten Schließens beschreiben Aamodt und Plaza mit einem Zyklus aus vier Phasen, wie er in Abbildung 6.1 gezeigt wird.

- In der *Retrieve*-Phase wird der neue Fall mit den Fällen der Fallbasis verglichen und die Ähnlichkeiten berechnet. Dabei können unterschiedliche Fallrepräsentationen und Ähnlichkeitsmaße zum Einsatz kommen.
- In der *Reuse*-Phase wird die Lösung des ähnlichsten Falls an die aktuelle Situation angepasst und als neue Lösung vorgeschlagen.
- In der *Revise*-Phase wird die Lösung angewendet und dadurch verifiziert. Bei erfolgreicher Verifizierung liegt ein neuer getesteter Fall mit einer bestätigten Lösung vor.
- In der *Retain*-Phase wird dieser neue Fall nun gelernt und in der Datenbasis gespeichert. Er steht nun zur Lösung neuer Problemsituationen zur Verfügung.

Angewandt auf die Suche stellt die Benutzersituation nach einer Suchanfrage eine Problemsituation dar, in der ein Benutzer möglicherweise Unterstützung benötigt. Lösungen für diese Situation sind mögliche taktische Vorschläge, die der Benutzer ausführen könnte. Die Situation wird daher mit gespeicherten Situationen in der Datenbasis verglichen und die Lösungen der ähnlichsten Fälle ermittelt. Diese werden auf die aktuelle Situation angepasst und dem Suchenden vorgeschlagen. Dieser kann den Vorschlag ausführen und dadurch die Lösung verifizieren. War der Vorschlag hilfreich, kann der Benutzer das bestätigen und die neue Kombination aus Situation und Vorschlag wird in der Fallbasis als Fall gespeichert.

Daneben kann es aber vorkommen, insbesondere da dem Benutzer mehrere Vorschläge zur Auswahl gestellt werden sollen, dass einige „Lösungen“ sich als besonders schlecht erweisen. Hier ist es notwendig, über das bloße Nichtlernen des neuen Falls hinaus schlechte Lösungen für zukünftige Entscheidungen zu berücksichtigen. Erweiterungen des fallbasierten Schließens, die



negative Evidenz miteinbeziehen, existieren insbesondere im Anwendungsfeld der Medizin. Eine Möglichkeit wird von Ontañón und Plaza [ONTAÑÓN und PLAZA 2003] in Form eines gewichteten *Votings* vorgeschlagen, bei dem die Ähnlichkeit eines neuen Falls sowohl zu positiven als auch zu negativen Fällen berücksichtigt wird.

### 6.1.1. Auswahl der Vorschläge

Für das Vorschlagswerkzeug wurden aus der existierenden Literatur (siehe Kapitel 2) Taktiken und Strategeme ausgewählt, die sich mit den Möglichkeiten von DAFFODIL umsetzen lassen und bei der Suche nützlich sein können. Dazu wurden auch Vorschläge aufgenommen, die auf den Erkenntnissen der vorangegangenen DAFFODIL-Evaluationen basierten [KLAS et al. 2004a, SCHAEFER et al. 2005]. Zunächst wurden dazu sechzehn verschiedene Vorschläge implementiert, die in Tabelle 6.1 wiedergegeben sind. Alle Vorschläge lassen sich in der Taxonomie von Bates einordnen und viele entsprechen direkt Taktiken aus ihrer Taxonomie, während andere zusätzliche Taktiken darstellen.

Diese Sammlung von Vorschlägen wurde für ein in Kapitel 8 beschriebenes Benutzerexperiment genutzt. Mit dem Experiment sollte die grundsätzliche Eignung des Werkzeugs, die Beschreibung und Präsentation der Vorschläge und die Bereitschaft von Benutzern, solche Vorschläge zu nutzen untersucht werden. Als Ergebnis des Benutzerexperiments wurden einige der Vorschläge angepasst und die Beschreibungen verbessert; außerdem wurden insgesamt 12 weitere Vorschläge in das Werkzeug aufgenommen.

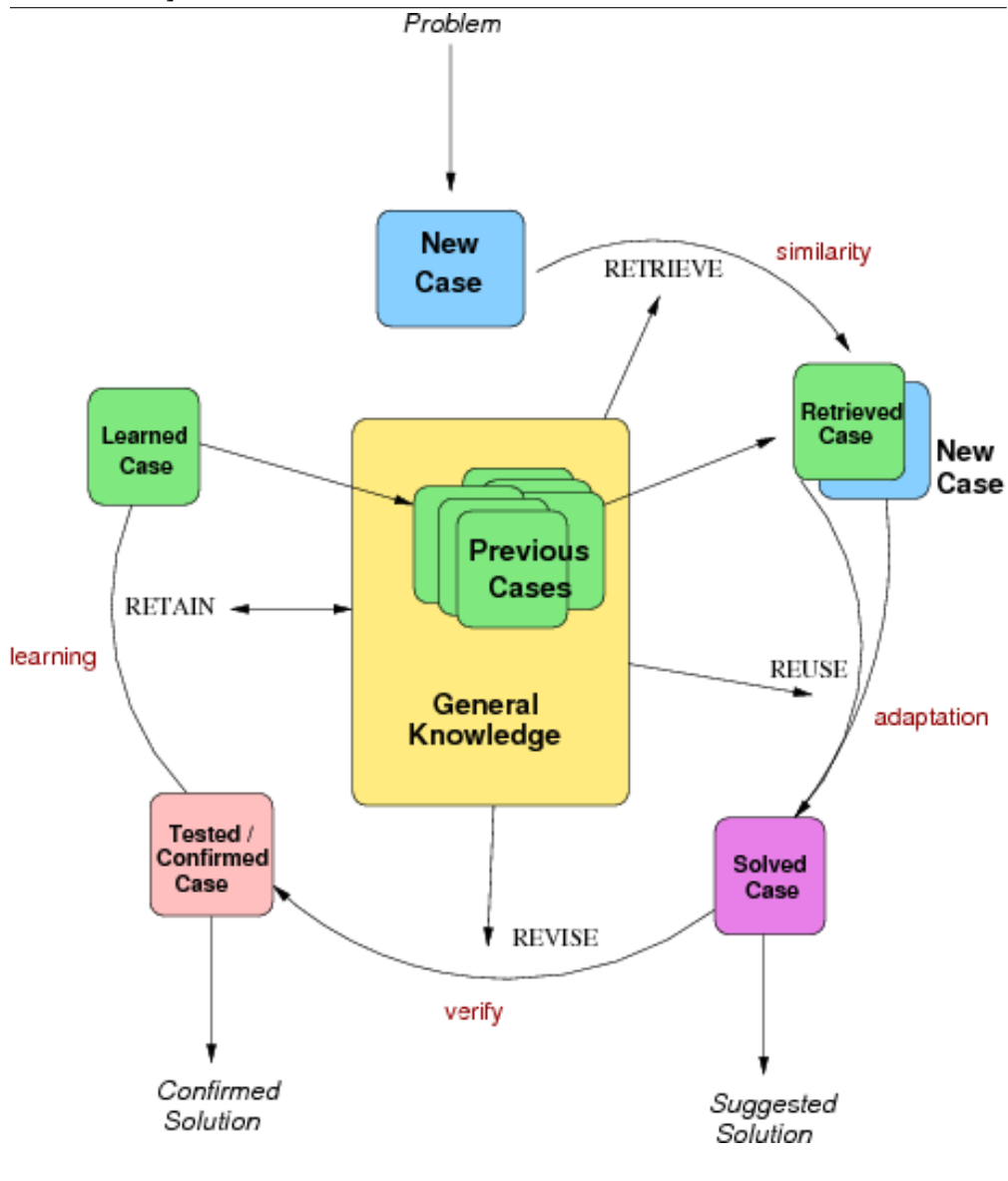
Wo es möglich war, wurden die Vorschläge so implementiert, dass sie an die aktuelle Situation angepasst und automatisch ausgeführt werden können. Nicht bei allen Vorschlägen war eine solche Vorgehensweise sinnvoll; dann beschränkt sich die Umsetzung auf einen generischen Vorschlag, der die jeweilige Beschreibung anzeigt. Eine aus drei oder mehr Begriffen bestehende Anfrage mag von der Nutzung eines Phrasenoperators profitieren, doch der Benutzer kann am besten einschätzen, welche der Begriffe tatsächlich zu einer Phrase gehören.

Zwei Beispiele für Vorschläge, die auf die Situation des Benutzers adaptiert werden, seien hier kurz beschrieben:

- Für das Strategem der Autorensuche werden vom System die häufigsten Autoren im aktuellen Suchergebnis ermittelt. War die letzte Suche des Benutzers bereits eine Autorensuche, so werden die Autoren aus der Liste entfernt, die in der Anfrage vorkamen. Für den häufigsten der verbleibenden Koautoren wird dann der Vorschlag einer weiteren Autorensuche gemacht.

## 6.1. VORSCHLÄGE DURCH FALLBASIERTES SCHLIESSEN

**Abbildung 6.1.** Der Zyklus beim fallbasierten Schließen nach [AAMODT und PLAZA 1994].



**Tabelle 6.1.** Erste Auswahl von Vorschlägen für DAFFODIL mit Einordnung nach Bates [BATES 1990].

<b>Strategeme</b>		
S1	Im häufigsten Journal aus der aktuellen Resultatmenge browsen.	Journal Run
S2	In den Proceedings der häufigsten Tagung aus der aktuellen Resultatmenge browsen.	—
S3	Mit Hilfe des Autorennetzwerks eine visuelle Darstellung der Koautoren für den häufigsten Autoren aus der aktuellen Resultatmenge generieren.	Author search
S4	Autorensuche durchführen für den häufigsten Autoren aus der aktuellen Resultatmenge (der nicht Teil der ursprünglichen Anfrage war).	Author search
S5	Mit Hilfe einer Klassifikation bessere Begriffe finden, um das Informationsbedürfnis zu beschreiben.	Index search
<b>Termtaktiken</b>		
T1	Mit dem Thesaurus einen spezielleren Suchbegriff finden.	SUB
T2	Mit dem Thesaurus einen allgemeineren Begriff finden.	SUPER
T3	Mit dem Thesaurus einen verwandten oder synonymen Term finden.	RELATE
T4	Anfrage einschränken durch Benutzung des Phrasen-Operators.	RESPACE
T5	Rechtschreibung variieren, z. B. um für die Unterschiede zwischen amerikanischem und britischem Englisch zu kompensieren.	RESPELL
<b>Anfragetaktiken</b>		
A1	Anfrage einschränken, indem der häufigste (neue) Autor aus der aktuellen Resultatmenge in die Anfrage aufgenommen wird.	—
A2	Anfrage einschränken, indem der häufigste (neue) Suchbegriff aus der aktuellen Resultatmenge in die Anfrage aufgenommen wird.	—
A3	Anfrage einschränken auf Publikationen aus den letzten Jahren.	—
A4	Anfrage erweitern durch Entfernen eines Suchbegriffs aus einer Konjunktion.	REDUCE
A5	Anfrage erweitern durch Ersetzen der impliziten Konjunktion durch eine Disjunktion.	PARALLEL
<b>Ideentaktiken</b>		
I1	Neue Suchbegriff extrahieren aus der aktuellen Resultatmenge und Anzeige als gewichtete Liste ( <i>Termwolke</i> ).	—

- Für die Taktik des Entfernens des Phrasenoperators wird die aktuelle Anfrage überprüft. Enthielt die Anfrage keinen Phrasenoperator, so wird der Vorschlag als nicht anwendbar erkannt und unterdrückt.

### 6.1.2. Berechnung der Ähnlichkeiten und Auswahl der Vorschläge

Die Beschreibung der Problemsituationen geschieht in der vorgenommenen Umsetzung des CBR durch Attribut-Werte-Paare. Dabei hängen die benutzten Parameter zur Beschreibung der Situation vom System ab, für das Vorschläge generiert werden sollen. Umsetzungen erfolgten für DAFFODIL (siehe Abschnitt 6.1.3) sowie für die Websuche am Beispiel Google (siehe Abschnitt 6.1.4). Allgemein können als Parameter numerische Werte, Termlisten, sowie Termlisten mit Gewichten auftauchen.

Eine neue Situation wird mit allen Situationen der Fallbasis verglichen, um eine Lösung zu finden. Um die Ähnlichkeit zweier Situationen zu berechnen, werden zunächst Ähnlichkeiten zwischen Parameterpaaren ermittelt. Dabei ergibt sich die Ähnlichkeit zwischen zwei numerischen Werten  $k_1$  und  $k_2$  als

$$sim_{num}(k_1, k_2) := \frac{\min(k_1, k_2)}{\max(k_1, k_2)}. \quad (6.1)$$

Die Ähnlichkeit zwischen zwei Listen (Sets)  $t_1$  und  $t_2$  von Termen wird bestimmt als Quotient der Anzahl gemeinsamer Terme zur Anzahl aller unterschiedlicher Terme in  $t_1$  und  $t_2$ :

$$sim_{set}(t_1, t_2) := \frac{|t_1 \cap t_2|}{|t_1 \cup t_2|} \quad (6.2)$$

Die Ähnlichkeit zwischen zwei gewichteten Termlisten oder Sets von Termen mit Häufigkeiten  $t_1$  und  $t_2$  wird bestimmt als normiertes inneres Produkt der Vektorrepräsentationen (*cosine similarity*).

$$sim_{wset}(t_1, t_2) := \frac{t_1 \cdot t_2}{|t_1| |t_2|} \quad (6.3)$$

Die Ähnlichkeit  $sim_T(a, c)$  zweier Situationen  $a$  und  $b$  wird dann berechnet als gewichtetes Mittel (mit Gewichten  $w_k$ ) der einzelnen Ähnlichkeitswerte  $sim_k$  zwischen den Aspekten einer Situation:

$$sim_T(a, b) := \frac{\sum_{k=0}^N w_k \cdot sim_k(a_k, b_k)}{\sum_{k=0}^N w_k} \quad (6.4)$$

**Tabelle 6.2.** Zusätzliche Vorschläge für DAFFODIL mit Einordnungen u.a. nach Bates [BATES 1990].

<b>Strategien</b>		
P1	Anfrage in konzeptuelle Facetten zerlegen und entweder einzeln anfragen oder kombinieren.	Building Blocks
P2	Ein als relevant gespeichertes Dokument benutzen, um von diesem aus weitere Dokumente zu finden.	Pearl Growing
P3	Häufige Begriffe aus den während der Suche gespeicherten Dokumenten extrahieren und für neue Anfragen nutzen.	Interactive Scanning
<b>Strategeme</b>		
S6	Benutzen der eingebauten Suche zur Ergänzung von Dokumentlisten, um ähnliche Dokumenten zu den Dokumenten der Zwischenablage zu finden (siehe 7.1.4).	—
S7	Ähnliche Anfragen anderer Benutzer verwenden, um zusätzliche relevante Dokumente zu finden (siehe 6.1.5).	—
<b>Beobachtungstaktiken</b>		
B1	Die Suchhistorie nutzen, um Wiederholungen zu vermeiden und besonders effektive Anfragen zu erkennen.	RECORD
<b>Strukturtaktiken</b>		
F1	Filter auf das Suchergebnis anwenden, um möglichst große Teile auszublenden.	CUT
<b>Termtaktiken</b>		
T6	Anfrage erweitern durch Entfernen des Phrasen-Operators.	RESPACE
T7	Benutzung von Termen aus den Verwandten Begriffen.	NEIGHBOR
<b>Anfragetaktiken</b>		
A6	Eine bestehende Überspezifikation der Anfrage beseitigen.	REDUCE
A7	Benutzung des Freitextfelds statt des Titelfelds.	—
A8	Benutzung des Titelfelds statt des Freitextfelds.	—
<b>Ideentaktiken</b>		
I2	Hilfe heranziehen durch Chat mit anderem Benutzer.	CONSULT
I3	Hilfe heranziehen durch Konsultation eines Bibliothekars.	CONSULT

### Auswahl und Ranking der Vorschläge

Sind alle Ähnlichkeiten bestimmt, so geschieht die Auswahl und die Gewichtung der Vorschläge in drei Phasen.

1. In der ersten Phase werden alle Vorschläge ausgewählt, für die zumindest ein positiver Fall in der Fallbasis existiert, der zur aktuellen Benutzersituation eine Ähnlichkeit größer einem Schwellenwert  $m_p$  (z. B. 0,5) besitzt.

Dabei wird zu jedem Vorschlag  $v$  der Fall  $b_v$  mit der größten Ähnlichkeit  $s_v := \text{sim}_T(a, b_v)$  gewählt.

2. Nun werden für alle bisher ausgewählten Vorschläge alle positiven und negativen Fälle aus der Fallbasis betrachtet, die diesen Vorschlag als Lösung besitzen.

Sei  $F^+(v)$  die Menge aller positiven Fälle, die den Vorschlag  $v$  als Lösung besitzen.

Sei  $F^-(v)$  die Menge aller negativen Fälle, die den Vorschlag  $v$  als Lösung besitzen

Das gesamte Vorschlagsgewicht  $p_v$  für den Vorschlag ergibt sich dann als

$$p_v := s_v * \frac{\sum_{b_i \in F^+(v)} \text{sim}_T(a, b_i)}{\sum_{b_i \in F^+(v)} \text{sim}_T(a, b_i) + \sum_{b_i \in F^-(v)} \text{sim}_T(a, b_i)} \quad (6.5)$$

- Sind alle Fälle, die diesen Vorschlag als Lösung besitzen, positiv, so ist  $p_v = s_v$ .
  - Durch die Auswahl der Vorschläge in der ersten Phase kann nicht eintreten, dass alle Fälle die den Vorschlag als Lösung besitzen, negativ sind. Das Vorschlagsgewicht ist aber auch für diesen Fall definiert und  $p_v = 0$ .
  - Gibt es sowohl negative als auch positive Fälle, die den Vorschlag als Lösung besitzen, so ist  $0 < p_v < s_v$ .
3. Alle Vorschläge mit einem Vorschlagsgewicht  $p_v$  größer dem Schwellenwert  $m_p$  werden in abfallender Reihenfolge ihres Vorschlagsgewichtes geordnet.

### 6.1.3. Umsetzung für DAFFODIL

Eine erste Umsetzung des Konzeptes erfolgte für DAFFODIL [KRIEWEL und FUHR 2007, KLAS et al. 2008] in Form eines Vorschlagsmoduls namens ASDL

(Adaptive Support for Digital Libraries), das sich als flexibel genug erwies, um auch für andere Systeme eingesetzt werden zu können. DAFFODIL bot eine gute Umgebung zur ersten Umsetzung durch seine reiche Werkzeugvielfalt, sowie durch das Loggingframework, welches das Sammeln von Informationen über den Fortschritt und die Situation des Benutzers erleichterte.

Das Vorschlagsmodul setzt sich aus drei Komponenten zusammen, wobei zwei als Agenten im DAFFODIL-Backend implementiert wurden und die dritte als Werkzeug im Benutzerinterface. Diese Unterteilung entspricht der von Jameson für die Adaption von Mensch-Maschine-Schnittstellen vorgestellten nach Afferenz, Inferenz und Efferenz [JAMESON 2001].

- der *Observing Agent* sammelt Informationen über die aktuelle Benutzersituation, indem er Nachrichten des Agentenframeworks auswertet (Afferenz);
- der *Reasoning Agent* hält die Fallbasis vor (die in einer relationalen Datenbank gespeichert ist) und vergleicht eine hereinkommende Situation mit den gespeicherten Fällen (Inferenz);
- die so gewichteten Vorschläge werden an die Benutzerschnittstelle geschickt, wo das *Vorschlagswerkzeug* diese anpasst (Efferenz) und sie als eine Rangliste den Benutzer präsentiert.

### **Observing Agent**

Der *Observing Agent* beobachtet den Nachrichtenfluss von DAFFODIL und wird bei einer Suchanfrage eines Benutzers aktiv. Um eine situationsspezifische Liste von Vorschlägen zu erstellen, wird die Suchsituation des Benutzers (siehe 3.1) ausgewertet. Eine Suchsituation setzt sich dabei aus der Anfrage des Benutzers und der Antwort des Systems zusammen. Hinzugenommen werden Informationen über den Verlauf der Suchsitzung.

Aus der Anfrage des Suchenden gehen in die Beschreibung der Situation ein

- die benutzten Suchbegriffe als Termliste,
- die benutzten Suchfelder,
- die Anzahl boolescher Operatoren, die vom Benutzer eingesetzt wurde, sowie
- die Informationsquellen, in denen gesucht wurde.

Hinzu kommen aus dem endgültigen Suchergebnis, das vom System dem Benutzer präsentiert wird,

- die Anzahl gefundener Dokumente,
- die Antwortzeit des Systems, sowie

## 6.1. VORSCHLÄGE DURCH FALLBASIERTES SCHLIESSEN

---

- gewichtete Listen mit den häufigsten Begriffen, Autoren, Journalen und Tagungen aus dem Resultat.

Schließlich kann der *Observing Agent* aus der Datenbank zur laufenden Sitzung des Benutzers die vom Loggingframework festgehaltene Suchhistorie mit berücksichtigen. Dabei werden die folgenden Informationen verwendet:

- die bisher benutzten Suchbegriffe aus den Anfragen (als gewichtete Liste) und
- die in den bisher gespeicherten Ergebnissen vorkommenden Begriffe (ebenfalls als gewichtete Liste).

Weitere Parameter einer Suchsituation lassen sich leicht hinzunehmen, allerdings wird die Berechnung der Ähnlichkeiten mit jedem zusätzlichen Parameter aufwändiger, ohne dass sich das Ergebnis notwendigerweise verbessert. Zusätzliche Parameter, die nicht umgesetzt wurden aber leicht aus der Suchhistorie zu entnehmen sind, könnten basierend auf Abschnitt 3.1 beispielsweise sein:

- die Gesamtzeit, die der Benutzer bereits für die Suchsitzung aufgewendet hat,
- die Durchschnittszeit, die der Benutzer üblicherweise in eine Suchsitzung investiert,
- die Anzahl der Suchanfragen in der Sitzung,
- die durchschnittliche Zahl von Ergebnissen bei den bisherigen Anfragen der Suchsitzung,
- die Anzahl bislang in der Sitzung gespeicherter Dokumente,
- die letzten vom Benutzer getätigten Aktionen bzw. benutzten Werkzeuge (siehe hierzu Abschnitt 6.2),
- Anzahl bisher befolgter Vorschläge.

Die vom *Observing Agent* zusammengestellte und parametrisierte Beschreibung der Suchsituation wird als Nachricht über das Agentenframework von DAFFODIL an den *Reasoning Agent* gesendet, der diese weiterverarbeitet.

### **Reasoning Agent**

Der *Reasoning Agent* hat Zugriff auf die Fallbasis in der Datenbank, hält jedoch aus Effizienzgründen eine Kopie der Fallbasis im Arbeitsspeicher und frischt diese nach jedem neu aufgenommenen Fall auf.



Trifft die Beschreibung einer Benutzersituation im Agenten ein, so werden wie in Abschnitt 6.1.2 beschrieben die Ähnlichkeiten zwischen der neuen Situation und den Situationen der Fallbasis ermittelt und basierend darauf ein Ranking von Vorschlägen erstellt. Daraufhin wird zunächst eine Benachrichtigung an den Benutzeragenten (die Benutzerschnittstelle) des Suchenden verschickt, dass Vorschläge zur aktuellen Situation vorliegen. Fordert der Benutzer die Vorschläge an, wird das Ranking einschließlich der zu den Vorschlägen gehörenden Beschreibungen gesendet. Für Vorschläge, die automatisch ausführbar sein sollen, wird auch der Name der Java-Klasse mitgeschickt, welche den Vorschlag umsetzt. Dieser Vorgang stellt im CBR-Zyklus die Phase des RETRIEVE dar.

Der *Reasoning Agent* ist darüber hinaus dafür zuständig, neue Fälle zu lernen, bei denen vom Benutzer Bewertungen über die Nützlichkeit abgegeben wurden. Der Agent setzt also auch die Phase des RETAIN im CBR-Zyklus um. Dazu wird die Beschreibung der Situation solange vorgehalten, bis in der gleichen Sitzung entweder neue Vorschläge angefordert werden oder die Sitzung vom Benutzer beendet wird. Im Falle einer abgegebenen Benutzerwertung wird die Situation zusammen mit dem bewerteten Vorschlag und der Wertung in der Datenbank gespeichert und die Fallbasis aufgefrischt.

Eine initiale Fallbasis wurde erzeugt, in dem zu jedem Vorschlag des Systems ein oder zwei ikonische, repräsentative Situationen erstellt wurden, in denen dieser Vorschlag hilfreich wäre. Für neue Vorschläge muss zumindest eine passende Situation in die Fallbasis aufgenommen werden, zu der dieser Vorschlag eine passende Lösung wäre.

Zur Initialisierung des Systems wurden zusätzlich generische Situationen benutzt, wie in Abschnitt 3.1.2 beschrieben. Dabei wurde der NULL-Wert des Datenbanksystems als ein „generischer“ Wert für einen oder mehrere Parameter benutzt, intern beschrieben durch ein besonderes Zeichen (\$). Dieser Wert soll ausdrücken, dass dieser Parameter zur Beschreibung der Situation nicht benutzt wird. Bei der Ähnlichkeitsberechnungen werden alle Teilähnlichkeiten, bei denen einer der Werte der generische Wert ist, auf 1 gesetzt. Der Vergleich zweier Situationen, bei denen für alle Parameter bei mindestens einer der beiden Situationen der generische Wert vorliegt, ergibt dann immer die Ähnlichkeit  $s = 1$ .

### **Vorschlagswerkzeug**

Das *Vorschlagswerkzeug* ist Teil der Benutzerschnittstelle von DAFFODIL und deckt die Phasen des REUSE und REVISE im CBR-Zyklus ab. Zusammen mit der Darstellung der Suchergebnisse wird dem Suchenden zunächst ein unaufdringlicher, aber gut sichtbarer Knopf unter der Resultatliste präsentiert (in Abbildung 5.4 ist dieser Knopf zur Rechten des Such-Fortschrittsbalken zu

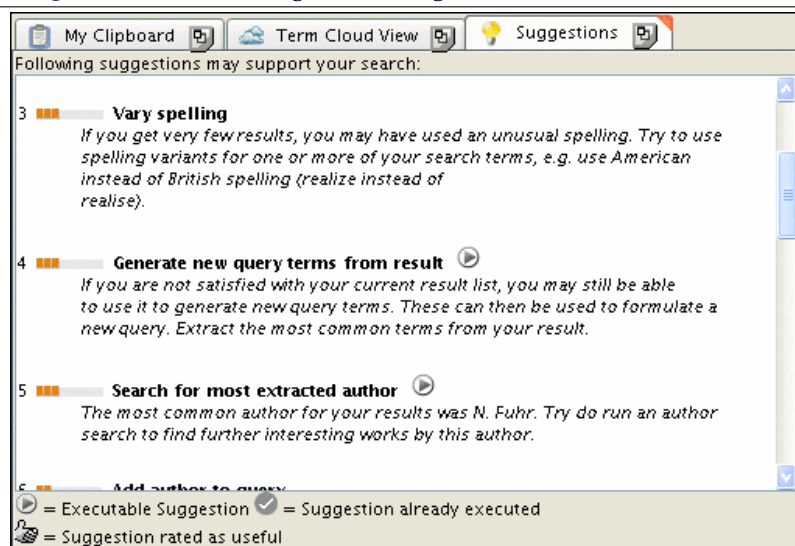
## 6.1. VORSCHLÄGE DURCH FALLBASIERTES SCHLIESSEN

sehen). Der Benutzer kann durch Anwahl des Knopfes die vorhandenen Suchvorschläge abrufen, die dann in Form einer nach abfallenden Vorschlagsgewichten geordneten Liste präsentiert werden (siehe Abbildung 6.2). Die Vorschlagsgewichte werden dabei zusätzlich durch einen kleinen, farbigen Balken neben jedem Vorschlag dargestellt, der die geschätzte Anwendbarkeit des Vorschlags auf die aktuelle Situation wiedergibt. Diese Darstellung wurde von der Präsentation der Ergebnisliste in DAFFODIL übernommen und ist dem Benutzer daher vertraut. Die Evaluation des Vorschlagssystem zeigte, dass diese Art der Darstellung auch leicht verstanden wurde (siehe Abschnitt 8.1).

Die Vorschläge werden dabei vor der Anzeige für den Benutzer einer Anpassung unterzogen, und für die konkrete Situation nicht anwendbare Vorschläge werden unterdrückt. Für direkt ausführbare Vorschläge wird ein Objekt der spezifischen Vorschlagsklasse instantiiert, welches die Umsetzung des Vorschlags innerhalb von DAFFODIL implementiert. Für die nicht automatisierten Vorschläge ist eine generische Klasse vorgesehen.

Die Darstellung jedes Vorschlags in der Liste setzt sich zusammen aus einem kurzen beschreibenden Titel und einer längeren Erklärung, sowie kleinen Grafiken (*Icons*), die anzeigen, ob der Vorschlag ausführbar ist, bereits angewendet und/oder bereits positiv oder negativ bewertet wurde. Die Beschreibung des Vorschlags erklärt zum einen, wie dieser umgesetzt werden kann, und zum anderen, mit welchem Ziel er vorgeschlagen wurde (z. B. um die problematische Situation einer leeren Resultatliste zu überwinden, oder um ein zu umfangreiches Ergebnis einzuschränken).

Abbildung 6.2. Das Vorschlagswerkzeug.



Der Benutzer kann automatisch ausführbare Vorschläge durch einfachen Doppelklick auf den Listeneintrag ausführen. Dabei bleibt die Vorschlagsliste solange präsent, wie der Benutzer keine neuen Vorschläge anfordert, so dass zu

einer Situation mehrere der Vorschläge ausgeführt werden können. Die Nützlichkeit (oder die fehlende Nützlichkeit) eines Vorschlags kann direkt in der Liste bewertet werden.

Die Entscheidung, Vorschläge nicht automatisch für jede Suche anzuzeigen, sondern erst auf Benutzeranforderung, fiel aus drei Gründen:

- In einer Benutzerbefragung sprach sich eine starke Mehrheit (10 von 12) der Teilnehmer gegen automatische Vorschläge aus. Darüber hinaus zeigten Jansen und McNeese [JANSEN und MCNEESE 2005], dass es eine nicht zu vernachlässigende Benutzergruppe gibt, die automatische Benutzerunterstützung grundsätzlich ablehnt.
- In einer Suchsituation gibt es mehr als eine mögliche „Lösung“. Dem Benutzer werden daher eine Reihe von Vorschlägen gemacht, von denen er mehrere einsetzen kann, um einen geeigneten zu finden. Eine automatische Anzeige neuer Vorschläge nach einer von einem Vorschlag ausgelösten Suche würde entweder den Zugriff auf die alten Vorschläge verhindern oder zu einer unübersichtlichen Anhäufung von Vorschlagslisten aus verschiedenen Situationen führen.
- Ebenso würde das automatische Ersetzen einer Vorschlagsliste nach einer Suche durch eine neue, aktualisierte verhindern, dass Benutzer die Vorschläge bewerten können. Dies ist aber eine wichtige Voraussetzung für die RETAIN-Phase des CBR-Zyklus.

### 6.1.4. Umsetzung für Websuche

Um zu zeigen, dass das Konzept der eines strategischen Vorschlagssystem auf Basis fallbasierten Schließens auch auf andere Suchsysteme übertragbar ist, erfolgte eine zweite Umsetzung. Diese sollte die Websuche unterstützen, bei der sich sowohl die Suchsituationen des Benutzers als auch die ihm möglichen Taktiken erheblich von denen bei der Suche mit DAFFODIL unterscheiden. Als Fallstudie wurde die Websuchmaschine Google<sup>1</sup> gewählt, als Mechanismus für die Umsetzung ein Browser-Addon für den Open-Source-Browser Firefox<sup>2</sup> [AWASUM 2008].

Die technische Umsetzung erfolgte in zwei wesentlichen Komponenten. Die Benutzerschnittstelle wurde als XUL-Plugin unter Verwendung von JavaScript und AJAX-Technologie in den Firefox-Browser integriert<sup>3</sup>. Das Backend basiert auf der für DAFFODIL entwickelten Implementierung, bietet aber eine HTTP-Schnittstelle in Gestalt eines integrierten Webservers. Die Kommunikation zwischen den beiden Komponenten erfolgt durch Nachrichten im JSON-Format.

---

<sup>1</sup><http://www.google.com>

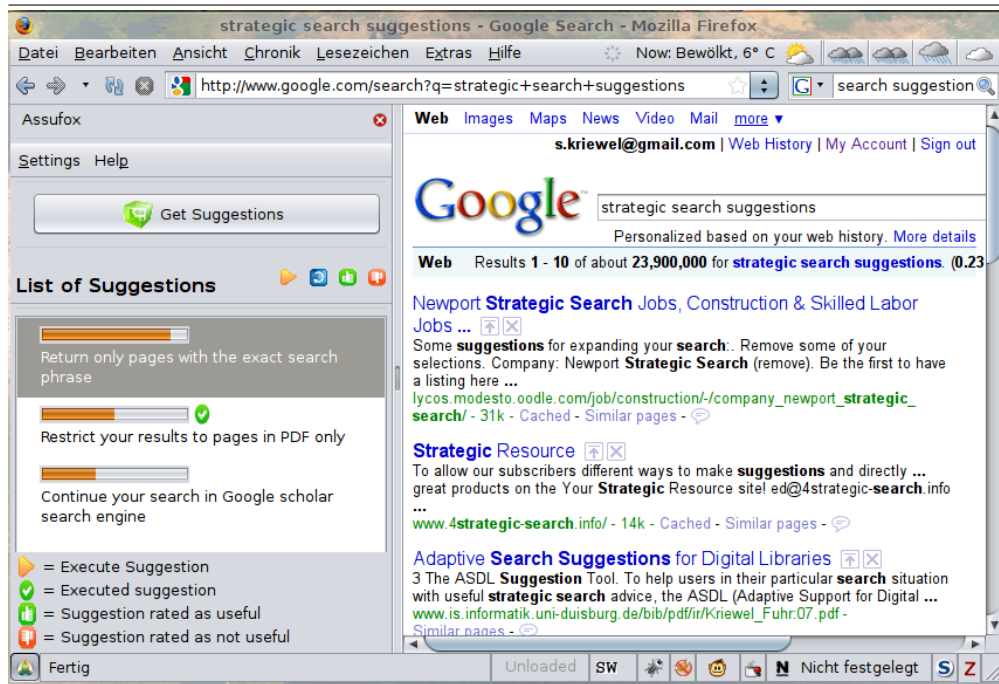
<sup>2</sup><http://www.mozilla-europe.org/de/firefox/>

<sup>3</sup><https://addons.mozilla.org/en-US/firefox/addon/9767>

**Tabelle 6.3.** Vorschläge für Google-Websuche, Einordnung nach Bates [BATES 1990].

<b>Strategem</b>		
GS1	Thematische Suche in den kategorisierten Seiten des Google Directory.	Catalog Subject Scan
<b>Anfragetaktiken</b>		
GA1	Anfrage erweitern unter Benutzung verwandten Begriffs.	NEIGHBOR
GA2	Anfrage erweitern unter Benutzung aller Synonyme von Anfragebegriffen.	PARALLEL
GA3	Anfrage erweitern durch Ersetzen der impliziten Konjunktion durch eine Disjunktion.	PARALLEL
GA4	Begriffe mit - aus der Anfrage ausschließen.	BLOCK
GA5	Nur nach dem exakten Begriff suchen.	SPECIFY
GA6	Anfrage einschränken auf Seiten aus dem letzten Jahr / den letzten 24 Stunden.	—
GA7	Anfrage einschränken auf Seiten, bei denen die Suchbegriffe im Titel / in der URL / im Seitentext / im Link vorkommen.	—
<b>Termtaktiken</b>		
GT1	Anfrage einschränken durch Benutzung von <i>Phrasen</i> .	RESPACE
GT2	Zur aktuellen Suche verwandte Begriffe finden.	RELATE
GT3	Thesaurus nutzen, um zu einem aktuellen Suchbegriff ein <i>Synonym</i> zu finden.	RELATE
<b>Strukturtaktiken</b>		
GF1	Auf Seiten aus einer bestimmten Domäne einschränken.	CUT
GF2	Auf Seiten in einer bestimmten Sprache einschränken.	CUT
GF3	Auf Seiten aus dem Land des Benutzers einschränken.	—
GF4	Auf Dateien in einem bestimmten Format (z. B. PDF) einschränken.	—
GF5	Google Scholar für die Suche in akademischen Quellen nutzen.	—
GF6	Google News für eine Suche in aktuellen Nachrichtenmeldungen nutzen.	—
GF7	Die Definition eines Suchbegriffs abfragen.	—
GF8	Suche in Blogs durchführen.	—
GF9	Suche in digitalisierten Büchern.	—

Abbildung 6.3. Das Vorschlagswerkzeug für Google.



Dabei werden die Vorschläge (wie in Abbildung 6.3 zu sehen) in einer Seitenleiste des Webbrowsers angezeigt. Darüber hinaus ist auch das Affferenzmodul als Teil des Browser-Addons implementiert. Es beobachtet die mit Firefox über Google getätigten Websuchen und sendet auf Anforderung des Benutzers eine Nachricht mit der Anfrage von Vorschlägen ans Backend. Dieses vergleicht die Situation des Benutzers (deren Parameter als Teil der Vorschlagsanforderung mitgesendet wurden) mit den in der Fallbasis gespeicherten Situationen und generiert wiederum ein Ranking von Vorschlägen, die es als Antwort an das Browser-Addon zurücksendet.

Die Umsetzung erlaubt ebenso wie die im vorangegangenen Abschnitt beschriebene die automatische Ausführung von Vorschlägen, sowie die Bewertung von Vorschlägen durch den Benutzer als hilfreich oder nicht hilfreich. Sie unterscheidet sich jedoch in den benutzten Parametern zur Beschreibung einer Suchsituation sowie in den verfügbaren (und umgesetzten) Vorschlägen. Diese sollen hier kurz beschrieben werden.

### Suchsituation im Webkontext

Nicht alle Parameter, die bei der Umsetzung des Vorschlagssystems für DAF-FODIL für die Charakterisierung einer Situation genutzt wurden, ließen sich sinnvoll auf den Webkontext übertragen.

- Die Antwortzeit ist im Allgemeinen bei Websuchen extrem gering und

## 6.1. VORSCHLÄGE DURCH FALLBASIERTES SCHLIESSEN

---

schwankt stärker bei Wiederholungen der gleichen Suchanfrage durch Latenz oder Auslastungsspitzen des Servers, als durch Variation der Anfrage. Daher wurde auf diesen Parameter verzichtet.

- Da die Unterstützung für die einfache Benutzerschnittstelle von Google erfolgte, wurde ebenso auf den Parameter der Suchfelder verzichtet.
- Durch die Beschränkung auf Google wurde der Parameter der Informationsquellen überflüssig.
- Ebenso wurde die Suchhistorie nicht mit berücksichtigt, da im Webkontext kein wirkliches Sitzungskonzept existiert.

Damit verbleiben als Parameter die benutzten Suchbegriffe, die Anzahl der Ergebnisdokumente, sowie die folgenden Informationen über die Suchergebnisse: die URLs und Domänen der Ergebniswebseiten, die extrahierten und nach Vorkommenshäufigkeit gewichteten Begriffe aus den Ergebnistiteln sowie aus den kurzen anfragespezifischen Ergebnisauszügen, die Google zu jedem Resultat anzeigt. Zusätzlich wurden noch automatisch ermittelte Synonyme der Anfragebegriffe mit berücksichtigt.

Daneben bieten sich eine Reihe weiterer Merkmale einer Suchsituation im Webkontext an, die jedoch nicht genutzt wurden:

- Sprache der Suchresultate, wenn sie vom Benutzer festgelegt wurde,
- benutzte Suchoperatoren wie
  - AND/OR**
  - Verneinung
  - ~ ähnliche Begriffe
  - “...” Phrasensuche
  - + exakte Suche,
- benutzte Einschränkungen wie
  - `allintitle`: Begriffe im Titel der Seite
  - `allinurl`: Begriffe in der URL der Seite
  - `allinanchor`: Begriffe im Link auf die Seite
  - `filetype`: Art der Datei
- benutzte Einschränkungen der Suchdomäne nach Sprache oder Lokalisation der Webseite,
- Suchbegriffe aus der Suchhistorie des Browsers während der letzten Stunde (oder einer anderen Zeitspanne, die statt einer Suchsitzung verwendet wird),

- Begriffe aus den Lesezeichen, die der Benutzer während der letzten Stunde neu angelegt hat (ebenso).

Die Berechnung der Ähnlichkeit zwischen zwei Situationen wurde basierend auf den gewählten Parametern wie in Abschnitt 6.1.2 beschrieben durchgeführt.

### Vorschläge für die Websuche

Auch die Auswahl der Suchvorschläge wurde auf die Websuche angepasst, dabei wurden insbesondere die Möglichkeiten genutzt, die von der Suchmaschine (Google) selbst zur Spezifikation der Anfrage angeboten werden. Da die z. B. von DAFFODIL angebotenen Werkzeuge zur direkten Umsetzung von Suchstrategien weitestgehend fehlen (mit Ausnahme einer Suche in einer Klassifikation, dem Google Directory<sup>4</sup>), wurden als Ersatz Vorschläge für die Suche in alternativen Suchdomänen mitaufgenommen:

- Google Scholar<sup>5</sup>
- Google News<sup>6</sup>
- Google Books<sup>7</sup>

Es wurde auch hier für die ausgewählten Vorschläge versucht, diese in die Taxonomie nach Bates einzuordnen (siehe Tabelle 6.3).

### Evaluation

Die Tauglichkeit des Firefox-Addons für die Unterstützung von Benutzern bei der Websuche mit Google wurde evaluiert [AWASUM 2008]. Dabei bearbeiteten zehn Studenten aus den Bereichen Informatik, Wirtschaftsinformatik, Communications Engineering und Mechanical Engineering eines von drei vorgegebenen Themen. Zu diesem sollten sie mit Hilfe von Google relevante Artikel und Webseiten finden und als Lesezeichen markieren. Für die Zwecke der Evaluation wurden Bewertungen von Vorschlägen nicht gespeichert und zwölf geordnete Vorschläge für jede Suchanfrage präsentiert, zu der Teilnehmer Unterstützung anforderten. Die Benutzer bewerteten durchschnittlich zwei Vorschläge aus der angezeigten Liste als nützlich, wobei der durchschnittliche Rang des besten als nützlich bewerteten Vorschlags 2,6 war (von zwölf angezeigten).

Acht der zehn Teilnehmer bewerteten die Arbeit mit dem Werkzeug in einer Reihe von Fragen als generell positiv und die Vorschläge als hilfreich. Dabei

---

<sup>4</sup><http://www.google.com/dirhp>

<sup>5</sup><http://scholar.google.com>

<sup>6</sup><http://news.google.com>

<sup>7</sup><http://books.google.com>

traten jedoch auch eine Reihe von Problemen mit den Beschreibungen der Vorschläge und der Interaktion mit dem Werkzeug zutage. Die Darstellung des Rankings wurde von einigen Benutzern nicht verstanden, da durch die geringe benutzte Fallbasis neue Situationen oft nur sehr geringe Ähnlichkeiten zu den gespeicherten aufwiesen. Es zeigte sich, dass eine ausreichend große Fallbasis Voraussetzung für die Nützlichkeit und Akzeptanz des Vorschlags-systems ist.

### 6.1.5. Vorschläge basierend auf Suchergebnissen

Als zusätzliche Möglichkeit, Suchvorschläge für Benutzer zu generieren, wurden Vorschläge untersucht, die auf der Ähnlichkeit des aktuellen Suchergebnisses des Benutzers zu früheren Suchergebnissen aus dem Suchlog des Systems basieren [BODNARIU 2009]. Die zugrundeliegende Annahme dabei war, dass Suchanfragen mit hoher Übereinstimmung der Ergebnisse zur aktuellen Anfrage des Benutzers geeignete Vorschläge darstellen.

Hierzu werden die geloggten Ergebnisse früherer Suchfragen (anderer Benutzer) als Pseudodokumente aufgefasst und die Titel der Dokumente aus den Ergebnissen als Pseudoterme. Für diese Pseudoterme und -dokumente wird dann eine invertierte Listenstruktur aufgebaut, welche benutzt werden kann, um zu einem neuen Suchergebnis ähnliche frühere Suchergebnisse zu finden.

Existiert ein solches früheres Ergebnis  $F$  mit hoher Ähnlichkeit zum aktuellen Ergebnis  $E$ , so können die nicht im Schnitt von  $F$  und  $E$  vorkommenden Dokumente als Ergänzung zur Resultatliste oder die zu  $F$  führende Anfrage als eine neue, alternative Anfrage vorgeschlagen werden. Die durchgeführten Experimente deuten darauf hin, dass diese Vorschläge vornehmlich für relativ kleine Ausgangsergebnisse  $E$  (weniger als 50 Dokumente) mit hoher *Precision* zu guten Resultaten führen. Eine Korrelation zwischen *Precision* der Ausgangsergebnisse oder Ähnlichkeit zwischen  $E$  und  $F$  und der *Precision* der vorgeschlagenen Ergebnisdokumente konnte allerdings nicht gezeigt werden. Hier spielte auch die relativ kleine Stichprobe des Experiments eine Rolle.

Innerhalb des adaptiven Vorschlagssystem finden die Vorschläge basierend auf Suchergebnissen Eingang als ein für bestimmte Fälle geeignetes Strategem (siehe Tabelle 6.2).

## 6.2. Vorschläge auf Basis von Suchpfaden

Als eine Alternative zu situationsbasierten Vorschlägen für die strategische Suche wurden auch pfadbasierte Suchvorschläge betrachtet [BODNARIU 2009]. Hier wird nicht der Suchkontext nach einer Suchanfrage als Grundlage für Empfehlungen genommen, sondern die Kette der Aktionen, die der Suchende



auf dem Weg zur aktuellen Situation durchgeführt hat. Dabei kann die aktuelle Situation nach der Benutzung eines beliebigen Werkzeugs erfasst werden, während die im vorangegangenen Unterkapitel vorgestellte Methodik erfordert, dass eine Suchanfrage und ein Suchergebnis vorliegt. Die pfadbasierte Methode zur Vorschlagsgenerierung ist daher flexibler einsetzbar.

Die Nutzung der Suchpfade basiert auf ähnlichen Ideen von Chalmers [CHALMERS 1999], die in Abschnitt 4.2 ausführlich behandelt wurden. Benutzer bewegen sich durch eine Nacheinanderausführung von Aktionen mit verschiedenen Werkzeugen auf ein Suchziel zu. Dabei hinterlassen sie einen Pfad, der aus der Reihung ihrer Suchaktivitäten entsteht. Überlagert man die Pfade zahlreicher erfolgreicher Benutzer, so entstehen ausgetretene Trampelpfade, die genutzt werden könnten, um aussichtsreiche nächste Aktionen vorzuschlagen. Die Einschränkung auf erfolgreiche Pfade soll verhindern, dass wiederkehrende Fehler erlernt werden, ist aber zugleich schwierig, da die Erfolgsbestimmung die Kenntnis des Suchziels des Benutzers erfordert. Das ist bei der Betrachtung von Suchlogs meist nicht klar zu erkennen, wenn die Aufgabe des Benutzers nicht (z. B. durch ein Experiment) vorgegeben wurde. Daher wurde diese Beschränkung zunächst außer Acht gelassen.

Hier seien zwei Beispiele für teilweise Suchpfade gegeben, die durch die Werkzeuge und Optionen von DAFFODIL möglich sind:

### **Beispiel 1:**

**Schritt 1:** Speichern eines Dokumentes in der Zwischenablage

**Schritt 2:** Speichern eines Dokumentes in der Zwischenablage

**Schritt 3:** Extrahieren von Begriffen aus gespeicherten Dokumenten

Benutzer, die diesen Pfad begangen haben, haben zunächst relevante oder möglicherweise relevante Dokumente gesammelt, und diese dann benutzt, um neue Suchbegriffe zu erhalten. Andere Benutzer, die den ersten Teil des Pfades begehen (also zwei Dokumente hintereinander ablegen), könnten als Vorschlag den Rat erhalten, auch den dritten Schritt durchzuführen.

### **Beispiel 2:**

**Schritt 1:** Betrachten eines Dokumentes aus dem Ergebnis

**Schritt 2:** Speichern eines Dokumentes in der Zwischenablage

**Schritt 3:** Suche nach Autoren durch Klick auf Link im Detail

In diesem Pfad haben Benutzer ein Ergebnisdokument betrachtet, dieses als (möglicherweise) relevant befunden, und es in der Zwischenablage abgelegt.

Anschließend führten sie eine Suchanfrage nach einem Autoren durch, den sie im Dokument entdeckt haben. Auch dies kann man als Vorschlag nutzen.

Bei der Betrachtung der DAFFODIL-Logs stellte sich heraus, dass Dreier-Kombinationen von Suchschritten am geeignetsten für auf Pfaden basierende Vorschläge sind [BODNARIU 2009]. Pfaden aus zwei Schritten fehlte der Kontext, um sinnvolle Vorschläge zu generieren, während längere Pfade statistisch entweder zu selten auftraten oder aus langen Ketten gleicher Aktionen bestanden. Es ist möglich, dass dies ein Artefakt des Systems ist, allerdings sind Dreiergruppen von Suchaktivitäten und Suchtaktiken schon früher als geeignete Einheit zur Analyse von Suchverhalten erkannt worden [QIU 1993, KESTRA et al. 1994, JANSEN und MCNEESE 2005].

Für die Umsetzung wurden die Logereignisse aus der Ebene der Dienste benutzt (siehe Abschnitt 5.4), da die Ereignisse auf der konzeptionellen Ebene zu abstrakt sind, um vom Benutzer in einen zielführenden Vorschlag umgesetzt werden zu können. Die vorhandenen Ereignisse wurden bereinigt, um Systemereignisse auszufiltern und nur Nutzeraktivitäten zu berücksichtigen. Außerdem wurden Testsitzungen entfernt. Die verbleibenden Sitzungen und Ereignisse wurden dann aufbereitet.

### 6.2.1. Vorgehensweise

Jede Aktivitätsart wird mit einem eindeutigen alphanumerischen Zeichen kodiert. Während die meisten Aktivitäten eins-zu-eins mit einem bestimmten DAFFODIL-Werkzeug korrespondieren, muss bei einigen Werkzeugen eine zusätzliche Aufteilung unternommen werden, um die unterschiedlichen Möglichkeiten, die das Werkzeug bietet, abzudecken. Eine komplette Sitzung kann dann als eine Zeichenkette kodiert werden, die der Abfolge der einzelnen Ereignisse entspricht (ein Beispiel ist in Tabelle 6.4 zu sehen).

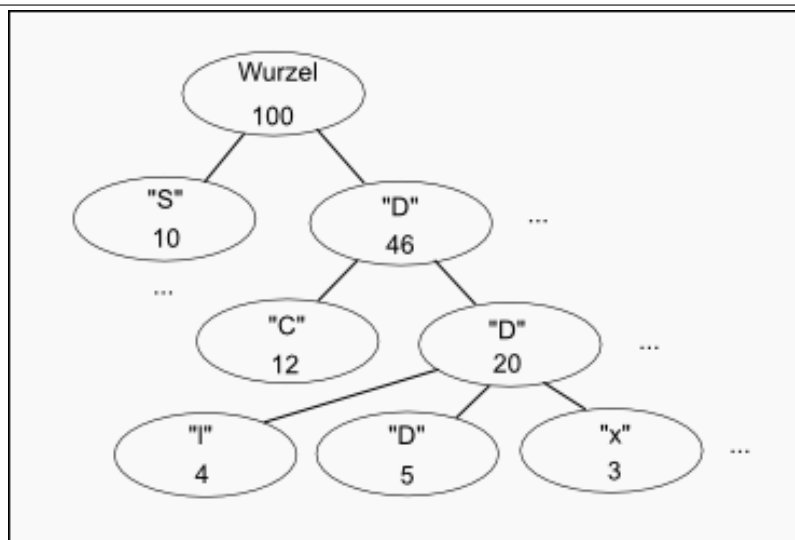
Die Stringrepräsentationen der Sitzungen werden nun in einen PAT-Tree eingefügt. PAT-Trees wurden erstmals von Gonnet, Baeza-Yates und Snider beschrieben [GONNET et al. 1992] und basieren auf Patricia-Tries. Es handelt sich bei ihnen um eine Index-Datenstruktur für die Stringsuche in Texten, wobei die Texte in sogenannte siStrings (*semi-infinite strings*) zerlegt und diese in einen Patricia-Trie eingefügt werden. Die siStrings entstehen, indem jede Textposition der Ausgangszeichenkette als Startposition eines siStrings betrachtet wird. Die Zeichenkette SDCDC hätte also die siStrings SDCDC, DCDC, CDC, DC und C. Diese Datenstruktur ist besonders effizient bei der Suche nach Präfixen oder längsten Wiederholungen in einem Text.

Dabei werden in dieser Anwendung alle Texte separat in siStrings zerlegt und dann einzeln in den PAT-Tree eingefügt. Als Erweiterung bekommt jeder Knoten (interne Knoten ebenso wie Blätter) ein Gewicht, durch das festgehalten

**Tabelle 6.4.** Beispielhafter Suchpfad: SDCDCx1SDPat.

S	Metadaten-Suche nach „Java AND language AND teaching“
D	Detailansicht von „The Case for Java as a first language“
C	Speichern von „The Case for Java as a first language“ in Zwischenablage
D	Detailansicht von „Experiences from Teaching Software Development in Java“
C	Speichern von „Experiences from Teaching Software Development in Java“ in Zwischenablage
x	Extrahieren häufiger Terme aus Zwischenablage
l	Hinzufügen von „education“ zur Anfrage über Link in Termwolke
S	Metadaten-Suche nach „Java AND (teaching OR education)“
D	Detailansicht von „Practice and Experience with Java in Education“
P	Speichern von „Practice and Experience with Java in Education“ in Handbibliothek
a	Extrahieren von Autoren aus Ergebnis
t	Extrahieren von Termen aus Ergebnis

**Abbildung 6.4.** Ausschnitt eines PAT-Trees mit Gewichten.



wird, in wievielen siStrings (entweder aus der gleichen Sitzung oder aus verschiedenen) die durch den Knoten repräsentierte Zeichenkette vorkommt. In Abbildung 6.4 beginnen von insgesamt 100 eingefügten siStrings 10 mit S und 46 mit D. Von letzteren beginnen 12 mit C und 20 mit D, usw.

Auf diese Art lässt sich sehr einfach zu einem beliebigen Suchpfad  $P$  der häufigste nächste Schritt finden. Zunächst wird der Suchpfad  $P$  in seine Zeichen-darstellung überführt. Anschließend wird diese Zeichenkette mit dem Algorithmus für Präfix-Suche im PAT-Tree gesucht. Der Teilbaum, der den letzten Knoten der gefundenen Zeichenkette als Wurzel  $W$  (mit Gewicht  $w$ ) besitzt, enthält nun alle Suchpfadteile, die  $P$  als Präfix haben. Die direkten Kindknoten dieses neuen Wurzelknotens (mit Gewichten  $k_1 \dots k_n$ ) sind dann mögliche Vorschläge für einen nächsten Schritt. Dabei lassen sich die Vorschlagsgewichte als  $k_i/w$  berechnen.

Auch ein längerer Teilpfade lässt sich so vorschlagen, indem von  $W$  aus jeweils der Kindknoten mit dem höchsten Gewicht gewählt wird.

### 6.2.2. Umsetzung für DAFFODIL

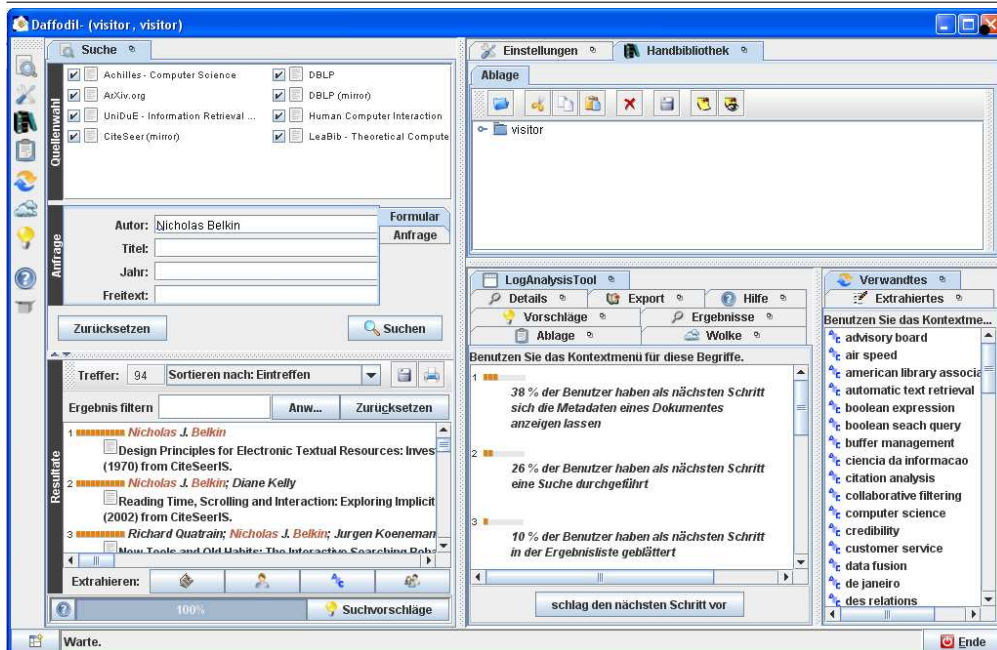
Das Vorschlagssystem wurde für DAFFODIL in Form eines neuen Agenten, sowie eines Werkzeugs in der Benutzerschnittstelle (siehe Abbildung 6.5) umgesetzt. Vorschläge werden nur auf Anforderung des Benutzers ermittelt. Dazu wird eine Mitteilung an den Agenten mit der ID der aktuellen Suchsitzung verschickt. Anders als beim in Abschnitt 6.1 vorgestellten System kann dieses Vorschlagssystem an jedem Punkt einer Suchsitzung Vorschläge generieren, nicht nur nach einer erfolgten Suche.

Der Agent hält die Datenstruktur aller indexierten Logsitzungen vor. Da das Einfügen einer neuen Sitzung mit  $n$  Ereignissen in  $O(n)$  erfolgt, kann die Datenstruktur nach jeder abgeschlossenen Sitzung aktualisiert werden. Wenn der Agent eine neue Anfrage nach Vorschlägen erhält, werden die Sitzungsdaten zur Sitzungs-ID aus der Log-Datenbank ausgelesen und die Sitzung als eine Zeichenkette kodiert. Diese Zeichenkette wird dann im PAT-Tree gesucht. Kann die gesamte Sitzung nicht gefunden werden, wird die Suche auf die letzten drei Schritte der aktuellen Sitzung eingeschränkt.

Die Kinderknoten des letzten Knotens der gefundenen Zeichenkette werden dann mitsamt den wie oben ermittelten Gewichten an die Benutzerschnittstelle geschickt. Diese bereitet die Informationen auf und präsentiert sie dem Benutzer auf ähnliche Weise wie die Vorschläge in Abschnitt 6.1. Auch hier werden die Gewichte der Vorschläge durch farbige Balken in unterschiedlicher Länge repräsentiert.

Da die auf diese Weise ermittelten Vorschläge für einen nächsten Suchschritt deutlich abstrakter sind als die taktischen und strategischen Vorschläge, wurde auf die Möglichkeit einer automatischen Ausführung verzichtet.

Abbildung 6.5. Vorschlagswerkzeug basierend auf Suchpfaden [BODNARIU 2009].



Die Evaluation zeigte gemischte Ergebnisse [BODNARIU 2009]. Dies ist auf zwei Hauptprobleme zurückzuführen:

1. Die Umsetzung der Vorschläge erfordert erheblich höheren kognitiven Aufwand durch den Benutzer. Die vorgeschlagenen nächsten Schritte sind abstrakt und müssen durch den Suchenden zunächst auf seine konkrete Situation übertragen werden.
2. Die Qualität der zugrundeliegenden Suchsitzungen war mangelhaft. Auch nach Entfernung von Testsitzungen überwogen Evaluations-sitzungen, die unter festen Rahmenbedingungen durchgeführt wurden, und bei denen die Teilnehmer selten Zugriff auf den vollen Werkzeug-Umfang von DAFFODIL hatten, gegenüber „echten“ Suchsitzungen. Dadurch überwogen stark bestimmte Aktivitätsarten und Muster, was zu einseitigen Vorschlägen von eingeschränkter Nützlichkeit führte.

Um dem ersten Problem zu begegnen, wäre es notwendig, konkrete auf die Situation bezogene Vorschläge aus der Nutzeraktivität abzuleiten. Hier könnte sich eine Kombination der beiden Verfahren anbieten. Das zweite Problem lässt sich nur durch eine größere Menge von Logdaten aus realistischen Sitzungen beheben.

### 6.2.3. Nutzung von Ergebnislogs in situationsbasierten Vorschlägen

Die beiden Ansätze zur Gewinnung von strategischen Vorschlägen lassen sich auch kombinieren, wobei allerdings im Rahmen dieser Arbeit keine Umsetzung erfolgte.

Die einfachste Kombination wäre eine Erweiterung der Parameter, mit denen eine Situation im CBR-Ansatz beschrieben wird. Dazu könnte ein zusätzlicher Parameter für den Suchpfad des Benutzers aufgenommen werden, beschränkt z. B. auf die letzten 3-5 Aktivitäten. Nimmt man weitere Parameter hinzu, die nicht von einer zuvor gestellten Suchanfrage abhängen (wie in Abschnitt 6.1 vorgeschlagen), so wäre es auch mit der Methode des fallbasierten Schließens möglich, Vorschläge für beliebige Situationen zu generieren.

Um die Ähnlichkeit zweier Suchpfade  $p_1$  und  $p_2$  zu bestimmen, könnte man ein auf N-Grammen basierendes Ähnlichkeitsmaß benutzen, etwa den Dice-Koeffizienten

$$sim_{dice}(p_1, p_2) := \frac{2|ng(p_1) \cap ng(p_2)|}{|p_1| + |p_2|}, \quad (6.6)$$

wobei  $ng(p_i)$  eine Zerlegung des Suchpfades  $p_i$  in N-Gramme ist. Dabei sollte allerdings auch die Position der N-Gramme für die Berechnung der Ähnlichkeit berücksichtigt werden. Eine Möglichkeit dazu ist, den Beitrag jedes gemeinsamen N-Gramms  $n$  mit dem Faktor

$$\frac{2}{1 + (pos(n, p_1) - pos(n, p_2))^2} \quad (6.7)$$

zu gewichten, wobei  $pos(n, p_i)$  die Position des N-Gramms  $n$  im Suchpfad  $p_i$  liefert [BREW und MCKELVIE 1996].

Eine andere Möglichkeit der Kombination würde erst einen nächsten Schritt anhand der bisher benutzten Suchpfade generieren und anschließend einen taktischen oder strategischen Ratschlag anhand der umfassenden Situation durch fallbasiertes Schließen ermitteln. Dabei kann der mit Hilfe des Suchpfad induzierte Schritt entweder

- als Filter dienen, um nur solche Vorschläge anzuzeigen, die konkrete Ausprägungen sind,
- als nachträglicher Gewichtungsfaktor dienen, um die Rangfolge der Vorschläge zu ändern, oder
- nur ein weiterer Parameter sein.

Die Kombination würde den Nachteil der pfadbasierten Methode überwinden, keine direkt anwendbaren Vorschläge zu produzieren, und würde konkrete, strategische Hilfestellung geben. Die Situation des Benutzers würde dabei stärker die gesamte Suchsitzung umfassen. Zusätzlich ließen sich so anders

als bei der einfachen CBR-Methode auch Lösungen für Situationen abseits der einfachen Suche finden.

### **6.3. Zusammenfassung**

In diesem Kapitel wurden mehrere Methoden vorgestellt, um Vorschläge zu generieren, mit denen Benutzer bei der strategischen Suche unterstützt werden sollen. Ein auf der Technik des fallbasierten Schließens basierendes System benutzt Taktiken und Strategeme aus der Literatur. Dadurch dass dem Suchenden aufgezeigt wird, welche strategischen Möglichkeiten er bei der Benutzung des Systems besitzt, und welche in einer bestimmten Situation besonders geeignet sind, wird eine Unterstützung bei der Bildung strategischen und prozeduralen Wissens erwartet. Die Evaluationen in Kapitel 8 werden zeigen, dass Benutzer mit Hilfe des Vorschlagssystems erfolgreicher suchen, mehr und fortgeschrittenere Möglichkeiten des Systems benutzen, und dieses Wissen dann auch in nicht unterstützten Suchen anwenden.

Das System erfordert dabei Bewertungen der Vorschläge zur Anreicherung der Fallbasis. Da es nicht immer leicht ist, Benutzer zur Abgabe von Feedback zu bewegen, insbesondere nachdem sie bereits den Nutzen durch den Vorschlag erfahren haben, ist hier darüber nachzudenken, welche Möglichkeiten des impliziten Feedbacks es gäbe, um positive oder negative Vorschlagsbewertungen daraus abzuleiten.

Die zweite vorgestellte Methode umgeht dieses Problem, indem sie die gesammelten Logdaten zur Generierung von Vorschlägen nutzt. Aus den Sitzungen früherer Benutzer werden die häufigsten Suchpfade ermittelt und diese Suchenden vorgeschlagen, die auf ähnlichen Wegen zu gehen scheinen. So können neue Benutzer in den „Fußstapfen“ anderer gehen und dabei Strategien erlernen.





## Kapitel 7.

# Unterstützende Nutzerführung

Während im vorangegangenen Kapitel das Hauptaugenmerk darauf lag, den Endbenutzer eines Suchsystems beim Finden geeigneter Suchstrategien und strategischer Suchoptionen zu unterstützen, soll nun die Umsetzung zweier Ansätze zur Unterstützung bei der Durchführung von Suchstrategien betrachtet werden. Mit Hilfe des in Abschnitt 5.3 vorgestellten Strategieframeworks für DAFFODIL können einfache, kurze Suchstrategeme automatisiert ausgeführt werden. Aufgabenabhängige Nutzerführung durch Software-Scaffolding soll insbesondere Suchnovizen helfen, ein Suchsystem für ausgewählte Aufgaben effektiv zu einzusetzen.

### 7.1. Automatische Ausführung einfacher Suchstrategeme

In [BATES 1990] diskutiert Marcia Bates verschiedene Ebenen der Suchunterstützung oder Systembeteiligung (*system involvement*). Auf der niedrigsten Stufe (Ebene 0) werden alle Suchaktivitäten vollständig vom Benutzer initiiert, geplant und umgesetzt. Auf Ebene 1 kann ein System verfügbare oder empfehlenswerte Suchaktionen auf Anfrage präsentieren und erklären, oder den Benutzer in der Benutzung des Systems oder Suche instruieren. Die Ebene 2 erlaubt die Durchführung einzelner Suchaktionen durch das System auf direkte Veranlassung des Benutzers. DAFFODIL stellt zahlreiche Werkzeuge zur Verfügung, über die ein Benutzer Suchstrategeme oder Taktiken durch das System ausführen lassen kann.

Die im vorangegangenen Kapitel beschriebenen Unterstützungsformen liegen nach Bates auf der Ebene 3, in der ein System die Suche des Benutzers überwacht und analysiert, um dann entweder auf Benutzeranfrage oder auf Initiative des System Vorschläge zu machen. Bates identifizierte jedoch auch eine höhere Ebene (4), bei der ein Suchsystem einzelne Aktionen automatisch ausführen kann. Diese Klassifikation kann zusätzlich mit der bereits in Abschnitt 2.3 vorgestellten Unterteilung von Suchaktivitäten kombiniert werden.

Aufbauend auf dem Strategieframework wurden für das Suchsystem DAFFODIL dabei insgesamt vier einfache Suchstrategien oder Suchstrategeme für

## 7.1. AUTOMATISCHE AUSFÜHRUNG EINFACHER SUCHSTRATEGIEN

---

die automatische Ausführung umgesetzt [FRANKMÖLLE 2004, KRIEWEL et al. 2005, ZHENG 2007]. Diese erlauben unerfahrenen Benutzern nach anfänglicher Definition ihres Suchbedürfnisses, die eigentliche Durchführung einer Suchstrategie dem System zu überlassen. Da der Suchende sein Suchbedürfnis oft nur vage beschreiben kann, ist diese Umsetzung nur für bestimmte Arten von Suchstrategien geeignet, bei denen während der Ausführung keine zusätzlichen Informationen durch den Benutzer bereitgestellt werden müssen.

In Anlehnung an die in Abschnitt 2.2 vorgestellten Klassifikationen von Informationsaufgaben nach ihrer Komplexität [BYSTRÖM und JÄRVELIN 1995] und dem Wissen von Suchenden [BHAVNANI et al. 2001b], unterteilt Frankmölle [FRANKMÖLLE 2004] Suchaufgaben dazu mit Hilfe einer zweidimensionalen Klassifikation. Die Dimensionen der Klassifikation sind die Komplexität der Suchaufgabe und das Problem- und Domänenwissen des Benutzers. Die Komplexität ist unterteilt in „einzelnes Dokument“, „Liste von Dokumenten“, „spezielles Thema“ und „Themenkomplex“. Das Problemwissen reicht von vollständiger Bekanntheit der zur Problemlösung notwendigen Information bis hin zum Mangel jeglicher zur Suche verwertbarer Information. Für Probleme, bei denen der Benutzer keine oder fast keine Information besitzt, die er für eine Suche verwerten kann, sind Browsingansätze geeignet, die durch das Strategieframework nicht unterstützt werden können.

Für vier andere Problemstellungen aus der Matrix, die sich durch Kombination aus Komplexität und Problemwissen ergibt, wurden jedoch konkrete Strategien für DAFFODIL in Form von Strategiemodulen umgesetzt und funktionell evaluiert:

- *Known-Item-Instantiation* für ein einzelnes bekanntes Informationsobjekt,
- Suche nach Information zu einer vagen Fragestellung, die durch ein einzelnes Informationsobjekt beantwortet werden kann,
- Suche nach einer Liste von Dokumenten mit gemeinsamen Eigenschaften,
- Ergänzung einer vorhandenen Dokumentliste um ähnliche Dokumente.

### 7.1.1. Suche nach einem bekannten Objekt (1-1)

Eine der einfachsten Suchaufgaben ist die Suche nach einem einzelnen Informationsobjekt, zu welchem dem Suchenden eine Reihe identifizierender Informationen bekannt sind. So könnten dem Benutzer etwa der Titel (oder wesentliche Teile des Titels), ein oder mehrere Autoren und das Erscheinungsjahr eines Dokumentes bekannt sein.

Der Benutzer spezifiziert für dieses Strategiemodul alle Metadaten, die ihm zum Dokument bekannt sind. Über ein Makro des Strategieframeworks zur

Quellenbestimmung wird gegebenenfalls anhand der bekannten Metadaten bereits eine Vorauswahl der Quellen getroffen, in denen gesucht werden soll, um die Bearbeitungszeit für die Suche gering zu halten.

Nun wird zunächst eine Suche unter Ausnutzung aller Angaben gestartet (die Taktik EXHAUST). Findet diese Suche bereits das Dokument und ist das Ergebnis eindeutig, so endet das Strategiemodul mit der Ausgabe des Ergebnisses. Werden zu viele Ergebnisse gefunden, so schränkt das Modul je nach Konfiguration die Ausgabe auf das Dokument mit dem höchsten Retrievalgewicht ein oder liefert die komplette Liste mit einer Fehlermeldung zurück, die ausdrückt, dass die Angaben zu keinem eindeutigen Ergebnis führten.

Wird das gesuchte Dokument jedoch unter Verwendung der vollständigen Angaben nicht gefunden, so wird die Suche gegebenenfalls zunächst unter Einbeziehung der zuvor ausgelassenen Quellen wiederholt. Bringen auch die neu hinzugenommenen Quellen nicht das gewünschte Ergebnis, so bedient sich das Strategiemodul der vagen Suche nach einer einzelnen Information, wie sie im folgenden Abschnitt beschrieben wird. Einzelne Teile der von Benutzer vorgegebenen Spezifikationen werden dafür unter Ausnutzung der von Bates beschriebenen Termtaktiken und Suchtaktiken gelockert, und es werden parallel mehrere vage Suchen gestartet. Die Ergebnisse werden zusammengeführt und das am besten zur ursprünglichen Spezifikation des Benutzers passende Dokument wird zurückgeliefert (wie schon oben erläutert können je nach Konfiguration auch die weiteren Dokumente zurückgeliefert werden).

### 7.1.2. Suche nach einer einzelnen Information (1-2)

Kann der Benutzer das gewünschte Informationsobjekt nur vage beschreiben, entweder weil ihm wichtiges Wissen über das Objekt fehlt oder weil es die Antwort auf eine definierte Frage des Benutzers enthält, das Informationsobjekt selbst aber dem Suchenden nicht bekannt ist, so kommt ein zweites Strategiemodul zum Einsatz. Wie das Modul 1-1 zur *Known-Item-Instantiation* liefert es im Erfolgsfall genau ein Informationsobjekt zurück. Auf das Strategiemodul wird auch automatisch zurückgegriffen, wenn unter Benutzung der vollständigen Spezifikation durch den Benutzer bei der Suche nach einem *known item* kein Ergebnis gefunden wurde.

Auch dieses Strategiemodul benutzt zunächst alle ihm übergebenen Informationen, um damit ein Dokument zu finden. Gelingt dies nicht, wird ein Makro zur Manipulation der Frageterme angestoßen, welches auf bekannten Taktiken zur Termerweiterung und Termauswahl basiert [FRANKMÖLLE 2004]. Hierzu werden parallel verschiedene Anfragen unter Einsatz unterschiedlicher Taktiken unternommen und die Ergebnisse zusammengeführt. Unter den Treffern der Anfragen wird das Ergebnis mit der höchsten Ähnlichkeit zur ursprünglichen Anfragespezifikation als Ausgabe gewählt.

### 7.1.3. Suche nach einer Liste von Dokumenten mit definierten Eigenschaften (2-1)

Bei der Suche nach einer Liste von bekannten Dokumenten berücksichtigt das Strategieframework zwei Fälle, wobei in beiden Fällen eine Anzahl  $n$  zu findender Dokumente durch den Benutzer vorzugeben ist. Im ersten Fall stellt sich die Suche als eine Reihe von *Known-Item-Instantiations* dar, wenn dem Suchenden eine Liste von ansonsten nicht notwendigerweise verwandten Dokumenten bekannt ist, die er finden möchte (z. B. eine Literaturliste). Hier ist die Zahl  $n$  bereits durch die einzelnen beschriebenen Dokumente gegeben. Der Ablauf des Moduls erstreckt sich dann auf den mehrfachen, parallelen Aufruf des Moduls 1-1.

Im zweiten Fall kann der Benutzer daran interessiert sein, eine Reihe von Informationsobjekten mit gemeinsamen Eigenschaften zu finden, z. B. Dokumente, die ein bestimmter Autor ohne Koautoren verfasst hat, oder Dokumente aus dem aktuellen Jahr, die sich mit einem vorgegebenen Thema beschäftigen – etwa im Rahmen einer Suchaufgabe, die *Monitoring* erfordert (siehe Abschnitt 2.2). Hierbei wird auf das Modul 1-2 zurückgegriffen, doch statt sich auf ein Dokument als Rückgabe zu beschränken, werden entweder alle auf die Spezifikation des Benutzers passenden Dokumente oder die  $n$  Dokumente mit dem höchsten Retrievalgewicht zurückgeliefert.

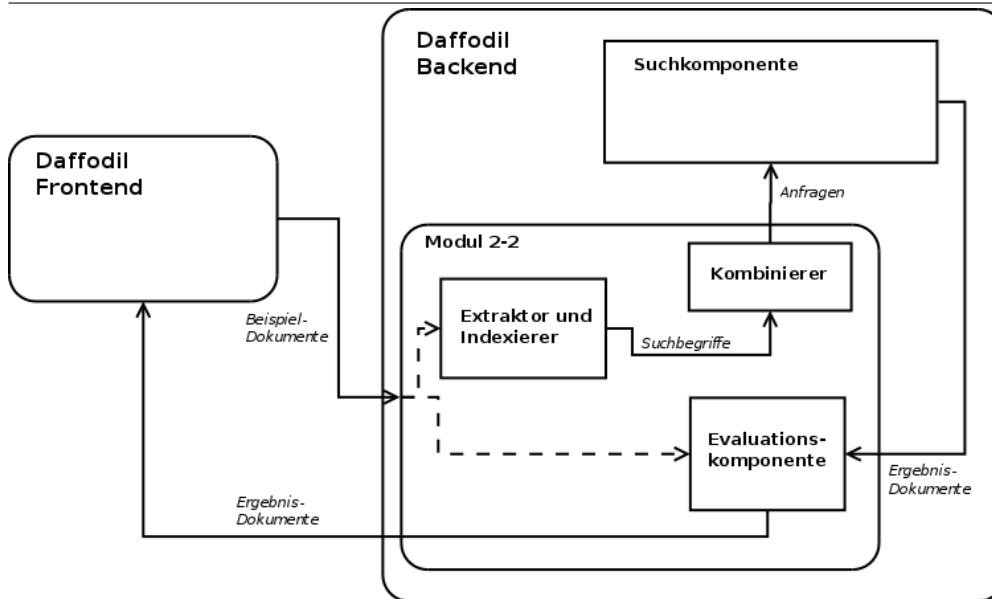
### 7.1.4. Suche nach einer Liste ähnlicher Dokumente (2-2)

Schließlich wurde im Rahmen des Strategieframeworks auch eine Strategie umgesetzt, bei der das Suchproblem durch den Benutzer nicht in Form von Suchbegriffen spezifiziert wird, sondern durch Vorgabe einer Liste von Informationsobjekten. Zu dieser Liste findet das Strategiemodul 2-2 ähnliche Dokumente [ZHENG 2007]. Die Strategie ist für zwei identifizierte Anwendungsfälle vorgesehen:

- Eine Suche des Benutzers lieferte zwar relevante Ergebnisse, aber nicht in ausreichender Zahl, um das Informationsbedürfnis zu befriedigen. Der Suchende ruft das Strategiemodul auf, um seine Resultatmenge um zusätzliche, ähnliche Ergebnisse zu ergänzen.
- Ein Benutzer hat z. B. in einer Reihe von früheren Suchen zu einem Thema Dokumente, Autoren, Begriffe oder andere Informationsobjekte gesammelt und diese in DAFFODIL entweder in seiner persönlichen Handbibliothek oder in der Zwischenablage abgelegt. Nun möchte er diese Objekte als Ausgangspunkt für eine Suche nach weiteren Dokumenten nutzen.

Eine solche Suche kann als Dokumentsuche anhand von Beispielen betrachtet werden („*query by example*“). Anders als sonst meist üblich (z. B. in Cite-

Abbildung 7.1. Schematische Darstellung von Modul 2-2 nach [ZHENG 2007].



Seer [BOLLACKER et al. 1998]), werden hier als Beispiele nicht einzelne Objekte, sondern Listen von Objekten übergeben. Abbildung 7.1 zeigt eine schematische Darstellung des Modulaufbaus. Dabei müssen zunächst Merkmale der Beispieldokumente extrahiert werden, die diese gegenüber anderen Dokumenten abgrenzen. Anschließend muss eine geeignete Form der Anfrage-Transformation gefunden werden, die dem Einsatz der Ähnlichkeitssuche für verteilte, heterogene Digitale Bibliotheken Rechnung trägt.

Dazu findet eine Extraktion der Terme statt, die in den Stringrepräsentationen der Beispielobjekte vorkommen, um dann aus diesen einen invertierten Index aufzubauen. Aus den Termen werden jene mit den 5 höchsten *Discrimination Scores* bezüglich einer Referenzkollektion ausgewählt und zu Anfragen kombiniert. Ein *Discrimination Score* ist ein Maß dafür, wie gut sich ein Indexierungsbegriff basierend auf einer gegebenen Dokumentenkollektion zur Trennung unterschiedlicher Dokumente eignet. Zur Vorberechnung der *Discrimination Score* basierend auf einer repräsentativen Teilmenge der Kollektion wird hier der *IfD score (Information for Discrimination)* nach Cai [CAI 2004] verwendet.

Die ausgewählten Suchbegriffe werden vom Kombinerer paarweise zu Konjunktionen zusammengefasst und entweder durch Disjunktion zu einer Anfrage verknüpft oder in Form paralleler Anfragen verschickt, wobei in letzterem Fall die Ergebnisse der einzelnen Anfragen vor der Weiterverarbeitung zusammengeführt werden. Die paarweise Kombination von Suchbegriffen erwies sich für den Einsatz in diesem Strategiemodul in der Evaluation anderen Kombinationsmethoden als überlegen [ZHENG 2007].

## 7.2. NUTZERFÜHRUNG DURCH SCAFFOLDING

---

*Beispiel:* Nach der Extraktion der Terme aus den Beispieldokumenten werden als die fünf Terme mit dem besten *Discrimination Score* die Suchbegriffe „daffodil“, „strategem“, „strategy“, „library“ und „support“ ausgewählt. Diese Begriffe werden nun paarweise zu Anfragen zusammengeführt: z. B. „daffodil AND strategem“, „strategy AND support“ oder „library AND support“. Schließlich werden die Teilanfragen mit OR kombiniert oder als parallele Anfragen ausgeführt.

Um aus den so erzielten Ergebnissen die besten Ergebnisse auszuwählen, wird von der Evaluationskomponente des Strategiemoduls die Ähnlichkeit der einzelnen Dokumente zu den vorgegebenen Beispielobjekten berechnet. Dabei werden mehrere Ähnlichkeitsmaße verwendet (Jaccard-Koeffizient, Kosinusmaß) und eine gewichtete Summe der normalisierten Werte gebildet.

Alle Dokumente mit einem Ähnlichkeitswert über einem konfigurierbaren Schwellenwert oder die Dokumente mit den  $n$  höchsten Ähnlichkeitswerten werden vom Strategiemodul als Dokumentenliste zurückgeliefert.

Evaluationen des Strategiemoduls 2-2 zeigten gemischte Ergebnisse [ZHENG 2007]. In einer Benutzerevaluation zeigten sich alle Benutzer zumindest teilweise zufrieden mit den gefundenen Dokumenten und in 78,5% der durchgeführten Suchen lag der Anteil relevanter Dokumente in der Liste ähnlicher Dokumente bei 50% oder besser.

### **7.2. Aufgabenabhängige Nutzerführung durch Scaffolding**

Während die Suchunterstützung durch automatische Ausführung von Suchstrategien darauf abzielt, dem Endbenutzer für einfache, nicht interaktive Strategien die eigentlichen Schritte der Suche abzunehmen, verfolgt der Scaffolding-Ansatz ein anderes Ziel. Beim software-realisierten Scaffolding sollen die Benutzer des Systems die effektive Nutzung erlernen, indem sie mit Hilfe sogenannter Scaffolds schrittweise durch eine Aufgabe geführt werden. Diese Methode richtet sich insbesondere an mit dem Suchsystem oder der Informationssuche noch völlig unerfahrene Benutzer. Durch den Einsatz von unterstützenden „Gerüsten“ sollen sie in die Lage versetzt werden, den Umgang mit dem System und einfache Suchstrategien während ihrer Arbeit mit dem System zu erlernen. So reduziert sich die Notwendigkeit passiven Wissens, das sich der Nutzer durch Tutorien, Handbücher oder Instruktionen aneignen muss.

Für eine vorgegebene Suchaufgabe wird hierzu eine beispielhafte Strategie in einzelne Teilschritte oder Teilziele zerlegt. Dabei sollte die Aufgabe innerhalb der *Zone of Proximal Development* des Benutzers oder Lernenden liegen, d. h. es sollte sich um eine Aufgabe handeln, die der Lernende zwar nicht alleine lösen

kann, zu deren Lösung ihn aber das Scaffolding befähigt. Dies könnte eine relativ einfache Suchaufgabe sein, wenn der Lernende mit den Werkzeugen des Systems völlig unvertraut ist und er sich zunächst das Systemwissen aneignen muss – oder eine komplexere Aufgabe, wenn ein Grundwissen über die Möglichkeiten des Systems und den Einsatz seiner Werkzeuge vorliegt und einfache Suchaufgaben bereits Teil des *Actual Development Levels* des Benutzers sind.

Ein ähnliches Konzept der Zerlegung der Suchaufgabe in Teilziele wird auch bei der Suchunterstützung durch die in Abschnitt 3.2.2 vorgestellten *Strategy Hubs* verwendet, um Endbenutzern prozedurales Suchwissen zu vermitteln. Im Rahmen des software-realisierten Scaffoldings soll der Benutzer nun vom System durch die Aufgabe geleitet werden, wobei weitere Scaffolds oder Unterstützungsmaßnahmen ihm helfen, sein Suchziel zu erreichen. Diese Scaffolds sollen den Erfolg des Lernenden ermöglichen und zugleich den Lernprozess unterstützen, indem sie prozedurales Wissen vermitteln.

Wichtig ist dabei, dass anders als etwa bei Software-Wizards [DRYER 1997] die Unterstützung durch Scaffolds im Rahmen des *Fading*-Prozesses nach und nach abgebaut wird.

### 7.2.1. Aufgabe

Die Umsetzung des Scaffolding-Konzepts zur Unterstützung bei der Durchführung von Suchstrategien für unerfahrene Nutzer erfolgte im Rahmen von DAFFODIL. Dazu wurde als Aufgabe die Suche nach Filmen zur Gestaltung eines thematischen Filmabends gewählt. Die Benutzer mussten hierzu eine inhaltsorientierte Suche durchführen, wobei die Lösung der Suchaufgabe mehrere Ergebnisse erforderte. Diese Ergebnisse waren zudem in Form von Zeitplänen zu organisieren.

Reiser [REISER 2002] legt dar, dass Software-Scaffolding insbesondere durch eine stärkere Strukturierung von Aufgaben unterstützen kann. Die Vorgehensweise bei der Suche zur gewählten Aufgabe wurde daher in eine Reihe von Schritten zerlegt. Dabei orientieren sich die Schritte an den drei Subprozessen in Marchioninis Modell der Informationssuche [MARCHIONINI 1995]: Definieren und Verstehen des Problems; Planen und Ausführen der Suche; Bewertung und Verwendung (die Bestandteile der drei Subprozesse wurden bereits in Kapitel 2 behandelt).

Im einzelnen besteht die Suchstrategie für die gewählte Aufgabe aus den folgenden fünf Schritten:

1. Beschreibung der Suchaufgabe durch den Benutzer,
2. Auswahl von Suchbegriffen,
3. Durchführung der Suche,

4. Organisation der Ergebnisse,
5. Aufbereitung der Ergebnisse.

Dabei handelt es sich nicht um eine rein lineare Strategie, sondern wie von Marchionini beschrieben muss dem Benutzer die Möglichkeit gegeben werden, aus jedem Teilprozess zu früheren Teilprozessen zurückzuspringen, um dort Änderungen durchzuführen. Dies wurde bei der Umsetzung berücksichtigt.

### 7.2.2. Umsetzung des Scaffolding-Konzepts

Zunächst sollen die zur Verfügung gestellten Scaffolds beschrieben werden. Wichtigste Gerüstbestandteile sind die schrittweise Führung durch die Suchaufgabe, sowie der Suchassistent, der in jedem Suchschritt Erklärungen anbietet. Assistent und Navigation sind in Abbildung 7.2 (a) zu sehen und sind während der gesamten Suchaufgabe am Rand des Programmes sichtbar. Die Navigation erlaubt dabei auch das Zurückspringen in frühere Teilschritte der Strategie, aber nicht das Überspringen von noch nicht bearbeiteten Schritten.

Komlodi und Soergel [KOMLODI und SOERGEL 2002] identifizieren drei Formate, in denen erfahrene Informationssuchende ihre Wissenstrukturen für die Zwecke einer Suchaufgabe organisieren:

- ein Umriss (oder eine Inhaltsübersicht oder Gliederung) der zu erledigenden Aufgabe, beispielsweise eines zu verfassenden Dokumentes, der die Themen und Fragestellungen zusammenfasst,
- Planungsnotizen für die Suche, die vor dem Beginn der Suche erstellt werden und beispielsweise eine Sammlung von Suchbegriffen und Synonymen umfassen,
- eine Organisationsstruktur für die Ablage der Suchergebnisse.

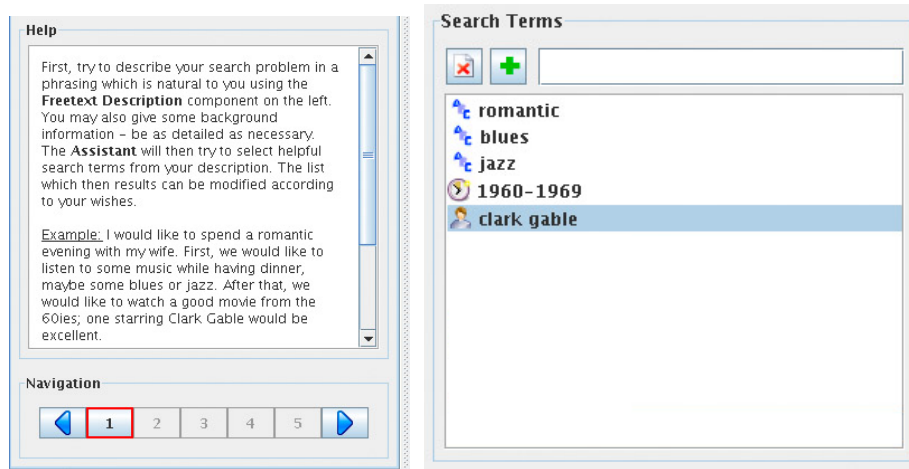
Der Scaffolding-Ansatz sollte diese drei Organisationsformate anbieten und den Suchnovizen dabei unterstützen, sich das Wissen zur Nutzung solcher organisatorischer Stützen anzueignen. Dazu wurden entsprechende Scaffolds und Werkzeuge bereitgestellt.

#### **Beschreibung der Aufgabe und Auswahl von Begriffen**

Im ersten Schritt der Strategie wird der Benutzer durch den Assistenten aufgefordert, seine Suchproblem zunächst in natürlicher Sprache zu beschreiben. Dabei werden Beispiele gegeben, wie das Suchsystem für die Zusammenstellung eines thematischen Filmabends genutzt werden kann. Diese Beschreibung der Suchaufgabe dient während der Arbeit mit dem System als Orientierung und soll helfen, die Fragestellung nicht aus dem Auge zu verlieren.



Abbildung 7.2. Bestandteile der Scaffolding-Erweiterung für DAFFODIL.



(a) Assistent.

(b) Planungswerkzeug.

Zwar erlaubt die gewählte Umsetzung keine strukturierte Beschreibung oder Gliederung, doch in der Intention entspricht sie dem Aufgabenumriss nach Komlodi und Soergel.

Als zusätzliche Werkzeuge neben der Eingabemaske werden eine domänen-spezifische Klassifikation mit Filmgenres und ein Werkzeug zur Anzeige verwandter Begriffe zur Verfügung gestellt. Die verwandten Begriffe werden in einer Termwolke dargestellt und aufgerufen, wenn Worte aus der Beschreibung des Benutzers oder der Klassifikation in das Werkzeug gezogen werden.

Beim Übergang zum zweiten Schritt extrahiert das System automatisch aus der Freitextbeschreibung geeignete Suchbegriffe (siehe Abbildung 7.2 (b)) und präsentiert sie dem Benutzer in einer Liste. Dort wo Suchbegriffe als besonders geeignet für bestimmte Suchfelder (Personen oder Jahresangaben) erkannt wurden, werden diese entsprechend gekennzeichnet. Dabei erklärt der Assistent, warum bestimmte Suchbegriffe geeignet, bzw. nicht geeignet sind, und wie der Benutzer mit Hilfe der verfügbaren Werkzeuge Synonyme und alternative Begriffe für seine Suche finden kann. Diese Begriffe können per Drag-n-Drop in die Liste mit Begriffen übernommen werden. Die Liste der gesammelten Begriffe lässt sich vom Benutzer umsortieren und es lassen sich existierende Begriffe entfernen. Sie bleibt auch im nächsten Schritt der Strategie erhalten und stellt eine Möglichkeit dar, Planungsnotizen für die Suche anzulegen.

### Durchführung der Suche

Wenn der Benutzer sich mit Hilfe des Planungswerkzeugs genügend Begriffe zurecht gelegt hat, kann er im nächsten Schritt Suchanfragen mit Hilfe des

Suchwerkzeugs von DAFFODIL durchführen. Das Planungswerkzeug unterstützt die Arbeit des Suchenden, indem die gewählten Begriffe automatisch den Feldern des Suchformulars zugeordnet und markiert werden. Die Begriffe können für eine Suche per Drag-n-Drop in die entsprechenden Felder übernommen werden.

Zusätzlich unterstützt wird dieser Schritt durch Beispiele für Suchanfragen im Assistenten, sowie durch das in Kapitel 6 beschriebene strategische Vorschlagswerkzeug, dessen Vorschläge auf mit den aktuell verfügbaren DAFFODIL-Werkzeugen anwendbare eingeschränkt werden. Alle gefundenen Dokumente können durch Drag-n-Drop in die Zwischenablage übernommen werden, die hier als unstrukturiertes Sammelkörbchen im Sinne des *Berrypickings* dient. Neu gefundene Personen oder Suchbegriffe können ins Planungswerkzeug übernommen werden, wo sie für weitere Suchen bereitstehen.

Schließlich erlaubt die Zwischenablage auf Knopfdruck die Extraktion von neuen Begriffen aus den Metadaten der abgelegten Dokumente, sowie die Ergänzung der bereits gesammelten Dokumente um eine Liste ähnlicher Dokumente durch Ausführung der in Abschnitt 7.1.4 beschriebenen automatischen Suchstrategie.

### **Organisation und Präsentation der Ergebnisse**

In den letzten Schritten der durch Scaffolding umgesetzten Strategie wird dem Benutzer ermöglicht, über die persönliche Handbibliothek seine Ergebnisse zu organisieren. Im Rahmen der unterstützten Suchaufgabe konnten so etwa Alternativen in unterschiedlichen Ordnern geplant werden.

Aus einem so vorbereiteten Ordner kann dann über ein Exportwerkzeug zur Aufbereitung und Präsentation der Ergebnisse ein Ablauf- oder Zeitplan erstellt werden. Die Organisation und Aufbereitung der Ergebnisse wird wie bereits die vorangegangenen Schritte durch Beispiele und Erklärungen über den Assistenten unterstützt. Zusammen bieten Handbibliothek und Exportwerkzeug eine Organisationsstruktur für die Ablage der Suchergebnisse.

### **7.2.3. Implementierung**

Die Implementierung der Scaffolding-Erweiterung für DAFFODIL besteht aus drei wesentlichen Komponenten:

- dem Assistenten einschließlich der Navigation,
- Perspektiven mit Werkzeugzusammenstellungen für einzelne Arbeitsschritte,
- einer Beschreibung des Ablaufs mit Festlegung von Scaffolds.

Zunächst wurde DAFFODIL dazu um das Konzept verschiedener Unterstützungsniveaus erweitert. Abhängig vom gewählten Unterstützungsniveau sollen existierende Werkzeuge angepasste Hilfestellungen geben können (bis hin zum Fehlen jeder Hilfestellung auf dem niedrigsten Unterstützungsniveau für Experten im Umgang mit dem System). Beispielsweise werden auf dem höchsten Niveau umfangreichere Erklärungen durch das Suchwerkzeug angeboten.

### Assistent und Navigation

Der Assistent zusammen mit der Navigation wurden als internes DAFFODIL-Werkzeug mit flexibler innerer Komponente umgesetzt. Damit fügt es sich zum einen direkt in die Werkzeugarchitektur des DAFFODIL-Frontends ein, ist zum anderen aber allgemein genug gehalten, um in verschiedenen Scaffolding-Szenarien zum Einsatz zu kommen.

Die beiden festen Bestandteile des Assistenten sind ein Hilfetext und eine Navigationsleiste. Der Hilfetext bietet für den aktuellen Schritt Erklärungen, die auf das Unterstützungsniveau angepasst werden. Die Navigationsleiste erlaubt, zum nächsten Schritt weiterzuspringen oder zu einem bereits früher bearbeiteten Schritt zurückgesprungen. Dazu kommt eine wechselnde Komponente mit für den aktuellen Schritt ausgelegten Scaffolds (z. B. Aufgabenbeschreibung oder Planungsnotizen).

Die Aufgabe und die Navigation werden im XML-Format beschrieben, wobei eine XML-Konfiguration des Assistenten die Beschreibungen für alle Unterstützungsniveaus zur Aufgabe enthält. Quelltext 7.1 zeigt einen Ausschnitt aus einer solchen Konfigurationsdatei. Zu jedem Niveau werden die folgenden Aspekte festgelegt:

- **step:** die zugehörigen Schritte, in welche die Aufgabe zerlegt wurde,
- **help:** die benutzten Hilfetexte des Assistenten in jedem Schritt,
- **component:** die interne Komponente des Assistenten für jeden Schritt,
- **input:** welche Eingaben aus anderen Schritten zu übernehmen sind,
- **perspective:** die zu benutzende Perspektive (siehe den folgenden Abschnitt).

Eine kommentierte DTD für geführte Aufgaben befindet sich in Anhang A der Arbeit (siehe Quelltext A.1).

### Perspektiven

Die einzelnen Schritte der Unterstützung sind als DAFFODIL-Perspektiven implementiert (siehe Abschnitt 5.2). Jede Perspektive enthält das Assistenten-Werkzeug, sowie zusätzlich die weiteren im aktuellen Schritt notwendigen

## 7.2. NUTZERFÜHRUNG DURCH SCAFFOLDING

---

### Quelltext 7.1 Beschreibung einer geführten Aufgabe.

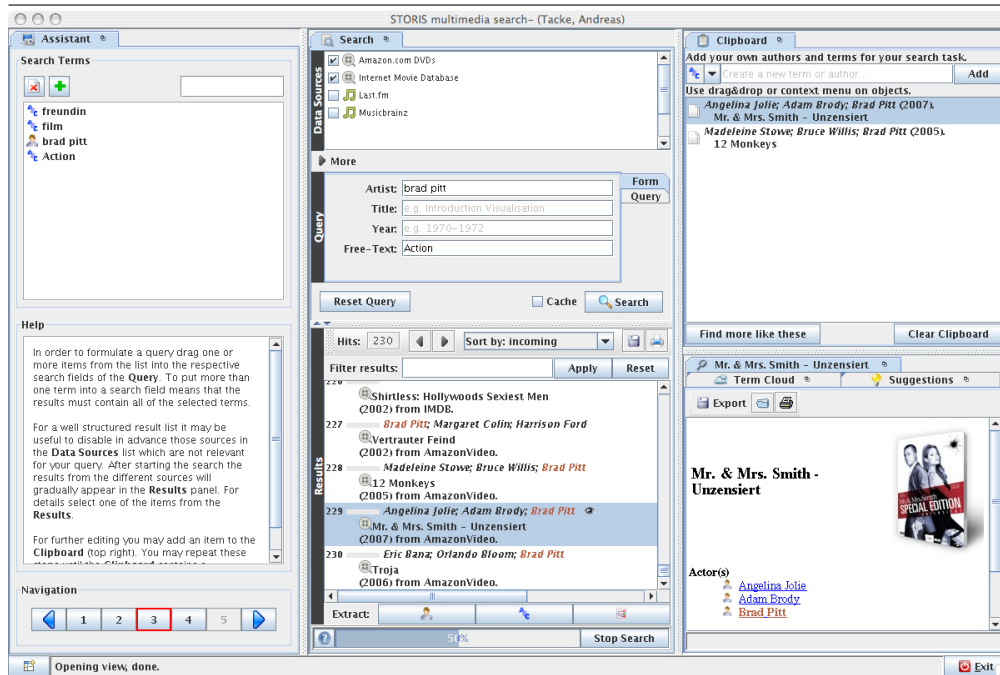
---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE task SYSTEM "http://www.example.com/guidedtask.dtd">

<task id="mediasearch">
  <!-- Strings in a task description are localized and stored in a
  property file. Use keys to reference them. -->
  <title>storis.task.mediasearch.title</title>
  <description>storis.task.mediasearch.description</description>
  <scaffold level="10" id="beginner">
    <!-- A step element has to have an "id" attribute containing
    the position of the step as an integer. The corresponding
    perspective is referenced by the "perspective"
    attribute containing the relative path to the xml file. -->
    <step id="1" perspective="beginner.xml">
      <title>storis.task.mediasearch.step1.title</title>
      <help>storis.task.mediasearch.step1.help</help>
      <component id="freetext"
        src="de.unidue.gframedl.gui.tools.storis.Freetext">
        <title>storis.task.mediasearch.freetext.title</title>
        <input step="2" component="termlist" />
      </component>
    </step>
    <step id="2" perspective="beginner2.xml">
      <title>storis.task.mediasearch.step2.title</title>
      <help>storis.task.mediasearch.step2.help</help>
      <component id="termlist"
        src="de.unidue.gframedl.gui.tools.storis.TermList">
        <title>storis.task.mediasearch.termlist.title</title>
        <input step="1" component="freetext" />
        <input step="3" component="termlist2" />
      </component>
    </step>
    <step id="3" perspective="beginner3.xml">
      <title>storis.task.mediasearch.step3.title</title>
      <help>storis.task.mediasearch.step3.help</help>
      <component id="termlist2"
        src="de.unidue.gframedl.gui.tools.storis.TermList">
        <title>storis.task.mediasearch.termlist.title</title>
        <input step="2" component="termlist" />
      </component>
    </step>
    <step id="4" perspective="beginner4.xml">
      <title>storis.task.mediasearch.step4.title</title>
      <help>storis.task.mediasearch.step4.help</help>
    </step>
    <step id="5" perspective="beginner5.xml">
      <title>storis.task.mediasearch.step5.title</title>
      <help>storis.task.mediasearch.step5.help</help>
    </step>
  </scaffold>

  <scaffold level="5" id="intermediate">
    ...
  </scaffold>
</task>
```

Abbildung 7.3. Perspektive für Schritt 3: Suche.



Werkzeuge in einer vorgegebenen Anordnung. Dabei wird eine generische Implementierung von Perspektiven benutzt, die ihren Aufbau dynamisch aus einer XML-Konfigurationsdatei liest. Dies erlaubt die flexible Konfiguration des Systems für unterschiedliche unterstützte Aufgaben.

In Quelltext 7.2 wird beispielhaft die Beschreibung einer solchen Perspektive für den dritten Schritt der beschriebenen Suchstrategie gezeigt. Eine kommentierte DTD für Perspektiven befindet sich in Anhang A der Arbeit (siehe Quelltext A.2). Die beschriebene Perspektive ist in Abbildung 7.3 zu sehen und besteht aus drei parallelen Bildschirmkacheln: links befindet sich der Assistent mit Navigation (STORISTool), in der Mitte das Suchwerkzeug. Die rechte Kachel ist erneut unterteilt in eine obere Hälfte mit der Zwischenablage und eine untere Hälfte mit verschiedenen anderen Werkzeugen.

## Fading

*Fading* ist eine wichtige Komponente des Scaffolding-Prozesses. Dabei muss das *Fading* nicht notwendigerweise kontinuierlicher Natur sein, sondern kann auch durch diskrete Abstufungen realisiert werden [GUZDIAL 1994]. Die Scaffolding-Umsetzung für DAFFODIL bedient sich dabei des bereits beschriebenen Konzeptes der Unterstützungsniveaus. Das Scaffolding eines neuen Benutzers beginnt während der Bearbeitung der ersten Aufgabe auf dem höchsten Unterstützungsniveau von 10 und kann dann bei späteren Arbeiten mit

dem System nach und nach reduziert werden, bis der Benutzer den sicheren strategischen Umgang mit dem Suchsystem erlernt hat.

Jackson und Kollegen [JACKSON et al. 1998] unterteilen *Fading* beim softwarebasierten Scaffolding in zwei Varianten. Bei adaptiven Umsetzungen wird ein explizites Benutzermodell verwendet und die Scaffolds der Benutzeroberfläche passen sich automatisch an das Verständnis des Benutzers an. Ein solches Modell des Wissens des Lernenden lässt sich aber in offenen Domänen oder für flexibel einsetzbare Systeme nur schwer spezifizieren oder evaluieren. Hingegen ist es bei adaptierbaren Umsetzungen der Benutzer selbst, der das *Fading* der Scaffolds initiiert. Dabei kann es für den Lernenden allerdings schwierig sein, seinen Fortschritt einzuschätzen und angemessene *Fading*-Entscheidungen zu treffen.

In der vorgestellten Umsetzung wurde zunächst eine adaptierbare Variante gewählt, bei der ein Benutzer zum Start der Aufgabe eine geringere Unterstützung wählen kann, sobald er sich beim Umgang mit dem System sicher genug fühlt. Erweiterungen könnten entweder Entscheidungsunterstützung durch das System bereitstellen, wie von Jackson und Kollegen vorgeschlagen [JACKSON et al. 1998], oder durch ein Benutzermodell echte Adaptivität ermöglichen.

### 7.2.4. Evaluation

Während der Entwicklung der Scaffolding-Erweiterung fand eine begleitende formative Evaluierung statt. Dabei wurden insgesamt 25 aus der Studierendenschaft der Universität Duisburg-Essen rekrutierte und mit dem Suchsystem DAFFODIL nicht vertraute Teilnehmer mit einzelnen Komponenten, sowie Mock-Ups des Gesamtsystems konfrontiert. Zur Datenerhebung wurden dafür Fragebögen mit einer fünfstufigen Likert-Skala von „sehr hilfreich“ (1) bis „überhaupt nicht hilfreich“ eingesetzt. Die Ergebnisse flossen in die Umsetzung ein.

Insbesondere die schrittweise Navigation mit ihrer Möglichkeit, zu bereits bearbeiteten Schritten zurückzukehren, wurde von den Teilnehmern der Evaluation positiv bis sehr positiv bewertet (Modalwert 1 bzw. 2). Gleiches gilt für Organisations- und Planungskomponenten wie das Planungswerkzeug, die Zwischenablage und das Exportwerkzeug, die durchweg als hilfreich beurteilt wurden (Modalwerte von 1-2).

Die endgültige Umsetzung wurde dann einer summativen Evaluation unterzogen, bei der zwölf Teilnehmer jeweils eine Suchaufgabe mit dem System auf höchstem und niedrigstem Unterstützungsniveau ausführten. Das System auf niedrigstem Unterstützungsniveau entsprach dabei der Standardperspektive von DAFFODIL. Die Reihenfolge, in der die Probanden die Systeme verwendeten, wurde rotiert.

**Quelltext 7.2** Perspektive für Schritt 3: Suche.

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE perspective SYSTEM "http://www.example.com/perspective.dtd">

<perspective id="Search and Copy">
  <pane id="ROOT">
    <splitpane id="FIRST" type="horizontal">
      <divideroffset>-250</divideroffset>
      <pane id="WEST" type="left">
        <tool id="tools.storis">
          de.unidue.gframedl.gui.tools.storis.STORISTool
        </tool>
      </pane>
      <pane id="EAST1" type="right">
        <splitpane id="SECOND" type="horizontal">
          <divideroffset>-160</divideroffset>
          <pane id="CENTER" type="left">
            <tool id="tools.search">
              de.unidue.gframedl.gui.tools.searching.SearchTool
            </tool>
          </pane>
          <pane id="EAST2" type="right">
            <splitpane id="SECOND" type="vertical">
              <divideroffset>-220</divideroffset>
              <pane id="NORTHEAST" type="top">
                <tool id="tools.clipboard">
                  de.unidue.gframedl.gui.tools.clipboard.BasketTool
                </tool>
              </pane>
              <pane id="SOUTHEAST" type="bottom">
                <tool id="tools.tagclouds">
                  de.unidue.gframedl.gui.tools.viewer.TagTool
                </tool>
                <tool id="tools.suggestion">
                  de.unidue.gframedl.gui.tools.suggest.SuggestionTool
                </tool>
                <tool id="tools.details">
                  de.unidue.gframedl.gui.tools.viewer.DetailTool
                </tool>
              </pane>
            </splitpane>
          </pane>
        </splitpane>
      </pane>
    </splitpane>
  </pane>
  <settings>
    <property key="tools.search.extract.authors">true</property>
    <property key="tools.search.extract.journals">>false</property>
    <property key="tools.search.extract.conferences">>false</property>
    <property key="tools.search.extract.keywords">true</property>
  </settings>
</perspective>
```

---

Die Ergebnisse der summativen Evaluation zeigten jedoch keine signifikanten Unterschiede in der Akzeptanz der Probanden oder in der Performanz bei der bearbeiteten Aufgabe. Die Probanden konnten mit beiden Systemen die Aufgaben gleich gut (oder gleich schlecht) lösen und es trat kein zeitlicher Vorteil zu Tage. Bei der qualitativen Ermittlung der Zufriedenheit mit den beiden Systemen gab es keinen signifikanten Vorteil für das System mit Scaffolding. Es zog allerdings kein Proband das System ohne Scaffolding vor, während die meisten beide gleich bewerteten. Die Ergebnisse sind zum einen durch die wenigen Teilnehmer der Studie und die geringe Anzahl durchgeführter Suchen begründet, zum anderen war aber auch die Suchaufgabe womöglich ungeeignet gewählt (nicht aus der *Zone of Proximal Development*), so dass selbst unerfahrene Suchende diese ohne Hilfe leicht bewältigen konnten und nicht durch das Scaffolding profitierten.

Eine über einen längeren Zeitraum angelegte Studie, ob bei der Benutzung des Suchsystems mit Software-Scaffolding ein erhoffter Lerneffekt auftritt, der über die Erlernung der Werkzeugbenutzung hinausgeht und eine Aneignung prozeduralen, strategischen Suchwissens durch den Benutzer zeigt, fand aus Zeitgründen im Rahmen dieser Arbeit nicht statt.

### **7.3. Zusammenfassung**

In diesem Kapitel wurde vorgestellt, wie mit Hilfe des Strategieframeworks die automatische Durchführung einfacher Suchstrategien in DAFFODIL möglich ist. Solche einfachen Strategien können entweder direkt als Werkzeuge angeboten, als Bausteine für komplexere Strategien verwendet oder in die in Kapitel 6 beschriebene vorschlagsbasierte Unterstützung aufgenommen werden. So lassen sich nicht nur einzelne Aktionen, Taktiken oder Strategeme als ausführbare Optionen vorschlagen, sondern ganze Teilstrategien.

Zur Erlernung strategischer, prozeduraler Fähigkeiten im Umgang mit DAFFODIL wurde software-basiertes Scaffolding vorgeschlagen und implementiert, eine Technik, die es ungeübten Benutzern erlaubt, komplexere Aufgaben zu bearbeiten und dabei prozedurales Wissen zu erwerben. Verschiedene der in dieser Arbeit vorgestellten Werkzeuge (siehe Abschnitt 5.1.1) und Methoden (siehe Abschnitt 6.1 und 7.1) wurden in eine geführte Umsetzung von Suchstrategien integriert. Dabei ist das grundlegende Konzept des Scaffolding zur Vermittlung prozeduraler Fähigkeiten bereits in einer Reihe von Anwendungen mit positivem Ergebnis evaluiert worden [GUZDIAL 1993, GUZDIAL 1994, QUINTANA et al. 2002, QUINTANA und ZHANG 2004b] und die allgemein gehaltene Umsetzung für DAFFODIL über in XML definierte Aufgabenbeschreibungen und Perspektiven in Kombination mit einem anpassbaren Assistentenwerkzeug bietet eine flexible Möglichkeit zur Implementierung verschiedener Scaffolding-Mechanismen für unterschiedliche Szenarien in einem Suchsystem für Digitale Bibliotheken.



**Teil III.**

**Evaluation**



# Kapitel 8.

## Evaluation

Zur Beurteilung der Tauglichkeit der entwickelten Unterstützungsansätze wurden zwei Benutzerexperimente durchgeführt, die sich auf die Untersuchung des Strategischen Vorschlagssystems richteten und die folgenden Hypothesen überprüfen sollten:

1. Erfolgreiche Suchaktionen erfahrener Benutzer können genutzt werden, um mittels fallbasiertem Schließen situationsangemessene Vorschläge zu machen, mit denen auch unerfahrene Benutzer zielführend Taktiken und Strategeme einsetzen können.
2. Ein Digital-Library-System kann unerfahrene Benutzer bei der Bildung einer Suchstrategie für ihr Informationsproblem unterstützen. Insbesondere können Benutzer aus strategischen Vorschlägen prozedurales Wissen erlangen, das sie später unabhängig in eigenen Suchen einsetzen können.

### 8.1. Strategisches Vorschlagssystem: Experiment 1

Das erste Benutzerexperiment zur Evaluation des Vorschlagswerkzeugs [KRIEWEL und FUHR 2007] war eine leichtgewichtige Studie mit zwei wichtigen Zielen:

1. einen ersten Eindruck über die Benutzerakzeptanz für strategische Vorschläge zu gewinnen, und
2. die Angemessenheit des Rankings der Vorschläge bezüglich der Ähnlichkeit der Benutzersituation zu den Situationen früherer Benutzer zu evaluieren.

Zur Vorbereitung des Experiments wurde das System trainiert. Dabei wurde eine Suchaufgabe gewählt, die thematisch keinen Bezug zur Suchaufgabe hatte, die für die Evaluation selbst genutzt wurde. Das Ziel war es, eine Fallbasis aufzubauen, in der für jeden Vorschlag zumindest eine Situation (ein Fall) existiert, für den der Vorschlag ein passendes Vorgehen darstellt. Es wurden für diese erste Evaluation keine negativen Fälle, sowie ein Schwellenwert  $m_p$  von

0 gewählt, so dass selbst Vorschläge mit niedrigem Gewicht in einer Situation präsentiert wurden.

Insgesamt zwölf Teilnehmer wurden für das Experiment rekrutiert. Dabei handelte es sich um zwei weibliche und zehn männliche Studierende der Universität Duisburg-Essen, aus den Studiengängen Angewandte Informatik und Angewandte Kommunikations- und Medienwissenschaften. Fünf der Studierenden verfügten bereits über ein abgeschlossenes Diplomstudium und verfolgten ein Promotionsstudium in Angewandter Informatik.

### 8.1.1. Versuchsaufbau

Die Versuchsteilnehmer wurden gebeten, eine komplexe, umfangreiche Suchaufgabe mit Hilfe der DAFFODIL-Software und des Vorschlagswerkzeugs zu bearbeiten. Dazu wurde ihnen eine einfache Perspektive des DAFFODIL-Systems mit einer ausgewählten Teilmenge der existierenden Werkzeuge zur Verfügung gestellt, und die Suchen selbst wurden über mehreren Artikelkollektionen aus der Domäne der Informatik durchgeführt. Der Zweck des Vorschlagswerkzeugs wurde ihnen erklärt und wo notwendig wurde eine kurze Einführung in das DAFFODIL-System (nicht aber in die Bedienung der Vorschlagssoftware) gegeben.

Die Suchaufgabe selbst war nur lose definiert, aber umfangreich genug, so dass selbst bei erfahrenen Suchern zu erwarten war, dass sie eine Reihe von Anfragen stellen und verschiedene Werkzeuge aus der von DAFFODIL bereitgestellten Sammlung würden benutzen müssen. Die Teilnehmer der Evaluation wurden gebeten,

„Artikel und zusätzliche Informationen wie z.B. wichtige Autoren, Suchbegriffe, zentrale Konferenzen oder Journalausgaben zu sammeln, um eine Semesterarbeit über Visualisierungsmethoden von Resultatlisten in Information-Retrieval-Systemen anfertigen zu können.“

Diese Suchaufgabe wurde gewählt, um den Suchenden eine simulierte, realistische Arbeitsaufgabe zu geben, die mit der verfügbaren Kollektion gelöst werden kann, und stellte ein nur leicht modifiziertes echtes Informationsproblem dar.

Jeder Teilnehmer hatte zur Bearbeitung der Aufgabe etwa 45 Minuten Zeit, wobei nach 40 Minuten ein Hinweis erfolgte, dass die Zeit für die Aufgabe zu Ende geht, und nach spätestens 50 Minuten die Suche abgebrochen wurde. Während der Suche waren die Versuchsteilnehmer gebeten, das Vorschlagswerkzeug zur Rate zu ziehen und es für die Generierung neuer Suchideen zu nutzen. Es war ihnen freigestellt, alle Vorschläge des Werkzeuges zu ignorieren und eigene Taktiken anzuwenden. Während des Experiments war ein

Beobachter anwesend, um technische Hilfe zum DAFFODIL-System zu geben und bei Systemfehlern einzugreifen. Keine strategische oder terminologische Hilfe wurde angeboten, und bei entsprechenden Fragen der Teilnehmer wurde auf das Vorschlagswerkzeug verwiesen.

Jede der Sitzungen wurde mit Hilfe des Logging-Frameworks des DAFFODIL-Systems (siehe Kap. 5.4) aufgezeichnet. Die durchschnittliche Sitzungslänge betrug 47 Minuten, in denen die Teilnehmer Suchanfragen stellten, Dokumente betrachteten und Ergebnisse speicherten. Insgesamt 198 Anfragen wurden von den Suchenden gestellt und 143 Objekte (Dokumente, Terme und Autoren) wurden als relevant abgespeichert (durchschnittlich 16,5 Anfragen und 11,9 gespeicherte Objekte pro Teilnehmer).

Im Verlauf ihrer Suchaufgaben benutzten die Teilnehmer das Vorschlagswerkzeug zur Generierung von Vorschlägen insgesamt 94-mal (etwa 7,8-mal pro Sucher oder ungefähr einmal pro zwei Anfragen). Nach dem Lesen der Vorschläge wurden 62 Vorschläge direkt (semi-automatisch) ausgeführt, weitere 29 Vorschläge wurden durch den Suchenden „von Hand“ umgesetzt.

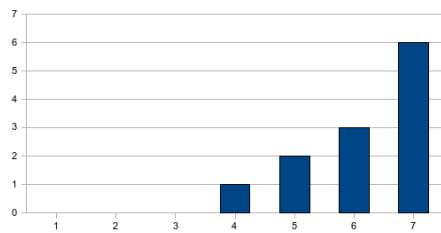
Das Experiment wurde mit einem zweiteiligen Fragebogen begleitet. Der erste, vor dem Experiment ausgeteilte Fragebogen, sollte Domänen- und Suchwissen der Teilnehmer ermitteln (Selbsteinschätzung und objektive Merkmale). Der zweite Teil des Fragebogens wurde nach dem Experiment vom Teilnehmer ausgefüllt und bildete die Grundlage für ein kurzes, etwa 10-minütiges Abschlussinterview. Dieser Teil enthielt insgesamt vier Fragenblöcke, in denen die Suchenden ihre Erfahrung mit dem Vorschlagswerkzeug bewerten sollten.

### **8.1.2. Ergebnisse**

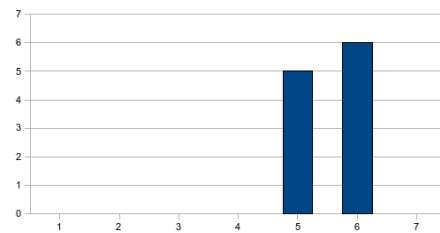
Aufgrund der geringen Teilnehmerzahl des Nutzerexperimentes sind die Resultate mit Vorsicht zu genießen. Einige der Ergebnisse deuteten allerdings eine vielversprechende Richtung an und rechtfertigten eine Weiterentwicklung und zweite Evaluation des Vorschlagswerkzeugs (siehe Abschnitt 8.2). Die meisten der Fragen benutzten eine 7-Punkt-Likertskala von „stimme gar nicht zu“/„fully disagree“ (mit dem Wert 1 kodiert) bis „stimme voll zu“/„fully agree“ (mit dem Wert 7 kodiert). Die Benutzer wurden durch den Fragebogen und im abschließenden Interview ermuntert, ihre Antworten näher auszuführen und konkrete Beispiele für Probleme oder Unzufriedenheiten mit dem Werkzeug zu nennen.

Die Fragen unterteilten sich in vier Blöcke: Klarheit der Vorschläge, Angemessenheit der Vorschläge in der aktuellen Situation, Nützlichkeit der Vorschläge für die konkrete Suchaufgabe und Benutzbarkeit der Vorschlags-Benutzerschnittstelle.

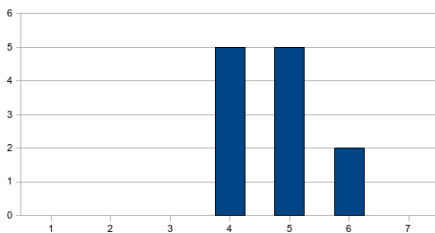
Abbildung 8.1. Evaluationsergebnisse.



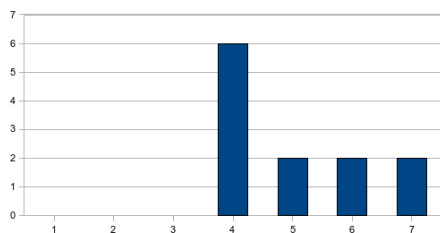
(a) Die Erklärungen waren klar und einfach zu verstehen.



(b) Die Resultate der Vorschläge entsprechen den Erwartungen.



(c) Die Vorschläge waren der Situation angemessen.



(d) Die Reihenfolge der Vorschläge war angemessen.

### Klarheit

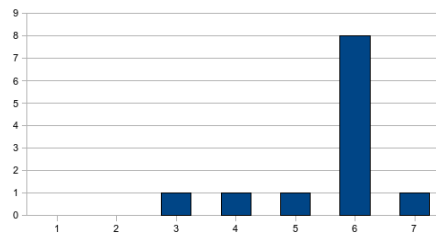
Nahezu alle Teilnehmer des Experimentes fanden die Vorschläge leicht verständlich. Die Erklärungen, warum ein spezifischer Vorschlag nützlich sein könnte, wurden allgemein für klar und nachvollziehbar gehalten (Modus 7, Median 6,5). Nur selten stießen die Teilnehmer auf Vorschläge, die sie anders als vom Autoren des Vorschlags erwartet interpretierten und bei denen sie infolgedessen überrascht von den Ergebnissen des Vorschlags waren (Modus und Median 6).

Ein besonderes Problem, das von mehreren Benutzern genannt wurde, war die Beschreibung des Vorschlags zur Extraktion neuer Anfrageterme aus der Resultatliste, bei dem drei Benutzern nicht klar war, wie die Liste der neuen Terme zustandekam.

### Angemessenheit

Da die Vorschläge nach Ähnlichkeit der aktuellen Situation zu den Situationen aus früheren Suchen ausgewählt und angeordnet wurden, war es insbesondere von Interesse, zu erfahren, ob diese *Ranking*-Methode die Vorschläge in einer dem Suchkontext angemessenen Weise ordnet. Obwohl die Resultate nur marginal positiv waren (Modus 4, Median 5), zeigte sich bei der Betrachtung der Detailantworten und im anschließenden Interview, dass diese Ergebnisse

Abbildung 8.2. Evaluationsergebnisse (Forts.).



(a) Die Vorschläge waren nützlich für die Suchaufgabe.

vor allem auf den Aufbau des Experimentes zurückzuführen waren. Fast alle berichteten Probleme bestanden mit sehr niedrig gewichteten Vorschlägen. Diese wurden im Experiment mit angezeigt, um den Teilnehmern eine größere Bandbreite an Vorschlägen in jeder Situation zu präsentieren. Eingeschränkt auf hoch gewichtete Vorschläge (mit einer Bewertung von 0,5 oder höher) wurden die Vorschläge als durchweg angemessen (Modus und Median 6) und die Reihenfolge als schlüssig (Modus 5, Median 5,5) bewertet.

### Nützlichkeit

Die hoch gewichtigen Vorschläge in jeder Situation wurden allgemein als nützlich und hilfreich bewertet (Median und Modus 6). Zehn der zwölf Teilnehmer erklärten, dass sie dank der Vorschläge Taktiken und Suchstrategeme eingesetzt hätten, die ihnen vorher nicht bekannt waren, die sie üblicherweise nicht benutzt hätten oder an die sie in der Situation nicht gedacht hätten. Alle zehn fanden diese „neuen“ Taktiken und Strategeme hilfreich und planten, diese unabhängig von den Vorschlägen bei zukünftigen Suchaufgaben zu benutzen.

Als besonders hilfreich wurden Vorschläge empfunden, die auf die verschiedenen Werkzeuge des DAFFODIL-Systems hinwiesen (Thesaurus, Klassifikationsbrowser, Autorennetzwerk), deren Zweck oder sogar Existenz für neue Nutzer oder Gelegenheitsbenutzer des Systems nicht offensichtlich war. Erfahrene Benutzer nannten vor allem die Extraktion häufiger Autoren, Konferenzen und Terme aus der Resultatliste, die durch das Vorschlagswerkzeug angestoßen wurden, als äußerst hilfreich.

Einige Benutzer erwähnten im Laufe des Interviews, dass sie zwar eventuell einige der vorgeschlagenen Taktiken auch ohne das Werkzeug angewandt hätten, dass sie das Vorschlagswerkzeug aber schneller und ohne langwieriges und frustrierendes Herumprobieren auf die verfügbaren Möglichkeiten in der aktuellen Situation aufmerksam gemacht hätte.

### Schnittstelle

Die visuelle Präsentation der Vorschläge wurde allgemein gut angenommen und die Teilnehmer verstanden sowohl das Ranking und die Darstellung der Vorschlagsgewichte, als auch die benutzten *Icons* für verfügbare Aktionen und Eigenschaften der Vorschläge. 75% der Benutzer bevorzugten, dass die Vorschläge ihnen nicht automatisch, sondern erst auf Nachfrage gezeigt wurden. Insbesondere nannten mehrere Benutzer es als vorteilhaft, dass die Vorschläge zu einer alten Situation erhalten blieben, bis explizit zu einer neuen Situation Vorschläge angefordert wurden, da sie dadurch mehrere Vorschläge zu einer Situation ausprobieren konnten. Der Verfügbarkeitsindikator erwies sich als deutlich genug, um Benutzer auf die Existenz neuer Vorschläge aufmerksam zu machen, war aber nicht zu aufdringlich, um von der eigentlichen Suche abzulenken.

Die Interaktion mit dem Werkzeug erwies sich als problematischer. Zwei deutliche Mängel wurden durch das Experiment aufgedeckt, welche die Benutzbarkeit einschränkten:

1. Das *Icon*, das einen Vorschlag als „automatisch ausführbar“ kennzeichnete, wurde von allen Teilnehmern verstanden. Aufgrund der Ähnlichkeit der Grafik mit dem „Abspiel“-Knopf vieler Medienwiedergabe-Programme versuchte allerdings ein Drittel der Benutzer einen Vorschlag durch Anklicken des *Icons* auszulösen, anstatt wie erwartet durch Doppelklick.
2. Obwohl die meisten der Benutzer willens waren, gute Vorschläge als hilfreich zu bewerten, hatten viele Probleme, die dafür vorgesehene Funktion finden, da sie im Kontextmenü des Vorschlags verborgen war.

Um diese Probleme zu beheben, scheint es sinnvoll, beide Aktionen (Ausführen und Bewerten) für alle Aktionen durch einfach erreichbare Knöpfe zur Verfügung zu stellen.

## 8.2. Strategisches Vorschlagssystem: Experiment 2

Für das zweite Benutzerexperiment [KRIEWEL und FUHR 2010] wurden die in Abschnitt 8.1 genannten Mängel behoben, und eine Reihe neuer Vorschläge hinzugenommen. Die zugrundeliegende Fallbasis war seit dem ersten Experiment durch die Benutzer des Systems auf insgesamt 150 Fälle angewachsen, davon 18 negative Fälle.

Ziel der Untersuchung war zum einen, festzustellen, ob Benutzer mit Hilfe situationsspezifischer Vorschläge erfolgreicher suchen können, und zum anderen, ob Benutzer Vorschläge auch unabhängig in späteren Suchen an angemessenen Stellen einsetzen würden. Insgesamt wurden drei Fragen untersucht zu den folgenden Teilhypothesen:



1. Die Unterstützung durch das Vorschlagswerkzeug führt dazu, dass Benutzer durch den Einsatz der erweiterten Möglichkeiten signifikant mehr relevante Dokumente finden als Benutzer ohne Unterstützung.
2. Die Unterstützung durch das Vorschlagswerkzeug führt dazu, dass Benutzer auch die erweiterten Möglichkeiten des Suchsystems nutzen. Spezifisch auf DAFFODIL bezogen wird vermutet, dass Benutzer signifikant häufiger erweiterte Werkzeuge neben der eigentlichen Suche (Klassifikation, Verwandte Begriffe, Thesaurus, Koautorensuche, Extraktion aus Ergebnis und relevanten Dokumenten) und signifikant häufiger zusätzliche Felder des erweiterten Suchformulars nutzen.
3. Die situationsspezifischen Vorschläge führen dazu, dass Benutzer in einer späteren Suche signifikant häufiger die erweiterten Möglichkeiten des Suchsystems anwendet, als Benutzer, die nie Vorschläge erhalten haben.

Für die Untersuchung der ersten Fragestellung wurde die Anzahl der von den Benutzern gespeicherten Dokumente betrachtet. Zur zweiten und dritten Fragestellung wurden die Suchhistorien der Benutzer analysiert und die Benutzung der unterschiedlichen DAFFODIL-Werkzeuge, sowie die Benutzung verschiedener Felder im Suchformular ausgewertet.

### 8.2.1. Versuchsteilnehmer

Für das Experiment wurden insgesamt 24 freiwillige Studierende der Studiengänge Angewandte Informatik, Angewandte Kognitions- und Medienwissenschaften und verwandter Studiengänge an der Universität Duisburg-Essen rekrutiert. Die Studenten, 20 von ihnen männlich und 4 weiblich, erhielten zusätzliche Übungspunkte für ihre Teilnahme. Das Alter der Teilnehmer lag zwischen 22 und 48 Jahren mit einem durchschnittlichen Alter von 27,25 (Standardabweichung 5,41, Modalwert 24).

Vor Beginn des Experiments wurde ein Fragebogen ausgegeben, auf dem die Teilnehmer unter anderem ihre bisherige Sucherfahrung angeben sollten. Auf einer fünf-stufigen Likertskala bewerteten sich 2 Teilnehmer als unerfahren (2), 6 als etwas erfahren (3), 15 als erfahren (4) und 1 als sehr erfahren (5). Niemand schätzte sich selbst als sehr unerfahren ein.

Da Studien [AULA und NORDHAUSEN 2006] jedoch gezeigt haben, dass Selbstevaluation von Suchfertigkeit nur ein schwacher Indikator für die tatsächliche Suchleistung darstellt (zumindest für einige Aufgabentypen), wurden die Teilnehmer auch gebeten, anzugeben, seit wie vielen Jahren sie regelmäßig Suchsysteme verwenden und wie häufig sie Suchsysteme einsetzen. Die durchschnittliche Erfahrung waren 4,75 Jahre mit einer Standardabweichung von 2,45. Nicht alle Teilnehmer gaben eine verwertbare Einschätzung ihrer Benutzungshäufigkeit von Suchmaschinen an, aber nur 6 benutzten Suchmaschinen

seltener als täglich. Unter den täglichen Suchmaschinennutzern, die eine genauere Abschätzung nannten, reichte die Häufigkeit von 1–35 Suchen pro Tag (durchschnittlich 11 pro Tag, Standardabweichung 11,84). Allerdings sind diese Suchen vielfach recht einfach und die derart gesammelte Erfahrung nicht für komplexe Suchen hilfreich.

Nach Suchsystemen gefragt, mit denen sie vertraut seien, nannten 23 der Teilnehmer Google, 4 nannten Yahoo und 2 MSN Search bzw. Bing. Drei der Studenten benutzten Desktop-Suchsysteme wie etwa Google Desktop oder Beagle für Linux. Drei Studenten benutzten Digitale Bibliotheken. Alle Benutzer waren mit DAFFODIL unerfahren oder kannten das System nicht. Keiner benutzte es regelmäßig.

### 8.2.2. Versuchsaufbau

Für die Studie kamen zwei Systeme zum Einsatz, die beide auf dem DAFFODIL-System basierten. Die Systeme waren bis auf den Einsatz des Vorschlagswerkzeuges in einem der beiden identisch. In beiden war das Suchwerkzeug für die Suche in sechs Digitalen Bibliotheken der Informatik-Domäne vorkonfiguriert. Von den in Abschnitt 5.1.1 beschriebenen Werkzeugen standen Suchwerkzeug, Extraktionswerkzeug, Koautorenwerkzeug, Thesaurus und Verwandte Begriffe, Klassifikationsbrowser, Suchhistorie, Körbchen und Detailansicht zur Verfügung.

Alle Teilnehmer erhielten drei komplexe Suchaufgaben, die sie in Sitzungen von jeweils 20 Minuten bearbeiten sollten. Während dieser Zeit waren sie angehalten, so viele relevante Dokumente wie möglich zu finden und zu speichern. Die Hälfte (12) der Teilnehmer nutzte das System mit Vorschlägen für die ersten beiden Aufgaben und das normale System für die letzte. Die andere Hälfte benutzte das normale System ohne Unterstützung für alle Aufgaben. Die Teilnehmer wurden vorab nicht über den Zweck der Studie informiert und es wurde ihnen nicht mitgeteilt, dass einige Suchunterstützung durch Vorschläge erhielten und einige nicht. Die Zuordnung der Teilnehmer zu den beiden Gruppen erfolgte zufällig. Keine signifikanten Unterschiede zwischen den Gruppen bezüglich durchschnittlicher Sucherfahrung (4,83 zu 4,67 Jahre,  $p = 0,872$ ) oder ihrer Selbsteinschätzung (beide Gruppen hatten einen Modalwert und einen Median von 4, was auf der verwendeten Likert-Skala „erfahren“ entsprach) konnten festgestellt werden.

Die Aufgaben wurden mit Hinblick auf die Dokumentensammlung ausgewählt, so dass die Suchaufgaben realen Arbeitsaufgaben ähnelten. Für eine der Aufgaben wurde das Adhoc-Topic Nr. 350 aus TREC-6 [HARMAN und VOORHEES 1998] angepasst übernommen, die anderen beiden Aufgaben wurden so ausgewählt, dass sie von ähnlicher Schwierigkeit und Komplexität waren (siehe Tabelle 8.1). Die Reihenfolge, in der die Teilnehmer die Aufgaben bearbeiteten, wurde rotiert.

Die Teilnehmer in beiden Gruppen erhielten eine identische Einführung in das Suchsystem DAFFODIL. Zusätzlich wurde den Teilnehmern in der Gruppe, die mit dem Vorschlags-Werkzeug arbeiten sollten, der Aufruf der Vorschläge demonstriert und erklärt, wie sie einen Vorschlag ausführen können. Das Bewerten der Vorschläge wurde für das Experiment deaktiviert. Vor jeder Aufgabe wurde den Teilnehmern eine schriftliche Beschreibung des zu bearbeitenden Suchszenarios gegeben. Falls notwendig, wurden Unklarheiten über die Aufgabe beantwortet, bevor der Teilnehmer mit der Suche begann. Während der Suche wurden vom Protokollanten nur technische Fragen beantwortet und keine Unterstützung in Form von Suchratschlägen oder Suchbegriffen gegeben.

**Tabelle 8.1.** Aufgabenbeschreibungen in Experiment 2

Task	Description
Gesundheit	Suchen Sie nach Artikeln, um zu beantworten, ob und inwieweit die ständige Arbeit an einem Computer bzw. Bildschirmarbeitsplatz zu gesundheitlichen Problemen führen kann. Relevante Dokumente sind Studien oder Berichte über gesundheitliche Probleme bei der Bildschirmarbeit. Dazu zählen etwa Augenüberbeanspruchung, Haltungsschäden, Muskel- oder Sehnenscheidenprobleme oder psychische Probleme (auch Stress).
Plagiate	Suchen Sie nach Artikeln, die Methoden beschreiben oder bewerten, um automatisch Software-Plagiate zu erkennen. Relevante Dokumente stellen Methoden zur Plagiaterkennung von Software-Quellcode vor, schildern wie existierende Verfahren für diese Aufgabe genutzt werden können oder beschreiben Evaluationen bzw. Vergleiche verschiedener Ansätze. Relevant sind auch Methoden zur Feststellung von Autorenschaft von Programmcode. Nicht relevant sind Methoden, die sich hauptsächlich oder ausschließlich mit der Plagiaterkennung bei digitalen Textdokumenten wie z. B. journalistischen oder wissenschaftlichen Artikeln befassen.
Java	Suchen Sie zum Aufbau und zur Untermauerung einer Argumentation für oder gegen Java als Unterrichtssprache im ersten Semester einer Informatikerausbildung nach geeigneten Artikeln. Relevant sind Artikel, die sich mit Java als Lehr- oder Unterrichtssprache auseinandersetzen. Ebenfalls relevant sind Vergleiche mit anderen Programmiersprachen für diese Zwecke, sowie Artikel zu allgemeinen Kriterien und Anforderungen an eine solche Lehrsprache. Nicht relevant sind Texte, die sich mit dem Erlernen und Unterrichten von Java beschäftigen.

### 8.2.3. Auswertung

Das Logging-Framework von DAFFODIL wurde benutzt, um alle Benutzeraktivitäten während der Suche festzuhalten. Zusätzlich wurden die Suchenden angehalten, während der Bearbeitung der Aufgaben laut zu denken. Ein Protokollant war während des Experiments anwesend, um diese Bemerkungen festzuhalten. Die Protokolle wurden später benutzt, um uneindeutige Logeinträge zu klären.

Aus den Suchlogs wurden zwei Teilhistorien extrahiert: (a) eine Historie aller Anfragen, die mit dem Suchwerkzeug von DAFFODIL gestellt wurden, (b) eine Historie aller Verwendungen der Werkzeuge DAFFODILS. Das Anfrage-log wurde mit Hilfe der Tabelle 8.2 handkodiert. Das Werkzeuglog wurde benutzt, um die durchschnittliche Werkzeugnutzung pro Aufgabe zu berechnen. Die Resultate sind in Tabelle 8.3 zu sehen.

Alle von den Teilnehmern während der einzelnen Aufgaben gespeicherten Dokumente wurden pro Aufgabe zusammengeführt. Zwei unabhängige Personen beurteilten diese Dokumente auf Relevanz. Diese blinden Relevanzurteile wurden dann genutzt, um die Zahl der relevanten Dokumente zu ermitteln, die jeder Suchende für jede Aufgabe gespeichert hat.

**Tabelle 8.2.** Moves zur Änderung der Anfrage (Unterstützung nur für Aufgaben 1 und 2)

Code	Move	Ø (Stdabw.)	Ø (Stdabw.)	Welch-Test		
		m. Unterst.	o. Unterst.	<i>t</i>	<i>df</i>	<i>p</i>
T	Term hinzu/weg	7,50 (5,95)	12,00 (7,62)	-1,61	20,78	0,122
S	Term ersetzen	11,08 (4,89)	16,92 (8,54)	-2,05	17,51	0,055
R	ganze Anfrage ersetzen	3,33 (2,57)	6,00 (4,05)	-1,93	18,64	0,069
O	Anfrage wiederholen	0,83 (1,11)	1,08 (1,31)	-0,50	21,44	0,620
F	zus./anderes Feld nutzen	7,08 (5,04)	0,92 (1,78)	3,99	13,71	0,001
B	Boolsch. Op. hinzu/weg	6,25 (6,16)	6,66 (6,92)	-0,16	21,71	0,877
C	Schreibweise korrigiert	2,25 (1,82)	2,42 (2,81)	-0,17	18,82	0,865
F <sub>3</sub>	Felder nutzen (Aufg. 3)	1,75 (1,42)	0,25 (0,62)	3,35	15,05	0,004

### 8.2.4. Ergebnisse

Neben der quantitativen Auswertung wurde zur qualitativen Erhebung der Nutzerzufriedenheit ein Postevaluations-Fragebogen ausgegeben. Hier sollten die Teilnehmer des Experimentes auf einer fünfstufigen Likertskala angeben, wie zufrieden sie mit dem Verlauf der Suche und ihren Ergebnissen waren. Bei gleichbleibendem Sucherfolg wäre ein Suchsystem vorzuziehen, bei dem die Suchenden insgesamt mit der Benutzung zufriedener waren.

Von den Teilnehmern, die Suchunterstützung erfahren hatten, antworteten 7, sie seien mit dem Verlauf der Suche zufrieden oder sehr zufrieden gewesen,

**Tabelle 8.3.** Benutzeraktionen während der Aufgaben (Unterstützung nur für Aufgaben 1 und 2)

Aktion	Ø (Stdabw.)	Ø (Stdabw.)	Welch-Test		
	m. Unterst.	o. Unterst.	<i>t</i>	<i>df</i>	<i>p</i>
<b>Grundlegende Aktionen</b>					
Anfrage ausführen	46,50 (9,76)	51,33 (19,89)	-0,82	17,01	0,423
Details betrachten	79,17 (45,47)	51,67 (28,50)	-1,78	18,49	0,092
Dokument speichern	38,17 (14,86)	23,25 (14,18)	2,52	21,95	0,019
<b>Vorschläge</b>					
Vorschläge anfordern	8,33 (3,11)	— (—)	—	—	—
Vorschlag ausführen	9,50 (4,01)	— (—)	—	—	—
<b>Fortgeschrittene Aktionen</b>					
Begriffe extrahieren	10,33 (6,53)	3,75 (3,89)	3,02	18,00	0,007
Thesaurus benutzen	4,91 (4,60)	0,50 (1,73)	2,99	12,58	0,011
Anfrage aus anderem Werkzeug ändern	1,17 (1,75)	0,92 (1,56)	0,37	21,73	0,716
	4,67 (2,87)	2,33 (2,31)	2,19	21,03	0,039
<b>Fortg. Aktionen (Aufg. 1)</b>	3,42 (2,39)	1,16 (1,34)	2,84	17,26	0,011
<b>Fortg. Aktionen (Aufg. 2)</b>	4,25 (1,24)	1,33 (1,44)	3,24	16,52	0,005
<b>Fortg. Aktionen (Aufg. 3)</b>	2,67 (1,78)	1,25 (1,29)	2,48	21,52	0,021

und 5 sie seien weder zufrieden noch unzufrieden. Bei den Teilnehmer ohne Suchunterstützung gaben 3 an, sie seien zufrieden oder sehr zufrieden gewesen, 4 weder noch und 5 antworteten, sie seien unzufrieden oder sehr unzufrieden mit dem Verlauf der Suche. Der Exakte Test nach Fisher ergab mit  $p = 0,067$  eine positive Korrelation zwischen den beiden Variablen Unterstützung und Zufriedenheit. Da  $p > 0,05$  ist, konnte die Null-Hypothese jedoch nicht verworfen werden, und es muss davon ausgegangen werden, dass keine Abhängigkeit vorliegt. Pearsons Chi-Quadrat-Test ergab eine Korrelation mit  $p = 0,040 < 0,05$ , jedoch ist dieser aufgrund der geringen Anzahl an Beobachtungen als unzuverlässig anzusehen.

Bei der subjektiven Einschätzung der Ergebnisse herrschte hingegen ein klareres Bild. 9 der Teilnehmer mit Unterstützung waren zufrieden oder sehr zufrieden mit ihren Ergebnissen, 3 waren weder zufrieden noch unzufrieden. Von den Teilnehmern ohne Unterstützung erklärten sich nur 2 als zufrieden oder sehr zufrieden, 5 waren weder zufrieden noch unzufrieden und ebenso viele waren entweder unzufrieden oder sehr unzufrieden mit ihren Ergebnissen. Der Exakte Test nach Fisher ergab hier mit  $p = 0,004$  eine positive Korrelation zwischen den Variablen, so dass davon ausgegangen werden kann, dass eine Abhängigkeit zwischen Unterstützung und Zufriedenheit mit den Ergebnissen existiert. Die quantitative Auswertung der Suchergebnisse bestätigte diesen Zusammenhang.

## 1. Fragestellung

Die erste untersuchte Frage war, ob die Suchenden, die Unterstützung durch das Vorschlags-Werkzeug erfahren hatten, bei ihren Suchen erfolgreicher wa-

ren. Wie in Tabelle 8.3 zu sehen ist, stellten Suchende in der Gruppe mit Unterstützung weniger Anfragen als jene in der Gruppe ohne (46,5 zu 51,33), aber der Unterschied war nicht signifikant (unter Benutzung des Welch-Test). Sie betrachteten mehr Dokumentdetails (79,17 zu 51,67), aber nicht signifikant mehr. Der Unterschied bei den gespeicherten Dokumenten (38,17 mit Unterstützung zu 23,25 ohne) ist signifikant mit  $p = 0,019$ , es ist aber nicht klar, ob dies einen guten Indikator für tatsächlichen Sucherfolg darstellt.

Daher wurden Relevanzurteile von unabhängigen Begutachtern herangezogen. Die Anzahl der durchschnittlich gespeicherten relevanten Dokumente für alle drei Aufgaben kann aus Tabelle 8.4 entnommen werden. Benutzer in der Gruppe, die Unterstützung erhielt, speicherten signifikant mehr relevante Dokumente für alle Aufgaben zusammen ( $p = 0,002$ ), sowie für die Aufgaben „Gesundheit“ und „Plagiate“ ( $p = 0,011$  und  $p = 0,022$ ). Für die Suchaufgabe „Java“ konnte kein signifikanter Unterschied im Sucherfolg festgestellt werden. Dies kann aber auch auf die geringe Zahl für die Aufgabe relevanter Dokumente in der benutzten Kollektion zurückzuführen sein.

Betrachtet man die gespeicherten Ergebnisse der Suchenden nicht isoliert, so muss man, wie von Järvelin [JÄRVELIN 2009] vorgeschlagen, auch die Kosten der Benutzer, die zur Erzielung des Sucherfolgs notwendig waren, in die Bewertung mit einbeziehen. Hierzu kann man das Verhältnis von relevanten Ergebnissen (Nutzen) zu der Anzahl der Anfragen oder zur Anzahl inspizierter Dokumente betrachten (Kosten). Dieses drückt aus, wieviel Aufwand der Suchende betreiben musste, um ein einzelnes relevantes Ergebnis zu erhalten.

Allerdings fehlen dazu Kostenfaktoren der verschiedenen Aktivitäten. Es lässt sich davon ausgehen, dass der Aufwand für eine Suchanfrage erheblich höher ist als für andere Aktionen des Benutzers, insbesondere aufgrund des Zeitaufwands für das Warten auf die Ergebnisse und für das Scannen der Resultatliste. Da die tatsächlichen Kosten allerdings unbekannt sind, lassen sich hier nur schwer Schlüsse ziehen. Betrachtet man allein das Verhältnis von relevanten Ergebnissen zu gestellten Anfragen, so zeigt sich ein klarer Unterschied zwischen den unterstützten und nicht unterstützten Benutzern (0,58 relevante Dokumente pro Anfrage zu 0,29,  $p = 0,01$ ). Bezieht man alle Aktionen mit ein und geht von einheitlichen Kosten für alle Aktionen aus, so führten die Benutzer mit Unterstützung im Schnitt zwar mehr Aktionen aus, aber im Verhältnis zum Nutzen waren sie dabei noch immer effizienter als die Benutzer ohne Unterstützung (0,14 relevante Dokumente pro Aktion zu 0,09,  $p = 0,032$ ).

Es scheint insgesamt, dass die Suchvorschläge in der präsentierten Form Nutzern grundsätzlich helfen können, erfolgreicher und mit tendenziell weniger Kosten zu suchen. Allerdings hilft diese Unterstützung offenbar nicht bei allen Aufgaben.

**Tabelle 8.4.** Von Suchenden gespeicherte relevante Dokumente

Aufgabe	Ø (Stdabw.)		Welch-Test		
	m. Unterstützung	o. Unterstützung	<i>t</i>	<i>df</i>	<i>p</i>
Gesundheit	8,25 (5,64)	3,00 (2,95)	2,86	16,61	0,011
Plagiate	14,92 (5,30)	8,25 (7,62)	2,49	19,63	0,022
Java	2,42 (1,78)	1,42 (1,62)	1,44	21,81	0,165
Gesamt	25,58 (9,61)	12,67 (8,95)	3,41	21,89	0,002
pro Anfrage	0,58 (0,24)	0,29 (0,26)	2,80	21,92	0,010
pro Aktion	0,14 (0,04)	0,09 (0,06)	2,30	19,55	0,032

## 2. Fragestellung

Die zweite betrachtete Fragestellung behandelte die Aktionen, die Benutzer während ihrer Suche durchführen. Die Ausgangshypothese war, dass Benutzer, die Suchvorschläge erhalten, die fortgeschrittenen Werkzeuge und erweiterten Suchoptionen von DAFFODIL signifikant häufiger benutzen, als Benutzer, die keine Suchvorschläge erhalten.

Die Evaluation der Suchhistorien zeigte, dass es keine statistisch signifikanten Unterschiede zwischen beiden Gruppen bezüglich der Aktionen zur Veränderung der Anfrage durch Hinzufügen oder Wegnehmen von Suchbegriffen oder booleschen Operatoren, durch Korrektur der Schreibweise, Wiederverwendung einer früheren Anfrage, Ersetzung eines oder mehrerer Suchbegriffe oder Ersetzung der gesamten Anfrage gab. Allerdings waren die Unterschiede in den beiden letzten Fällen nur marginal nicht signifikant: Benutzer, die *keine* Vorschläge erhielten, ersetzten Anfragebegriffe durchschnittlich 16,92 mal pro Suche im Gegensatz zu 11,08 mal ( $p = 0,055$ ) und ersetzten die gesamte Anfrage im Schnitt 1,31 mal pro Suche zu 0,83 mal ( $p = 0,062$ ).

Andererseits verwendeten Benutzer, die Vorschläge erhielten, signifikant häufiger das Jahresfeld zu Einschränkung der Anfrage, wechselten zwischen Freitext- und Titelfeld oder benutzten das Autorenfeld (7,08 zu 0,92,  $p = 0,001$ ). Obwohl beide Gruppen die gleiche schriftliche Einführung in das Suchformular des Programms erhielten und alle Suchfelder während des gesamten Experiments durchgehend sichtbar waren, war zu beobachten, dass die Benutzer ohne Vorschläge sich grundsätzlich auf ein Suchfeld beschränkten, welches sie zu Beginn für ihre erste Anfrage gewählt hatten (entweder „Titel“ oder „Freitext“). Bei diesem Suchfeld blieben sie dann für alle Suchen während aller drei Aufgaben, auch wenn die Ergebnisse nicht zufriedenstellend waren.

Ein Blick auf die Benutzung der fortgeschrittenen Werkzeuge während des Experiments liefert ein ähnliches Bild. Keine signifikanten Unterschiede zeigten sich bei der Benutzung des Thesaurus zum Nachschlagen von allgemeineren oder spezielleren Begriffen, von Synonymen oder Wortdefinitionen. Benutzer des Vorschlags-Werkzeug benutzten aber signifikant häufiger die Extraktionsmöglichkeiten von DAFFODIL, um neue Begriffe oder häufige Autoren

### 8.3. LOGAUSWERTUNGEN

---

aus Suchresultaten oder gespeicherten, relevanten Dokumenten zu extrahieren (4,91 zu 0,5 mal,  $p = 0,01$ ). Sie nutzten auch mehr Begriffe aus anderen Werkzeugen (wie der Darstellung verwandter Begriffe, den extrahierten Begriffen, dem Thesaurus oder der Klassifikation) direkt in ihrer Suche (4,67 zu 2,33 mal,  $p = 0,039$ ).

#### 3. Fragestellung

Die letzte untersuchte Frage war, ob Benutzer des Vorschlags-Werkzeug die fortgeschrittenen Suchaktionen auch unabhängig weiter nutzen würden, nachdem sie diese für vorangegangene Aufgaben als Vorschläge erhalten haben. Obgleich man diese Frage eigentlich nur durch ein Langzeitexperiment beantworten kann, wurde ein erstes Experiment im Rahmen dieser Studie durchgeführt. Für die Benutzer der Gruppe mit Unterstützung wurden die Vorschläge während der letzten Aufgabe abgeschaltet, so dass keine Vorschläge angezeigt wurden.

Man kann bei dieser Gruppe einen klaren Rückgang der Benutzung fortgeschrittener Aktionen zwischen Aufgabe 2 und Aufgabe 3 erkennen (siehe Tabelle 8.3), aber sie verwendeten trotzdem noch signifikant häufiger solche fortgeschrittenen Aktionen als Benutzer der Gruppe ohne Vorschläge (2,67 zu 1,25,  $p = 0,02$ ). Außerdem war auch die Benutzung zusätzlicher Suchfelder höher bei den Teilnehmern der Gruppe, die zuvor Vorschläge erhalten hatte (1,75 zu 0,25,  $p = 0,004$ ).

Das lässt sich als ein leichter Lerneffekt interpretieren, der aus dem Erhalt von situativ passenden Suchvorschlägen resultiert. Dieser führte dazu, dass Benutzer zuvor vorgeschlagene Aktionen später unabhängig für eine andere Aufgabe einsetzten. Natürlich muss man dabei berücksichtigen, dass die drei Aufgaben in direkter Folge ausgeführt wurden, daher bleibt es ungewiss, ob die Benutzer das erworbene Wissen für spätere Aufgaben wieder abrufen könnten. Andererseits hätte sich in einer Langzeitstudie, bei der Suchende das Vorschlags-Werkzeug über einen langen Zeitraum hinweg einsetzen, möglicherweise auch ein ausgeprägter Effekt beobachten lassen.

### 8.3. Logauswertungen

Neben den zuvor beschriebenen Evaluationen der Vorschlagssoftware wurden auch einige Auswertungen des Benutzerverhaltens auf Basis des Loggingframeworks (siehe Abschnitt 5.4) durchgeführt. Zu diesem Zweck wurden sowohl Suchsitzungen ohne Gebrauch des Vorschlagswerkzeugs, als auch solche mit Gebrauch des Vorschlagswerkzeugs betrachtet. Ausgangsbasis der Auswertungen waren die in 9 Jahren des Systemeinsatzes (2001-2009) gesammel-



ten Aktivitätslogs aus insgesamt 78.877 Suchsitzungen mit dem DAFFODIL-System und 658.576 einzelnen Logereignissen.

Diese Suchsitzungen und Logereignisse wurden bereinigt, um technische Fehler, Funktionstests und nicht verwertbare Sitzungen zu entfernen:

- 41.707 Logereignisse, die durch Fehler beim Logging keiner Sitzung zugeordnet wurden,
- 65.758 Logereignisse in 37.386 Suchsitzungen, die ohne Ereignisse für Start und Ende nur ein Ereignis umfassten,
- 4.890 Logereignisse aus designierten Funktionstests,
- 53.902 Logereignisse, die Beginn und Ende einer Suchsitzung anzeigten, sowie
- 245.450 Logereignisse, die fehlerhaft oder nicht kategorisiert wurden,

fielen durch diese Bereinigung aus den betrachteten Daten heraus.

**Tabelle 8.5.** Häufigste benutzte Dienste oder Werkzeuge

Werkzeug	Alle Sitzungen		ohne Vorschläge		mit Vorschlägen	
	Anzahl	%-Anteil	Anzahl	%-Anteil	Anzahl	%-Anteil
Suche	131 508	53,27	126 447	53,04	5 061	59,77
Handbibliothek	57 363	23,24	56 422	23,67	941	11,11
Konferenzen	12 496	5,06	12 408	5,20	88	1,04
Koautoren	9 607	3,89	9 275	3,89	332	3,92
Thesaurus	8 048	3,26	7 835	3,29	213	2,52
Hilfe	3 010	1,22	2 950	1,24	60	0,71
Journale	2 124	0,86	2 117	0,89	7	0,08
Zwischenablage	1 180	0,48	521	0,22	659	7,78
Export	568	0,23	555	0,23	13	0,15
Vorschläge	455	0,18	0	0,00	455	5,37
Annotationen	234	0,09	231	0,10	3	0,04
andere	20 276	8,21	19 640	8,24	636	7,51

**Tabelle 8.6.** Häufigste Ereignisse auf konzeptueller Ebene

Konzept	Alle Sitzungen		ohne Vorschläge		mit Vorschlägen	
	Anzahl	%-Anteil	Anzahl	%-Anteil	Anzahl	%-Anteil
Search	88 027	35,66	86 275	36,19	1 752	20,69
Display	45 617	18,48	43 110	18,08	2 507	29,61
Inspect	35 758	14,48	33 923	14,23	1 835	21,67
Store	29 236	11,84	28 215	11,84	1 021	12,06
Annotate	9 396	3,81	9 396	3,94	0	0,00
Help	4 583	1,86	3 683	1,54	900	10,63
Author	991	0,40	991	0,42	0	0,00
andere	33 261	13,47	31 808	13,76	453	5,35

Von den verbleibenden 41.368 Sitzungen fanden 173 unter Benutzung des Vorschlagswerkzeuges statt. Dabei betrug die durchschnittliche bereinigte Sitzungslänge 5,9 Aktionen. Bei den durch Vorschläge unterstützten Sitzungen

### 8.3. LOGAUSWERTUNGEN

---

betrug die durchschnittliche Sitzungslänge 48,95 Aktionen (dabei ist zu berücksichtigen, dass diese Zahlen durch umfangreiche Sitzungen im Rahmen der Benutzerexperimente zustandekommen). Die durchschnittliche Dauer einer Suchsitzung in Sekunden betrug 394,69 (699,31 bei Beschränkung auf Sitzungen mit Unterstützung).

In den Tabellen 8.5 und 8.6 sind einige grundlegende Daten über die geloggen Suchsitzungen zusammengestellt. Es ist jeweils die Anzahl der benutzten Werkzeuge und der Ereignisse auf konzeptioneller Ebene, sowie ihr Anteil an allen Werkzeugaktionen und Ereignissen angegeben. Dabei erfolgt die nach Benutzung des Vorschlagswerkzeugs unterteilte Angabe nur der Vollständigkeit halber. Aufgrund der kleinen Zahl an Sitzungen mit dem Vorschlagswerkzeug, von denen ein großer Anteil im Rahmen von Experimenten mit vorgegebenen Bedingungen stattfanden, ist keine direkte Vergleichbarkeit gegeben.

Nur auf den ersten Blick überraschend ist, dass in den Sitzungen ohne Vorschläge zwar häufiger das Suchwerkzeug zum Einsatz kam, aber weniger Suchen durchgeführt wurden. Tatsächlich korrespondieren aber Werkzeuge und konzeptionelle Ereignisse nicht direkt. Das Suchwerkzeug wird beispielsweise für Suchen ebenso genutzt wie für „Display“-Ereignisse (Sortieren und Extrahieren). Während also in den Sitzungen ohne Vorschläge relativ gesehen weniger gesucht wurde, verwendeten die Benutzer das Suchwerkzeug für konzeptionell andere Aktionen. Der hohe Anteil an Hilfe-Ereignissen für Sitzungen mit dem Vorschlagswerkzeug in Tabelle 8.6 spiegelt die Benutzung des Werkzeuges selbst wieder, während sich in den Sitzungen ohne Vorschlagswerkzeug die Hilfe-Ereignisse auf die Verwendung der Online-Dokumentation beschränkten.

Wie bereits in Abschnitt 6.2 ausgeführt, sind die Logereignisse auf der Ebene der Dienste am geeignetsten, um aus ihnen konkrete, von einem menschlichen Nutzer umsetzbare Vorschläge abzuleiten. Für die Zwecke der Auswertung wurde neben der Ebene der Dienste oder Werkzeuge, insbesondere die konzeptionelle Ebene betrachtet (siehe Abschnitt 5.4). Die Tabellen 8.7 und 8.8 zeigen die häufigsten Dreierketten von Logereignissen auf den angesprochenen Abstraktionsebenen, die in den ausgewerteten Suchsitzungen mindestens 250-mal auftraten. Dabei sind Dreierketten jeweils drei direkt hintereinander folgende Logereignisse innerhalb einer Suchsitzung (entweder auf der Dienstebene oder der konzeptuellen Ebene). Wie bei Jansen und McNeese [JANSEN und MCNEESE 2005] wurden also auch hier Ausschnitte von Suchsitzungen mit zwei Übergängen betrachtet.

Daneben wurden auch Zweierketten von Ereignissen auf der Ebene der Dienste untersucht [BODNARIU 2009]. Die Ergebnisse der Auswertung sind in Tabelle 8.9 zusammengefasst. Diese zeigt, welche Ereignisarten besonders häufig auf andere Ereignisarten folgen. So kann man sehen, dass auf eine Inspektion von Dokumentdetails häufiger ein Speicherereignis als eine erneute Suche folgt, am häufigsten aber eine weitere Inspektion von Dokumentdetails.

**Tabelle 8.7.** Vorkommen von Dreierketten (konzeptuelle Ereignisse)

1. Ereignis	2. Ereignis	3. Ereignis	Häufigkeit	%-Anteil	
annotate	navigate	annotate	1 660	1,55	
	navigate	navigate	443	0,41	
	annotate	annotate	281	0,26	
display	display	display	16 252	15,17	
		search	515	0,48	
		store	294	0,27	
		search	display	1 100	1,03
		store	store	687	0,64
			search	226	0,21
		inspect	inspect	467	0,44
help	help	help	2 416	2,25	
	search	display	354	0,33	
	inspect	inspect	8 499	7,93	
inspect		store	694	0,65	
		search	490	0,46	
		store	inspect	858	0,80
		search	display	472	0,44
			search	293	0,27
		navigate	navigate	3 658	3,41
			annotate	932	0,87
navigate		display	367	0,34	
		navigate	2 120	1,98	
		search	986	0,92	
		search	search	26 568	24,79
			display	1 029	6,52
			store	404	0,38
		display	display	6 983	6,52
search		store	801	0,75	
		search	618	0,58	
		store	1 082	1,01	
			search	279	0,26
		navigate	navigate	343	0,32
		store	store	13 983	13,05
			search	1 175	1,10
		inspect	store	793	0,74
		search	display	786	0,73
			search	526	0,49
		store	391	0,36	
store		search	13 983	13,05	
		inspect	793	0,74	
		search	786	0,73	
			search	526	0,49
		store	391	0,36	

### 8.3. LOGAUSWERTUNGEN

**Tabelle 8.8.** Vorkommen von Dreierketten (Werkzeuge)

1. Werkzeug	2. Werkzeug	3. Werkzeug	Häufigkeit	%-Anteil		
Extraktion	Suche	Suche	180	0,07		
Export	Export	Export	238	0,10		
Handbibliothek	Handbibliothek	Handbibliothek	17 965	7,33		
		Suche	1 330	3,63		
	Suche	Suche	8 905	3,63		
		Handbibliothek	301	0,54		
		Konferenzen	185	0,08		
	Hilfe	Hilfe	159	0,06		
		Thesaurus	158	0,06		
		Hilfe	2 280	0,93		
	Journale	Journale	Journale	1 032	0,42	
	Koautoren	Koautoren	Koautoren	8 337	3,40	
Konferenzen	Handbibliothek	Suche	1 557	0,64		
		Handbibliothek	925	0,45		
	Konferenzen	Konferenzen	1 113	0,45		
		Suche	206	0,38		
		Handbibliothek	183	0,07		
	Suche	Suche	Suche	558	0,23	
			Suche	65 599	26,77	
		Handbibliothek	Handbibliothek	2 928	1,34	
			Zwischenablage	543	0,22	
			Extraktion	537	0,22	
			Konferenzen	195	0,08	
			Thesaurus	160	0,07	
			Handbibliothek	Handbibliothek	2 053	0,84
				Suche	898	0,37
			Zwischenablage	Zwischenablage	Zwischenablage	616
	Thesaurus	256			0,10	
	Extraktion	Vorschläge		168	0,07	
		Suche		167	0,07	
Koautoren	Koautoren	162		0,07		
	Konferenzen	158		0,06		
Thesaurus	Thesaurus	6 212		2,53		
	Suche	219		0,09		
Vorschläge	Suche	Suche		216	0,09	
Zwischenablage	Zwischenablage	Zwischenablage		5 205	2,12	
		Suche	333	0,14		
	Suche	Suche	334	0,14		
		Zwischenablage	219	0,09		

**Tabelle 8.9.** Häufigste Folgeereignisse

Ereignis	Folgeereignisse
annotate	navigate (50%) search (30%) annotate (14%)
authoring	search (39%) authoring (38%) store (15%)
display	display (83%) search (7%) store (5%)
help	help (79%) search (16%)
inspect	inspect (77%) store (12%) search (7%)
navigate	navigate (54%) annotate (35%) search (6%)
search	search (65%) display (28%) store (4%)
store	store (80%) search (11%) inspect (5%)

Insgesamt fällt bei der Betrachtung der Auswertungen auf, wie häufig zwei oder drei gleichartige Ereignisse mehrfach hintereinander auftreten. Dies ist auf unterschiedliche Gründe zurückzuführen. Während z. B. mehrere aufeinanderfolgende Speicherereignisse darauf zurückzuführen sein können, dass ein Benutzer aus einer Ergebnisliste zunächst eine Reihe vielversprechender Informationsobjekte in die Zwischenablage oder seine Handbibliothek ablegt, bevor er sich die Details der Dokumente ansieht, sind andere Dopplungen auf Fehler im System oder im Benutzerverhalten zurückzuführen (z. B. doppelte Anfragen mit gleichen Suchbegriffen). Daher wurden zusätzlich in [BODNARIU 2009] auch Ketten von Ereignissen unter Auslassung gleichartiger, aufeinander folgender Ereignisse betrachtet. Die Tabelle 8.10 zeigt solche Ketten für Nutzeraktivitäten. Dabei steht jeder Eintrag für eine Reihe gleichartiger Ereignisse in Folge.

**Tabelle 8.10.** Vorkommen von „echten“ Zweier-Übergängen (Aktionen)

1. Aktion	2. Aktion	3. Aktion	Häufigkeit
Browsing	Suche	Browsing	19 303
Suche	Sortieren	Suche	18 043
Sortieren	Suche	Sortieren	12 642
Exportieren	Suche	Exportieren	8 428
Details	Suche	Sortieren	7 096
Thesaurus	Suche	Sortieren	5 524
Koautoren	Suche	Sortieren	5 450
Objekt angelegt	Suche	Sortieren	778
Konferenzen	Suche	Sortieren	606
Autoren extrahiert	Terme extrahiert	Konferenzen extrahiert	565
Terme extrahiert	Autoren extrahiert	Journale extrahiert	429
Zitationssuche	Export	Zitationssuche	298
Speichern	Details	Suche	288
Journale extrahiert	Autoren extrahiert	Terme extrahiert	255
Konferenzen extrahiert	Journale	Autoren extrahiert	197
Annotation	Suche	Sortieren	146

Schließlich wurden auch die in Experiment 8.2 aufgezeichneten Logsitzungen untersucht. Dabei wurden die 24 unterstützten und 24 nicht unterstützten Sitzungen für die jeweils ersten beiden Suchaufgaben der Probanden betrachtet. Für die Sitzungen standen dabei mit Ausnahme des Vorschlagswerkzeugs die gleichen Werkzeuge zur Verfügung. Die Tabellen 8.11 und 8.12 zeigen die häufigsten Ereignisfolgen aus den nicht unterstützten, bzw. den unterstützten Sitzungen mit einer relativen Häufigkeit von mindestens 1% an allen Ereignisfolgen. Zahlen zur Häufigkeit einzelner Aktionen und Werkzeugbenutzungen wurden bereits im vorangegangenen Abschnitt genannt.

Die Ausführung eines Vorschlags führte am häufigsten zu einer Suche mit anschließender Sortierung des Ergebnis (29), zum Betrachten zweier Dokumentdetails in Folge (19) und zum Übernehmen eines Vorschlagsergebnisses in die Anfrage, gefolgt von der Ausführung eines weiteren Vorschlags (17). Während die Unterschiede bei den meisten der Dreierfolgen nicht signifikant sind, kann

## 8.4. ZUSAMMENFASSUNG

man sehen, dass sich nur bei den unterstützten Sitzungen häufige Nutzung des Thesaurus und der Extraktionsfunktion findet. Dies bestätigt die in Tabelle 8.3 gezeigten Zahlen. Die Benutzungen von Thesaurus und Begriffsextraktion waren dabei für die unterstützten Nutzer eingebunden in wiederkehrende Muster.

**Tabelle 8.11.** Dreierketten in Experiment 2 (nicht unterstützt)

1. Aktion	2. Aktion	3. Aktion	Anzahl	%-Anteil
Details	Details	Details	459	20,53
		Speichern	218	9,53
		Suche	85	9,75
		Sortieren	28	3,80
		Sortieren	18	1,25
		Speichern	152	6,79
		Suche	35	1,56
		Sortieren	87	3,89
		Sortieren	169	7,56
		Sortieren	92	4,11
Speichern	Details	Suche	38	1,69
		Details	169	7,56
		Sortieren	30	1,34
Suche	Sortieren	Details	166	7,42
		Suche	123	5,50
		Sortieren	77	3,44
	Suche	Suche	64	2,86

## 8.4. Zusammenfassung

Die durchgeführten Evaluationen des strategischen Vorschlagswerkzeugs zeigen den Nutzen eines Systems, das Endnutzern situationsabhängig zielführende Taktiken und Strategeme zur Ausführung anbieten kann. Im Umgang mit DAFFODIL unerfahrene Benutzer waren mit Hilfe des Vorschlagswerkzeugs in der Lage, Suchaufgaben signifikant erfolgreicher zu absolvieren, gemessen an der Zahl relevanter Dokumente, die sie innerhalb einer vorgegebenen Zeit gefunden und gespeichert haben. Ebenso nutzten durch Vorschläge unterstützte Suchende mehr der in Kapitel 5 vorgestellten weiterführenden Werkzeuge zur Umsetzung ihrer Suchstrategien und griffen auf andere Taktiken zurück als nicht unterstützte Suchende.

Mit der Benutzung des Vorschlagswerkzeugs zeigten sich sowohl unerfahrene wie erfahrene Benutzer DAFFODILS zufrieden. Selbst erfahrene Benutzer hoben hervor, dass sie durch Vorschläge auf Optionen hingewiesen wurden, die sie entweder nicht oder erst nach längerem Probieren selbstständig eingesetzt hätten, und die sie bei ihrer Suche voran brachten. Eine große Mehrheit der Benutzer entdeckte Taktiken oder Möglichkeiten, die ihnen neu waren oder an deren Anwendbarkeit sie in der konkreten Situation nicht gedacht hätten. Die

**Tabelle 8.12.** Dreierketten in Experiment 2 (unterstützt)

1. Aktion	2. Aktion	3. Aktion	Anzahl	%-Anteil	
Details	Details	Details	355	22,38	
		Speichern	141	8,89	
		Suche	52	3,27	
	Speichern	Speichern	Details	94	5,92
			Suche	20	1,26
			Speichern	19	1,19
			Sortieren	38	2,39
		Suche	Suche	20	1,26
			Details	73	4,61
			Sortieren	60	3,78
Sortieren	Vorschläge	Suche	17	1,07	
		Vorschlag ausführen	40	2,52	
		Details	73	4,61	
Speichern	Suche	Sortieren	60	3,78	
		Suche	17	1,07	
		Details	73	4,61	
Suche	Sortieren	Vorschlag ausführen	40	2,52	
		Details	116	7,31	
		Sortieren	21	1,32	
Suche	Sortieren	Details	76	4,79	
		Suche	69	4,35	
		Vorschläge anfordern	43	2,71	
		Extrahieren	23	1,45	
		Speichern	17	1,07	
		Sortieren	44	2,77	
		Suche	36	2,27	
	Vorschläge	Suche	Vorschlag ausführen	20	1,26
			Thesaurus	18	1,13
			Thesaurus	18	1,13
Thesaurus	Suche	Sortieren	29	1,82	
		Details	19	1,19	
		Details	19	1,19	
Vorschlag ausführen	Anfrage übernehmen	Vorschlag ausführen	17	1,06	
		Suche	27	1,70	
		Details	19	1,19	
Vorschläge	Vorschlag ausführen	Suche	27	1,70	
		Details	19	1,19	
		Anfrage übernehmen	16	1,00	

Art und Weise der Präsentation erwies sich als durchweg erfolgreich, kleinere Mängel der Benutzerschnittstelle waren leicht behebbar.

Ob Suchende durch das Digital-Library-System beim Erwerb prozeduralen Suchwissens unterstützt werden können, das sie später auch unabhängig einsetzen können, bleibt offen und müsste in einer langfristig angelegten Studie in Kombination mit der Scaffolding-Erweiterung des Systems untersucht werden. Die ersten Ergebnisse der Evaluation zeigten zumindest auf, dass Benutzer des Vorschlagssystems auch in einer nicht unterstützten Aufgabe signifikant erfolgreicher waren als Benutzer, die nie Unterstützung durch Vorschläge erfahren hatten.





## Kapitel 9.

### Zusammenfassung und Ausblick

Ausgangsproblem dieser Arbeit war das mangelnde strategische oder prozedurale Suchwissen vieler Endbenutzer von Informationssystemen. Zahlreiche Nutzerstudien und Experimente, sowie Untersuchungen von interventionsfreien, nutzerinitiierten Suchen haben gezeigt, dass Endnutzer durch fehlende Suchexpertise auf ineffiziente und ineffektive Suchtaktiken und Suchstrategien zurückgreifen. Ausgehend davon sollte untersucht werden, wie ein Suchsystem unerfahrene Benutzer bei der Bildung von Suchstrategien für ihr Informationsproblem unterstützen kann.

Dazu wurde eine Reihe von Modellen zur Beschreibung des menschlichen Informationssuchverhaltens untersucht und verglichen. Im Vordergrund standen dabei solche Modelle, die Suche als einen Prozess oder insbesondere als Problemlösungsprozess beschreiben. Zu Beginn des Prozesses steht eine Suchaufgabe als Informationsproblem, das es für den Benutzer zu erkennen und zu verstehen gilt. Durch Definition des Problems kann der Suchende eine Strategie zu seiner Lösung entwickeln bzw. dabei durch ein System unterstützt werden. Ein wichtiger Aspekt ist die Zerlegung des Problems in Teilprobleme oder Teilaufgaben.

Während Modelle der Informationssuche das Suchverhalten von Informationssystem-Nutzern auf einer sehr hohen Abstraktionsebene beschreiben, wurden daneben auch mehrere Klassifikationssysteme für Suchtaktiken, Heuristiken und Strategien betrachtet. Dabei zeigte sich das Klassifikationssystem nach Bates als sehr flexibel. Es wurde für andere Klassifikationen von Taktiken und Suchheuristiken erläutert, wie diese sich in die Bates'sche Klassifikation einordnen lassen.

Suchexperten verfügen über das strategische Wissen, um diese Taktiken situativ angemessen im Rahmen von erfolgreichen Suchstrategien einzusetzen. Einige aus der Literatur bekannte Suchstrategien von erfahrenen Informationssuchenden wurden vorgestellt. Suchnovizen hingegen sind meist nicht in der Lage, ein Suchproblem geeignet in Teilprobleme zu zerlegen, eine geeignete Suchstrategie für das Informationsproblem oder einzelne Teilprobleme zu entwickeln, oder innerhalb eines Suchprozesses auf angemessene Taktiken zurückzugreifen, um auf eine bestimmte Situation zu reagieren.

---

Um die Suchunterstützung auf die aktuelle Situation eines Benutzers abzustimmen, war es notwendig, Suchsituationen zu definieren. Es wurden hierbei sowohl Attribute betrachtet, die sich durch das System direkt erfassen lassen, als auch so genannte Benutzerparameter, die einen wichtigen aber nur schwierig oder gar nicht automatisch zu charakterisierenden Teil der Situation darstellen. Außerdem wurde ein Ordnungssystem für solche Benutzer- und Systemaktionen, die zu Änderungen von Suchsituationen führen können, vorgestellt.

Die Untersuchungen zur Benutzerunterstützung bei der Bildung und Durchführung von Suchstrategien und -taktiken fanden (mit einer Ausnahme) im Rahmen des DAFFODIL-Systems zur Suche in verteilten Digitalen Bibliotheken statt. Das System wurde im Rahmen der Arbeit um eine Reihe von Erweiterungen ergänzt. Neue Werkzeuge zur Planung und Durchführung von Strategien wurden implementiert: eine Zwischenablage als Repräsentation des Körbchens in Marcia Bates' *Berrypicking*-Strategie, eine stärker interaktive Ergebnisanzeige, die Durchführung bekannter Strategeme erleichtert, eine Visualisierung von Begriffsmengen mit Hilfe von Termwolken, ein Strategieframework, sowie eine grafische Suchhistorie basierend auf einem mehrschichtigen Logmodell. Die Oberfläche des Systems wurde mit aufgabenabhängigen Perspektiven in Form von *Tiled Panes* neu gestaltet und die externe Definition von Perspektiven durch XML-Dateien ermöglicht.

Basierend auf der parametrisierten Beschreibung von Suchsituationen und unter Einsatz bekannter Taktiken, die auf das Suchsystem DAFFODIL bezogen wurden, ist ein Vorschlagssystem für Suchtaktiken, -strategeme und Suchstrategien entwickelt worden. Es benutzt die Technik des fallbasierten Schließens, um aus den erfolgreichen Aktionen anderer Benutzer ein angemessenes Vorgehen für die aktuelle Situation des Suchenden vorzuschlagen. Daneben wurden auch Vorschläge basierend auf den Suchergebnissen und auf Suchpfaden früherer Benutzer implementiert und untersucht, sowie die Kombination der verschiedenen Vorgehensweisen diskutiert.

In einer zweiteiligen Evaluation des Vorschlagssystems wurden zunächst das grundsätzliche Konzept und die Benutzerschnittstelle mit positivem Ergebnis untersucht. In einem Folgeexperiment mit 24 Benutzern sind drei Hypothesen untersucht worden, die sich aus der Ausgangsfragestellung dieser Arbeit ergaben:

1. Sind Suchende mit Unterstützung erfolgreicher, d. h. finden sie mehr relevante Dokumente?
2. Benutzen Suchende mit Unterstützung häufiger fortgeschrittene Taktiken oder Werkzeuge?
3. Helfen situativ angepasste Vorschläge auch bei späteren Suchen ohne Unterstützung?

Mit einigen Abstrichen konnten dabei alle Fragen positiv beantwortet werden. Insbesondere zeigten sich für zwei der drei bearbeiteten Aufgaben signifikante Verbesserungen bezüglich der innerhalb der Bearbeitungszeit gefundenen Dokumente. Darüberhinaus griffen Suchende verstärkt auf die erweiterten Funktionen oder Optionen des Suchsystems DAFFODIL zurück. Aus den erfolgreichen Suchaktionen früherer Benutzer konnten also Vorschläge abgeleitet werden, die auch wenig erfahrenen Nutzern den zielführenden Einsatz des Systems ermöglichten.

Um das Vorschlagssystem in einen größeren Zusammenhang einer geführten Suchunterstützung einzubetten, wurde das erprobte Lehrkonzept des software-realisierten Scaffoldings gewählt. Anhand einer beispielhaften Aufgabe wurde prototypisch als *Proof of Concept* Scaffolding für die Suche in DAFFODIL umgesetzt. Dabei wird einem unerfahrenen Suchenden aufgezeigt, wie sich eine Suchaufgabe in Teilprobleme gliedern lässt und wie man die Werkzeuge des Systems zur Planung und Organisation der Suche einsetzen kann. Das entwickelte Strategieframework zur Beschreibung und automatischen Umsetzung einfacher Suchstrategeme wurde ebenfalls beispielhaft in den Scaffolding-Prototypen integriert. Mit Hilfe des modularen Strategiesystems wurde ein Werkzeug implementiert, das eine existierende Dokumentenliste um zu diesen Dokumenten ähnliche ergänzt.

Der Wert der Umsetzung liegt dabei primär in der Bereitstellung einer flexiblen Architektur mit definierbaren Unterstützungsniveaus. Diese Architektur erlaubt den Einsatz von Software-Scaffolding für beliebige wohldefinierte Suchaufgaben, die sich mit DAFFODIL bearbeiten lassen. Über die beispielhaft implementierten Scaffolds hinaus können verschiedene Unterstützungsmaßnahmen leicht in das System integriert und auf ihre Eignung für das Erlernen von prozeduralem Suchwissen hin untersucht werden.

Insgesamt wurden verschiedene Methoden aufgezeigt und umgesetzt, mit denen ein DL-System unerfahrene Benutzer bei der Bildung und bei der Durchführung einer Suchstrategie unterstützen kann. Die implementierten Werkzeuge und Frameworks erlauben darüber hinaus die Untersuchung weiterer Unterstützungsszenarien.

### 9.1. Ausblick

Weitere Untersuchungen auf Basis von Logpfaden können interessante Erkenntnisse über die Suchmuster von erfahrenen Suchenden liefern. Hierzu wäre es notwendig, Suchexperten zu identifizieren und die Suchen solcher Nutzer in den Logs zu isolieren. Statt Vorschläge basierend auf den häufigsten Suchpfaden aller Benutzer zu generieren, würde man sich diese dann aus den Suchen der erfahrenen Benutzer ableiten. Eine automatische Identifizierung von erfolgreichen Suchen könnte durch eine weitere Einschränkung der

zur Pfadgeneration benutzten Logsitzungen zusätzlich zu einer qualitativen Verbesserung der Vorschläge führen. Kombinationen der pfadbasierten Vorschläge mit taktischen Vorschlägen basierend auf CBR wurden in Abschnitt 6.2.3 diskutiert und bieten eine interessante Möglichkeit für weitere Untersuchungen.

Zusätzliche strategische und taktische Vorschläge können in das Vorschlagswerkzeug integriert werden, wobei die vorgestellte Literatur hier noch eine Reihe Möglichkeiten zur Erweiterung aufzeigt. Auch eine Umsetzung für andere Suchsysteme als DAFFODIL ist möglich, wie beispielsweise für den Suchdienst *The European Library*<sup>1</sup>, der browsenden und suchenden Zugriff auf die Bestände der europäischen Nationalbibliotheken bereitstellt, oder für die Online-Kataloge von Universitätsbibliotheken. Für die Websuche mit Google wurde bereits beispielhaft die generelle Eignung des Konzeptes für andere Suchsysteme aufgezeigt. Dabei sind jedoch gegebenenfalls die strategischen Vorschläge und die Parameter einer Suchsituation auf die Möglichkeiten des Suchsystems anzupassen.

Da es nicht immer leicht ist, Benutzer zur Abgabe von Feedback zu bewegen, insbesondere nachdem sie bereits den Nutzen durch einen Vorschlag erfahren haben, wäre es nützlich, zu untersuchen, ob positive oder negative Vorschlagsbewertungen durch implizites Feedback erlangt werden können. So könnte etwa das Speichern eines Dokumentes aus einer Ergebnisliste, die aus der Ausführung eines Vorschlags resultierte, als positiver Indikator für diesen Vorschlag betrachtet werden, während eine leere Ergebnismenge in Folge eines Vorschlags zu einer negativen Bewertung des Vorschlags für die aktuelle Benutzersituation führen würde.

Eine Langzeitstudie über den Einsatz des Systems fehlt derzeit noch. Diese könnte die Frage untersuchen, ob das CBR-basierte Vorschlagssystem den Benutzer beim *Erlernen* eines strategischen Umgangs unterstützt und dieser nach längerer Benutzung in der Lage ist, expertenähnliche Suchstrategien ohne Unterstützung des Systems einzusetzen. In einer solchen Studie kann das Vorschlagswerkzeug mit der vorgestellten Scaffolding-Erweiterung kombiniert werden, um Benutzer zunächst bei ihren Suchen mit strategischen Vorschlägen zu unterstützen und diese nach und nach durch Fading zurückzunehmen.

Weitere Scaffolds für zusätzliche Unterstützungsmaßnahmen im Rahmen des DAFFODIL-Systems können in das Scaffolding-Framework integriert werden. Für verschiedene Aufgaben ließen sich dann Unterstützungsmaßnahmen auf ihre Eignung für den Einsatz im Scaffolding vergleichend evaluieren.

Durch Integration der unterschiedlichen Werkzeuge und Methoden, die in dieser Arbeit vorgestellt wurden, lässt sich ein umfassend unterstützendes Informationssystem umsetzen, das neuen, unerfahrenen Benutzern durch Scaffolding beim Erwerb des notwendigen strategischen Wissens hilft und erfahre-

---

<sup>1</sup><http://search.theeuropeanlibrary.org>

ne Benutzer durch kontextspezifische Vorschläge und teilweise automatisierte Suchstrategeme bei der Durchführung schwieriger Suchaufgaben assistiert.



## **Teil IV.**

# **Anhänge und Verzeichnisse**





# Anhang A.

## DTDs

### Aufgabenbeschreibung (task.dtd)

Die DTD A.1 beschreibt eine durch Scaffolding innerhalb von DAFFODIL unterstützte Aufgabe. Dabei befinden sich alle sprachabhängigen Bildschirmausgaben und -texte innerhalb einer Sprachdatei und werden über Schlüssel referenziert. Zur Laufzeit von DAFFODIL wird eine der Benutzersprache entsprechende Sprachdatei benutzt, um Strings zu lokalisieren.

Die Aufgabenbeschreibung enthält Schritte für mindestens ein Unterstützungsniveau, wobei für jeden Schritt eine dedizierte Perspektive und eine Komponente für den Assistenten angegeben werden können.

### Perspektivenbeschreibung (perspective.dtd)

Die DTD für eine aufgabenabhängige Perspektive (A.2) beschreibt den Aufbau des DAFFODIL-Frontends in Gestalt von *Tiled Panes* (siehe Abschnitt 5.2). Die einzelnen Kacheln werden als *Panes* konfiguriert, die entweder senkrecht oder waagrecht in zwei weitere *Panes* unterteilt werden oder ein einzelnes Werkzeug enthalten können. Zusätzlich können Systemeinstellungen festgelegt werden, die nur für diese Perspektive gelten.

---

## Quelltext A.1 task.dtd

---

```
<!-- task description containing title and an explanation
      of the task in form of a key to language file,
      and a list of scaffold levels -->
<!ELEMENT task ( title, description, scaffold+ ) >
<!ATTLIST task id NMTOKEN #REQUIRED >
<!ELEMENT title ( #PCDATA ) >
<!ELEMENT description ( #PCDATA ) >

<!-- each scaffold level is described as a list of steps;
      the level is a numeric value between 0 (no support)
      and 10 (highest level of support) -->
<!ELEMENT scaffold ( step+ ) >
<!ATTLIST scaffold level NMTOKEN #REQUIRED >

<!-- a single step during a scaffolding process,
      with a display title and a help text given as
      property keys for the language file, and
      a component that is used within the assistant -->
<!ELEMENT step ( title, help, component? ) >
<!ATTLIST step id NMTOKEN #REQUIRED >
<!-- filename for the perspective to be used during this step -->
<!ATTLIST step perspective NMTOKEN #REQUIRED >

<!-- components of the assistant -->
<!ELEMENT component ( title, input* ) >
<!ATTLIST component id NMTOKEN #REQUIRED >
<!-- full path of the Java class -->
<!ATTLIST component src NMTOKEN #REQUIRED >

<!-- help text, given as a property key to a language
      file -->
<!ELEMENT help ( #PCDATA ) >

<!-- input determines from which component of other steps the output
      should be used a input in this step, the other steps can come
      earlier or later (if the user goes back some steps -->
<!ELEMENT input EMPTY >
<!ATTLIST input component NMTOKEN #REQUIRED >
<!ATTLIST input step NMTOKEN #REQUIRED >
```

---

---

**Quelltext A.2 perspective.dtd**

---

```
<!-- top level element, contains a description of tiled
      setup (pane) and optionally settings information -->
<!ELEMENT perspective ( pane, settings? ) >
<!ATTLIST perspective id CDATA #REQUIRED >

<!-- a pane is a single area of the perspective, it can
      contain splitplanes and tools -->
<!ELEMENT pane ( splitpane | tool )* >
<!ATTLIST pane id NMTOKEN #REQUIRED >
<!-- type: left or right for panes in a horizontal
      splitpane, top or bottom for panes in a vertical
      splitpane -->
<!ATTLIST pane type NMTOKEN #IMPLIED >

<!-- a splitpane consists of two parts: either left and
      right, or top or bottom, depending on the type
      attribute -->
<!ELEMENT splitpane ( divideroffset, pane+ ) >
<!ATTLIST splitpane id NMTOKEN #REQUIRED >
<!-- type: horizontal or vertical -->
<!ATTLIST splitpane type NMTOKEN #REQUIRED >

<!-- offset for the splitplane divider -->
<!ELEMENT divideroffset ( #PCDATA ) >

<!-- a Daffodil tool, the id is a property name, the
      #PCDATA should contain the full class path of the
      Java class -->
<!ELEMENT tool ( #PCDATA ) >
<!ATTLIST tool id NMTOKEN #REQUIRED >

<!-- container element for property settings -->
<!ELEMENT settings ( property+ )>

<!-- a Daffodil property/value pair -->
<!ELEMENT property ( #PCDATA ) >
<!ATTLIST property key CDATA #REQUIRED >
```

---



## Literaturverzeichnis

- [AAMODT und PLAZA 1994] A. AAMODT und E. PLAZA (1994). *Case-based reasoning : Foundational issues, methodological variations, and system approaches*. AI Communications, 7(1):39–59.
- [ARMELLINI et al. 2000] M. ARMELLINI, G. BRAJNIK, M. D. FANT, S. MIZARRO und C. TASSO (2000). *System Specifications of the Information Retrieval Assistant*. Technischer Bericht, Department of Mathematics and Computer Science, University of Udine.
- [AULA und NORDHAUSEN 2006] A. AULA und K. NORDHAUSEN (2006). *Modeling successful performance in Web searching*. Journal of the American Society for Information Science and Technology, 57(12):1678–1693.
- [AWASUM 2008] M. AWASUM (2008). *Vorschläge für Google-Suchen als Firefox-Erweiterung*. Bachelorarbeit, Universität Duisburg-Essen.
- [BATES 1979a] M. J. BATES (1979a). *Idea Tactics*. Journal of the American Society for Information Science, 30(5):280–289.
- [BATES 1979b] M. J. BATES (1979b). *Information Search tactics*. Journal of the American Society for Information Science, 30(4):205–214.
- [BATES 1989] M. J. BATES (1989). *The design of browsing and berrypicking techniques for the online search interface*. Online Review, 13(5):407–424. <http://www.gseis.ucla.edu/faculty/bates/berrypicking.html>.
- [BATES 1990] M. J. BATES (1990). *Where Should the Person Stop and the Information Search Interface Start?*. Information Processing and Management, 26(5):575–591.
- [BELKIN et al. 1990] N. J. BELKIN, S. J. CHANG, T. DOWNS, T. SARACEVIC und S. ZHAO (1990). *Taking account of user tasks, goals and behavior for the design of online public access catalogs*. In: *Proc. of the 53rd ASIS Annual Meeting*, S. 69–79.
- [BELKIN 1996] N. BELKIN (1996). *Intelligent Information Retrieval: Whose intelligence?*. In: J. KRAUSE, M. HERFURTH und J. MARX, Hrsg.: *Herausforderungen an die Informationswirtschaft: Informationsverdichtung, Informationsbewertung und Datenvisualisierung*, S. 25–31. Universitätsverlag Konstanz, Konstanz.

- [BELKIN 1993] N. J. BELKIN (1993). *Interaction with Texts: Information Retrieval as Information Seeking Behavior*. In: G. KNORZ, J. KRAUSE und C. WOMSER-HACKER, Hrsg.: *Information Retrieval '93. Von der Modellierung zur Anwendung. Proc. d. 1. Tagung Information Retrieval*, S. 55–66, Konstanz.
- [BELKIN 2000] N. J. BELKIN (2000). *Helping People Find What They Don't Know*. *Communications of the ACM*, 43(8):58–61.
- [BELKIN et al. 1995] N. J. BELKIN, C. COOL, A. STEIN und U. THIEL (1995). *Cases, Scripts, and Information-Seeking Strategies: On the Design of interactive Information Retrieval Systems*. *Expert Systems with Applications*, 9(3):379–395. <http://www.scils.rutgers.edu/~belkin/articles/eswa.pdf>.
- [BELKIN et al. 1993] N. J. BELKIN, P. MARCHETTI und C. COOL (1993). *BRAQUE: Design of an interface to support user interaction in information retrieval*. *Information Processing and Management*, 29(3):325–344.
- [BHAVNANI et al. 2006] S. K. BHAVNANI, C. K. BICHAKJIAN, T. M. JOHNSON, R. J. LITTLE, F. A. PECK, J. L. SCHWARTZ und V. J. STRECHER (2006). *Strategy hubs: Domain portals to help find comprehensive information*. *Journal of the American Society for Information Science and Technology*, 57(1):4–24.
- [BHAVNANI et al. 2003] S. K. BHAVNANI, B. K. CHRISTOPHER, T. M. JOHNSON, R. J. LITTLE, F. A. PECK, J. L. SCHWARTZ und V. J. STRECHER (2003). *Strategy hubs: next-generation domain portals with search procedures*. In: *Proceedings of the conference on Human factors in computing systems*, S. 393–400. ACM Press.
- [BHAVNANI et al. 2001a] S. K. BHAVNANI, K. DRABENSTOTT und S. M. ATHOTA (2001a). *General and Efficient Strategies for Information Retrieval*. In: *Proceedings of ASIST'2001*, S. 680–681.
- [BHAVNANI et al. 2001b] S. K. BHAVNANI, K. DRABENSTOTT und D. RADEV (2001b). *Towards a unified framework of IR tasks and strategies*. In: *Proceedings of ASIST*, S. 340–354.
- [BHAVNANI und JOHN 2000] S. K. BHAVNANI und B. E. JOHN (2000). *The Strategic Use of Complex Computer Systems*. *Human-Computer Interaction*, 15:107–137.
- [BODNARIU 2009] C. BODNARIU (2009). *Analyse von Ereignislogs einer Digitalen Bibliothek zur Benutzerunterstützung*. Diplomarbeit, Universität Duisburg-Essen.
- [BOLLACKER et al. 1998] K. D. BOLLACKER, S. LAWRENCE und C. L. GILES (1998). *CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications*. In: *Proceedings of the second international conference on Autonomous agents*, S. 116–123, New York. ACM.

- [BORGMAN et al. 1989] C. L. BORGMAN, D. O. CASE und C. T. MEADOW (1989). *The Design and Evaluation of a Front-End User Interface for Energy Researchers*. Journal of the American Society for Information Science, 40(2):99–109.
- [BRAJNIK 1999] G. BRAJNIK (1999). *Information Seeking as Explorative Learning*. MIRA.
- [BRAJNIK et al. 1996] G. BRAJNIK, S. MIZZARO und C. TASSO (1996). *Evaluating User Interfaces to Information Retrieval Systems: A Case Study on User Support*. In: H.-P. FREI, D. HARMAN, P. SCHÄUBLE und R. WILKINSON, Hrsg.: *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, S. 128–136, New York. ACM.
- [BRAJNIK et al. 2002] G. BRAJNIK, S. MIZZARO, C. TASSO und F. VENUTI (2002). *Strategic Help in User Interfaces for Information Retrieval*. Journal of the American Society for Information Science and Technology, 53(5):343–358.
- [BREW und MCKELVIE 1996] C. BREW und D. MCKELVIE (1996). *Word-Pair Extraction for Lexicography*. In: *Proc. of the 2nd Intl Conf. on New Methods in Language Processing*, S. 45–55.
- [BYSTRÖM und JÄRVELIN 1995] K. BYSTRÖM und K. JÄRVELIN (1995). *Task complexity affects information seeking and use*. Information Processing Management, 31(2):191–213.
- [CAI 2004] D. CAI (2004). *IfD — Information for Discrimination*. Doktorarbeit, University of Glasgow, Glasgow, Scotland.
- [CARD et al. 1983] S. CARD, T. MORAN und A. NEWELL (1983). *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA.
- [CATLEDGE und PITKOW 1995] L. D. CATLEDGE und J. E. PITKOW (1995). *Characterizing browsing strategies in the World-Wide Web*. Computer Networks and ISDN Systems, 27(6):1065–1073.
- [CHALMERS 1999] M. CHALMERS (1999). *Comparing Information Access Approaches*. Journal of the American Society for Information Science, 50(12):1108–1118. Special Topic Issue: The 50th Anniversary of the Journal of the American Society for Information Science - Part 2: Paradigms, Models, and Methods of Information Science.
- [CHALMERS 2001] M. CHALMERS (2001). *Paths and Contextually Specific Recommendations*. In: *Proceedings of the Second DELOS Network of Excellence Workshop on Personalisation and Recommender Systems in Digital Libraries*, S. 19–24. DELOS - Network of Excellence on Digital Libraries.
- [CHALMERS 2003] M. CHALMERS (2003). *Informatics, architecture and language*, S. 315–341. Springer-Verlag, London, UK.

- [CHALMERS et al. 1998] M. CHALMERS, K. RODDEN und D. BRODBECK (1998). *The Order of Things: Activity-Centred Information Access*. Computer Networks, 30(1-7):359–367.
- [CHEN und DHAR 1991] H. CHEN und V. DHAR (1991). *Cognitive process as a basis for intelligent retrieval systems design*. Information Processing and Management, 27(5):405–432.
- [CHEN et al. 1996] M. S. CHEN, J. S. PARK und P. S. YU (1996). *Data mining for path traversal patterns in a web environment*. In: *Sixteenth International Conference on Distributed Computing Systems*, S. 385–392.
- [COOL und BELKIN 2002] C. COOL und N. J. BELKIN (2002). *A Classification of Interactions with Information*. In: H. BRUCE, R. FIDEL, P. INGWERSEN und P. VAKKARI, Hrsg.: *Emerging frameworks and methods. Proceedings of the Fourth International Conference on Conceptions of Library and Information Science (COLIS4)*, S. 1–15, Greenwood Village. Libraries Unlimited.
- [COOLEY et al. 1997] R. COOLEY, J. SRIVASTAVA und B. MOBASHER (1997). *Web Mining: Information and Pattern Discovery on the World Wide Web*. In: *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*.
- [COVE und WALSH 1988] J. F. COVE und B. C. WALSH (1988). *Online text retrieval via browsing*. Information Processing and Management, 24(1):31–37.
- [DRABENSTOTT 2003] K. M. DRABENSTOTT (2003). *Do Nondomain Experts Enlist the Strategies of Domain Experts*. Journal of the American Society for Information Science and Technology, 54(9):8836–854.
- [DRYER 1997] D. C. DRYER (1997). *Wizards, guides, and beyond: rational and empirical methods for selecting optimal intelligent user interface agents*. In: *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces*, S. 265–268, New York, NY, USA. ACM.
- [ELLIS 1989] D. ELLIS (1989). *A behavioural approach to information retrieval system design*. Journal of Documentation, 45(3):171–212.
- [ELLIS et al. 1993] D. ELLIS, D. COX und K. HALL (1993). *A Comparison of the Information Seeking Patterns of Researchers in the Physical and Social Sciences*. Journal of Documentation, 49(4):356–369.
- [FENICHEL 1980] C. H. FENICHEL (1980). *The process of searching online bibliographic databases: a review of research*. Library Research, 2(2):107–127.
- [FIDEL 1985] R. FIDEL (1985). *Moves in online searching*. Online Review, 9(1):61–74.
- [FIELDS et al. 2004] B. FIELDS, S. KEITH und A. BLANDFORD (2004). *Designing for Expert Information Finding Strategies*. In: S. FINCHER, P. MARKOPOULOS, D. MOORE und R. A. RUDDLE, Hrsg.: *BCS HCI*, S. 89–102. Springer.



- [FRANKMÖLLE 2004] S. FRANKMÖLLE (2004). *Strategien zur Suche in Digitalen Bibliotheken*. Diplomarbeit, Universität Dortmund, FB Informatik.
- [FUHR et al. 2000] N. FUHR, N. GÖVERT und C.-P. KLAS (2000). *An Agent-Based Architecture for Supporting High-Level Search Activities in Federated Digital Libraries*. In: *Proceedings 3rd International Conference of Asian Digital Library*, S. 247–254, Taejon, Korea. KAIST.
- [FUHR et al. 2002] N. FUHR, C.-P. KLAS, A. SCHAEFER und P. MUTSCHKE (2002). *Daffodil: An Integrated Desktop for Supporting High-Level Search Activities in Federated Digital Libraries*. In: *Research and Advanced Technology for Digital Libraries. 6th European Conference, ECDL 2002*, S. 597–612, Heidelberg et al. Springer.
- [GAUCH und SMITH 1991] S. GAUCH und J. B. SMITH (1991). *Search Improvement via Automatic Query Reformulation*. *ACM Transactions on Information Systems*, 9(3):249–280.
- [GAUCH und SMITH 1993] S. GAUCH und J. B. SMITH (1993). *An Expert System for Automatic Query Reformation*. *Journal of the American Society for Information Science*, 44(3):124–136.
- [GENDERA 2006] O. GENDERA (2006). *Visualisierung von Suchpfaden*. Diplomarbeit, Universität Dortmund, FB Informatik.
- [GONNET et al. 1992] G. H. GONNET, R. A. BAEZA-YATES und T. SNIDER (1992). *New Indices for Text: PAT Trees and PAT Arrays*. In: *Information Retrieval. Data Structures & Algorithms*, S. 66–82. Prentice Hall, Englewood Cliffs.
- [GÖVERT et al. 2000] N. GÖVERT, N. FUHR und C.-P. KLAS (2000). *Daffodil: Distributed Agents for User-Friendly Access of Digital Libraries*. In: J. BORBINHA und T. BAKER, Hrsg.: *Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2000)*, Bd. 1923 d. Reihe *Lecture Notes in Computer Science*, S. 352–355, Heidelberg et al. Springer.
- [GREENBERG und COCKBURN 1999] S. GREENBERG und A. COCKBURN (1999). *Getting Back to Back: Alternate Behaviors for a Web Browser's Back Button*. In: *Proceedings of the 5th Annual Human Factors and the Web Conference*.
- [GUZDIAL 1993] M. GUZDIAL (1993). *Emile: Software-realized Scaffolding for Science Learners Programming in Mixed Media*. Doktorarbeit, University of Michigan, Ann Arbor, MI, USA.
- [GUZDIAL 1994] M. GUZDIAL (1994). *Software-Realized Scaffolding to Facilitate Programming for Science Learning*. *Interactive Learning Environments*, 4(1):1–44.
- [HALTTUNEN 1993] K. HALTTUNEN (1993). *Scaffolding performance in IR Instruction: exploring learning experiences and performance in two learning environments*. *Journal of Information Science*, 25(3):307–332.

- [HALTTUNEN 1995] K. HALTTUNEN (1995). *Two Information Retrieval Learning Environments - Their Design and Evaluation*. Doktorarbeit, University of Tampere.
- [HARMAN und VOORHEES 1998] D. HARMAN und E. M. VOORHEES, Hrsg. (1998). *The Sixth Text REtrieval Conference (TREC-6)*, Gaithersburg, Md. 20899. National Institute of Standards and Technology.
- [HARTER 1986] S. P. HARTER (1986). *Online information retrieval: concepts, principles, and techniques*. Academic Press Professional, Inc., San Diego, CA, USA.
- [HARTER und PETERS 1985] S. P. HARTER und A. R. PETERS (1985). *Heuristics for online information retrieval: a typology and preliminary listing*. *Online Review*, 9(5):407–424.
- [HASSAN-MONTERO und HERRERO-SOLANA 2006] Y. HASSAN-MONTERO und V. HERRERO-SOLANA (2006). *Improving Tag-Clouds as Visual Information Retrieval Interfaces*. In: *Proc. InSciT 2006*, Merida, Spain.
- [HAWKINS und WAGNER 1982] D. T. HAWKINS und R. WAGNER (1982). *Online bibliographic search strategy development*. *Online*, 6(3):12–19.
- [HEARST 1999] M. A. HEARST (1999). *User Interfaces and Visualization*. In: *Modern Information Retrieval*. Addison Wesley.
- [HEARST 2009] M. A. HEARST (2009). *The Evaluation of Search User Interfaces*, Kap. 2. Cambridge University Press.
- [HEARST und ROSNER 2008] M. A. HEARST und D. ROSNER (2008). *Tag Clouds: Data Analysis Tool or Social Signaller?*. In: *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, S. 160–169, Los Alamitos, CA, USA. IEEE Computer Society.
- [HIGHTOWER et al. 1998] R. R. HIGHTOWER, L. T. RING, J. I. HELFMAN, B. B. BEDERSON und J. D. HOLLAN (1998). *PadPrints: graphical multiscale Web histories*. In: *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, S. 121–122, New York, NY, USA. ACM.
- [HSIEH-YEE 1993] I. HSIEH-YEE (1993). *Effects of Search Experience and Subject Knowledge on Online Search Behavior: Measuring the Search Tactics of Novice and Experienced Searchers*. *Journal of the American Society for Information Science*, 44(3):161–174.
- [JACKSON et al. 1998] S. L. JACKSON, J. KRAJCIK und E. SOLOWAY (1998). *The Design of Guided Learner-Adaptable Scaffolding in Interactive Learning Environments*. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*, S. 187–194, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [JAMESON 2001] A. JAMESON (2001). *Systems That Adapt to Their Users: An Integrative Perspective*. Saarland University, Saarbrücken.

- [JANSEN 2005] B. J. JANSEN (2005). *Seeking and implementing automated assistance during the search process*. *Inf. Process. Manage.*, 41(4):909–928.
- [JANSEN 2008] B. J. JANSEN (2008). *The Methodology of Search Log Analysis*. In: B. J. JANSEN, A. SPINK und I. TAKSA, Hrsg.: *Handbook of Research on Web Log Analysis*, Information Science Reference, S. 99–120. IGI Global.
- [JANSEN und MCNEESE 2005] B. J. JANSEN und M. D. MCNEESE (2005). *Evaluating the effectiveness of and patterns of interactions with automated searching assistance: Research Articles*. *Journal of the American Society for Information Science and Technology*, 56(14):1480–1503.
- [JÄRVELIN 2009] K. JÄRVELIN (2009). *Explaining User Performance in Information Retrieval: Challenges to IR Evaluation*. In: L. AZZOPARDI, G. KAZAI, S. E. ROBERTSON, S. M. RÜGER, M. SHOKOUHI, D. SONG und E. YILMAZ, Hrsg.: *ICTIR*, Bd. 5766 d. Reihe *Lecture Notes in Computer Science*, S. 289–296. Springer.
- [JÄRVELIN und INGWERSEN 2004] K. JÄRVELIN und P. INGWERSEN (2004). *Information seeking research needs extension towards tasks and technology*. *Information Research*, 10(1). paper 212, available at <http://InformationR.net/ir/10-1/paper212.html>.
- [JÄRVELIN und WILSON 2003] K. JÄRVELIN und T. WILSON (2003). *On conceptual models for information seeking and retrieval research*. *Information Research*, 9(1). paper 163, available at <http://InformationR.net/ir/9-1/paper163.html>.
- [JOHN und KIERAS 1996] B. E. JOHN und D. E. KIERAS (1996). *Using GOMS for User Interface Design and Evaluation: Which Technique?* *ACM Trans. ACM Trans. Comput.-Hum. Interact.* 3.
- [KELLAR et al. 2006] M. KELLAR, C. WATTERS und M. SHEPHERD (2006). *A goal-based classification of web information task*. In: *Proceedings of ASIS&T 2006*, Austin, TX.
- [KIESTA et al. 1994] M. D. KIESTA, M. J. W. STOKMANS und J. KAMPHUIS (1994). *End-users searching the online catalogue: The influence of domain and system knowledge on search patterns*. *The Electronic Library*, 12(6):335–343.
- [KLAS 2007] C.-P. KLAS (2007). *Strategische Unterstützung bei der Informationssuche in Digitalen Bibliotheken*. Doktorarbeit, University of Duisburg-Essen.
- [KLAS et al. 2006a] C.-P. KLAS, H. ALBRECHTSEN, N. FUHR, P. HANSEN, S. KAPIDAKIS, L. KOVÁCS, S. KRIEWEL, A. MICSIK, C. PAPANTHEODOROU, G. TSAKONAS und E. JACOB (2006a). *A Logging Scheme for Comparative Digital Library Evaluation..* In: J. GONZALO, C. THANOS, M. F. VERDEJO und R. C. CARRASCO, Hrsg.: *Research and Advanced Technology for Digital Libraries. Proc. of the 10th European Conference on Digital Libraries (ECDL 2006)*, *Lecture Notes in Computer Science*, S. 267–278, Heidelberg et al. Springer.

- [KLAS et al. 2006b] C.-P. KLAS, N. FUHR, S. KRIEWEL, H. ALBRECHTSEN, G. TSAKONAS, S. KAPIDAKIS, C. PAPTAEODOROU, P. HANSEN, L. KOVACS, A. MICSIK und E. JACOB (2006b). *An experimental framework for comparative digital library evaluation: the logging scheme*. In: *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, S. 308–309, New York, NY, USA. ACM Press.
- [KLAS et al. 2004a] C.-P. KLAS, N. FUHR und A. SCHAEFER (2004a). *Evaluating Strategic Support for Information Access in the DAFFODIL System*. In: R. HEERY und L. LYON, Hrsg.: *Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2004)*, Lecture Notes in Computer Science, S. 476–487, Heidelberg et al. Springer.
- [KLAS et al. 2007] C.-P. KLAS, S. KRIEWEL und N. FUHR (2007). *An Experimental Framework for Interactive Information Retrieval and Digital Libraries Evaluation*. In: *DELOS Conference on Digital Libraries, Tirrenia, Pisa (Italy)*. available at <http://146.48.87.21/OLP/UI/1.0/Disseminate/1194810992pfVSLInHz4/a221194810992YLolEYIH>.
- [KLAS et al. 2008] C.-P. KLAS, S. KRIEWEL und M. HEMMJE (2008). *An Experimental System for Adaptive Services in Information Retrieval*. In: *Proceedings of the 2nd International Workshop on Adaptive Information Retrieval (AIR 2008)*.
- [KLAS et al. 2004b] C.-P. KLAS, S. KRIEWEL und A. SCHAEFER (2004b). *Daffodil - Nutzerorientiertes Zugangssystem für heterogene digitale Bibliotheken*. dvs Band.
- [KOENEMANN und BELKIN 1996] J. KOENEMANN und N. J. BELKIN (1996). *A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness*. In: *CHI*, S. 205–212.
- [KOMLODI 2002] A. KOMLODI (2002). *The Role of Interaction Histories in Mental Model Building and Knowledge Sharing in the Legal Domain*. *Journal of Universal Computer Science*, 8(5):557–566. available at [http://www.jucs.org/jucs\\_8\\_5/the\\_role\\_of\\_interaction](http://www.jucs.org/jucs_8_5/the_role_of_interaction).
- [KOMLODI 2000] A. KOMLODI (2000). *Search history for user support in information-seeking interfaces*. In: *CHI '00: CHI '00 extended abstracts on Human factors in computing systems*, S. 75–76, New York, NY, USA. ACM Press.
- [KOMLODI 2004] A. KOMLODI (2004). *Task management support in information seeking: A case for search histories..* *Computers in Human Behavior*, 20:163–184.
- [KOMLODI et al. 2007] A. KOMLODI, G. MARCHIONINI und D. SOERTEL (2007). *Search history support for finding and using information: User interface design recommendations from a user study*. *Information Processing and Management*, 43:10–29.

- [KOMLODI und SOERGEL 2002] A. KOMLODI und D. SOERGEL (2002). *Attorneys Interacting with Legal Information Systems: Tools for Mental Model Building and Task Integration*. In: *Proceedings of the American Society for information science and technology*, S. 152–163.
- [KRAUSE 1997] J. KRAUSE (1997). *Das WOB-Modell*. In: *Vages Information Retrieval und graphische Benutzeroberflächen: Beispiel Werkstoffinformation.*, S. 59–88. Universitätsverlag, Konstanz.
- [KRAWCZAK et al. 1987] D. KRAWCZAK, P. SMITH und S. SHUTE (1987). *EP-X: a demonstration of semantically based search of bibliographic databases*. In: *SIGIR '87: Proceedings of the 10th annual international ACM SIGIR conference on Research and development in information retrieval*, S. 263–271, New York, NY, USA. ACM.
- [KRIEWEL und FUHR 2007] S. KRIEWEL und N. FUHR (2007). *Adaptive Search Suggestions for Digital Libraries*. In: *Asian Digital Libraries: Looking Back 10 Years and Forging New Frontiers (ICADL 2007)*, S. 220–229.
- [KRIEWEL und FUHR 2010] S. KRIEWEL und N. FUHR (2010). *An evaluation of an adaptive search suggestion system*. In: *32nd European Conference on Information Retrieval Research (ECIR 2010)*.
- [KRIEWEL et al. 2005] S. KRIEWEL, C.-P. KLAS, S. FRANKMÖLLE und N. FUHR (2005). *A Framework for Supporting Common Search Strategies in DAFFODIL*. In: [RAUBER et al. 2005], S. 527–528.
- [KRIEWEL et al. 2004] S. KRIEWEL, C.-P. KLAS, A. SCHAEFER und N. FUHR (2004). *Daffodil - Strategic Support for User-Oriented Access to Heterogeneous Digital Libraries*. *D-Lib Magazine*, 10(6). available at <http://www.dlib.org/dlib/june04/kriewel/06kriewel.html>.
- [KUHALTHAU 1991] C. C. KUHALTHAU (1991). *Inside the Search Process: Information Seeking from the User's Perspective*. *Journal of the American Society for Information Science*, 42(5):361–371.
- [KUHALTHAU 1993] C. C. KUHALTHAU (1993). *Seeking Meaning: A Process Approach to Library and Information Services*. Ablex Publishing, Norwood, NJ.
- [KUHALTHAU 2002] C. C. KUHALTHAU (2002). *Information Search Process: A Search for Meaning rather than Answers*. available at <http://www.scils.rutgers.edu/~kuhlthau/Search%20Process.htm>.
- [KUO et al. 2007] B. Y.-L. KUO, T. HENTRICH, B. M. GOOD und M. D. WILKINSON (2007). *Tag Clouds for Summarizing Web search results*. In: *Proceedings of the WWW 2007*, S. 1203–1204, New York, NY, USA. ACM.
- [LALMAS und RUTHVEN 1999] M. LALMAS und I. RUTHVEN (1999). *A Framework for Investigating the Interaction in Information Retrieval*. In: *Proceedings of the 9th European-Japanese Conference on Information Modelling and Knowledge Bases (EJ-IMKB)*, Iwate, Japan.

- [LI und BELKIN 2008] Y. LI und N. J. BELKIN (2008). *A faceted approach to conceptualizing tasks in information seeking*. *Information Processing and Management*, 44(6):1822–1837.
- [LIN 1997] X. LIN (1997). *Map Display for Information Retrieval*. *Journal of the American Society for Information Science*, 48(1):40–54.
- [MAIN et al. 2001] J. MAIN, T. DILLON und S. SHIU (2001). *A Tutorial on Case Based Reasoning*. In: S. PAL, T. DILLON und D. YEUNG, Hrsg.: *Soft Computing in Case Based Reasoning*, S. 1–28, London. Springer.
- [MANGLANO et al. 1998] V. MANGLANO, M. BEAULIEU und S. ROBERTSON (1998). *Evaluation of Interfaces for IRS: Modelling End-User Searching Behaviour*. *Colloquium on Information retrieval (IRSG)*, S. 137–146.
- [MANNILA et al. 1997] H. MANNILA, H. TOIVONEN und A. I. VERKAMO (1997). *Discovery of Frequent Episodes in Event Sequences*. *Data Mining and Knowledge Discovery*, 1(3):259–289.
- [MARCHIONINI 1989] G. MARCHIONINI (1989). *Information-seeking strategies of novices using a full-text electronic encyclopedia*. *Journal of the American Society for Information Science*, 40(1):54–66.
- [MARCHIONINI 1995] G. MARCHIONINI (1995). *Information seeking in electronic environments..* Cambridge University Press, New York, NY, USA.
- [MARKEY 2007a] K. MARKEY (2007a). *Twenty-five years of end-user searching, Part 1: Research findings*. *Journal of the American Society for Information Science and Technology*, 58(8):1071–1081.
- [MARKEY 2007b] K. MARKEY (2007b). *Twenty-five years of end-user searching, Part 2: Future research directions*. *Journal of the American Society for Information Science and Technology*, 58(8):1123–1130.
- [MEADOW und COCHRANE 1981] C. T. MEADOW und P. COCHRANE (1981). *Basics of Online Searching*. Wiley, New York, NY, USA.
- [MEADOW et al. 1982a] C. T. MEADOW, T. T. HEWETT und E. S. AVERSA (1982a). *A Computer Intermediary for Interactive Database Searching. I. Design*. *Journal of the American Society for Information Science*, 33(5):325–332.
- [MEADOW et al. 1982b] C. T. MEADOW, T. T. HEWETT und E. S. AVERSA (1982b). *A Computer Intermediary for Interactive Database Searching. II. Evaluation*. *Journal of the American Society for Information Science*, 33(6):357–364.
- [MEHO und TIBBO 2003] L. I. MEHO und H. R. TIBBO (2003). *Modeling the information-seeking behavior of social scientists: Ellis's study revisited*. *Journal of the American Society for Information Science and Technology*, 54(6):570–587.
- [MÜLLER 2001] F. MÜLLER (2001). *DAFFODIL: Basisarchitektur für ein Agentensystem für digitale Bibliotheken*. Diplomarbeit, Universität Dortmund, FB Informatik.

- [OLLERENSHAW 2002] M. OLLERENSHAW (2002). *Redefining the Browser History in Hypertext Terms*. In: *MultimediaSystems02 Conference*.
- [ONTAÑÓN und PLAZA 2003] S. ONTAÑÓN und E. PLAZA (2003). *Justification-based multiagent learning*. In: N. M. TOM FAWCETT, Hrsg.: *The Twentieth International Conference on Machine Learning (ICML 2003)*, S. 576–583. AAAI Press, AAAI Press.
- [PHARO 2002] N. PHARO (2002). *The Search Situation and Transition method schema: a tool for analyzing Web information search processes*. Doktorarbeit, University of Tampere, Finland. <http://acta.uta.fi/pdf/951-44-5355-7.pdf>.
- [PHARO 2004] N. PHARO (2004). *A new model of information behaviour based on the Search Situation Transition schema*. *Information Research*, 10(1). paper 203, available at <http://InformationR.net/ir/10-1/paper203.html>.
- [PITKOW und PIROLI 1999] J. E. PITKOW und P. PIROLI (1999). *Mining Longest Repeating Subsequences to Predict World Wide Web Surfing*. In: *USENIX Symposium on Internet Technologies and Systems*.
- [PLAISANT et al. 1999] C. PLAISANT, B. SHNEIDERMAN, K. DOAN und T. BRUNS (1999). *Interface and data architecture for query preview in networked information systems*. *ACM Transactions on Information Systems*, 17(3).
- [QIU 1993] L. QIU (1993). *Markov Models of Search State Patterns in a Hypertext Information Retrieval System*. *Journal of the American Society for Information Science*, 44(7):413–427.
- [QUINTANA et al. 2000] C. QUINTANA, E. FRETZ, J. KRAJCIK und E. SOLOWAY (2000). *Evaluation criteria for scaffolding in learner-centered tools*. In: *CHI '00: CHI '00 extended abstracts on Human factors in computing systems*, S. 189–190, New York, NY, USA. ACM.
- [QUINTANA et al. 2002] C. QUINTANA, J. KRAJCIK und E. SOLOWAY (2002). *A Case Study to Distill Structural Scaffolding Guidelines for Scaffolded Software Environments*. In: *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, S. 81–88, New York, NY, USA. ACM.
- [QUINTANA und ZHANG 2004a] C. QUINTANA und M. ZHANG (2004a). *The digital ideakeeper: integrating digital libraries with a scaffolded environment for online inquiry..* In: H. CHEN, H. D. WACTLAR, C. CHIH CHEN, E.-P. LIM und M. G. CHRISTEL, Hrsg.: *JCDL*, S. 388. ACM.
- [QUINTANA und ZHANG 2004b] C. QUINTANA und M. ZHANG (2004b). *Idea-Keeper notepads: scaffolding digital library information analysis in online inquiry*. In: *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, S. 1329–1332, New York, NY, USA. ACM.
- [RAMIREZ et al. 2002] A. RAMIREZ, J. B. WALTHER, J. K. BURGOON und M. SUNNAFRANK (2002). *Information-Seeking Strategies, Uncertainty, and*

- Computer-Mediated Communication. Toward a Conceptual Model.* Human Communication Research, 28(2):213–228.
- [RAUBER et al. 2005] A. RAUBER, C. CHRISTODOULAKIS und A. M. TJOA, Hrsg. (2005). *Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2005)*, Lecture Notes in Computer Science, Heidelberg et al. Springer.
- [REISER 2002] B. J. REISER (2002). *Why Scaffolding Should Sometimes Make Tasks More Difficult for Learners.* In: *Proceedings of Computer Support for Collaborative Learning (CSCL 2002)*, S. 255–264.
- [RUDNER 2000] L. RUDNER (2000). *Who Is Going to Mine Digital Library Resources? And How?.* D-Lib Magazine, 6(5).
- [SCHAEFER 2008] A. SCHAEFER (2008). *Arbeitsteilung zwischen direkter Manipulation und proaktiven Software-Agenten in Informationssystemen.* Doktorarbeit, University of Duisburg-Essen.
- [SCHAEFER et al. 2005] A. SCHAEFER, M. JORDAN, C.-P. KLAS und N. FUHR (2005). *Active Support For Query Formulation in Virtual Digital Libraries: A case study with DAFFODIL.* In: [RAUBER et al. 2005].
- [SHNEIDERMAN 1999] B. SHNEIDERMAN (1999). *Supporting Creativity with Advanced Information-Abundant User Interfaces.* In: *Human-Centred Computing, Online Communities and Virtual Environment.*
- [SHNEIDERMAN und PLAISANT 2009] B. SHNEIDERMAN und C. PLAISANT (2009). *Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th edition).* Addison-Wesley.
- [SMITH 1989] P. J. SMITH (1989). *Knowledge-Based Search Tactics for an Intelligent Intermediary System.* ACM Transactions on Information Systems, 7(3):246–270.
- [SOLSO et al. 2005] R. L. SOLSO, M. K. MACLIN und O. H. MACLIN (2005). *Cognitive Psychology (International Edition).* Pearson Education International, Allyn & Bacon, Boston, MA, USA.
- [SORMUNEN et al. 1998] E. SORMUNEN, J. LAAKSONEN, J. KEKÄLÄINEN, H. KEMPPAINEN, A. PIRKOLA und K. JÄRVELIN (1998). *The IR Game - A Tool for Rapid Query Analysis in Cross-Language IR Experiments.* In: *Proceedings of Joint Workshop on Cross-Language Issues in Artificial Intelligence and Issues in Cross Cultural Communication - Towards Culturally Situated Agents*, S. 22–32.
- [SPINK et al. 2001] A. SPINK, D. WOLFRAM, B. J. JANSEN und T. SARACEVIC (2001). *Searching the Web: The Public and Their Queries.* Journal of the American Society for Information Science and Technology, 52(3):226–234.



- [STELMASZEWSKA und BLANDFORD 2002] H. STELMASZEWSKA und A. BLANDFORD (2002). *Patterns of interactions: user behaviour in response to search results*. In: *Proceedings of the JCDL Workshop on Usability*.
- [STELMASZEWSKA et al. 2005] H. STELMASZEWSKA, A. BLANDFORD und G. BUCHANAN (2005). *Designing to change users' information seeking behaviour: a case study*. In: S. Y. CHEN und G. D. MAGOULAS, Hrsg.: *Adaptable and Adaptive Hypermedia Systems*, S. 1–18. Idea Publishing.
- [SULLIVAN et al. 1990] M. V. SULLIVAN, C. L. BORGMAN und D. WIPPERN: (1990). *End-users, mediated searches and front-end assistance programs on dialog: A comparison of learning, performance and satisfaction*. *Journal of the American Society for Information Science*, 41(1):27–42.
- [TIDWELL 2005] J. TIDWELL (2005). *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly Media, Inc.
- [TIPS 1999] TIPS (1999). *Information Retrieval Assistant User Requirements*. Department of Mathematics and Computer Science, University of Udine. TIPS Documentation, Workpackage 2, Task 2.1, UR-R1 D2.1.1.
- [TWIDALE et al. 1995] M. TWIDALE, D. NICHOLS, G. SMITH und J. TREVOR (1995). *Supporting collaborative learning during information searching*. In: *Proceedings of Computer Support for Collaborative Learning (CSCL 95)*, S. 367–374.
- [VIGIL 1983] P. J. VIGIL (1983). *The psychology of online searching*. *Journal of the American Society for Information Science*, 43(4):281–287.
- [VYGOTSKY 1978] L. VYGOTSKY (1978). *Mind in Society*. Harvard University Press, Cambridge.
- [WAGERS 1989] R. WAGERS (1989). *Can easy searching be good searching? A model for easy searching*. *Online*, 13(3):78–85.
- [WERSECK 2006] C. WERSECK (2006). *Evaluation von Digitalen Bibliotheken*. Diplomarbeit, Universität Duisburg-Essen, Institut für Informatik und Interaktive Systeme.
- [WEXELBLAT 1999] A. WEXELBLAT (1999). *History-Based Tools for Navigation*. In: *Proceedings of the Hawaii International Conference On System Sciences (HICSS-32), IEEE Computer Society Press*.
- [WEXELBLAT und MAES 1999] A. WEXELBLAT und P. MAES (1999). *Footprints: History-Rich Tools for Information Foraging*. In: *CHI 99*, S. 270–277. ACM Press.
- [WILDEMUTH 2004] B. M. WILDEMUTH (2004). *The Effects of Domain Knowledge on Search Tactic Formulation*. *Journal of the American Society for Information Science and Technology*, 55(3):246–258.
- [WILDEMUTH und MOORE 1995] B. M. WILDEMUTH und M. E. MOORE (1995). *End-user search behaviors and their relationship to search effectiveness*. *Bulletin of the Medical Library Association*, 83(3):294–304.

- [WILSON 1999] T. D. WILSON (1999). *Models in Information Behaviour Research*. Journal of Documentation, 55(3):249–270. <http://informationr.net/tdw/publ/papers/1999JDoc.html>.
- [WOOLDRIDGE und JENNINGS 1995] M. WOOLDRIDGE und N. R. JENNINGS, Hrsg. (1995). *Intelligent Agents: Theories, Architectures, and Languages*. Springer, Heidelberg et al.
- [XIE 1997] H. I. XIE (1997). *Planned und situated aspects in Interactive IR: Patterns of user interactive intentions and information seeking strategies*. In: C. SCHWARTZ und M. RORVIG, Hrsg.: *Proceedings of the ASIS annual meeting*, S. 100–122, Medford, NJ, USA. Learned Information Inc.
- [XIE 2000] H. I. XIE (2000). *Shifts of Interactive Intentions and Information-Seeking Strategies in Interactive Information Retrieval*. Journal of the American Society for Information Science, 51(9):841–857.
- [XIE 2007] H. I. XIE (2007). *Shifts in information-seeking strategies in information retrieval in the digital age. A planned-situational model*. Information Research., 12(4).
- [XIE und COOL 2009] I. XIE und C. COOL (2009). *Understanding help seeking within the context of searching digital libraries*. Journal of the American Society for Information Science and Technology, 60(3):477–494.
- [ZHENG 2007] Y. ZHENG (2007). *Automatisierte Ähnlichkeitsuche zur Ergänzung von Dokumentlisten*. Diplomarbeit, Universität Duisburg–Essen.

## Abbildungsverzeichnis

5.1. Suchwerkzeug. . . . .	65
5.2. Extrahierte Terme. . . . .	67
5.3. Ursprüngliche Umsetzung des DAFFODIL-Desktops. . . . .	71
5.4. DAFFODIL-Desktop mit aufgabenspezifischen Perspektiven. . .	72
5.5. Modulare Suchstrategien im Rahmen des Strategieframeworks [KRIEWEL et al. 2005]. . . . .	74
5.6. Ein Makro des Strategieframeworks [KRIEWEL et al. 2005]. . . .	75
5.7. Der Loggingdienst von DAFFODIL [KLAS et al. 2006a]. . . . .	76
5.8. Visualisierungsvarianten einer Nutzersitzung. . . . .	79
6.1. Der Zyklus beim fallbasierten Schließen nach [AAMODT und PLAZA 1994]. . . . .	84
6.2. Das Vorschlagswerkzeug. . . . .	92
6.3. Das Vorschlagswerkzeug für Google. . . . .	95
6.4. Ausschnitt eines PAT-Trees mit Gewichten. . . . .	101
6.5. Vorschlagswerkzeug basierend auf Suchpfaden [BODNARIU 2009]. . . . .	103
7.1. Schematische Darstellung von Modul 2-2 nach [ZHENG 2007]. . .	111
7.2. Bestandteile der Scaffolding-Erweiterung für DAFFODIL. . . . .	115
7.3. Perspektive für Schritt 3: Suche. . . . .	119
8.1. Evaluationsergebnisse. . . . .	128
8.2. Evaluationsergebnisse (Forts.). . . . .	129



## Tabellenverzeichnis

2.1. Beispielhafte Suchtaktiken nach Bates [BATES 1979a, BATES 1979b]. . . . .	17
2.2. Beispielhafte Suchstrategeme nach Bates [BATES 1989]. . . . .	20
2.3. Fidels operationale Suchtaktiken in Bates' Taxonomie. . . . .	27
2.4. Fidels konzeptionelle Suchtaktiken in Bates' Taxonomie. . . . .	29
3.1. Ordnungssystem für Benutzer- und Systemaktionen. . . . .	36
6.1. Erste Auswahl von Vorschlägen für DAFFODIL mit Einordnung nach Bates [BATES 1990]. . . . .	85
6.2. Zusätzliche Vorschläge für DAFFODIL mit Einordnungen u.a. nach Bates [BATES 1990]. . . . .	87
6.3. Vorschläge für Google-Websuche, Einordnung nach Bates [BATES 1990]. . . . .	94
6.4. Beispielhafter Suchpfad: SDCDCx1SDPat. . . . .	101
8.1. Aufgabenbeschreibungen in Experiment 2 . . . . .	133
8.2. Moves zur Änderung der Anfrage (Unterstützung nur für Aufgaben 1 und 2) . . . . .	134
8.3. Benutzeraktionen während der Aufgaben (Unterstützung nur für Aufgaben 1 und 2) . . . . .	135
8.4. Von Suchenden gespeicherte relevante Dokumente . . . . .	137
8.5. Häufigste benutzte Dienste oder Werkzeuge . . . . .	139
8.6. Häufigste Ereignisse auf konzeptueller Ebene . . . . .	139
8.7. Vorkommen von Dreierketten (konzeptuelle Ereignisse) . . . . .	141
8.8. Vorkommen von Dreierketten (Werkzeuge) . . . . .	142
8.9. Häufigste Folgeereignisse . . . . .	142
8.10. Vorkommen von „echten“ Zweier-Übergängen (Aktionen) . . . . .	143
8.11. Dreierketten in Experiment 2 (nicht unterstützt) . . . . .	144
8.12. Dreierketten in Experiment 2 (unterstützt) . . . . .	145



# Index

- Ähnlichkeitsmaß, 84, 89, 94, 96, 101, 110, 126
- Actual Development Level, 47, 111
- Afferenz, 87, 91
- Agent, 61, 62, 68, 87–89, 99
- ASDL, 86
- Begriffe, verwandte, 65, 92, 113, 129, 130, 136
- Case-Based Reasoning, 3, 40, 79, 80, 84, 89, 91, 101, 102, 123, 146, 148
- CBR, *siehe* Case-Based Reasoning
- Chatfunktion, 66, 85
- Coaching, kollaboratives, 39, 40
- CORBA, 61, 68
- Daffodil, 2, 3, 61, 62, 64, 65, 67–70, 73, 78, 81, 83–89, 91, 93, 95, 97–99, 105, 108, 111, 113–115, 117, 118, 120, 124, 129, 130, 132, 135, 142, 146, 147, 153
- Decision Task, 11
- Dice-Koeffizient, 101
- Digital-Library-System, 1, 2, 35, 123, 130
- Discrimination Score, 109
- DLS, *siehe* Digital-Library-System
- Effektivität, 1, 25, 37, 38, 133, 147
- Efferenz, 87
- Effizienz, 1, 37, 88
- Endnutzer, 1, 2, 21, 25, 37, 38, 49, 105, 110, 142, 145
- EP-X, 39
- Extraktion, 62, 64, 66, 83, 114, 127, 129, 130, 135
- Fähigkeitsbereich, 47
- Fähigkeitshorizont, 50
- Fälle, negative, 81, 86, 102, 128
- Fading, 46, 47, 50, 111, 117, 118
- Fallbasiertes Schließen, *siehe* Case-Based Reasoning
- Fallbasis, 79, 80, 84, 86–89, 91, 95, 102, 123, 128
- Filtern, 62, 64, 67, 75, 85
- FIRE, 39, 40
- Firefox, 91, 95
- Footprints, 56
- Formular, 62, 129, 135
- GOMS-Modell, 15, 16, 74
- Google, 84, 91–95, 130
- Handbibliothek, persönliche, 62, 66, 108, 114
- IdeaKeeper, 48
- IIDA, 38
- Inferenz, 87
- Infon, 54
- Information Retrieval, 7, 13, 107
- Information-Retrieval-System, 1, 2, 19, 25, 37–39, 42, 52, 54, 61
- Informationsbedürfnis, 1, 7, 8, 10, 14, 29, 31, 32, 35, 36, 56, 106, 108
- Informationsobjekt, 1, 18, 19, 31, 53, 55, 61, 66, 67, 75, 106–108

- Informationsproblem, 2, 3, 8, 9, 123, 145
- Informationssuche, 3, 7–10, 14, 22, 29, 30, 35, 42, 49, 52, 62, 110, 111, 145
- Informationssuchstrategie, 9, 13, 16, 19, 40, 55
- IRS, *siehe* Information-Retrieval-System
- ISS, *siehe* Informationssuchstrategie
- Journalbrowser, 65, 75, 83
- Kachel, *siehe* Pane
- Klassifikationsbrowser, 65, 75, 95, 113, 127, 129, 130, 136
- Known-Item-Instantiation, 20, 21, 106–108
- Koautorennetz, 65, 83, 127, 129, 130
- Konferenzbrowser, 65, 83
- Konversationsmodell, 41
- Lernen, 18, 52
  - exploratives, 10
- Loggingebene, 73, 75, 76, 97, 138, 146
- Makro, 70, 106, 107
- Markov-Modelle, 56
- MERIT, 40
- Move, 14–16, 56, 61
- N-Gramme, 101
- Navigieren, 75
- Nearest-Neighbor-Methode, 79
- Nutzerintervention, 2
- Pane, 68, 117
- PAT-Tree, 98–100
- Perspektive, 68, 115, 117–119, 124, 146
- Pfadmodell, 54, 55, 79
- Potential Development Level, 47
- Problemlösungsmodell, 9, 10, 51, 111, 145
- Problemlösungsverhalten, 9, 51
- Problemsituation, 1, 30, 34, 35, 37, 79, 80, 84, 90
- Prozessmodell, 7, 10
- Query by Example, 108
- Recommender-System, 55
- Scaffolding, 16, 46–48, 110–115, 118, 120
  - adaptierbar, 48, 118
  - adaptiv, 48, 118
  - Makroebene, 49
  - Mikroebene, 49
  - software-realisiert, 2, 3, 48, 49, 105, 110–114, 118, 120, 147
- Shift, 18, 19, 21
- siString, 98, 99
- Situation, 127
- Software-Wizard, 111
- Sortieren, 62, 75
- Strategem, 2, 14, 15, 18, 21, 22, 28, 36, 39, 52, 61, 67, 79, 81, 83, 85, 92, 96, 102, 105, 120, 123, 127, 142, 146
  - Area Scanning, 20, 21, 65
  - Author Searching, 20, 65, 81, 83
  - Briefsearch, 22
  - Citation Searching, 20, 21
  - Cited Author, 22
  - Footnote Chasing, 20, 21
  - Index Searching, 83
  - Journal Run, 83
  - Subject Searching, 20, 21, 92
- Strategie, *siehe* Suchstrategie
  - analytisch, 16, 18, 22, 36, 37, 61, 62
  - Berrypicking, 22, 24, 66, 114, 146
  - browsend, 16, 22, 24, 61, 62, 106
  - Building Blocks, 23, 85
  - Cited Publication, 23
  - Closed-Loop Relevance Clustering, 24
  - Interactive Scanning, 24, 85



- 
- kollaborativ, 66, 73
  - Navigieren, 24
  - Pearl Growing, 23, 85
  - Scanning, 24
  - Screen Browsing, 21
  - Successive Facets, 23
  - Trial-and-Error, 21
  - Strategierahmen, 70, 105, 106, 108, 109, 120, 146, 147
  - Strategy Hub, 41, 111
  - Suchaktivität, 14, 61, 67, 73, 76, 96, 97, 105
  - Suchassistent, 112, 114, 115, 117, 120
  - Suchaufgabe, 3, 10–12, 28, 32, 36, 37, 55, 70, 106, 108, 111, 112, 124, 130, 145
  - Suchdomäne, 1, 14, 15, 95
  - Suchexperte, 1, 2, 35, 36, 41, 76, 145
  - Suchexpertise, *siehe* Wissen, prozedural, 129
  - Suchhistorie, 3, 30–33, 43, 51–54, 56, 66, 67, 76, 85, 87, 93, 94, 129, 130, 146
  - Suchlogs, 3, 56, 73, 76, 87, 96, 97, 100, 102, 125, 132
  - Suchpfad, 4, 51, 54–56, 73, 74, 76, 79, 96, 97, 99–103, 146
  - Suchplan, *siehe* Suchstrategie
  - Suchproblem, 2, 3
  - Suchprozess, 7, 8, 34, 42, 55
  - Suchschnittstelle, 1, 36, 39, 51, 52, 54, 56, 61, 62, 87, 89, 99
  - Suchsituation, 29–33, 36, 44, 51, 54, 72, 79–81, 86–89, 91, 93, 94, 102, 126, 146
  - Suchstrategie, 1–3, 10, 12–16, 18, 19, 21–23, 25, 28–30, 35, 36, 38, 40–42, 46, 49, 51–53, 61, 62, 66, 67, 70, 73, 74, 79, 85, 105, 110, 111, 114, 117, 120, 123, 142, 145, 148
  - Suchsystem, 1, 2, 15, 26, 35, 51, 110, 112, 118, 129, 132, 145
  - Suchverhalten, 7, 9, 10, 15–19, 22, 26, 29, 37, 145
  - Suchwerkzeug, 61, 62, 113, 117, 130
  - Systemunterstützung, 2, 3, 7, 10, 30, 34, 37–39, 42, 45, 61, 67, 70, 105, 132, 134, 146
  - Taktik, 2, 13, 15, 16, 25–28, 36, 39, 56, 61, 67, 79, 81, 91, 97, 102, 105, 107, 120, 123, 124, 127, 142, 145, 146
    - File Structure Tactics, 17, 26, 85, 92
    - Idea Tactics, 17, 21, 25, 26, 66, 83, 85, 92
    - Monitoring Tactics, 17, 85
    - Search Formulation Tactics, 17, 26, 83, 85, 92
    - Term Tactics, 17, 26, 83, 85, 92
  - Termwolke, 64, 67, 83, 113, 146
  - Thesaurus, 65, 67, 75, 83, 127, 129, 130, 136
  - Tiled Panes, 68, 146
  - Transaktionslog, 43, 56
  - Unterstützungsniveau, 114, 115, 117, 118, 153
  - Verhaltensmodell, 7
  - Vorschläge, 2, 3, 28, 31, 42, 56, 79–81, 86, 88–91, 94–97, 99–102, 105, 114, 120, 123–125, 127, 128, 130, 134–136, 142, 146
  - Wissen
    - Aufgabenwissen, *siehe* Wissen, Domänenwissen
    - Domänenwissen, 7, 10–12, 23, 32, 35, 36, 41, 70, 73, 125
    - prozedural, 1, 2, 7, 12, 37, 41, 47, 49, 111, 120, 123, 125, 145, 147
    - Systemwissen, 7, 35, 41, 111
  - Zone der Intervention, 8

## Index

---

Zone of Proximal Development, 46,  
47, 110

Zwischenablage, 62, 64, 66, 67, 85,  
97, 108, 114, 117, 130, 146