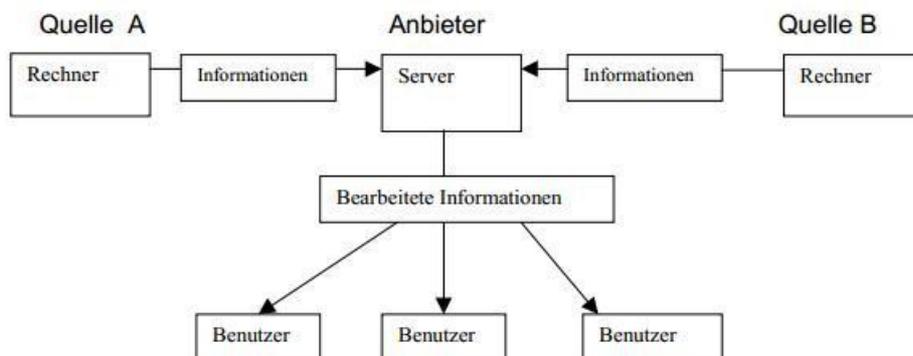


II - V Anwendungen der Informationserfassung und Bearbeitung ¹

II.1 Vorbemerkung

Wichtige Informationsquellen für den Wissenschaftler sind die im Internet zugänglichen bibliographischen Datenbanken und Suchmaschinen. Ihre auf Webseiten angezeigten Suchergebnisse, die mehrere tausend Treffer umfassen können, sind direkt verwertbar. Wie kann man auf diese zugreifen, verarbeiten und mit anderen Informationen und Daten verknüpfen, um zusätzliche Informationen zu gewinnen? Der Zugriff auf entfernte Rechner ist für Informationsanbieter des Internet kein Problem, sobald sie eine Zugriffserlaubnis auf die Daten eines anderen Anbieters haben, d.h. vom eigenen Rechner oder Server zu den entfernten Rechnern einer Quelle A oder B, wie das folgende Schema I zeigt.

Schema I: Datenübernahme und Bearbeitung bei Internetanbietern



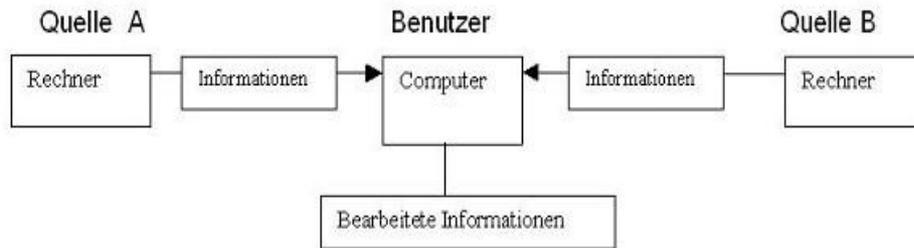
Der Rechner oder Server des Anbieters erhält Informationen aus den verschiedenen Quellen A oder B, bearbeitet und schickt diese an die einzelnen Computer der Benutzer.

Ihr eigener Computer kann zwar auch als Server fungieren, aber dieser hat normalerweise keinen direkten Zugriff auf den Rechner eines Anbieters. Ausserdem: Die Informations- und Datenbanksysteme im Internet verfügen über sehr leistungsfähige und schnelle Rechner, mit denen diese Informationen gespeichert, verarbeitet und gesendet werden. Mit diesen kann ihr eigener Computer nicht konkurrieren.

Andererseits sind die handelsüblichen Computer in den letzten Jahren erstaunlich leistungsfähig geworden, was die Schnelligkeit und die Speicherkapazität betrifft. Es ist also jetzt sogar sehr leicht möglich, zwar nicht auf einen entfernten Rechner, aber auf andere Weise direkt auf diese Informationen zuzugreifen und selbst sehr grosse Datenmengen in einfachen Formularen zu speichern. Das ist neuartig und erstaunlich und soll in dieser Schrift gezeigt werden. Schema II zeigt vereinfacht die Datenübernahme, wie sie mit der hier beschriebenen Technologie erfolgt.

¹ Die hier gezeigten Demonstrationsbeispiele und Webanwendungen finden Sie im Internet von der Seite <http://www.multisuchsystem.de/informationen.html> aus bzw. von den in den Fussnoten angegebenen Adressen!

Schema II: Datenübernahme und Bearbeitung mit der hier dargestellten Technik



Informationssysteme sind im allgemeinen in schwer verständlichen und schwer erlernbaren Programmiersprachen programmiert. Ausserdem ist der verwendete Code normalerweise nicht einsehbar. In diesem Bereich waltet, um es etwas überspitzt auszudrücken, eine elitäre und esoterische Atmosphäre.

Glücklicherweise existiert aber die sehr leicht verständliche und erlernbare Programmiersprache JavaScript. In einer Zeitung las ich vor Jahren, dass mit JavaScript jetzt quasi eine revolutionäre Programmiersprache entstanden sei, die es fortan auch Laien wie mir erlauben würde, eigene Programme zu entwickeln, also quasi eine Demokratisierung des bisher nur für Eingeweihte zugänglichen Programmierens. Das hat mich motiviert, mich mit dieser Programmiersprache zu beschäftigen. Nachdem ich mir die grundlegenden Kenntnisse angeeignet hatte, wollte ich die in JavaScript enthaltenen Möglichkeiten auf die vorhin genannten Informationen im Internet anwenden. Man kann diese zwar wie üblich in einer Datei speichern, kann aber auf diese mit JavaScript leider nicht direkt zugreifen. Ausserdem wäre es umständlich, immer erst Webseiten zu speichern, bevor man auf ihren Inhalt zugreifen kann. Erst nach einiger Überlegung kam ich, wohlgermerkt als Programmieranfänger, auf die Idee, die Informationen durch Copy und Paste zu übernehmen und in ein Formularfeld einzufügen. Als dies auch mit relativ grossen Datenmengen problemlos gelang, war mir klar, dass damit ein sehr wichtiger Schritt getan war. Wenn die Informationen erst einmal in einem Formularfeld erfasst sind, können sie beliebig weiter bearbeitet werden. Sollen sie allerdings anschliessend auch gespeichert werden, so ist dies nicht mit reinem JavaScript, sondern mit der Erweiterung durch die Active-X-Technologie möglich, die allerdings auf den IE beschränkt ist.

Auf diese Weise können auch sehr grosse Datenmengen sofort in Formularfeldern erfasst, bearbeitet und gespeichert werden. Dass Formularfelder als Datenbanken fungieren und ausschliesslich mit JavaScript und ActiveX operieren können, ist bisher noch ungewöhnlich und kaum bekannt. In Hand- und Lehrbüchern fand ich bisher dafür noch keine Beispiele. Bisher wird JavaScript eher für blosser Erweiterungen von Webseiten, für Animationen usw. eingesetzt. Als vollwertige Programmiersprache wird JavaScript nicht ernst genommen, oder vielleicht sogar als Spielzeug betrachtet. Andererseits hat sich JavaScript weiter entwickelt. Viele Erweiterungen sind hinzugekommen, die JavaScript sogar als sehr schwierig erscheinen lassen, die aber in dieser Schrift nicht berücksichtigt zu werden brauchen, weil alle hier besprochenen Anwendungen mit dem ganz einfachen JavaScript auskommen, das sich auch über Jahre hinweg kaum geändert hat.

Nach dem Durchlesen dieser Schrift sollten Sie in der Lage sein, selbst eigene

Informations- und Datenbanksysteme mit grossen Datenbeständen zu erstellen, was phänomenal und - sollte diese hier beschriebene Technik erst einmal grössere Verbreitung finden - geradezu revolutionär erscheint. Sie selbst werden hiermit aufgerufen, dazu beizutragen.

Mit Ihrem Computer übernehmen Sie direkt die Informationen der Quellen A und B. Diese werden in Ihrem Computer gespeichert oder nur temporär übernommen, nach verschiedenen Kriterien bearbeitet und durchsuchbar gemacht.

Die Quelle A sind in den von mir dargestellten Beispielen in Kap. V die von verschiedenen Anbietern (z.B. der EZB) angezeigten Zeitschriftenlisten, die Quelle B die von Datenbanken angezeigten Suchergebnisse. Diese beiden Quellen werden durch ein auf Webformularen basierendem JavaScript-Programm aufgeschlüsselt und die einzelne Elemente miteinander verknüpft. Vor allem werden die Zeitschriftentitel aus den Zitaten erfasst und mit den Titeln der übernommenen Zeitschriftenlisten verknüpft. So werden die zusätzlichen Informationen gewonnen, die in Datenbankergebnissen nicht oder nur unzureichend gegeben sind, z.B. die Verfügbarkeit von Volltexten und deren Zugänglichkeit. So können auch umfangreiche Literaturverwaltungssysteme leicht erstellt und diese wiederum mit der Zeitschriftenverwaltung verbunden werden, so dass ganz neuartige Systeme, wie es die *Contents Zeitschriften und Aufsatzverwaltung*² darstellt, entstehen.

II.2 Einfache Beispiele

Sobald sich in einem Texteingabefeld eines Webformulars Texte, Suchergebnisse von Datenbanken, Zeitschriftenlisten oder sonstige Daten befinden, können diese als durchgehende Zeichenkette in einem sogenannten Stringobjekt zusammengefasst und mit JavaScript oder einer anderen Programmiersprache beliebig verändert und bearbeitet werden. Wird z.B. eine Zeitschriftenliste von mehreren Tausend Seiten in ein solches Eingabefeld hineinkopiert, kann diese innerhalb eines Programmcodes als Stringobjekt in einer frei gewählten Benennung, z.B. A, erfasst und abgerufen werden:

A=document.forma.area.value;

Das Eingabefeld ist hier mit *document.forma.area* definiert, dessen Inhalt mit der Eigenschaft *value* ausgelesen wird.

Natürlich können sehr grosse Datenmengen nicht so leicht und schnell, von einer gewissen Grösse an auch gar nicht mehr, durch Copy und Paste in ein Eingabefeld übernommen werden. Dann ist das übliche Vorgehen zu wählen: die zu erfassenden Daten werden in einer Datei gespeichert und diese Datei wird anschliessend ausgelesen. Das Auslesen einer Datei ist aber nicht mit JavaScript, sondern nur mit dem ActiveX-Objekt des Internet Explorers und dies auch nur in einer lokalen Anwendung, d.h. im eigenen Computer, möglich. Von dieser Möglichkeit wird bei der Verwaltung der KnowledgeBase der hier dargestellten Datenbanksysteme auch vielfach Gebrauch gemacht, wie in Kap. V.4 erläutert wird.

Nachdem Daten und Informationen auf Webseiten oder im Textformat durch Copy und Paste in ein Formular übernommen worden sind, stehen sie zunächst nur temporär,

² <http://www.multisuchsystem.de/contentsLit.html>

also vorübergehend, für eine Bearbeitung zur Verfügung. Sollen sie dauernd zur Verfügung stehen, so kann dies, wie gesagt, mittels des ActiveX-Objektes geschehen. Eine kurzgefasste Einführung in dessen Anwendung finden Sie in Kap. I.

Das Erfassen von Daten mit Copy und Paste und mit Hilfe eines einfachen Webformulars erweist sich als ein besonders effektives Verfahren, das für viele Anwendungen der Computerarbeit praktikabel ist. Einige von ihnen werden hier vorgestellt, Natürlich werden dabei möglicherweise relevante, auf der Oberfläche einer Webseite nicht sichtbare, d.h. im SourceCode verborgene Informationen wie Hyperlinkverweise, Identifiers usw. nicht miterfasst. Man könnte zwar auch diesen verborgenen Quellcode anzeigen lassen und durch Copy und Paste übernehmen, was aber etwas umständlich und in diesen Beispielen auch nicht relevant ist, weil alle hier besprochenen Anwendungen auf solche zusätzlichen Informationen wie ISSN und andere Metadaten verzichten können. Eine kurzgefasste Einführung in das Programmieren mit JavaScript finden Sie in Kap. I.1.

II.2.1 Webformulare. Texte oder Liste in ein Eingabefeld einfügen

Webformulare sind grundlegende Bestandteile dieses Copy und Paste-Verfahrens. Ein Webformular in seiner einfachsten Form besteht aus mindestens einem Eingabefeld und einem Schalter, Button genannt. Dieser ist oft der Schalter zum Abschicken des Formulars, ein sogen. Submit-Button. Ein einfaches Formular kann aus blossem HTML-Code bestehen ohne JavaScript. Bei den meisten der in den Beispielen gezeigten Webformularen dagegen wird über diese Start-Buttons jeweils eine entsprechende JavaScript-Funktion ausgelöst, wodurch die in das Eingabefeld eingefügten oder eingegebenen Werte, Texte, Daten usw. erfasst und bearbeitet werden, wie das folgende Beispiel zeigt.

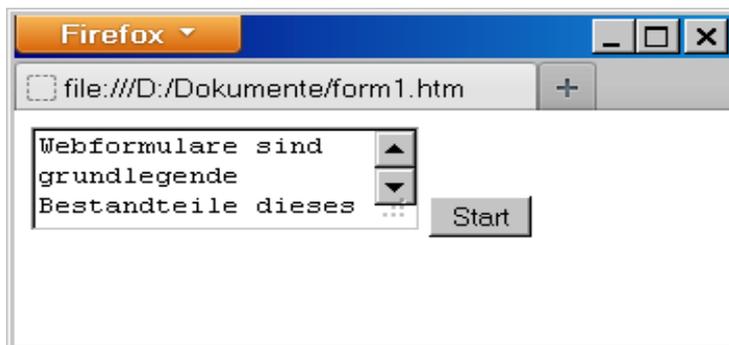


Abb. II.1 Formular mit Eingabefeld ³

Ein beliebiger Text wird markiert, kopiert (Copy) und in das Eingabefeld des Formulars eingefügt (Paste). Klickt man auf den Start-Button, wird der Text ohne jede Formatierung, aber sonst in der gleichen Weise angezeigt wie im Original. Das gleiche geschieht, wenn Sie die Oberfläche einer Webseite übernehmen. Auch hier werden alle Formatierungen ausgelassen. Man kann dieses mit verschiedenen Web- oder Dokumentseiten ausprobieren.

³ <http://www.multisuchsystem.de/form1.htm>

II.2.2 Trefferliste einer Datenbank (Current Contents) kopieren und einfügen

Suchfrage: formation/%3Cin%3Eti
Suche in folgenden Datenbeständen gestartet:
[Current Contents Connect](#)

Treffer 1 - 100 von 48262 for SET #1 (TS=

Masek, J; Turek, I; Kudrnovsky, J, et al.
[Compositional dependence of the formation energies of substitutional and interstitial Mn in partially compensated \(Ga,Mn\)As](#)
ACTA PHYS POL A, 105: (6) 637-644 JUN 2004

Beljonne, D; Ye, AJ; Shuai, Z, et al.
[Chain-length dependence of singlet and triplet exciton formation rates in organic light-emitting diodes](#)
ADV FUNCT MATER, 14: (7) 684-692 JUL 2004

Shin, HJ; Jeong, DK; Lee, JG, et al.
[Formation of TiO2 and ZrO2 nanotubes using atomic layer deposition with ultraprecise control of the wall thickness](#)
ADVAN MATER, 16: (14) 1197-+ JUL 19 2004

Fügt man diese Liste vollständig per Copy und Paste in das obige Eingabefeld ein und klickt man auf den Start-Button, so stellt die Alert-Anzeige die Liste mit allen Zeilenumbrüchen dar. In der anschliessend unformatiert ausgeschriebenen Darstellung werden die Zeilenumbrüche durch Sternchen ersetzt:

```
Suchfrage: formation/%3Cin%3Eti*Suche in folgenden Datenbeständen gestartet:**Current Contents Connect**Current Contents Connect**Treffer 1 - 100 von 48262 for SET #1 (TS=***Masek, J; Turek, I; Kudrnovsky, J, et al.*Compositional dependence of the formation energies of substitutional and interstitial Mn in partially compensated (Ga,Mn)As*ACTA PHYS POL A, 105: (6) 637-644 JUN 2004***Beljonne, D; Ye, AJ; Shuai, Z, et al.*Chain-length dependence of singlet and triplet exciton formation rates in organic light-emitting diodes*ADV FUNCT MATER, 14: (7) 684-692 JUL 2004***Shin, HJ; Jeong, DK; Lee, JG, et al.*Formation of TiO2 and ZrO2 nanotubes using atomic layer deposition with ultraprecise control of the wall thickness
```

Die einzelnen Treffer sind durch 3 Sternchen von einander getrennt, die einzelnen Teile innerhalb eines Treffers durch 1 Sternchen. Dadurch wird die Struktur der Liste gekennzeichnet. Eine wichtige Voraussetzung für die weitere Bearbeitung der Daten.

Der HTML-Quellcode für das oben verwendete einfache Webformular mit einem mehrzeiligen Eingabefeld und einem Startbutton sieht in vereinfachter, aber funktionsfähiger Schreibweise so aus:

```
<html>  
<form name=forma>  
<textarea name=area>  
</textarea>  
<input type=button value=Start onclick="starte()">  
</form>  
</html>
```

Innerhalb dieses Codes ist der Eventhandler *onclick* bei der Definition des Buttons eingefügt. Durch das Klicken auf diesen Button wird über die frei gewählte Funktionsbezeichnung *starte()* das JavaScript-Programm gestartet.

Der JavaScript-Quellcode für dieses Beispiel, ebenfalls in etwas vereinfachter Schreibweise:

```
<script>
function starte()
{
T=document.forma.area.value;
alert(T);
Ta=T.replace(/n/g, "");
document.write(Ta);
}
</script>
```

Beachten Sie, dass der hier angegebene JavaScript-Code sich innerhalb des oben angegebenen HTML-Codes, d.h. zwischen *<html>* und *</html>* befindet und mit diesem zusammen den gesamten Quellcode für dieses Formular bildet!

Das in einem Webformular enthaltene mehrzeilige Eingabefeld⁴ ist durch die Paste-Aktion fähig, auch eine grosse Datenmenge von mehreren Tausend KB auf einmal aufzunehmen. Diese verblüffende Tatsache kann man ausnutzen, um die im Internet angezeigten Trefferlisten von Datenbanken oder Listen von Zeitschriften, Zitaten, Texten usw. zu erfassen und zu bearbeiten. Auch im Textformat vorliegende Dateien von Aufsätzen, ja ganze Bücher können auf diese Weise erfasst, durchsucht und verändert werden. Zum Beispiel können die für ein Register relevanten Begriffe erfasst und alphabetisch geordnet werden, das Vorkommen von bestimmten Wörtern kann untersucht, eine Abkürzungsliste erstellt werden usw.

II.2.3 Zugriff auf fest eingespeicherten Text im Eingabefeld

Man kann den zu bearbeitenden Text wie oben beschrieben in ein Eingabefeld einfügen. Man kann ihn ebenso als festen Wert einspeichern, um auf diesen wiederholt, z.B. zwecks einer Recherche, zuzugreifen. Dazu wird der Text zwischen den beiden Textarea-Tags des Quellcodes im Editor eingefügt und anschliessend der ganze Quellcode zusammen mit dem eingefügten Text gespeichert, wie Abb. II.2 zeigt.

Der hier eingespeichert Text besteht aus 5 Sätzen. Der Anfang ist noch im Eingabefeld zu erkennen. Über den Bildlaufpfeil kann man die folgenden Teile des Textes sichtbar machen. Der hier eingeschriebene Text kann auch aus sehr vielen Sätzen bestehen. Es können auch sehr umfangreiche Texte, Listen oder sonstige Daten in ein Formular fest eingespeichert werden, so dass dieses Formular wie eine Datenbank fungiert. Dabei werden die zu speichernden Daten zwischen die beiden Textarea-Tags des Formulars hineinkopiert und dieses dann gespeichert.

⁴ Inzwischen können auch einzeilige Eingabefelder grosse Datenmengen aufnehmen, was noch vor einiger Zeit nicht möglich war



Abb. II.2 Formular mit eingespeichertem Text

Der HTML-Quellcode sieht so aus:

```

<textarea name=area>
Das Internet - und innerhalb .....
.....

es folgt der ganze hineinkopierte Text!
</textarea>
  
```

Wenn Sie das Formular starten, wird in einem Alert-Fenster zunächst die Anzahl der Sätze angezeigt, danach der in dem Eingabefeld enthaltene Text ausgeschrieben. Diesmal werden nicht die Zeilenumbrüche, sondern die einzelnen Sätze jeweils durch ein Sternchen gekennzeichnet.

II.2.4 Eine einfache Suchmaschine

Bei einer Suchmaschine werden ein oder mehrere Begriffe in einem gegebenen Text, einer Liste, einem Datenbankspeicher usw. gesucht. Dabei ist es gleichgültig, ob das zu Suchende in einem gegebenen Text oder in einer umfangreichen strukturierten Datenbank zu suchen ist. Das Prinzip ist immer das gleiche und im Grunde sehr einfach. In diesem Beispiel ist der oben genannte zu durchsuchende Text quasi als Datenbank zwischen den beiden Textarea-Tags eines Formulars, wie bereits gezeigt, gespeichert. Im selben Formular muss ausserdem für die zu suchenden Begriffe ein Eingabefeld vorhanden sein, in diesem Fall ein einzeiliges Eingabefeld :

```
<input type=text name=eingabe>
```

Diese Befehlszeile bezeichnet immer ein einzeiliges Eingabefeld, hier mit der frei gewählten Benennung *eingabe*.

Die Suchfunktion wird mittels eines JavaScripts ausgeführt, das wiederum über einen Button mittels des Event-Handlers *onclick* gestartet wird. Der Code dazu lautet:

```

function starte3()
{
T=document.forma2.S1.value;
E=document.forma3.eingabe.value; // Das in das Eingabefeld eingegebene Suchwort wird über
                                // die Value-Eigenschaft in der Variablen E gespeichert
TA=T.split(/\.\s/);
  
```

```

alert(TA.length);
for(x=0;x<TA.length;x++)
{
ES=new RegExp (E,"gi"); // Das Suchwort E soll überall (g=global) und ohne Rücksicht auf
                        Gross- oder Kleinschreibung (i=insensible) gesucht werden; dafür
                        wird der reguläre Ausdruck ES gebildet
Treffer=TA[x].match(ES); //Mit der Match-Methode wird E in jedem Arrayelement TA[x] gesucht
                        und das Ergebnis in Treffer gespeichert
if(Treffer)           // Falls Treffer gefunden werden
{
                        //Die beiden folgenden Anweisungen befinden sich innerhalb einer
                        For-Schleife, daher geschweifte Klammern
TB=TA[x].replace (ES,"<b>"+E+"</b>"); // ES durch fett geschriebenes E ersetzt
document.write("<br>"+TB+".");      //Ausschreiben der Treffer TB
}}}

```

Als Ergebnis werden alle Sätze angezeigt, in denen die markierten Suchbegriffe vorkommen.

II.2.5 Wörter anzeigen mit Häufigkeit des Vorkommens

Eine weitere Anwendungsmöglichkeit kann darin bestehen, für einen längeren Text oder sogar ein ganzes Buch eine Wortliste zu erstellen, die evtl. für die Anlegung eines Registers benutzt werden kann. Das dafür erforderliche Programm-Skript, das dem vorigen Skript etwas ähnlich ist, kann etwa so aussehen:

```

<script>
function starte4()
{
T=document.forma2.S1.value;
T1=T.replace(/ä/g,"ae"); //Hier und in den folgenden Zeilen: Umlaute und ss werden aufgelöst
T2=T1.replace(/ü/g,"ue");
T3=T2.replace(/ö/g,"oe");
T4=T3.replace(/ss/g,"ss");
TR=T4.match(/[A-Z]\w+/g); // //Alle Wörter mit Grossbuchstaben am Anfang werden global (g)
gesucht
TA=TR.sort(); // Die gefundenen Wörter werden alphabetisch sortiert
for(x=0;x<TA.length;x++)
if(TA[x+1]!=TA[x]) // Mehrfach gefundene Wörter bleiben unbeachtet
{
A=new RegExp(TA[x],"g"); // Jedes Wort soll nochmals global gesucht werden
Treffer=T4.match(A);
document.write("<br>"+TA[x]+" "+Treffer.length); //Treffer.length gibt die Häufigkeit des
Vorkommens eines Wortes an}}
</script>

```

Als Ergebnis werden alle Wörter angezeigt, die in dem oben (1.2) eingespeicherten Text vorkommen und sofern sie mit Grossbuchstaben beginnen. Dass hierbei auch die am Anfang eines Satzes, d.h. nach einem Punkt grossgeschriebenen Wörter wie *Aber*, *Da*, *Das* usw. ausgeschrieben werden, lässt sich in diesem Beispiel leider nicht vermeiden. Zusätzlich wird hinter jedem Wort angegeben, wie häufig dieses vorkommt.

III Listen und Datenbanken

In Datenbanken sind Daten gesammelt und gespeichert. Daten können z.B. Texte, Zeitschriftentitel oder Aufsatzzitate sein. Wesentlich für Datenbanken ist, dass ihre Inhalte strukturiert sind und dass sie bearbeitet, ergänzt, verändert und vor allem durchsucht werden können.

Datenbanken sind Tabellen vergleichbar:

Index	Datenfeld 1	Datenfeld 2	Datenfeld 3
1 [Datensatz:]	Autor	Titel	Zeitschrift
2 [Datensatz:]	Autor	Titel	Zeitschrift
3 [Datensatz:]	Autor	Titel	Zeitschrift

Die obige Tabelle zeigt vereinfacht das Schema einer Aufteilung von 3 Zeitschriftenaufsätzen (= Datensätzen) auf die Datenfelder Autor, Titel und Zeitschrift. Dazu kommt eine vorangehende Spalte für die Index-Nummer.

Die einzelnen Datensätze oder Elemente einer Datenbank müssen eine Indexierung und zumindest ein Datenfeld aufweisen. Die Indexierung erlaubt die Zugriffsmöglichkeit auf einen Datensatz, und in den Datenfeldern ist der Inhalt eines Datensatzes enthalten. Datenbanken weisen eine kleinere oder grössere Strukturierung auf, d.h. sie sind durch mehrere oder eine Vielzahl von Datenfeldern gegliedert, was einer Tabelle mit entsprechend vielen Spalten entspricht. Listen z.B. von Zeitschriftentiteln können unter bestimmten Bedingungen wie Datenbanken behandelt werden. Eine einfach untereinander geschriebene Liste von Zeitschriftentiteln ist zunächst noch keine Datenbank. Es fehlt die Einteilung in Index und Dateninhalt. Wie dies auf sehr einfache Weise mit Hilfe des Arrayobjekts erreicht werden kann, wird im folgenden erläutert.

Eine Zeitschriftenliste weist bereits die für ein Arrayobjekt typischen Eigenschaften auf, d.h. sie besteht aus einer Anzahl von Elementen bzw. Datensätzen *E* mit jeweils verschiedenem Inhalt:

E1 (.....) E2(.....) E3 (.....) E4 (.....) usw.

Die einzelnen Elemente brauchen noch nicht einmal näher mit Nummern usw. bezeichnet zu werden, um aus ihnen ein Array-Objekt zu bilden. Es genügt, dass sie durch ein bestimmtes Zeichen oder durch eine bestimmte Anordnung voneinander abgegrenzt sind.

III.1 Erfassen von Zeitschriftentitellisten

Überwiegend werden Zeitschriftentitellisten durch einfaches Untereinanderschreiben angezeigt, manchmal mit, manchmal ohne zusätzliche Numerierung, wie z.B. die im WWW veröffentlichten Zeitschriftenlisten vieler Verlage oder die Fachlisten der Elektronischen Zeitschriftenbibliothek (EZB). Hier ein Ausschnitt aus einer ihrer Webseiten:

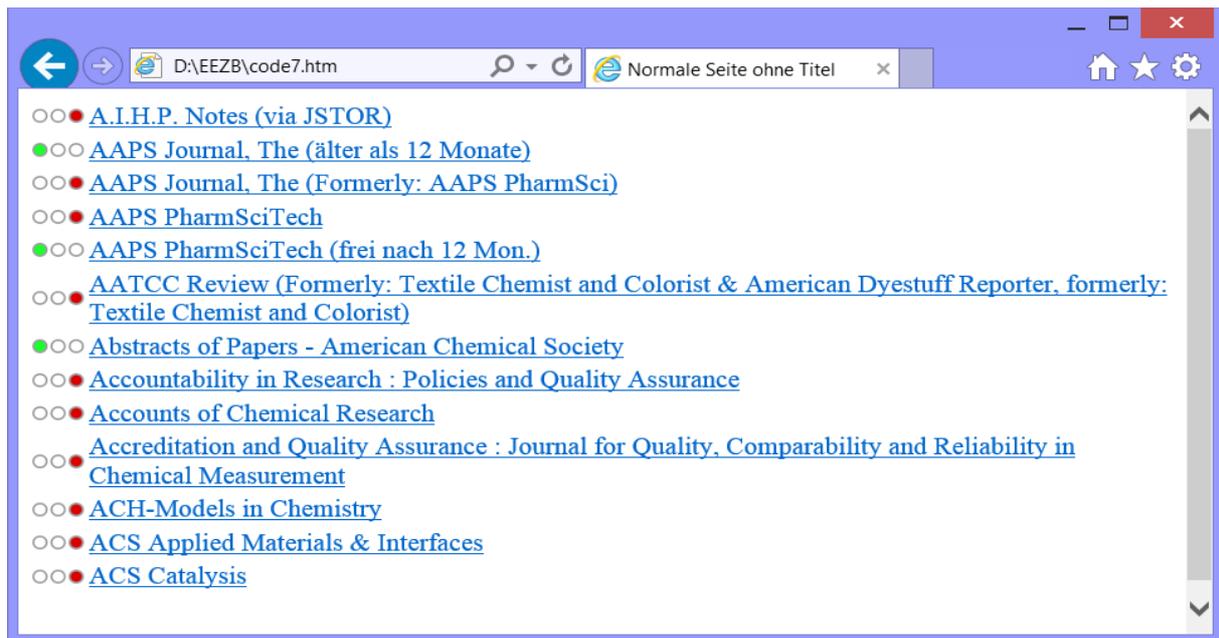


Abb. II.3 Zeitschriftenliste der EZB

Wenn diese Liste kopiert und in das Eingabefeld des Formulars⁵ eingefügt wird, so werden nach Klicken auf den Start-Button die einzelnen Titel der Liste angezeigt und ausgeschrieben. Der Code hierfür ist denkbar einfach:

```
a=document.forma.area.value;// die eingelesene Liste
aa=a.split(\n); // Array gebildet durch neue Zeile
alert(aa.length); // Zahl der Elemente
for(x=0;x<aa.length;x++) // Anzeige und Ausschreiben in der For-Schleife
{
alert(aa[x]);
document.write("<br>" +aa[x]);
}
```

Dieses Beispiel zeigt, dass, sobald eine einfach untereinander geschriebene Liste von Zeitschriftentiteln einer Webseite durch Copy und Paste in das Formularfeld eingefügt ist, sie mit der Value-Methode als durchgehende Zeichenkette bzw. Stringobjekt gelesen werden kann, und dass sofort durch die Zeilenstruktur ein Array gebildet wird, dessen Elemente dann in der For-Schleife nacheinander angezeigt werden können. Das ist das verblüffend einfache Prinzip, wie die in Kap. V beschriebenen Datenbanken erstellt werden können. Natürlich sind im Einzelfall noch erhebliche Anpassungen vorzunehmen, wie dort beschrieben.

Relevante Informationsquellen - speziell für die Literatursuche und -beschaffung -sind neben den Listen von Literaturzitataten bzw. den Suchergebnissen von Datenbanken eben diese Zeitschriftenlisten, die von Verlagen und Informationsdiensten oder auch, wie im Falle der EZB, von einer Bibliothek angeboten werden. Wenn diese verschiedenen Quellen in geeigneter Weise bearbeitet und miteinander verbunden werden, können zusätzliche Informationen entstehen, die in den originalen Listen nicht enthalten sind. So können z.B. zahlreiche Hyperlinkverknüpfungen entstehen, wie in Kap. IV gezeigt wird.

⁵ form2.htm

III.1.2 Alphabetisches Ordnen einer Liste



Firefox

Normale Seite ohne Titel

>Geol Soc Am
Bull=00167606*BTU *GÖ:

Start

Im Eingabefeld ist eine ungeordnete Zeitschriften-Liste untergebracht.

Klicken Sie auf den Start-Button, wird diese in einem Alert-Fenster angezeigt und anschließend in alphabetischer Anordnung ausgeschrieben.

Abb. II.4 Formular zum alphabetischen Ordnen einer Liste

Eine Listen- oder Datenbankbearbeitung kann darin bestehen, dass aus einer gegebenen Datenmenge die relevanten Teile entnommen und neu angeordnet werden, dass Teile verändert oder entfernt werden, dass z.B. bestimmte Zeichen wie Bindestriche oder Klammern oder bestimmte Buchstabenfolgen entfernt werden. Ein besonders einfaches und gleichzeitig sehr effektives Beispiel ist das alphabetische Anordnen einer Liste. In JavaScript müssen dafür nur wenige Zeilen geschrieben werden.

```
1) T=document.forma.area.value;  
2) Ta=T.split(">"),  
3) Tb=Ta.sort(),  
4) for(x=0;x<Tb.length-1;x++)  
5) if(Tb[x]!=Tb[x+1])  
6) document.write("<br>" + Tb[x]);
```

Die in das Eingabefeld *area* eingefügte Liste wird in Zeile 1 ausgelesen. Zeile 2 macht aus der Zeichenkette oder dem String *T* über das Trennungszeichen ">" ein Array. In 3) wird dieser Array alphabetisch angeordnet. In der For-Schleife wird jedes Element *Tb[x]* mit dem nächstfolgenden *Tb[x+1]* verglichen und nur dann in 6) ausgeschrieben, wenn sie verschieden sind, ausgedrückt durch *!=*.

So werden neben der erreichten Alphabetisierung gleichzeitig auch Doppelintragungen entfernt. Dieses ist, wie schon gesagt, ein besonders einfaches, aber auch sehr effizientes Beispiel, das Sie an einem Beispiel ausprobieren können.⁶

⁶ <http://www.multisuchsystem.de/code8.htm>

III.1.3 Zwei Listen miteinander vergleichen

Um kompliziertere Bearbeitungen handelt es sich, wenn die einzelnen Elemente innerhalb einer Liste verglichen, Einträge mit abweichenden oder identischen Teildaten gekennzeichnet werden usw. Dazu gehört auch die Veränderung oder Erweiterung einer Liste durch bestimmte Daten, z.B. die Ergänzung von Zeitschriftentiteln durch die ISSN, was dadurch erreicht werden kann, dass die zu ergänzende Liste mit einer anderen Liste verglichen und aus dieser die fehlenden ISSN-Daten entnommen werden.

III.1.3.1 Zwei Zeitschriftenlisten über die ISSN miteinander vergleichen

Im folgenden Beispiel wird die Bestandsliste elektronischer Zeitschriften einer einzelnen Bibliothek mit einer umfassenden Bestandsliste von über 9000 Titeln verglichen.

Liste A: Ein Ausschnitt aus der Liste einer Bibliothek (UB Bochum):

```
>Annals of Nuclear Energy 03064549 03064549
>Annals of Oncology (-2001) 09237534 09237534
>Annals of Operations Research 02545330
>Annals of Otolaryngology, Rhinology & Laryngology 00034894
```

Liste B: Ein Ausschnitt aus der Bestandsliste des Multisuchsystems:

```
>ANNALS OF NUTRITION AND
METABOLISM=02506807=KARG*DD*ER*HL*J*MHH*OL*RE*TUM*TIB*GÖ
>ANNALS OF OCCUPATIONAL HYGIENE=00034878*FREI(AFTER 2 YEARS
>ANNALS OF ONCOLOGY=09237534=HW=KLUW*AA*BHU*ER(-2001*TUM(-2001*TIB(-2001*FZK(-
2001*GÖ(-2001
>ANNALS OF OPERATIONS
RESEARCH=02545330=KLUW*AA*BHU*ER*HA*HL*J*MHH*MP*OL*POT*RE*TUM*TIB*GÖ
>ANNALS OF OTOLARYNGOLOGY RHINOLOGY AND LARYNGOLOGY=00034894=PUB*AA*DD
```

Liste A gibt ausser dem Titel eine oder mehrere ISSN sowie ggf. eine zeitliche Beschränkung (z.B. -2002) an. Die Bestandsliste B enthält ausserdem eine Anzahl von Kennzeichen für die besitzenden Bibliotheken (*AA*BHU*ER*HA*HL), Verlagsangaben (=KLUW =KARG =PUB usw.) sowie ebenfalls zeitliche Angaben. In Liste B ist im allgemeinen nur jeweils eine ISSN angegeben. Liste B kann auf Grund ihrer Strukturierung durch entsprechende Zeichen (> =, * () als Datenbank fungieren.

Die einzelnen Titel der Liste B sollen mit denen der Liste A verglichen, bei identischen Titeln soll ein Kennzeichen hinzugefügt werden, das den Besitz für eine bestimmte Bibliothek angibt, hier *BO für die UB Bochum.

Da beide Listen bereits alphabetisch geordnet sind und von wenigen Ausnahmen abgesehen jeder Zeitschriftentitel nur einmal aufgeführt wird bzw. mehrere gleichlautende Titel hinter einander stehen, braucht nur eine der beiden Listen in ein Array umgewandelt zu werden. Auf die andere Liste kann als einheitliche Zeichenkette, als Stringobjekt, zugegriffen werden. Es wird also in diesem Fall nur für die Liste B ein Array gebildet. Aus jedem Element wird die ISSN ermittelt, die jeweils in der gesamten Liste A gesucht wird. Liste A bleibt also als String bestehen.

Für diese Prozedur werden im wesentlichen nur wenige Codezeilen benötigt:

```

A=document.forma.area.value; // Auslesen der A-Liste
TB=document.forma.area2.value; // Auslesen der B-Liste
TBa=TB.split(">"); // aus der B-Liste wird ein Array gebildet, Unterscheidungsmerkmal ist das
>-Zeichen
for(x=0;x<TBa.length;x++) // Definition der For-Schleife
{
is=TBa[x].match(/d\d\d\d\d\d\d\d/); // aus jeder Zeile der B-Liste wird die ISSN gelesen
p=TA.indexOf(is); // dies ISSN wird in der A-Liste gesucht
if(p!=-1)
Tx="<b>"+TBa[x]+"*BO+</b>"; // falls gefunden, wird an die Zeile das Kennzeichen *BO
angefügt
if(p==-1)
Tx=TBa[x]; // falls nicht gefunden, bleibt die Zeile unverändert
document.write("<br>"+Tx); // die gesamte Bestandsliste wird mit den Hinzufügungen
ausgeschrieben

```

Bei der Demonstration dieses Beispiels (Abb. II.5) werden beide Listen nur bis zum Buchstaben A verglichen. Zusätzlich zum oben gezeigten Code kommen nur noch die Code-Zeilen für einen Trefferzähler und ein neu zu öffnendes Fenster hinzu. In der ausgeschriebenen Bestandsliste sind die Treffer markiert, am Ende einer Trefferzeile ist als Kennzeichen *BO+ hinzugefügt. Die Trefferzahl wird bei der Demonstration ganz unten angegeben.⁷

III.1.3.2 Zwei Zeitschriftenlisten mittels des Zeitschriftentitels miteinander vergleichen

Zeitschriftenliste der Bibliothek:	>Abacus 14676281 00013072 >Accountancy 00014664 >Accounts of Chemical Research 15204898 00014842 >Acta Anaesthesiologica Scandinavica
Bestandsliste des Multisuchsystems:	>AACN CLINICAL ISSUES=10790713*DD >AAPG BULLETIN=01491423*GÖ >AAPS PHARMSCI=15221059*FREI >AAPS PHARMSCITECH=15309932*FREI >ABA BANKING J=01945947*ER

Starte

Abb II.5 Formular für den Vergleich von 2 Zeitschriftenlisten

Wesentlich schwieriger ist der Vergleich zweier Listen, wenn nur eine dieser Listen die Zeitschriftentitel mit ISSN aufführt und wenn deren Schreibweisen mehr oder weniger von einander abweichen. Im oberen Beispiel war die Zeitschriftenliste der Elektronischen Zeitschriftenbibliothek (EZB) zu sehen, die in dieser Anzeigeform

⁷ <http://www.multisuchsystem.de/code9.htm>

lediglich die Titel der Zeitschriften ohne ISSN aufführt. Um diese Liste mit einer Gesamtliste mit ISSN, die ca. 29000 Titel umfasst⁸, zu vergleichen, müssen die Schreibweisen beider Listen einander angeglichen werden. Die oben genannte Bestandsliste B basiert übrigens ebenfalls im wesentlichen auf der Schreibweise dieser Gesamtliste.

Im folgenden wird die durch das Bearbeitungsprogramm erreichte Angleichung gezeigt. Auf die in der Schreibweise der EZB-Liste aufgeführten eingegebenen Titel folgen die durch das Programm ausgeschriebenen Titel in der Schreibweise der Gesamtliste G, die ebenfalls in einigen Punkten (z.B. bei Serienangaben) abgeändert wurden.

Auf Doppelpunkt, Bindestrich oder Schrägstrich folgende Zusätze zum Titel werden entfernt:

METABOLISM - CLINICAL AND EXPERIMENTAL
METABOLISM 00260495

BJU INTERNATIONAL : BRITISH J OF UROLOGY
BJU INTERNATIONAL 14644096

ARCHIVE OF APPLIED MECHANICS / INGENIEUR ARCHIV
ARCHIVE OF APPLIED MECHANICS 09391533

In runden und eckigen Klammern Stehendes wird entfernt:

ACM TRANSACTIONS ON DATABASE SYSTEMS (ACM TODS)
ACM TRANSACTIONS ON DATABASE SYSTEMS 03625915

J OF SYNCHROTRON RADIATION [SYNCHROTRON RADIATION ONLINE]
J OF SYNCHROTRON RADIATION 09090495

Unterreihen werden in den EZB-Listen in sehr unterschiedlicher Weise angegeben. SERIES, PART und SECTION werden in beiden Listen entfernt, in den EZB-Listen ausserdem die ergänzende Angabe hinter dem Doppelpunkt. Serienhinweise werden normalerweise nur mit einem Buchstaben oder einer Ziffer angegeben. Beispiele:

PROCEEDINGS ROYAL SOCIETY OF LONDON SERIES A : MATHEMATICAL PHYSICAL AND ENGINEERING SCIENCES (1996 -)
PROCEEDINGS ROYAL SOCIETY OF LONDON A 13645021

PHYSICA A : STATISTICAL MECHANICS AND ITS APPLICATIONS
PHYSICA A 03784371

PHYSICA STATUS SOLIDI (A) - APPLIED RESEARCH
PHYSICA STATUS SOLIDI A 00318965

J DE PHYSIQUE II
J DE PHYSIQUE 2=1155-4312

Die Demonstration zeigt Ihnen, wie es funktioniert: ⁹

⁸ Diese Liste beruht im wesentlichen auf der Zeitschriftenliste der JADE-Datenbank

⁹ <http://www.multisuchsystem.de/code10.htm>

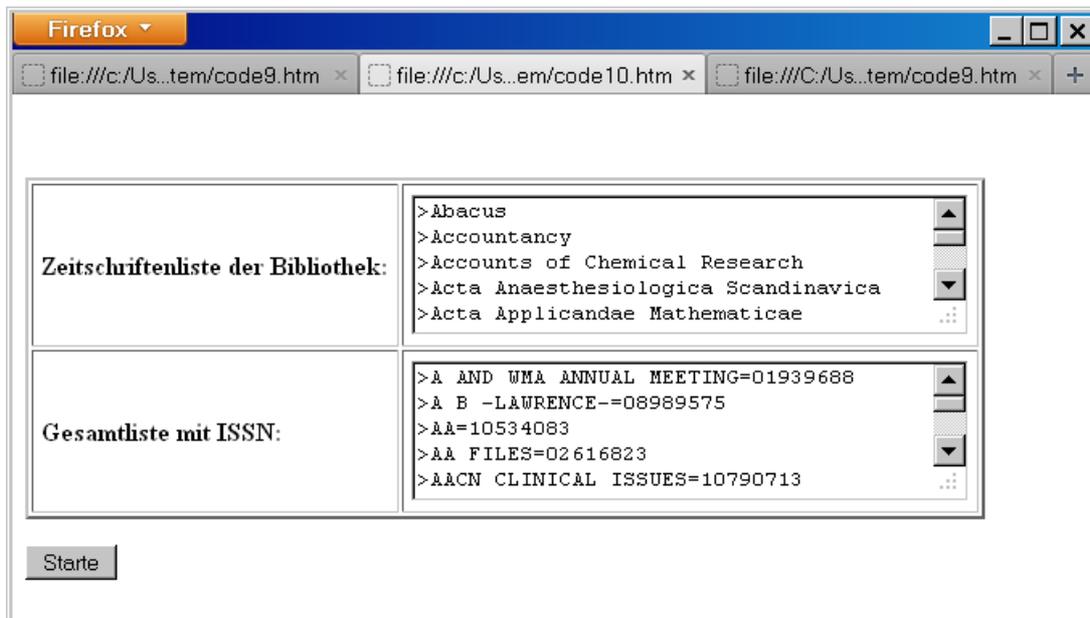


Abb. II.6 Vergleich einer Liste mit einer Gesamtliste

Bei diesem Beispiel werden von 169 Titeln 133 Treffer erzielt. Der überwiegende Teil der nicht gefundenen Titel ist in der Gesamtliste nicht enthalten. Einzelne Titel wurden wegen einer abweichenden Schreibweise nicht gefunden: z.B. *ANGLO SAXON ENGLAND* anstatt *ANGLOSAXON ENGLAND* , *ANTI CANCER DRUGS* anstatt *ANTICANCER DRUGS* , *ARCHIVES OF SEXUAL BEHAVIQR* anstatt *ARCHIVES OF SEXUAL BEHAVIQR* .

Das hier verwendete Programm in einer stark vereinfachten Version:

```

<script>
function starte()
{
treffer=0;
F=window.open("fenster.htm");
T=document.forma.area.value;// Einlesen der EZB-Liste
G=document.forma.area2.value; // Einlesen der Gesamtliste mit ISSN
Ga=G.replace(/s- +/g,""); //die zwischen Bindestrichen stehenden Erscheinungsorte werden entfernt
Gb=Ga.replace(/-\s/g," "); // Bindestrich mit folgender Leerstelle (bei Körperschaften) werden durch Leerstelle ersetzt
Gc=Gb.replace(/JOURNAL/g,"J");
TA=T.split(/r/); // Zeichenkette T wird in ein Array TA verwandelt; Zeilenumbruch (/r) als Trennungsmerkmal
for(x=0;x<TA.length;x++)
{
T1=TA[x].replace(/ä|Ä/g,"a");
T2=T1.replace(/ö/g,"o");
T3=T2.replace(/ü/g,"u");
T4=T3.replace(/ss/g,"ss");
T5=T4.replace(/ç/g,"c");
T6=T5.replace(/é/g,"e");

T7=T6.replace(/of\sthe\s/g,"");
T8=T7.replace(/,\sThe\b|,\sthe\b|,\sTHE\b|,\sLa\b|,\sLe\b|,\sDer\b|,\sDie\b|de\s/g,"");
}
}

```

```

T9=T8.replace(/Journal|journal|JOURNAL/g, "J");
T10=T9.replace(/&/g, "and");
T11=T10.replace(/\(.+\)|s|:.+|,/g, ""); // Zeichenfolgen in runden Klammern oder nach
Doppelpunkt sowie Kommas werden entfernt
T11=T10.replace(/-/," ");
Z=T12.toUpperCase();
Za=Z+"=";
ZA=Za.replace("=", "") // z.B. ZA=>ABACUS=
p=Gc.indexOf(ZA);
if(p!=-1)
{
treffer++;
pa=Gc.indexOf(">",p+2); // in der abgewandelten Gesamtliste Gc wird das nächstfolgende
>Zeichen ab der Position p+2 gesucht
ZG=Gc.substring(p,pa); // daraus wird der Teilstring gebildet
is=ZG.match(/=\d\d\d\d\d\d\d\d\d\d/w/); // die ISSN wird entnommen
F.document.write("<br>"+Z+is); // die gefundenen Treffer werden ausgeschrieben
}
if(p==-1)
document.write("<br>"+Z); // die nichtgefundenen Treffer werden ausgeschrieben
}
F.document.write("<p><b>"+treffer+" Treffer</b>");
</script>

```

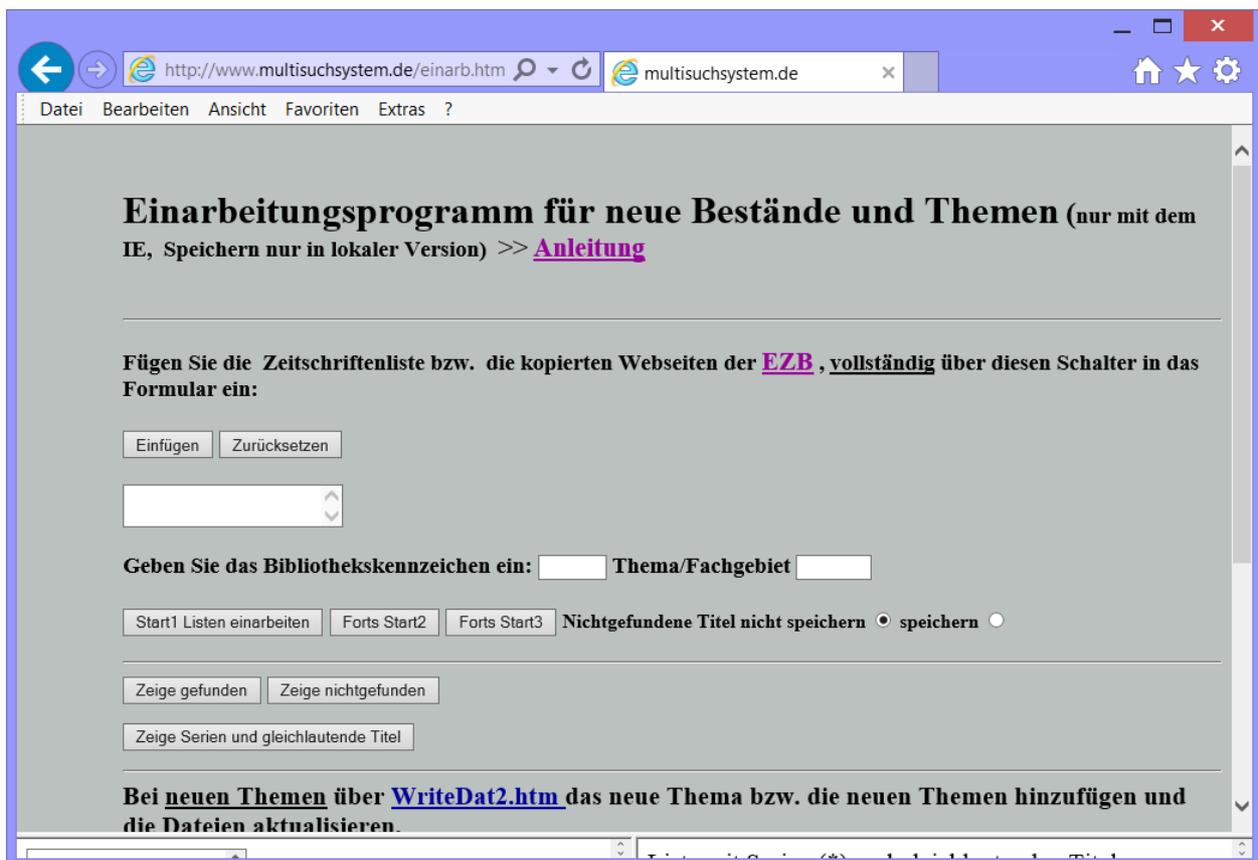


Abb. II.7 Einarbeitungssystem für Zeitschriftenlisten

Die erweiterte Version dieses Programms, das mit einer weit grösseren Genauigkeit von den hier gesuchten Titeln alle bis auf ganz wenige ermittelt, ist in das Einarbeitungs-

system des Multisuchsystems integriert, über das EZB-Listen ohne ISSN direkt in eine Bestandsliste eingearbeitet werden: Abb.II. 7 ¹⁰

Dieses Beispiel zeigt, dass es durchaus möglich ist, Zeitschriftenlisten allein über die Titel zu vergleichen. Die Trefferquote ist dabei nicht viel niedriger als bei einem Vergleich über die ISSN. Allerdings ist dafür ein wesentlich grösserer Programmieraufwand nötig.

IV Linksysteme

IV.1 Verknüpfungen

Ein Linksystem ist zunächst nichts anderes als ein Verlinkungs- oder Verknüpfungssystem im WWW, basierend auf der Hyperlinktechnik. Man unterscheidet statische und dynamische Verknüpfungen. Bei Hyperlinkverweisen handelt es sich überwiegend um statische Verknüpfungen. Dynamische Verknüpfungen dagegen entstehen dadurch, dass eine gegebene Internetadresse dynamisch durch bestimmte Daten verändert und dann angesteuert wird. Das geschieht z.B. bei der Übermittlung von Suchwörtern an Suchmaschinen und Datenbanken. Dazu ein einfaches Beispiel.

Die Internetadresse des Suchformulars von Google ist:

```
http://www.google.de/search?q=+&ie=UTF-8&oe=UTF-8&hl=de&btnG=Google+Suche&meta=
```

An der Stelle *q=* wird nun ein Stichwort eingefügt, etwa *production*.

Die Adresse ändert sich dann in:

```
http://www.google.de/search?q=production+&ie=UTF-8&oe=UTF-8&hl=de&btnG=Google+Suche&meta=
```

Mit JavaScript lassen sich sehr leicht solche Verknüpfungen konstruieren. Dabei werden im wesentlichen folgende Codezeilen verwendet:

```
L="http://www.google.de/search?q=+&ie=UTF-8&oe=UTF-8&hl=de&btnG=Google+Suche&meta=";  
La=L.replace("q=", "q="+a);verweis="<b>Google</b>".link(La);document.write(verweis);
```

L ist die Internetadresse von Google. *La* ist die veränderte Adresse, wobei *a* das Suchwort ist. *verweis* ist der ausgeschriebene Hyperlinkverweis, bei dessen Anklicken die betreffende Suche in Google gestartet wird.

Das lässt sich über ein einfaches Suchformular demonstrieren (Abb.II.7)¹¹

Im Formurlarfeld ist *selection* eingegeben. Starten Sie das Formular, wird zunächst *Google* als Hyperlink ausgeschrieben. Klickt man auf den Hyperlink, erscheint die Ergebnisanzeige von Google und in der Adressenzeile des Browsers der Suchstring:

¹⁰ <http://www.multisuchsystem.de/einarb.htm>

¹¹ <http://www.multisuchsystem.de/suchn.htm>

<https://www.google.de/search?q=selection+&ie=UTF-8&oe=UTF-8&hl=de&btnG=Google+Suche&meta=>

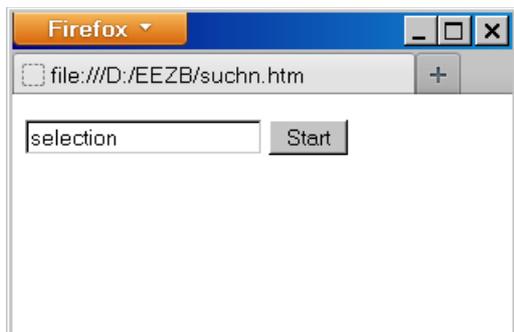


Abb. II.7

Der vollständige Code für dieses Formular lautet:

```
<htm>
<form name="forma">
  <p><strong><input type="text" size="20" name="E"> <input
    type="button" value="Start" onclick="starte()"> </strong></p>
</form>
<script>
function starte()
{
a=document.forma.E.value;
L="http://www.google.de/search?q="+&ie=UTF-8&oe=UTF-
8&hl=de&btnG=Google+Suche&meta=";
La=L.replace("q=", "q="+a);
verweis="<b>Google</b>".link(La);
document.write(verweis);
}
</script>
</htm>
```

Diese Verknüpfungen sind, wie schon erwähnt, generell in Suchdiensten, Internetdatenbanken und anderen Internetangeboten zu finden. So enthalten Suchergebnisse von Datenbanken vielfach Hyperlinks, die zur Homepage und elektronischen Version einer Zeitschrift hinführen oder z.B. deren Standort in einem bibliothekarischen Verbundkatalog ermitteln. In vielen Datenbanken führen Hyperlinks auch direkt zum Volltext eines Aufsatzes hin, falls eine Lizenz für eine Zeitschrift besteht. Mit dieser mit dem englischen Fachausdruck *Reference Linking* bezeichneten bibliographischen Verknüpfungstechnik wird mehr oder weniger das Problem gelöst, wie ein auf einer Webseite zitierter oder von einer Datenbank als Treffer angezeigter Aufsatz so mit einer URL verbunden werden kann, dass als Ziel des Hyperlinks die entsprechende elektronische Version angesteuert wird:

Abb. II.8 zeigt eine Einzeltitelanzeige der JADE-Aufsatzdatenbank mit den Verknüpfungen: Verfassernamen, Fernleihe (Standortermittlung) und WWW-Ausgabe des Anbieters:

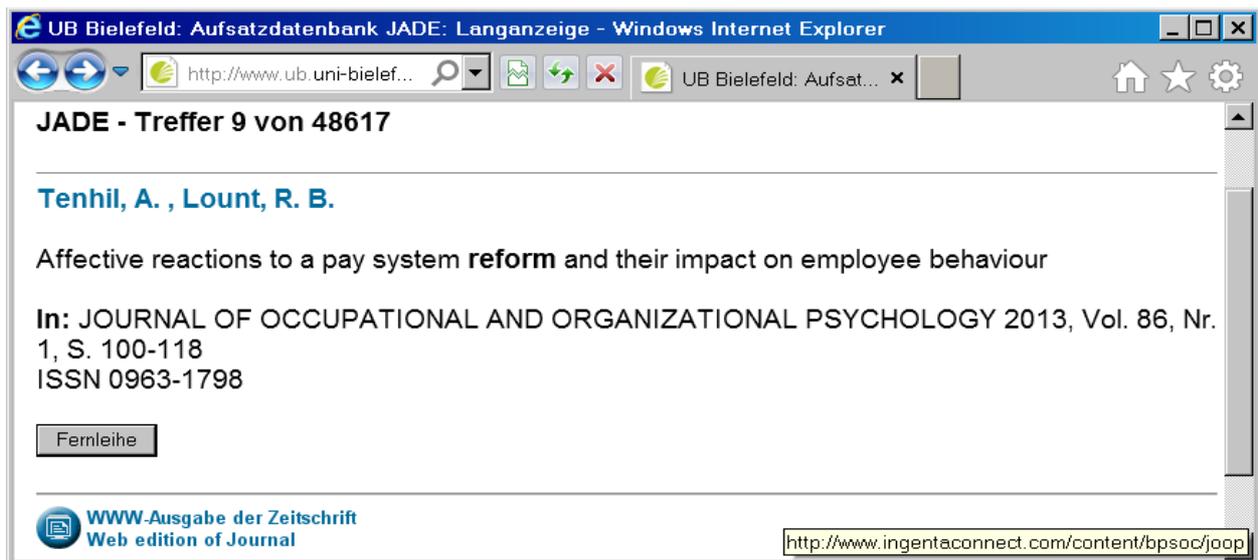


Abb. II.8 Einzeltitelanzeige mit Hyperlinks

Solche Verknüpfungen finden Sie auch zahlreich in den in Kap. V dargestellten Datenbanksystemen des Multisuchsystems. Dabei werden jeweils Suchwörter, Zeitschriftentitel oder ISSN-Angaben mit URL-Adressen so verknüpft, dass die Homepages der Zeitschriftenanbieter, vielfach sogar die einzelnen Online-Aufsätze angewählt und gleichzeitig auch die Verfügbarkeitsnachweise angezeigt werden, wie im folgenden Abschnitt erläutert wird.

IV.1.1 Verknüpfungen über Suchadressen, Open-URL und Google Scholar

Der Zugang vom vorliegenden Aufsatznachweis zum Volltextangebot des Anbieters kann in manchen Fällen über deren Suchmaschinen erfolgen. Hierbei werden in den offenen Suchstring der Suchadressen dieser Systeme einige Suchbegriffe aus dem vorliegenden Titel eingefügt. Fast immer kann auf diese Weise das betreffende Zitat, bei mehreren Treffern meist ganz oben auf der Ergebnisliste aufgeführt, gezielt angesteuert werden, z.B. bei PubMed:

```
LM="http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=search&db=PubMed&term=";
LM1=LM.replace("term=", "term="+e);
```

Hier wird der in *LM* gespeicherte Suchstring von PubMed hinter *term=* durch die Eingabe eines Suchbegriffs *e* ergänzt.

Der Code eines entsprechenden Formulars ist denkbar einfach:¹²

```
<htm>
<form name=forma>
<input type=text name=E>
<input type=button value="Suche"
onclick="starte()">
</form>
<script>
function starte()
{
```

¹² <http://www.multisuchsystem.de/suchgoogle.htm>

```
e=document.forma.E.value;
LM="http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=search&db=PubMed&term=";
LM1=LM.replace("term=", "term="+e);
window.location=LM1;
}
</script>
</htm>
```

Voraussetzung für diese Verknüpfung ist, dass die Suchadresse des Systems nicht, wie leider oft, verborgen und verschlüsselt, sondern offen ist wie hier bei PubMed .

IV.1.1.1 Open-URL

Eine ganz ähnliche Verknüpfungsmöglichkeit ergibt sich durch die Anwendung der sogenannten Open-URL, einem Standardformat für die Adressierung im Internet, die ähnlich aufgebaut ist wie die vorige Suchadresse. Auch hier wird die vorliegende Adresse durch einige Angaben, z.B. durch Titelwörter, Verfasser, Jahr usw., ergänzt. So lautet z.B. die Open-URL bei Springerlink:

```
LinkSb="http://www.springerlink.com/openurl.asp?
genre=article&issn=&volume=&issue=&spage=";
```

Hier müssen die ISSN, Band, Heft und Seitenangabe ergänzt werden. Diese Open-URL wäre an sich eine gute Verknüpfungsmöglichkeit. Leider ist sie aber vom jeweiligen Anbieter abhängig, der nicht immer seine Volltexte über eine Open-URL zugänglich macht, wie hier z.B. SpringerLink. Ausserdem sind relativ genaue Angaben, wie ISSN, Band, Heft und Seite erforderlich, die nicht immer und auch etwas schwierig den Zitaten zu entnehmen sind. Der weit verbreitete Linkresolver SFX von ExLibris verwendet ebenfalls eine solche Open-URL mit den eben genannten Nachteilen.

IV.1.1.2 Google und Google Scholar

Vorteile gegenüber der Open-URL kann eine andere, noch weniger bekannte Art der Verknüpfung bieten, die ich in den Datenbanksystemen des Multisuchsystems verwendet habe. Es ist dies die Verknüpfung über Google Scholar und Google. Bei diesen funktioniert die Verknüpfung äusserst schnell, zuverlässig und direkt. Es zeigt sich, dass Google Scholar und Google fast immer einen Zugang zum Volltext eines Artikels liefern, falls wesentliche Daten wie Sachtitel, Verfasser und Zeitschriftentitel dem Zitat entnommen und in den offenen Suchstring dieser Suchmaschinen eingefügt werden. Oft genügt schon ein Teil des Sachtitels eines Aufsatznachweises. Die ISSN und andere Metadaten sind bei Google und Google Scholar nicht zu verwenden. Sie sind in den von mir entwickelten Datenbanksystemen überflüssig geworden, und das könnte auch anderswo die Verknüpfungspraxis beeinflussen.

IV.1.3 Multiples Linksystem als Linkresolver

Das Multisuchsystems E-Connect bietet im Multiplen Linksystem¹³ Verknüpfungen zu den Trefferlisten einer Reihe von Datenbanken. Das Linksystem funktioniert hierbei ähnlich wie kommerzielle Linkresolver, die ebenfalls nicht nur den Zugang zum Volltext herstellen, sondern auch die Verfügbarkeit nach dem Durchsuchen einer Bestandsliste ermitteln und anzeigen (s. Kap. V.2)

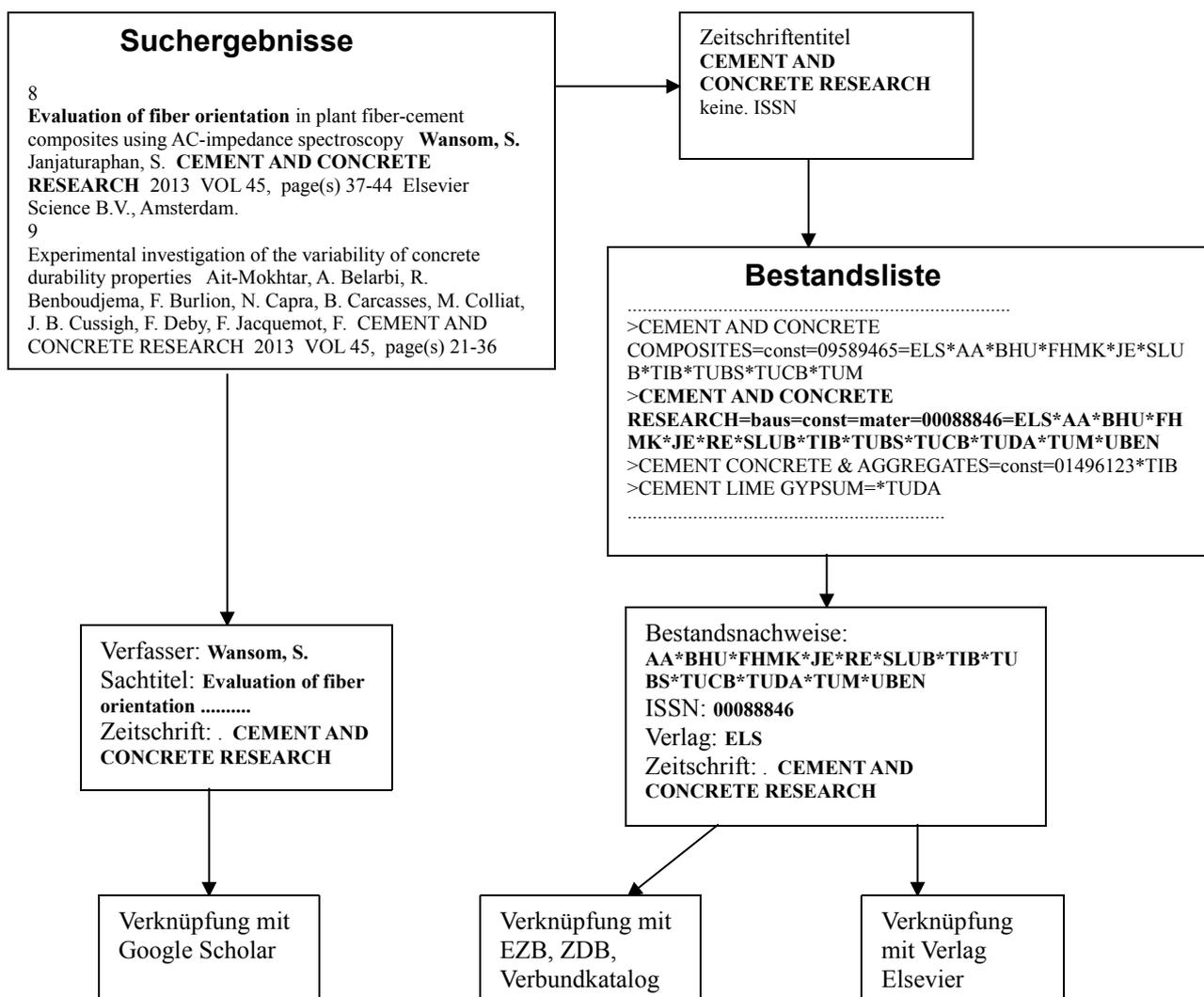
¹³ <http://www.multisuchsystem.de//multipleA.htm>

IV.2 Funktionsweise der Linksysteme

Das folgende Schema zeigt die Funktionsweise eines Linksystems am Beispiel von Suchergebnissen der British Library Direct. Links sind 2 Zitate der Datenbank zu sehen, die einzeln abgefragt werden. Jeweils wird der Zeitschriftentitel ermittelt und dieser in der Bestandsliste gesucht. Dort ist der Zeitschriftentitel mit thematischen Angaben, mit der ISSN, mit dem Verlagshinweis und den Standortnachweisen angegeben. Aus diesen werden die gewünschten Verknüpfungen für die EZB, ZDB und Bibliothekskataloge gebildet. Die Verknüpfung für Google Scholar wird dagegen direkt aus den im Zitat gegebenen Angaben zu Verfasser, Sachtitel und Zeitschriftentitel gewonnen.

Die einzelnen Arbeitsschritte

- Die gegebene Suchergebnisliste wird in die einzelnen Zitate aufgeteilt (im linken Feld oben jeweils durch die Numerierung 8, 9 usw. mit Absatz)
- Die Daten aus einem Zitat werden erfasst (zunächst der Zeitschriftentitel, evtl. auch die ISSN, wenn vorhanden)
- Nach dem Zeitschriftentitel (hier CEMENT AND CONCRETE RESEARCH), evtl auch der ISSN, wird die Bestandsliste durchsucht: Bestandsnachweise, evtl. wie hier auch Verlag und ISSN werden ermittelt.



- Die Verknüpfungen mit den verschiedenen Internetdiensten werden hergestellt:
 Google Scholar-Adresse mit Zeitschriftentitel, Verfasser und Sachtitel ergänzt
 Adressen von EZB, ZDB und Katalogen mit Zeitschriftentitel oder ISSN (wenn vorhanden) ergänzt
 Verlagsadresse von Elsevier hier nur mit Zeitschriftentitel oder ISSN ergänzt
 (bei Springer-Zss und vorhandener Open-URL mit Band, Heft und Seite, die aus den Zitaten zusätzlich ermittelt werden müssen)

IV.2.1 Arbeitsweise eines einfachen Linksystems

Das folgende Beispiel zeigt, wie ein einfaches Linksystem¹⁴ arbeitet. Im oberen Texteingabefeld des Formulars sind bereits einige Treffer einer Datenbank fest eingespeichert

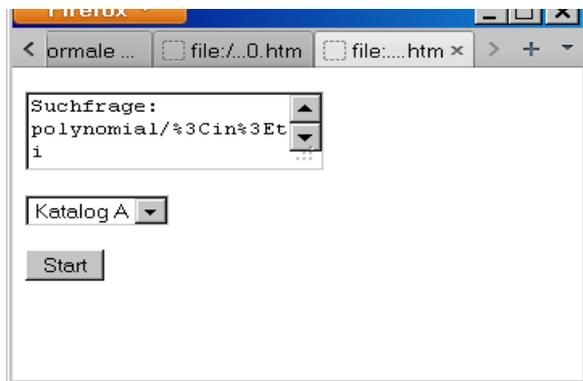


Abb. II.9

Diese werden durch die folgenden Zeilen eingeleitet:

```
Suchfrage: polynomial/%3Cin%3Eti
Suche in folgenden Datenbeständen gestartet:
Current Contents Connect
Current Contents Connect
Treffer 1 - 100 von 1250 for SET #1 (TS=
(Die ersten beiden Treffer:)
Cash, GG
Relationship between the Hosoya polynomial and the
hyper-Wiener index
APPL MATH LETT, 15: (7) 893-895 OCT 2002

Wolf, C
Hausdorff and topological dimension for polynomial
automorphisms of C-2
ERGOD THEOR DYN SYST, 22: 1313-1327 Part 4 AUG 2002
```

Die beiden Treffer werden durch 2 Sternchen als Trennzeichen in ein Array verwandelt:

Cash, GG*Relationship between the Hosoya polynomial and the*hyper-Wiener index*APPL

¹⁴ <http://www.multisuchsystem.de/Formular1.htm>

MATH LETT, 15: (7) 893-895 OCT 2002***Wolf, C*Hausdorff and topological dimension for polynomial*automorphisms of C-2*ERGOD THEOR DYN SYST, 22: 1313-1327 Part 4 AUG 2002

In jedem einzelnen Element des Arrays (=Treffer) markiert das letzte Sternchen den Anfang des Zeitschriftentitels:

Cash, GG*Relationship between the Hosoya polynomial and the*hyper-Wiener index*APPL MATH LETT, 15: (7) 893-895 OCT 2002

Die einzelnen ermittelten Teile des Treffers:

Titel: Cash, GG*Relationship between the Hosoya polynomial and the*hyper-Wiener index
Erscheinungsvermerk: , 15: (7) 893-895 OCT 2002
Zeitschriftentitel: APPL MATH LETT

Der Zeitschriftentitel wird durch ein Komma begrenzt, dahinter folgt der Erscheinungsvermerk.

Das entsprechende Skript lautet demnach:

```
<script>
function starte()
{
trefferE=0;
Link2="http://kvk.ubka.uni-karlsruhe.de/hylib-bin/kvk/nph-kvk2.cgi?maske=kvk-
last&title=UB+Karlsruhe%3A+KVK+Ergebnisanzeige&header=http%3A%2F%2Fwww.ubka.uni-
karlsruhe.de%2Fkvk%2Fkvk%2Fkvk-header_de_04_07_02.html&spacer=http%3A%2F
%2Fwww.ubka.uni-karlsruhe.de%2Fkvk%2Fkvk%2Fkvk-spacer.html&footer=http%3A%2F
%2Fwww.ubka.uni-karlsruhe.de%2Fkvk%2Fkvk%2Fkvk-footer_de_04_07_02.html&kvk-
session=X7YZ1GHU&kataloge=ZDB&TI=&AU=&SB=&CI=&SS=&ST=&PU=&sortiert=nein&css
=http%3A%2F%2Fwww.ubka.uni-karlsruhe.de%2Fkvk%2Fkvk%2Fkvk-
neu.css&target=_blank&Timeout=60";//Suchadresse des KVK
if(document.forma.D1[0].selected) //falls Katalog A ausgewählt
Link2a=Link2;
if(document.forma.D1[1].selected) // falls Katalog B ausgewählt
Link2a=Link2.replace("kataloge=ZDB","kataloge=GBV");
if(document.forma.D1[2].selected)// falls Katalog C ausgewählt
Link2a=Link2.replace("kataloge=ZDB","kataloge=BVB");

T=document.forma.area.value;
Ta=T.replace(/\n/g,""); // neue Zeilen durch * ersetzt
Tb=Ta.replace(/\r/g,""); // Zeilenumbrüche entfernt
pa=Tb.search(/\*$/); // Sternchen am Ende der Eingabe wird gesucht
if(pa!=-1)
Tc=Tb.substring(0,pa-4); // falls gefunden, wird der String um einige Positionen verkürzt
if(pa==--1)
Tc=Tb; // andernfalls bleibt der String bestehen
TA=Tc.split("*****"); // Arraybildung
for(x=0;x<TA.length;x++)
if(TA[x].search(/Time\sFrame|Update\sList|Current\sContent|Copyright/)==-1) // überflüssige
Arrayelemente werden ausgesondert
if(TA[x].search(/\ww+w+/)!=-1) // hier ebenso
{
trefferE++; // Definieren des Zählers

```

```

Tx=TA[x].replace(/\s+/g, ""); // Sternchen mit folgender Leerstelle entfernt
p1=Tx.lastIndexOf(""); // das letzte enthaltene Sternchen gesucht
p2=Tx.indexOf(",",p1); // ein nachfolgendes Komma gesucht
if(p2!=-1) // falls vorhanden
{
Titel=Tx.substring(0,p1); // Sachtitel ermittelt
S=Tx.slice(p2); // Erscheinungsangaben ermittelt
Z=Tx.substring(p1+1,p2); // Zeitschriftentitel ermittelt
}
if(p2===-1) // falls kein Komma folgt, d.h. wenn sich ein Sternchen im letzten Teil von Tx befindet
{
Txa=Tx.substring(0,p1); // der Teilstring Txa wird gebildet ohne dieses Sternchen
Txb=Tx.slice(p1); // der Reststring, der nach diesem Sternchen folgt, wird gebildet
p1a=Txa.lastIndexOf(""); // hier und in den folgenden Zeilen werden die o.g. Angaben aus Txa
ermittelt
p2a=Txa.indexOf(",",p1a);
Titel=Txa.substring(0,p1a);
S=Txa.slice(p2)+Txb;
Z=Txa.substring(p1a+1,p2a);
}
ZA=Z.replace(/\s/g, "+"); // Leerstellen im Zeitschriftentitel werden durch + ersetzt
Link2b=Link2a.replace("TI=", "TI="+ZA); // in die Hyperlinkadresse wird dieser Wert eingesetzt
document.write("<br>"+Z.link(Link2b)); // der durch link zum Hyperlink verwandelte
Zeitschriftentitel wird ausgeschrieben
}}
</script>

```

In dem zugehörigen Formular können Sie über ein Menü drei verschiedene Kataloge anwählen: A=ZDB, B=GVB, C=BVB. Wenn Sie das Formular starten, werden die aus den Treffern extrahierten Zeitschriftentitel als Hyperlinks ausgeschrieben. Klicken Sie auf einen dieser Titel, wird die Verbindung über *Link2a* (erweitert mit dem Titel zu *Link2b*) entsprechend dem im Menü ausgewählten Katalog hergestellt, eine Suche mit dem betreffenden Zeitschriftentitel eingeleitet und sofort das Suchergebnis gezeigt. In diesem Beispiel wird nicht zu allen Titeln ein positives Ergebnis erzielt. Beachten Sie, dass in diesem Beispiel abgekürzte Zeitschriftentitel gegeben sind, die teilweise nur über die ZDB gefunden werden. In einigen Fällen, vor allem wenn ein Zeitschriftentitel mit einem Sachbegriff identisch ist, werden auf diese Weise auch zu viele Treffer gefunden.

Eine zu empfehlende sehr interessante Demonstration dieses Linksystems zeigt Abb.II.10. Über den Schalter *Starte JavaScript* wird eine Webseite mit dem unten aufgeführten Programmcode geöffnet. Darin wird der Quellcode des o.g. Linksystems mit einigen Variablen als Hyperlinks angezeigt. Klickt man auf diese, so wird jeweils der Inhalt dieser Variablen innerhalb des Programmablaufs angezeigt. So erscheinen beim Anklicken von *Titel* nacheinander die Sachtitel der eingefügten Zitate. Über den untersten Hyperlink *document.write* wird das Ergebnis des Linksystems ausgeschrieben (Abb. II.11). Die Zeitschriftentitel sind hier mit der EZB verknüpft, daneben befindet sich noch die Verknüpfung mit der ZDB.

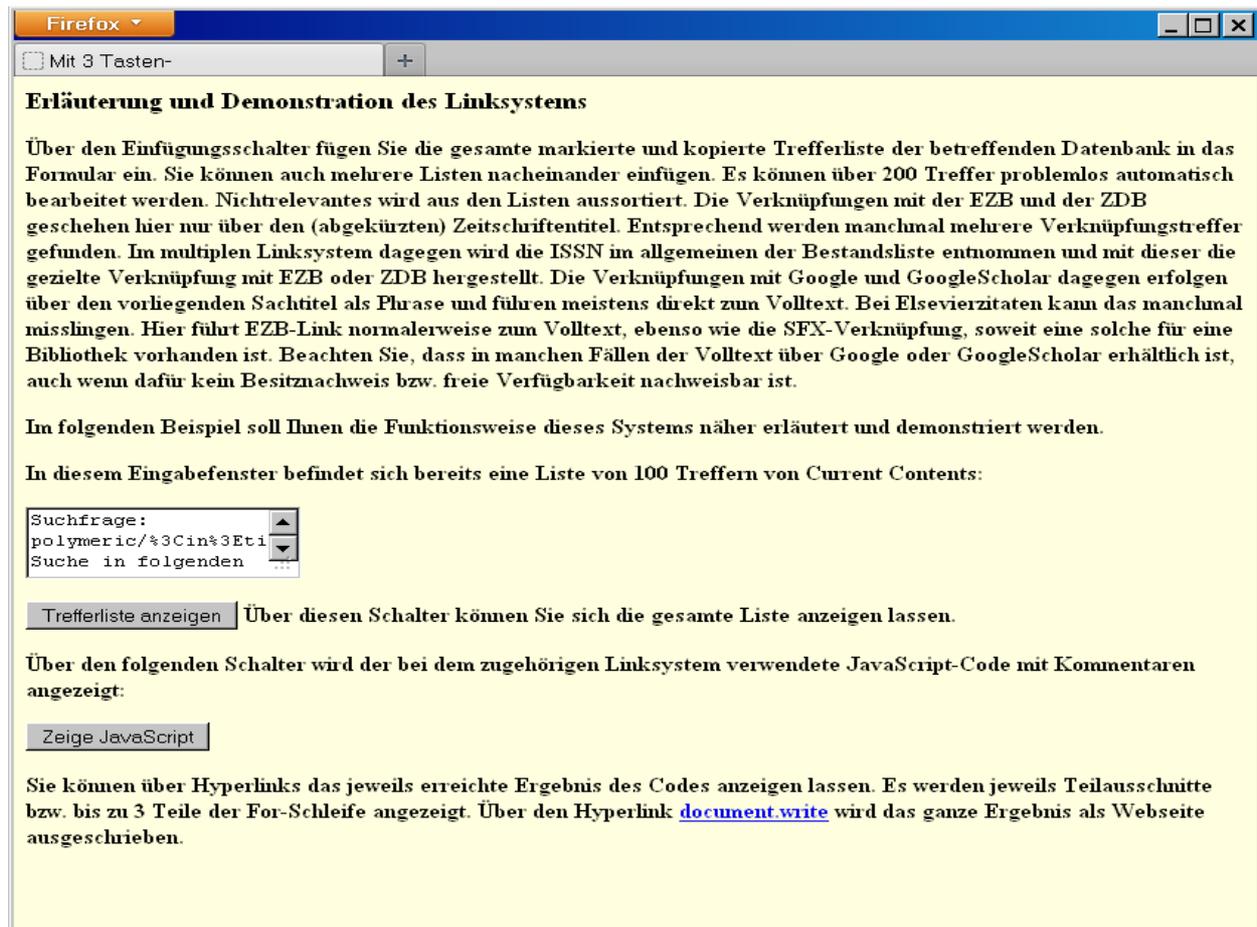


Abb. II.10 Formular für die Demonstration eines Linksystems ¹⁵

```
function starte()
{
window.setTimeout("fort()",2000);
}
function fort()
{
TL1=top.unten.document.forma.area2.value; //Adresse der EZB
TL2=top.unten.document.forma.area3.value; //Adresse der ZDB
T=top.unten.document.forma.area.value; /*Trefferliste der Datenbank, im unteren Frame des
Fensters befinden sich 3 Felder, mit diesen 3 Zeilen werden die darin enthaltenen Daten
gelesen*/
Ta1=T.replace(/\n/g, ""); //neue Zeilen werden durch Sternchen gekennzeichnet
Ta1a=Ta1.replace(/(TS=/, "");
Ta2=Ta1a.replace(/--/g, "");
Ta3=Ta2.replace(/r/g, ""); //Zeilenumbrüche werden beseitigt
Ta4=Ta3.replace(/\*\s/g, "****");
Ta=Ta4.replace(/\*\s\s/g, "****");
TA=Ta.split("****"); /*die Liste wird in ein Array verwandelt, d.h. in die einzelnen Treffer
aufgeteilt, 3 Sternchen als Trennungsmerkmal*/
for(x=0;x<TA.length;x++)
if(TA[x].search(/Time\sFrame|Update\sList|Current\sContent|Copyright/)===-1)
if(TA[x].search(/w\w\w+/)!=-1)
```

¹⁵ <http://www.multisuchsystem.de/Erlmultipl.htm>

```

{
Txa=TA[x].replace(/\s+/g, "");
Tx=Txa.replace(/\|/g, "");
Txx=Tx.replace(/\*/g, " ");
p1a=Tx.lastIndexOf("");
p2=Tx.indexOf(", ", p1a);
if(p2!=-1)
{
Titel=Tx.substring(0,p1a); //der Sachtitel wird ermittelt
S=Tx.slice(p2); //die Erscheinungsangaben der Zeitschrift werden ermittelt
Z=Tx.substring(p1a+1,p2); //der Zeitschriftentitel wird ermittelt
}
if(p2===-1) /*die folgenden Zeilen innerhalb der Klammer ermitteln dieselben Angaben, wenn die
Bedingung negativ ist*/
{
Tx1=Tx.substring(0,p1a);
Tx2=Tx.slice(p1a);
p1=Tx1.lastIndexOf("");
Txa=Tx1+Tx2;
p2=Txa.indexOf(", ", p1);
Titel=Txa.substring(0,p1);
S=Txa.slice(p2);
Z=Txa.substring(p1+1,p2);
}
ZA=Z.match(/\w+/g); //der Zeitschriftentitel als Array
ZAA=ZA.join("+"); // die einzelnen Teile des Zeitschriftentitels werden mit Pluszeichen
verbunden
Link1=TL1.replace(/KT=/, "KT="+ZAA); //die Adresse der EZB wird durch den Zeitschriftentitel
ergänzt
verweis1=Zh.link(Link1); //der Hyperlinkverweis auf die EZB
verweis2=TL2; //der Hyperlinkverweis auf die ZDB
verweis2a=verweis2.replace(/TI=/, "TI="+ZAA); //dieser wird durch den Zeitschriftentitel ergänzt
Txa=Tx.replace(/\*/g, " "); //in dem Text des Treffers werden die Sternchen entfernt
document.write("<p><strong>"+Txa+"<br>"+verweis1+" "+verweis2a+"<hr>"); //die durch das
Linksystem bearbeitete Trefferliste wird ausgeschrieben
}
}
}

```

Harvey, MA; Baggio, S; Garland, MT, et al. Weak C-H center dot center dot center dot O interactions leading to measurable effects in polymeric sulfates [M(SO₄)L(H₂O)(₂)](n) (M = Cd, Zn, Cu; L = bipy, phen, dmp) AUST J CHEM, 56: (11) 1161-1165 2003

[AUST J CHEM](#) [ZDB](#)

Jin, RH Functional polymeric micelles formed from a novel cationic star block copolymer CHEMPHYSCHEM, 4: (10) 1118-1121 OCT 17 2003

[CHEMPHYSCHEM](#) [ZDB](#)

Zhao, GH; Lei, XQ; Gong, CK, et al. In-situ preparation of integrated polymeric pora-u PLOT columns and their applications in gas chromatography CHROMATOGRAPHIA, 58: (7-8) 465-469 OCT 2003

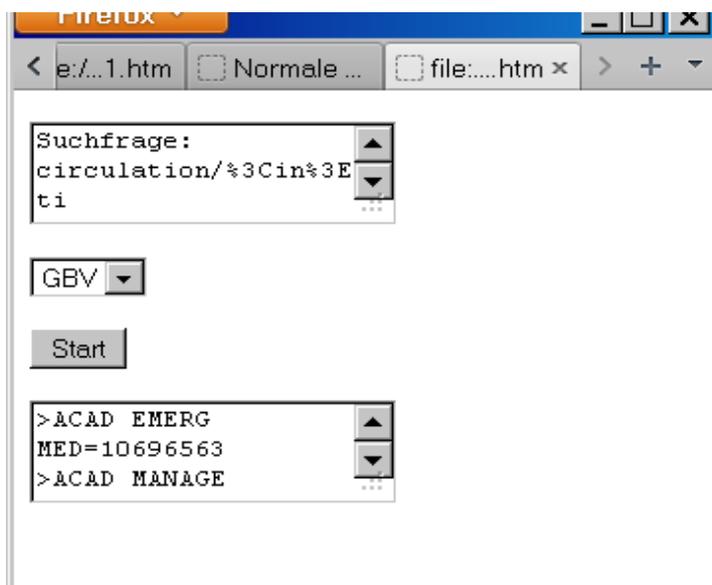
[CHROMATOGRAPHIA](#) [ZDB](#)

Abb. II.11 Ausgeschriebenes Ergebnis des Linksystems. Ausschnitt

IV.2.2 Arbeitsweise eines erweiterten Linksystems

Das vorige Beispiel zeigt, dass die Verknüpfung von Zeitschriftentiteln, zumal von abgekürzten, mit der Suchadresse eines Internetdienstes nicht immer am vorteilhaftesten ist. Es können zu viele oder gleichlautende Treffer erzielt werden. Besser wäre eine Verknüpfung mit der ISSN, die aber in Zitaten nicht immer angezeigt wird. Daher sollten die aus den Treffern extrahierten Zeitschriftentitel vor der Verknüpfung möglichst mit einer Bestandsliste verglichen werden, aus der die ISSN entnommen werden. Ausserdem könnten aus der Bestandsliste noch weitere Daten entnommen werden, die zu weiteren Verknüpfungen und zusätzlichen Informationen führen. So könnten über Verlags- und Publikationsangaben die Volltexte der einzelnen Zitate angesteuert werden, und über die Besitznachweise von Bibliotheken könnte ermittelt werden, ob ein Aufsatz im Volltext für eine bestimmte Bibliothek verfügbar ist oder nicht bzw. ob dieser allgemein frei verfügbar ist. Natürlich lassen sich unabhängig davon andere Verknüpfungen, z.B. über Google und Google Scholar, einrichten, die keine zusätzlichen, aus einer Bestandsliste zu ermittelnden Daten benötigen.

Ein solches durch eine Bestandsliste erweitertes Linksystem soll an einem vereinfachten Beispiel demonstriert werden.¹⁶ Abb.II.12 zeigt das einfache Formular, das im Unterschied zum vorangehenden unten ein weiteres Formularfeld zeigt, in dem die Anfänge einer Bestandsliste, der die Zeitschriftentitel mit dem Buchstaben A enthält, zu erkennen sind.



The image shows a browser window with a search form. The search query is "circulation/%3Cin%3E ti". Below the search field is a dropdown menu set to "GBV" and a "Start" button. At the bottom, there is a list of search results: ">ACAD EMERG", "MED=10696563", and ">ACAD MANAGE".

Abb II.12 Formular eines Linksystems mit Bestandsliste

Auch hier ist die Trefferliste von Current Contents bereits fest in das obere Formularfeld eingespeichert. Wenn Sie das Formular starten, werden die Treffer mit 2 Hyperlinks ausgeschrieben. Der Zeitschriftentitel führt zu einer Verbindung mit der EZB und der Eintrag *Katalog* zu einer solchen mit dem ausgewählten Verbundkatalog. Kann eine Verknüpfung nicht über die ISSN erfolgen, weil der Zeitschriftentitel nicht in der Bestandsliste gefunden wurde, so wird eine solche über den Titel hergestellt. Das ist in der angezeigten Ergebnisseite bei Treffern 6 bis 8 der Fall und zwar für die Zeitschrift [ARTERIOGENESIS](#), die nicht in der Bestandsliste gefunden wurde. Allerdings kommt auch über den Titel kein Suchergebnis zustande, weil es sich hierbei um keine

¹⁶ <http://www.multisuchsystem.de/Formular2.htm>

Zeitschrift, sondern um einen Sammelband handelt. In der letzten Zeile eines Treffers sind die Kennzeichen derjenigen Bibliotheken angegeben, die eine Lizenz auf die elektronische Version der angegebenen Zeitschrift besitzen.

Der Code hierfür lautet:

```
<form name="forma">
<p><textarea name="area" rows="2" cols="20">

</textarea> </p>

<p><select name="D1" size="1">
<option selected>GBV</option>
<option>BVB</option>
<option>ZDB</option>
</select> </p>
<p><input type="button" name="B1" value="Start"
onclick="starte()"> </p>

<p><textarea name="liste" rows="1" cols="20">
</textarea></form>
<script>
function starte()
{
trefferE=0;
Link1="http://www.bibliothek.uni-regensburg.de/ezeit/searchres.phtml?
bibid=&frames=&colors=7&offset=0&KT=&KT_bool=OR&KW=&Notations%5B
%5D=all&selected_colors%5B%5D=1&selected_colors%5B%5D=2&selected_colors%5B
%5D=4&input_date=&PU=&IS=&howmany=25";
Link1a=Link1.replace("bibid=", "bibid=RUBO");
Link2="http://kvk.ubka.uni-karlsruhe.de/hylib-bin/kvk/nph-kvk2.cgi?maske=kvk-
last&title=UB+Karlsruhe%3A+KVK+Ergebnisanzeige&header=http%3A%2F%2Fwww.ubka.uni-
karlsruhe.de%2Fkvk%2Fkvk%2Fkvk-header_de_04_07_02.html&spacer=http%3A%2F
%2Fwww.ubka.uni-karlsruhe.de%2Fkvk%2Fkvk%2Fkvk-spacer.html&footer=http%3A%2F
%2Fwww.ubka.uni-karlsruhe.de%2Fkvk%2Fkvk%2Fkvk-footer_de_04_07_02.html&kvk-
session=X7YZ1GHU&kataloge=GBV&TI=&PY=&AU=&SB=&CI=&SS=&ST=&PU=&sortiert=nei
n&css=http%3A%2F%2Fwww.ubka.uni-karlsruhe.de%2Fkvk%2Fkvk%2Fkvk-
neu.css&target=_blank&Timeout=60";
if(document.forma.D1[0].selected)
Link2a=Link2;
if(document.forma.D1[1].selected)
Link2a=Link2.replace("kataloge=GBV", "kataloge=BVB");
if(document.forma.D1[2].selected)
Link2a=Link2.replace("kataloge=GBV", "kataloge=ZDB");
B=document.forma.liste.value;
T=document.forma.area.value;
Ta=T.replace(/n/g, "");
Tb=Ta.replace(/r/g, "");
Tc=Tb.replace(/*+$/,"");
TA=Tc.split("****");
for(x=0;x<TA.length;x++)
if(TA[x].search(/Suchfrage|Suche|sin|Time|sFrame|Update|sList|Current|sContent|
Copyright/)==-1)
if(TA[x].search(/w\w+/)!=-1)
{
```

```

trefferE++;
y=x;
Tx=TA[x].replace(/^\s+/g, "");
p1=Tx.lastIndexOf("");
p2=Tx.indexOf(", ", p1);
if(p2!=-1)
{
Titel=Tx.substring(0, p1);
S=Tx.slice(p2);
Z=Tx.substring(p1+1, p2);
}
if(p2== -1)
{
Txa=Tx.substring(0, p1);
Txb=Tx.slice(p1);
p1a=Txa.lastIndexOf("");
p2a=Txa.indexOf(", ", p1a);
Titel=Txa.substring(0, p1a);
S=Txa.slice(p2a)+Txb;
Z=Txa.substring(p1a+1, p2a);
}
Za=Z.replace(/AMER\b/, "AM");
ZA=Za.replace(/s/g, "+");
p=B.indexOf(">"+ZA+"=");
if(p== -1)
{
Link1b=Link1a.replace("KT=", "KT="+ZA);
Link2b=Link2a.replace("TI=", "TI="+ZA);
document.write("<p><b>"+y+" "+Titel+" "+Z.link(Link1b)+S+" Katalog".link(Link2b)+" (über
Titel!)</b>");
}
if(p!=-1)
{
pa=B.indexOf(">", p+1);
BS=B.substring(p, pa);
is=BS.match(/d\d\d\d\d\d\d\d\d\d/w/);
IS=is[0].substring(0, 4)+"-"+is[0].substring(4, 10);
Link1b=Link1a.replace("IS=", "IS="+IS);
Link2b=Link2a.replace("ss=", "ss="+IS);
document.write("<p><b>"+y+" "+Titel+" "+Z.link(Link1b)+S+" Katalog".link(Link2b)+"</b>");
}
}
}
}
</script>

```