

Studienarbeit

*InfoNews*

Eine Elektronische Institutszeitung  
mit Archiv

Cand. Inform. Axel Arne Guicking

22. Mai 2000

Betreuer:

Dipl. Inform. Christian Betz

Prof. Dr. Frank Puppe

---

---

Lehrstuhl für Künstliche Intelligenz  
und Angewandte Informatik

Bayerische Julius-Maximilians-Universität Würzburg

---

---



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Zeitungen im Internet</b>	<b>3</b>
2.1	Druckausgaben vs. Online-Zeitungen . . . . .	3
2.2	Reine Online-Zeitungen . . . . .	4
2.3	Die Online-Zeitung <i>InfoNews</i> . . . . .	5
2.4	Suchmöglichkeiten in Online-Zeitungen . . . . .	6
<b>3</b>	<b>HTTP und Java Servlets</b>	<b>8</b>
3.1	Das <i>Hypertext Transport Protocol</i> . . . . .	8
3.1.1	Anfragen („Requests“) . . . . .	8
3.1.2	Antworten („Responses“) . . . . .	8
3.1.3	<b>GET</b> . . . . .	10
3.1.4	<b>POST</b> . . . . .	10
3.1.5	Weitere HTTP-Methoden . . . . .	10
3.2	Dynamische Generierung von Webseiten . . . . .	11
3.2.1	Herkömmlicher Ansatz: CGI . . . . .	11
3.2.2	Alternative zu CGI: Java Servlets . . . . .	12
3.2.3	Vorteile von Servlets gegenüber CGI . . . . .	13
3.3	Das Java Session Tracking API . . . . .	15
3.3.1	Überblick . . . . .	16
3.3.2	Benutzer-Authentifizierung . . . . .	16
3.3.3	Versteckte Formularfelder . . . . .	17
3.3.4	„URL Rewriting“ . . . . .	17
3.3.5	Cookies . . . . .	18
3.3.6	Session Tracking mit Java Servlets . . . . .	18
<b>4</b>	<b>Funktionalität von <i>InfoNews</i></b>	<b>19</b>
4.1	Artikeltypen . . . . .	19
4.1.1	Allgemein . . . . .	21
4.1.2	Beamer-Reservierung . . . . .	21
4.1.3	Konferenz/Tagung . . . . .	21
4.1.4	Mailingliste . . . . .	22
4.1.5	Veröffentlichung . . . . .	23
4.1.6	Vortrag . . . . .	23
4.1.7	Programmfehler . . . . .	24
4.1.8	Person . . . . .	24
4.1.9	Benutzerdaten . . . . .	24
4.2	Benutzerregistrierung . . . . .	25

4.3	Login . . . . .	25
4.4	Anzeige der neuesten Artikel . . . . .	25
4.5	Anzeige eines einzelnen Artikels . . . . .	26
4.6	Anzeige aller Artikel in der Archivansicht . . . . .	28
4.7	Suchen im Archiv . . . . .	29
4.8	Einreichen eines neuen Artikels . . . . .	31
4.9	Logout . . . . .	32
<b>5</b>	<b>Architektur von <i>InfoNews</i></b>	<b>33</b>
5.1	Die Artikel . . . . .	33
5.2	Die Programmebene . . . . .	34
5.2.1	ArticleSubmitter . . . . .	35
5.2.2	ArticleProcessor . . . . .	37
5.2.3	ArticlePresenter . . . . .	38
5.2.4	ArticleFinder . . . . .	39
5.2.5	UserManager . . . . .	40
5.2.6	InfoManager . . . . .	42
5.3	Die Abstraktionsebene . . . . .	42
5.4	Die Datenebene . . . . .	44
<b>6</b>	<b>Erweiterungsmöglichkeiten</b>	<b>46</b>
6.1	Technische Ergänzungen . . . . .	46
6.2	Ontologien . . . . .	50
6.3	Persönliche Benutzerprofile . . . . .	51
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>53</b>
<b>A</b>	<b>Installation und Konfiguration</b>	<b>55</b>
A.1	Installation . . . . .	55
A.2	Konfiguration . . . . .	56

# 1 Einleitung

Betrachtet man heute im Internet vertretene Zeitungen wie den Spiegel<sup>1</sup>, die Süddeutsche<sup>2</sup> oder auch die Würzburger Mainpost<sup>3</sup>, so fällt auf, dass die Möglichkeiten des Internets, allen voran Querverweise (Hyperlinks) und multimediale Objekte, nur in geringem Umfang auf den Internetseiten verwendet werden [7]. Der Grund dafür liegt überwiegend in der Tatsache, dass Artikel der Printausgabe, die in das Online-Angebot übernommen werden, häufig nur minimal an dieses Kommunikationsmedium angepasst werden.

In dieser Arbeit wird das System *InfoNews* vorgestellt, das versucht, diesen Rückstand dadurch aufzuholen, dass es als reines Online-Informationssystem ausschließlich im Internet verfügbar ist<sup>4</sup> und die Artikel von potenziell jedem Benutzer geschrieben werden können. Damit wird des weiteren erreicht, dass die einzelnen Artikel stets aktuell sind; über die Möglichkeit, auf einen Artikel zu antworten, können ansatzweise auch „Diskussionsfäden“ erstellt werden.

Das System versucht automatisch bei der Anzeige eines Artikels Hyperlinks zum Autor, anderen Artikeln sowie zu dem System bekannten Personen hinzuzufügen. Zweck dieses Vorgehens ist es, dem Leser des Artikels den Zugriff auf weitere Informationen, die zum Teil außerhalb des Systems im World Wide Web (WWW) verteilt sind, zu erleichtern oder zumindest Anhaltspunkte zu geben, an wen er sich bei Bedarf wenden kann. Da eine solche Artikelseite bei Aufruf jedes Mal neu generiert wird, wird garantiert, dass Verweise immer zu den aktuellsten Informationen erstellt werden.

Ferner kann der Autor eines Artikels bei Bedarf selbst Links zu seiner Meinung nach interessanten und informativen Websites angeben, um den Leser auf zusätzliche Informationsangebote im Internet hinzuweisen. Diese werden dann entsprechend im Artikeltext aufgeführt.

Die Motivation des in dieser Arbeit vorgestellten Informationssystems liegt im verbesserten Informationsaustausch einzelner Benutzer und Benutzergruppen. Um diesen möglichst effektiv und übersichtlich zu gestalten, beschränkt sich das System dabei auf einen relativ eng umgrenzten Fachbereich – z. B. den Lehrstuhl für Informatik VI.

Das System kann von einem normalen Webbrowser wie dem Netscape

---

<sup>1</sup>URL: <http://www.spiegel.de/>

<sup>2</sup>URL: <http://www.sueddeutsche.de/>

<sup>3</sup>URL: <http://www.mainpost.de/>

<sup>4</sup>URL: <http://l6-lassie-dos.informatik.uni-wuerzburg.de/infonews/index.html>

Navigator<sup>5</sup> oder dem Microsoft Internet Explorer<sup>6</sup> genutzt werden. Es basiert überwiegend auf HTML-Formularen, über die die gewünschten Informationen des Benutzers (z. B. beim Schreiben eines neuen Artikels oder der Suche nach bestimmten Daten) dem System mitgeteilt werden.

Damit erfordert dieses System nur wenige Computerkenntnisse, sodass es auch leicht in anderen Bereichen wie beispielsweise in der Biologie oder Akustik eingesetzt werden kann, in denen die Nutzung des Internets zur Informationsbeschaffung teilweise noch weniger verbreitet ist.

---

<sup>5</sup>URL: <http://www.netscape.com/computing/>

<sup>6</sup>URL: <http://www.microsoft.com/windows/ie/>

## 2 Zeitungen im Internet

Das World Wide Web bietet insbesondere für Zeitungen sowohl Vorteile als auch Nachteile. Die Vorteile umfassen neben Multimedialität, Verweise zu internen und externen Quellen, Suchfunktionen auch die Möglichkeit einer ständigen Aktualisierung (z. B. „News-Ticker“) und der Verbreitung von Zusatzangeboten wie kostenlosen Kleinanzeigen im Netz oder Chat-Foren.

Das Risiko bei Online-Versionen gedruckter Zeitungen ist, dass zunehmend mehr Benutzer des Internets auf diese meist kostenlose Alternative zurückgreifen können und damit die Einnahmen durch die gedruckte Ausgabe zurückgehen.

Dieser Trend, der besonders von Kritikern Mitte der neunziger Jahre vorhergesagt wurde, hat sich bisher jedoch kaum bewahrheitet. Die Hauptursache dafür liegt darin, dass der Komfort einer Online-Zeitung trotz der oben genannten Vorteile noch weit hinter dem einer gedruckten Ausgabe zurückliegt. Das Lesen am Bildschirm ist deutlich anstrengender und meist mit Kosten für den Internet-Zugang verbunden. Ferner ist für viele (vor allem ältere) Menschen der Umgang mit dem Computer und Internet noch zu ungewohnt [7].

Im folgenden Abschnitt werden die Unterschiede zwischen Druckausgaben und Online-Versionen von verschiedenen Zeitungen betrachtet, bevor auf reine Online-Zeitungen und das hier vorgestellte System *InfoNews* eingegangen wird. Abschließend werden kurz die bisher realisierten Suchmöglichkeiten in Online-Zeitungen vorgestellt.

### 2.1 Druckausgaben vs. Online-Zeitungen

Bei den bereits genannten Vorteilen der Online-Zeitungen gegenüber einer gedruckten Ausgabe spielen bei den Benutzern die kostenlose Verfügbarkeit und die Aktualität die wichtigste Rolle.

Weiterhin ist die Nutzung von verschiedenen Zeitungen – insbesondere auch ausländischen – und Querverweisen zu weiterführenden Informationen im WWW ein wichtiger Aspekt, den die Leser von Online-Zeitungen als Vorzug gegenüber einer Druckausgabe in einer Befragung angeben [7]. Wichtigste Sparten sind wie bei Druckausgaben auch bei Online-Versionen die „klassischen“: Politik, Wirtschaft, Regionales sowie Wissenschaft und Technik.

Zusätzliche Angebote wie Kontakt zur Redaktion per E-Mail, Gästebücher und interaktive Spiele werden hingegen nur am Rande genutzt.

Als ein großer Nachteil wird die Tatsache empfunden, dass die meisten der im Internet vertretenen deutschen Zeitungen nur einen Teil ihrer Artikel im Netz bereitstellen – dies gilt durchaus auch für wissenschaftliche internationale Journals wie beispielsweise das „Journal of Sound and Vibration“<sup>1</sup>.

## 2.2 Reine Online-Zeitungen

Während bei gedruckten Zeitungen die zusätzliche Online-Version im Internet in erster Linie dazu dient, einen weiteren Marktsektor zu besetzen und das Serviceangebot für die Leser zu erhöhen, beispielsweise durch Online-Abonnements, liegt bei reinen Online-Zeitungen der Schwerpunkt auf dem Informationsaustausch einzelner Leser.

Aus diesem Grund existieren bei letzteren häufig sogenannte Themen-„Threads“, in denen auf einen konkreten anderen Artikel Bezug genommen wird – innerhalb dieser „Artikelbäume“ kann meistens über Querverweise zwischen den einzelnen Artikeln geblättert werden.

Dieser Ansatz erinnert stark an Newsgroups, in denen ein ganz ähnlicher Informationsaustausch über reine Textnachrichten (mit E-Mails vergleichbar) erfolgt. In der Tat können Webzeitungen als eine Kombination von Newsgroups und dem Word Wide Web aufgefasst werden.

Es existieren mehrere grundlegende Unterschiede zu herkömmlichen Zeitungen und ihrem Online-Angebot. Die wichtigsten werden im folgenden kurz beschrieben.

### Keine Redaktion im eigentlichen Sinne

Im Gegensatz zu konventionellen Zeitungen besitzt eine reine Online-Zeitung keine Redaktion – herkömmliche Zeitungen haben dagegen häufig sogar eine eigene Abteilung für das Online-Angebot. Für die Auswahl und den Inhalt der erscheinenden Artikel sind bei letzteren Redakteure zuständig, bei Web-Zeitungen kann potenziell jeder Leser selbst Artikel schreiben.

In der Regel ist dafür eine (kostenlose) Registrierung erforderlich, damit der Missbrauch des Systems minimiert wird. Da der Inhalt der Artikel dementsprechend weder kontrolliert noch überarbeitet wird, erfolgt auch keine Filterung bezüglich der „Güte“ des Artikels.

---

<sup>1</sup>URL: <http://www.apnet.com/jsv>

### Rund-um-die-Uhr-Aktualisierung

Obwohl auch bei Online-Versionen konventioneller Zeitungen bereits auf die ständige Aktualisierung gegenüber ihrem gedruckten Pendant hingewiesen wurde, erfolgt in der Regel eine Aktualisierung der Webseiten nur ein- bis zweimal pro Tag [7]. Grund dafür ist neben den Arbeitszeiten der Redakteure die Tatsache, dass die technische Bearbeitung für die Internet-Präsentation und Programmierung der eigenen Online-Angebote (noch) sehr viel Zeit in Anspruch nimmt.

Bei Web-Zeitungen ist es möglich, zu jeder Zeit einen Artikel zu schreiben, der anschließend unverzüglich den anderen Lesern zur Verfügung steht.

### Layout wird vom System übernommen

Das Einreichen eines Artikels erfolgt meist auf Basis von reinem Text, evtl. mit der Möglichkeit, Links zu anderen Webseiten anzugeben. Der Artikel wird anschließend vom System aufbereitet und in eine WWW-gerechte Form übersetzt – d. h. das Layout des Artikels wird automatisch generiert.

Damit übernimmt der Autor gleichzeitig die Aufgabe des Journalisten und des Redakteurs; die im vorigen Abschnitt erwähnte aufwändige technische Bearbeitung in den Online-Redaktionen wird in diesem Fall auf ein Minimum gesenkt.

## 2.3 Die Online-Zeitung InfoNews

Das in dieser Arbeit vorgestellte Informationssystem ist eine Web-Zeitung, die außerdem versucht, die Vorzüge des Internets noch vermehrt mit denen der bereits im vorigen Abschnitt beschriebenen Eigenschaften von Online-Zeitungen zu vereinen. Künftige Erweiterungen können außerdem das System mit Methoden des intelligenten Information Retrieval noch direkter auf die einzelnen Benutzer anpassen (vgl. Kapitel 6 ab Seite 46).

Im Vordergrund bei der Implementierung standen drei Aspekte, die in praktisch eingesetzten Web-Zeitungen wie Telepolis<sup>2</sup> nur in Ansätzen unterstützt werden: Zum ersten *kontrollierter Informationsfluss*, darunter wird in diesem Zusammenhang der Einsatz von Benutzergruppen verstanden, die dem Autor ermöglichen festzulegen, welche Benutzer später auf diesen Artikel zugreifen können.

---

<sup>2</sup>URL: <http://www.heise.de/tp/default.html>

Zum zweiten fügt das System eigenhändig *Hyperlinks* zu Personen und anderen Artikeln hinzu, wenn diese im Artikeltext genannt werden. Anhand der gespeicherten Informationen in der Artikeldatenbank erkennt das System die Namen von ihm bekannten Personen. Falls entsprechende Daten verfügbar sind, werden bei der Darstellung der Webseite mit dem Artikeltext automatisch Links zur Homepage und E-Mail-Adresse dieser Person in den Text eingefügt. Entsprechend wird mit den Titeln anderer Artikel verfahren, wodurch eine Vernetzung der einzelnen Texte erreicht werden kann, die über die von Newsgroups bekannten Thementhreads hinausgeht.

Durch diese erweiterte Funktionalität versucht das System, dem *Info-News*-Benutzer die gezielte Suche nach ihn interessierenden Informationen mit den Möglichkeiten des World Wide Web zu erleichtern.

Zum dritten verfügt das System über verschiedene *Artikeltypen*, die in der vorliegenden Fassung dem geplanten Einsatzgebiet am Lehrstuhl für Informatik VI angepasst sind. Anhand dieser Artikeltypen wie beispielsweise „Konferenz“ oder „Vortrag“ kann der Autor anderen Benutzern auf einfache Weise wichtige Detailinformationen wie Veranstaltungsort und Termin zur Verfügung stellen. Die Architektur des Systems ist so angelegt, dass bei Bedarf weitere Artikeltypen erstellt und auf einfache Weise in das System integriert werden können (vgl. Kapitel 5 ab Seite 33).

## 2.4 Suchmöglichkeiten in Online-Zeitungen

Jede der Online-Zeitungen verfügt über eine Eigenschaft, die grundsätzlich ein Vorteil der elektronischen Datenverarbeitung ist: die Möglichkeit eine große Datenmenge vollständig und schnell nach bestimmten Begriffen zu durchsuchen.

Je nach System ist diese Funktion mehr oder weniger komfortabel gestaltet. Sie reicht von der Suche nach einem einzelnen Wort über die Suche mit sogenannten Wildcards (Platzhalter für einzelne Zeichen und Zeichenketten) bis hin zu ausgefeilten Verfahren, wo mithilfe von logischen Operatoren z. B. auch die Suchergebnisse negiert werden können.

Meist ist bereits auf der Titelseite der jeweiligen Online-Zeitung ein kleines Textfeld zu sehen, mit dem eine erste Beschränkung auf einen Teil der Daten erreicht werden kann. Über einen Verweis kann außerdem zu einem komplexeren Suchformular gewechselt werden, in dem zusätzliche Suchoptionen angegeben werden können, z. B. die Beschränkung auf einen bestimmten Zeitraum oder die gewünschte Unterscheidung zwischen Groß- und Kleinschreibung.

Des Weiteren kann häufig über ein kleines Formular, das auf der Seite mit der Liste gefundener Artikel eingeblendet ist, diese Auswahl weiter eingeschränkt werden – logisch gesehen ist das die UND-Verknüpfung.

Darüber hinausgehende Ansätze verwenden Methoden des intelligenten Information Retrieval, beispielsweise Ontologien, wie in Kapitel 6.2 auf Seite 50 beschrieben. Mit diesen Techniken ist es möglich, anhand der Semantik des Inhalts von Webseiten nach Daten zu suchen; die Eingabe der Suchkriterien kann auch in diesem Fall über Suchformulare eingegeben werden [3]. Ein solches System ist On2Broker<sup>3</sup>, das genauer in Kapitel 6.2 beschrieben wird.

IN unterstützt zur Zeit wie die meisten Online-Zeitungen eine Suche nach einzelnen Begriffen. Je nach Kontext wird dabei ein kleines Suchformular angezeigt, mit dem auf einfache Weise im gesamten Artikel-Archiv nach einer Phrase gesucht werden kann, oder ein umfangreiches, wo weitere Optionen und Einstellungen zur Suche angegeben werden können.

Die Suchmöglichkeiten in *InfoNews* sind ein Kernpunkt der möglichen Erweiterungen des Systems (s. Kapitel 6). Im Vordergrund steht dabei der Einsatz der im vorigen Absatz genannten intelligenten Suchverfahren, um die Vorteile einer Webzeitung mit denen einer effizienten Suchmaschine zu vereinen.

Diese Kombination ist bereits Forschungsschwerpunkt vom „KMi Planet“ bzw. „Planet-Onto“ der Open University in Milton Keynes [1].<sup>4</sup> Auch auf dieses Projekt wird bei der Diskussion möglicher Erweiterungen näher eingegangen.

---

<sup>3</sup>URL: <http://www.aifb.uni-karlsruhe.de/www-broker>

<sup>4</sup>URL: <http://kmi.open.ac.uk/planet/index.html>

## 3 HTTP und Java Servlets

*InfoNews* basiert überwiegend auf den beiden Technologien *Hypertext Transport Protocol (HTTP)* und *Java Servlets*. In diesem Kapitel werden darum einige Grundlagen dazu erläutert und mit herkömmlicher Client-Server-Programmierung im WWW (CGI) verglichen. Den Abschluss bildet ein Abschnitt über Programmierung von Java Servlets und ein paar diesbezügliche Besonderheiten.

### 3.1 Das Hypertext Transport Protocol

Das *Hypertext Transport Protocol (HTTP)* ist wie andere im Internet eingesetzte Übertragungsprotokolle, z. B. das *File Transfer Protocol (FTP)* oder das *News Network Transport Protocol (NNTP)*, auf das Standardprotokoll im Internet, TCP/IP (*Transmission Control Protocol/Internet Protocol*), aufgesetzt und ermöglicht die Übertragung von Multimedia-Daten (Grafiken, Sounds und Videos).

Der Datenaustausch zwischen Client und Server über HTTP basiert auf Anfragen des Clients (Webrowsers) und einer daraufhin vom Server generierten Antwort.

#### 3.1.1 Anfragen („Requests“)

Eine Anfrage besteht aus einem Header und eventuellen zusätzlichen Daten. Der Header enthält Informationen über die eigentliche Anfrage, u. a. die HTTP-Aktion (Methode), die der Server ausführen soll, die URL der angeforderten Datei sowie weitere Informationen über den Client selbst (Software, Betriebssystem, akzeptierte MIME-Dateitypen).

Ein Request-Header kann beispielsweise folgendermaßen aussehen:

```
GET /index.html HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 4.0; Windows 95)
Accept: image/gif, image/jpeg, text/*, */*
```

Nach dem Header werden ggf. weitere Daten geschickt, z. B. der Inhalt eines HTML-Formulars.

#### 3.1.2 Antworten („Responses“)

Nachdem der Server die Anfrage ausgewertet hat, schickt er eine Antwort, die ebenfalls aus einem Header mit Meta-Informationen und der Antwort an sich besteht.

Der Header enthält als wichtigsten Bestandteil einen Status-Code mit Beschreibung, der den Client über Erfolg oder Misserfolg seiner Anfrage informiert. Insgesamt existieren mehr als 30 solcher Antwort-Codes, die häufigsten davon sind:

**200 (OK):**

Die Anfrage wurde verstanden und erfolgreich beantwortet.

**401 (Unauthorized):**

Die Anfrage benötigt eine kennwortgestützte Anmeldung per HTTP.

**403 (Forbidden):**

Die Anfrage wurde verstanden, der Server weigerte sich aber, diese auszuführen.

**404 (Not Found):**

Die angeforderte Datei wurde nicht gefunden.

**408 (Request Timeout):**

Der Server erwartete ein Request des Clients, der nicht in einem bestimmten Zeitraum erfolgte.

**500 (Internal Server Error):**

Bei der Beantwortung des Requests trat ein interner Fehler auf – dies geschieht z. B., wenn bei der Ausführung eines Servlets eine Ausnahme nicht abgefangen wurde.

**503 (Service unavailable):**

Der HTTP-Server kann aufgrund zu vieler Requests vorübergehend die Anfrage nicht beantworten.

Außerdem enthält der Header u. a. Informationen über die eigene Serversoftware, das Datum der letzten Änderung der angeforderten Datei, den MIME-Typ der Datei und die Länge der Antwort in Bytes.

Beispiel eines Response-Headers:

```
HTTP/1.1 200 OK
Date: Saturday, 08-Jan-00 11:58:25 GMT+01:00
Server: JavaWebServer/2.0
MIME-version: 1.0
Content-type: text/html
Content-length: 2084
Last-modified: Friday, 07-Jan-00 13:47:03 GMT+01:00
```

Nach dem Header folgt der Inhalt der angeforderten Datei (sofern die Anfrage erfolgreich verarbeitet wurde).

HTTP verfügt über mehrere Methoden, die den angesprochenen Server dazu veranlassen, bestimmte Aktionen auszuführen. Die mit Abstand wichtigsten HTTP-Methoden sind **GET** und **POST**, die – einfach ausgedrückt – für die Anforderung von Informationen bzw. für das Schicken von Informationen an den Server verwendet werden.

### 3.1.3 GET

Die Methode **GET** kann zur näheren Beschreibung in der URL weitere Daten enthalten (dies ist der sogenannte „Query string“), z. B. die Loginkennung des Benutzers bei Online-Bestellungen. Vorteil dieser Methode ist, dass diese zusätzlichen Daten direkt in der URL angegeben sind – so kann der Benutzer beispielsweise diese Seite direkt als Lesezeichen im Browser angeben. Der Nachteil daran ist aber, dass zum einen die Daten sichtbar sind (d. h. zur Übertragung eines Passworts ist die **GET**-Methode unbrauchbar) und zum anderen die Länge der URL bei manchen Webservern auf ca. 240 Zeichen beschränkt ist.

### 3.1.4 POST

Für das Schicken größerer Informationspakete verfügt HTTP darum über die **POST**-Methode. Diese Methode schickt die Daten direkt im Rumpf der Anfrage, d. h. die URL wird nicht verändert, sodass die Daten für den Client (bzw. Benutzer) „unsichtbar“ sind. Bei der Auswertung von HTML-Formularen wird darum häufig die **POST**-Methode verwendet.

### 3.1.5 Weitere HTTP-Methoden

Es existieren noch weitere HTTP-Methoden, die jedoch nur selten zum Einsatz kommen und nicht von allen Webservern unterstützt werden:

**HEAD:**

Zeigt dem Client nur den Antwort-Header.

**PUT:**

Speichert eine Datei direkt auf dem Server.

**DELETE:**

Löscht eine Datei auf dem Server.

**TRACE:**

Sendet die Anfrage zurück an den Client.

**OPTIONS:**

Fragt den Server, welche Optionen und Methoden er unterstützt.

Das Hypertext Transport Protocol bildet die Grundlage für den Datenaustausch im World Wide Web. Im Zuge der ständig wachsenden Datenmengen spielen inzwischen dynamisch generierte Webseiten eine wichtige Rolle. Dazu läuft auf dem Server ein Programm, das je nach Anfrage eine bestimmte Aufgabe ausführt und das Ergebnis als HTML-Seite zurück an den Webbrowser des Clients schickt.

Häufig greifen solche Programme auf Datenbanken zu, man denke z. B. an ein Online-Reisebüro, mit dessen Hilfe man einen Flug nach Madrid buchen kann. Auch *InfoNews* greift auf eine Datenbank zu, in der die einzelnen Artikel gespeichert sind.

In den folgenden Abschnitten werden zwei Technologien beschrieben, solche Programme zu entwickeln.

## 3.2 Dynamische Generierung von Webseiten

### 3.2.1 Herkömmlicher Ansatz: CGI

Die konventionelle Art, Programme im WWW zur Verfügung zu stellen, um dynamisch HTML-Seiten zu generieren, ist das Common-Gateway-Interface (CGI). Darunter versteht man eine Schnittstelle, die von der Server-Software unterstützt werden muss, was jedoch bei allen gängigen Webservern der Fall ist.

Es gibt viele Beispiele für CGI-Programme, von einfachen Zugriffszählern über Gästebücher und Suchmaschinen bis hin zu komplexen Datenbankanwendungen [6].

Was für ein Art von Programm letztendlich hinter dem jeweiligen CGI-Programm steckt, ist unabhängig vom Webserver, meistens sind es aber (bei kleinen Programmen) Perl-Scripte oder Unix-Shellscripte, da diese sich besonders gut eignen, Operationen mit Zeichenketten zu automatisieren.

Der prinzipielle Ablauf eines Requests an ein CGI-Programm ist in Abbildung 1 schematisch dargestellt.

Die CGI-Schnittstelle stellt verschiedene Umgebungsvariablen zur Verfügung, mit denen das Programm zur Laufzeit u. a. Informationen über den Client, den Server und die jeweils verwendete Software abrufen (und

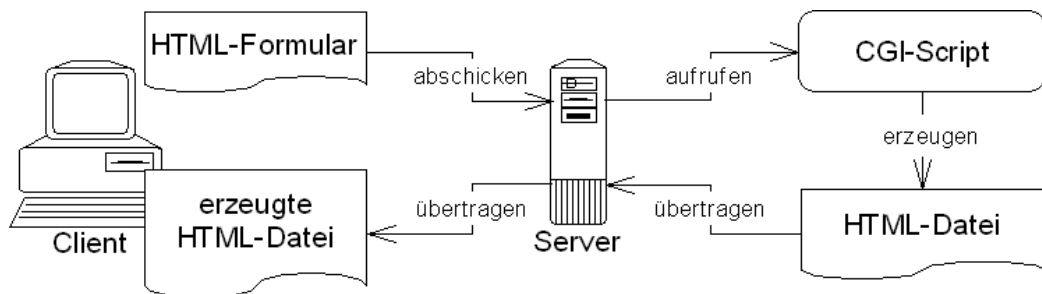


Abbildung 1: Generelle Funktionsweise eines CGI-Scripts.

teilweise auch selbst setzen) kann. Quelle dieser Informationen ist entweder der Request des Clients (s. Abschnitt 3.1) oder die Server-Software.

Um beispielsweise eine bestimmte Datenbank anzusprechen und darin nach einem angegebenen Begriff zu suchen, den ein Benutzer in ein HTML-Formular in seinem Webbrowser eingegeben hat, kann das CGI-Programm die Umgebungsvariable **QUERY\_STRING** auswerten und so den Suchbegriff feststellen. Anschließend wird aus dem Suchbegriff eine SQL-Anfrage erzeugt, die dann an eine bestimmte Datenbank weitergereicht wird. Das Ergebnis dieser Anfrage wird schließlich vom CGI-Programm in HTML-Code übersetzt und als Antwort zurück an den Client geschickt.

### 3.2.2 Alternative zu CGI: Java Servlets

Obwohl sich bereits Ende 1996 abzuzeichnen begann, dass Java als komfortable Programmiersprache für viele Anwendungen besonders im Internet (Applets) nützlich sein würde, war die Unterstützung für Java-Code auf Serverseite jedoch nur sehr vereinzelt anzutreffen. Es gab verschiedene Ansätze einzelner Firmen, die Programmierung auf Webservern mithilfe einzelner Java-Erweiterungen von der technischen auf eine höhere Ebene zu verlagern. Diese Erweiterungen waren dabei aber auf den jeweiligen Webserver zugeschnitten und nur für ganz bestimmte Aufgaben zuständig [5].

Diese beiden entscheidenden Nachteile waren Anfang 1997 ausschlaggebend dafür, dass JavaSoft<sup>1</sup> (inzwischen Teil von Sun Microsystems) die erste Version ihres Java Servlet API („Advanced Programming Interface“) verabschiedete, um die verschiedenen Trends auf einen gemeinsamen Standard zu bringen. Mit Servlets ist es möglich, Java-Code auf einem Webserver (genauer: in der Java Virtual Machine des Servers) ausführen

<sup>1</sup>URL: <http://java.sun.com/>

zu lassen und dabei die Funktionalität des Servlet API auszunutzen – mit allen Vorzügen, die Java bietet, u. a. Objektorientierung, Plattformunabhängigkeit, Sicherheit und Robustheit.

Das Servlet API stellt Methoden zur Verfügung, mit denen man auf die einzelnen Informationen des Anfrage- und Antwort-Headers zugreifen kann, ähnlich den Umgebungsvariablen bei der CGI-Programmierung (s. voriger Abschnitt).

Ferner kann man auf einfache Weise die verschiedenen Aktionen implementieren, die bei Anforderung der jeweiligen HTTP-Methode (**GET**, **POST** etc.) ausgeführt werden sollen.

Einsatzgebiete für Servlets gibt es viele – seien es nun einfache Servlets zur Verwaltung registrierter Benutzer einer Software-Firma, größere miteinander interagierende Servlets, die das Online-Shopping so komfortabel gestalten – oder eben Online-Zeitungen.

Inzwischen werden Servlets von allen gängigen Webservern unterstützt. Seit der ersten Veröffentlichung wurde die Funktionalität von Servlets ständig erweitert, die aktuelle Version des Servlet API ist 2.2 (*InfoNews* verwendet jedoch aus Kompatibilitätsgründen die Version 2.0).

Die Entwicklung von Java ist noch lange nicht abgeschlossen, sodass neben dem Core API auch das Servlet API noch erweitert und verbessert werden wird. Zur Zeit werden vor allem HTTP-Servlets unterstützt; sowohl diese Unterstützung kann bei späteren Releases erweitert und verbessert als auch die Unterstützung anderer Servlets (bzw. Protokolle) hinzugefügt werden.

### 3.2.3 Vorteile von Servlets gegenüber CGI

Die wichtigsten Schlagworte zur Beschreibung der Vorzüge von Servlets gegenüber anderen Techniken sind Portabilität, Eleganz, Effizienz, Sicherheit, Flexibilität und Erweiterbarkeit. Jeder dieser Begriffe wird im folgenden kurz erläutert – für eine ausführlichere Beschreibung siehe z. B. [5].

#### Portabilität

Während normale Java-Applets und -Applikationen in der Praxis leider selten auf Anhieb auf allen Zielpattformen laufen (meist durch das „Abstract Windowing Toolkit“, AWT, verursacht, das auf die Grafikroutinen des jeweiligen Betriebssystems zugreift), sind Servlets in dieser Hinsicht weniger anfällig.

Der Grund liegt zum einen darin, dass Servlets keine AWT-Klassen verwenden und zum anderen nur auf der Server-Plattform laufen müssen,

man also bei Zugriff auf Systemressourcen nicht alle potenziellen Client-Plattformen berücksichtigen muss.

### **Sprachmächtigkeit und Eleganz**

Durch die Tatsache, dass Java im Vergleich zu anderen Programmiersprachen noch sehr jung ist (Ende 1995 wurde die erste offizielle Version verabschiedet), stellt das „Core API“ von Java eine Funktionalität zur Verfügung, die mit älteren Programmiersprachen häufig nur über Umwege erreicht werden kann.

Bei nicht im Core API vorhandenen Funktionen helfen oft Java-Pakete von Fremdanbietern weiter, z. B. bei der Textsuche nach regulären Ausdrücken, wie von Perl bereits bekannt.

Der Code von Servlets ist einfach und übersichtlich – dies wird überwiegend durch das Java Servlet API, eine der Standarderweiterungen des Core API, erreicht. Selbst komplexere Operationen wie das Setzen und Verwalten von Cookies und „Session Tracking“ werden durch Klassen und deren Methoden dem Programmierer bereits fertig zur Verfügung gestellt.

Im Gegensatz dazu ist ein CGI-Programm wesentlich aufwändiger zu erstellen, da jegliche Kommunikation zwischen Client und Server „von Hand“ implementiert werden muss. Auch andere Fähigkeiten von Java wie Objekt-Serialisierung und der Zugriff auf Datenbanken lassen sich mit anderen Programmiersprachen häufig nur weniger elegant und einfach bewältigen.

### **Effizienz und Dauerhaftigkeit**

Ein weiterer Vorteil von Servlets liegt in der Effizienz: ein einmal geladenes Servlet bleibt als einzelne Instanz im Speicher des Servers, so dass eine spätere Anfrage eines Clients sofort beantwortet werden kann. Während beim CGI jede einzelne Anfrage einen eigenen Prozess auf dem Server startet, kann ein Servlet bei mehreren simultanen Requests durch Starten eines weiteren Threads dem Client wesentlich schneller antworten.

Durch das Verbleiben im Speicher auch bei Nichtbenutzung eines Servlets können zeitaufwändige Verbindungen z. B. zu Datenbanken die ganze Zeit aufrecht erhalten werden, während ein CGI-Programm jeweils eine neue Verbindung aufbauen muss.

### Sicherheit

Servlets sind als Java-Programme streng typisiert, d. h. falls beispielsweise Zahlen als Strings übertragen werden, werden sie als Zahlen verwaltet, sodass es zu keinen Zuweisungsfehlern zur Laufzeit kommen kann. Da Java über keine Zeiger verfügt, werden Probleme beim Speicherzugriff minimiert – auch durch den Garbage Collector, der automatisch den Speicherplatz von nicht mehr referenzierten Objekten wieder freigibt.

Durch das in Java durchgängige Konzept der Ausnahmebehandlung können viele Fehler wie Division durch Null oder Laden einer nicht vorhandenen Datei abgefangen werden und den Client bzw. Benutzer informieren.

Auf einer weiteren Ebene kann sich ein Server durch einen Java Security Manager vor unsicheren Servlets schützen, z. B. um Dateioperationen zu überwachen.

### Flexibilität und Erweiterbarkeit

Die Flexibilität von Servlets besteht in ihren unterschiedlichen Anwendungsmöglichkeiten:

- Generierung ganzer Webseiten, die als Ganzes zum Client geschickt werden – dies wird häufig bei der Verarbeitung von Daten aus HTML-Formularen verwendet.
- „Server-side includes“ (SSI): Durch Einbinden eines Servlets mithilfe des `<SERVLET>`-Tags in eine ansonsten statische Webseite kann diese „etwas dynamischer“ gestaltet werden. Das Servlet wird auf dem Server ausgeführt.
- „Java Server Pages“ (JSP): Ähnliche Funktionsweise wie SSI mit dem Unterschied, dass der Code direkt in die statische Webseite eingefügt wird, d. h. der Browser muss das Ausführen des Servlet-Codes unterstützen. Die Syntax entspricht der von Microsofts „Active Server Pages“ (ASP).
- „Servlet Chaining“: Dient der Filterung von Inhalten, indem mehrere Servlets hintereinander ausgeführt werden – das letzte in der Kette schickt die Antwort an den Client zurück.

## 3.3 Das Java Session Tracking API

Durch die im vorigen Abschnitt beschriebenen Vorteile gegenüber herkömmlichen Techniken und das umfangreiche Java Servlet API ist man

als Programmierer in der Lage, auf ein recht umfangreiches Sortiment von Methoden zurückzugreifen, die die Programmierung deutlich vereinfachen.

Um einen Eindruck von diesem Komfort zu vermitteln, wird hier exemplarisch ein Teil des Servlet API, das Session Tracking, herausgegriffen und etwas genauer vorgestellt.

### 3.3.1 Überblick

Wird über das Hypertext Transport Protocol ein Request eines Clients von einem Server beantwortet, so ist letzterer nicht in der Lage festzustellen, von wem genau die Anfrage gesendet wurde. Zwar enthält der Request-Header die IP-Adresse des Client-Rechners, dies ist aber meist nicht ausreichend, z. B. kann die IP-Adresse die eines Proxy- oder Firewall-Rechners sein, oder der Rechner verwaltet mehrere Benutzer.

Um dieses Problem der Zuordnung zu lösen, muss jeder Client eine eindeutige Identifizierung mit dem Request mitschicken, sodass der Server in der Lage ist festzustellen, wer welche Anfrage geschickt hat.

Dieses Verfahren wird als „Session Tracking“ bezeichnet; im folgenden werden kurz die herkömmlichen Verfahren beschrieben, wie sie häufig in CGI-Scripts Verwendung finden (s. Abschnitt 3.2.1). Anschließend wird ein Teil des Java Servlet API erläutert, das Session Tracking API, das auf den herkömmlichen Verfahren aufbaut, dem Programmierer aber einfache Werkzeuge (sprich: Methoden) zur Verfügung stellt, um Session Tracking in Servlets einzusetzen, ohne sich mit der eigentlichen Verarbeitung von Sessions beschäftigen zu müssen.

### 3.3.2 Benutzer-Authentifizierung

Eine Möglichkeit, einen Request eindeutig zuzuordnen, ist das Verfahren der Benutzer-Authentifizierung. Dabei gibt der Webserver den Zugriff auf bestimmte Ressourcen nur dann frei, wenn der Benutzer am Client-Rechner sich mittels eines Login-Formulars mit Kennung und Passwort einloggt.

Kennung und Passwort werden ab diesem Zeitpunkt automatisch vom Browser im Request-Header mitgeschickt, sodass der Server darauf zugreifen kann und eine Sitzung dieses Benutzers verwalten kann.

Vorteil dieser Methode ist, dass sie sehr einfach zu implementieren ist. Der größte Nachteil ist, dass jeder Benutzer einen eigenen Account benötigt und sich jeweils Einloggen muss – dieses Verfahren ist daher für

Session Tracking relativ ungeeignet – andere Verfahren ermöglichen zudem auch die Verwaltung von anonymen Zugriffen.

### 3.3.3 Versteckte Formularfelder

Versteckte Felder in HTML-Formularen werden wie andere Eingabefelder mit dem Typ `hidden` in einem Formular mit folgendem Code definiert:  
`<INPUT TYPE=HIDDEN NAME="name" VALUE="value">`

Es handelt sich dabei also um Name/Wert-Paare von Strings, die im Browser-Fenster für den Benutzer unsichtbar sind. Diese Daten werden wie normale Parameter an den Webserver übertragen, der sie dementsprechend auf die gleiche Weise wie sichtbare Parameter auslesen kann und anhand dieser Daten die Requests unterscheiden kann.

Die Vorteile von versteckten Formularfeldern liegen auf der Hand: Neben anonymem Session Tracking erfordert die Unterstützung auf Serverseite keinerlei zusätzliche Fähigkeiten der Software. Der größte Nachteil von versteckten Daten in HTML-Formularen liegt darin, dass dies nur bei einer Folge von dynamisch erzeugten Formularen funktioniert – bei völlig statischen Webseiten versagt diese Technik.

### 3.3.4 „URL Rewriting“

Wie bereits der Name dieses Verfahrens vermuten lässt, werden URLs in einer Webseite dahingehend modifiziert, dass sie Informationen wie z. B. eine eindeutige Session-ID als weiteren Parameter oder als zusätzliche Pfad-Information enthält. Dies geschieht durch Anhängen an die ursprüngliche URL; beispielsweise kann bei einem dynamisch generierten Formular die URL zum Abschicken des Formulars um eine ID-Nummer ergänzt werden, die ihrerseits aus einem anderen Formular extrahiert oder vom Servlet neu generiert wurde.

Die Vor- und Nachteile ähneln denen von versteckten Formularfeldern; ein Unterschied ist, dass URL Rewriting nicht auf Formulare beschränkt ist und dass – mit Einschränkungen – auch statische HTML-Seiten damit verarbeitet werden können (dies erfordert aber eine umsichtige Konfiguration des Webserver). Hauptnachteil von URL Rewriting ist die relativ komplizierte Implementierung.

### 3.3.5 Cookies

Cookies ist die am weitesten verbreitete Methode, um vom Webserver auf zusätzliche Informationen auf dem Client zugreifen zu können. Unter ihnen versteht man eine kurze Information, die vom Webserver als Teil einer Antwort an den Client geschickt wird und die dort vom Browser lokal gespeichert wird. Diese Information wird von da an automatisch vom Browser bei jedem weiteren Request zu dem Server mitgesendet. Auf diese Weise werden Cookies häufig als Session Tracking-Verfahren eingesetzt, da sie eine einfache und elegante Möglichkeit bieten, den Client eindeutig zu identifizieren.

Auch Cookies haben einen Nachteil, und zwar werden sie nicht von allen Browsern akzeptiert. Dies wird meistens dadurch verursacht, dass der Benutzer (z. B. aus Sicherheitsgründen) den Browser so konfiguriert hat, dass Cookies generell abgelehnt werden. In diesem Fall muss man als Programmierer auf die vorher besprochenen Techniken zurückgreifen.

### 3.3.6 Session Tracking mit Java Servlets

Das Java Servlet API implementiert das Session Tracking auf Basis von Cookies und URL Rewriting. Je nach verwendetem Webserver kann dieser außerdem auf URL Rewriting zurückgreifen, wenn Cookies vom anfragenden Browser nicht akzeptiert werden (z. B. der JavaWebServer).

Mit jedem Benutzer ist ein eindeutiges Session-Objekt (`HttpSession`) verknüpft. In diesem Objekt können beliebige Java-Objekte gespeichert werden, die in Zusammenhang mit dem Benutzer stehen.

Mit der Methode `HttpServletRequest.getSession(boolean)` wird anhand des Benutzers, der den Request ausführte, das entsprechende Session-Objekt geholt – über den Parameter kann angegeben werden, ob ein neues Session-Objekt angelegt werden soll, falls für diesen Benutzer noch keines existiert.

Über die Methoden `putValue`, `getValue` und `removeValue` der Klasse `HttpSession` kann auf die gespeicherten Objekte im Session-Objekt zugegriffen werden.

Es existieren eine Reihe weiterer Methoden, um z. B. auf den Zustand eines Session-Objekts zuzugreifen oder es für ungültig zu erklären. Für eine genauere Beschreibung s. z. B. [5] oder die Java Servlet API-Dokumentation<sup>2</sup>.

---

<sup>2</sup>URL: <http://java.sun.com/products/servlet/index.html>

## 4 Funktionalität von *InfoNews*

*InfoNews* ist in erster Linie ein Informationssystem, das die verschiedenen Vorzüge des Internets, insbesondere des World Wide Web und E-Mail, ausnutzt, um Informationen über das Institut für Informatik dem Benutzer zu präsentieren. Die von den Benutzern eingereichten Artikel sind monatsweise in einzelne Ausgaben gegliedert.

Hauptaugenmerk lag bei der Entwicklung des Systems auf einer leichten und überschaubaren Bedienbarkeit sowie auf der automatischen Aufbereitung der Informationen.

Konkret hat ein Benutzer von *InfoNews* die Möglichkeit, das System nach bestimmten Informationen zu durchsuchen und selbst Informationen für andere Anwender zur Verfügung zu stellen. Dies geschieht auf Basis von HTML-Formularen in einem Webbrowser, die auf der Serverseite von Java Servlets verarbeitet werden.

Der grobe Ablauf entspricht dem von CGI-Scripts und wurde bereits in Abschnitt 3.2.1 auf Seite 11 erläutert (vgl. Abbildung 1).

Die Architektur wird in Kapitel 5 ab Seite 33 ausführlich beschrieben; in diesem Kapitel werden die einzelnen Fähigkeiten und Möglichkeiten von *InfoNews* vorgestellt.

### 4.1 Artikeltypen

Das System unterstützt verschiedene Arten von Informationen, die dem Einsatzgebiet am Lehrstuhl für Informatik VI, allgemeiner: an einer Universität, angepasst sind. Realisiert wird dies durch verschiedene Artikeltypen, die ein registrierter Benutzer einreichen kann. Je nach der Information, die ein Benutzer bekanntgeben möchte, kann er einen der folgenden Typen auswählen:

**Allgemein:**

Allgemeiner Artikel (ohne Zusatzinformationen).

**Beamer-Reservierung:**

Reservierung des Beamers am Lehrstuhl für Informatik VI.

**Konferenz/Tagung:**

Ankündigung einer Konferenz oder Tagung.

**Mailingliste:**

Bekanntgabe einer Mailingliste zu einem bestimmten Thema.

**Veröffentlichung:**

Erscheinen einer Veröffentlichung.

**Vortrag:**

Ankündigung eines einzelnen Vortrags aus einer bestimmten Vortragsreihe.

**Fehler/Verbesserungsvorschlag:**

Mitteilung eines Fehlers über *InfoNews* selbst.

**Person:**

Daten zu einer bestimmten Person.

**Benutzerdaten:**

Registrierung eines neuen Benutzers oder Änderung der eigenen Benutzerdaten.

*InfoNews* ist so strukturiert, dass diese Artikeltypen über eine definierte Schnittstelle mit dem System „kommunizieren“, sodass weitere Typen leicht zu einem späteren Zeitpunkt hinzugefügt werden können (vgl. dazu Kapitel 5.1 auf Seite 33). Jeder Typ verfügt dabei über die folgenden Daten:

**Titel:**

Dieser Text wird in der Archivansicht und auf der Seite der aktuellen Ausgabe angezeigt.

**Untertitel (optional):**

Der Untertitel erscheint nur in der Einzelansicht mit dem kompletten Artikel; er entspricht einem zweiten Titel, wie man ihn aus herkömmlichen Zeitungen kennt.

**Text:**

Hier steht der eigentliche Text, der vom System automatisch mit Verweisen zu Personen und anderen Artikeln versehen wird. Auf der Startseite werden jeweils die ersten Zeilen der drei neuesten Artikel angezeigt. Je nach Typ sollte hier mehr oder weniger ausführlich das Thema des Artikels beschrieben werden. Bei einem Vortrag z. B. eine kleine Inhaltsangabe, wie es bei Aushängen für Kolloquiumsvorträge gemacht wird.

**Benutzergruppen:**

Der Autor kann festlegen, welche Benutzergruppen auf diesen Artikel zugreifen dürfen. Standardmäßig sind alle Gruppen ausgewählt.

Gegebenenfalls wird der Benutzer zur Eingabe von zusätzlichen typspezifischen Informationen aufgefordert, z. B. bei einem Vortrag u. a. Ort und Termin. Die einzelnen Artikeltypen werden im folgenden kurz vorgestellt.

### 4.1.1 Allgemein

Dieser Artikeltyp stellt eine Möglichkeit dar, nicht näher spezifizierte Informationen bekanntzugeben. Die Daten bestehen lediglich aus denen, die bereits im vorigen Abschnitt beschrieben wurden.

### 4.1.2 Beamer-Reservierung

Als konkretes Anwendungsbeispiel dient dieser Typ, mit dessen Hilfe der Beamer des Lehrstuhls reserviert werden kann.

Die Daten, die in diesem Zusammenhang eingegeben werden können, sind:

**Ausleiher:**

Nachname der Person, die den Beamer ausleiht (wird versucht, einer gespeicherten Person zuzuordnen).

**Häufigkeit:**

Hier kann ausgewählt werden, ob der Beamer regelmäßig – z. B. für eine Vorlesung – oder nur einmalig benötigt wird. Als Auswahlmöglichkeiten stehen „einmalig“, „täglich“ und „wöchentlich“ zur Verfügung.

**Termin:**

Je nach der gewählten Häufigkeit wird hier der Termin angegeben: Bei „einmalig“ ist das gewünschte Datum anzugeben, bei „wöchentlich“ ist der Wochentag auszuwählen.

**Uhrzeit:**

Hier ist die genaue Uhrzeit anzugeben, von wann bis wann der Beamer benötigt wird.

**Veranstaltung (optional):**

Hier kann bei Bedarf der Name der Veranstaltung angegeben werden.

### 4.1.3 Konferenz/Tagung

Um eine demnächst stattfindende Tagung oder Konferenz anzukündigen kann dieser Artikeltyp verwendet werden. Die Daten, die hierfür eingegeben werden können, sind im einzelnen:

**Start- und Enddatum:**

Gibt den Termin an, an der die Tagung stattfindet. Evtl. falsch eingegebene Daten werden korrekt umgerechnet (z. B. wird der 31. Februar 2000 zum 2. März und der 31. Februar 2001 zum 3. März).

**Veranstaltungsort (optional):**

Gibt den Ort an, wo die Tagung stattfindet, also beispielsweise ein Institut oder eine Stadt.

**Veranstalter (optional):**

Gibt den Veranstalter der Konferenz an, z. B. die Gesellschaft für Informatik.

**Kontaktperson (optional):**

Gibt die Person an, an die man sich bei konkreten Fragen wenden kann. Eingegeben wird dabei nur der Nachname, das System prüft anschließend, ob über diese Person bereits Daten wie E-Mail-Adresse und Homepage gespeichert sind. Ist dies der Fall und der Nachname ist eindeutig, werden bei Anzeige des Artikels automatisch entsprechende Hyperlinks erzeugt. Andernfalls wird entweder ein Auswahlfeld aller in Frage kommenden Personen oder das Formular zur Angabe von Personendaten (s. Abschnitt 4.1.8) angezeigt.

**Weitere Informationen im WWW (optional):**

Sofern es eine Seite im Internet mit näheren Informationen wie Annahmeschluss für Vorträge oder Hinweise zur Teilnahme gibt, so kann man die URL hier angeben.

#### 4.1.4 Mailingliste

Möchte ein Benutzer eine Mailingliste zu einem bestimmten Thema bekanntgeben, kann er dieses relativ umfangreiche Formular verwenden. Um ein automatisches Abonnieren („subscribe“ bzw. „unsubscribe“) zu ermöglichen, müssen die Daten für die jeweiligen E-Mails angegeben werden. Im einzelnen sind das:

**Thema:**

Hier ist das Thema der Mailingliste anzugeben, z. B. „LISP“ oder „Robotik“.

**E-Mail-Adresse:**

Dies ist die Adresse, an die eine subscribe- oder unsubscribe-Mail geschickt werden muss.

**Mail subject/body:**

Hier werden das Subject bzw. der Text der Mails zum Abonnieren und Abbestellen der Mailingliste eingetragen. Dies ist in der Regel ein fest definierter Text, der automatisch durch ein Filterprogramm auf dem Server, auf dem der Mail-Account eingerichtet ist, ausgewertet wird.

**Weitere Informationen im WWW (*optional*):**

Auch hier kann man bei Bedarf eine Internetadresse mit detaillierten Informationen angeben.

### 4.1.5 Veröffentlichung

Um eine Veröffentlichung bekanntzugeben, kann dieser Typ verwendet werden; als Daten sind dabei der Nachname des Erstautors anzugeben und optional die URL, unter der man sich die Veröffentlichung herunterladen kann. Auch hier versucht das System, den Namen des Autors aufzulösen, wie bereits im Abschnitt 4.1.3 beschrieben.

### 4.1.6 Vortrag

Mithilfe dieses Artikeltyps kann ein Vortrag aus einer bestimmten Vortragsreihe angekündigt werden.

Es werden die folgenden Daten gespeichert, von denen lediglich der Organisator optional ist:

**Thema:**

Enthält das genaue Thema des Vortrags – dieser Eintrag wird in der Regel ähnlich dem Titel sein.

**Termin:**

Hier kann der genaue Termin angegeben werden; wie bei einer Konferenz wird auch hier ein falsches Datum automatisch korrigiert.

**Veranstaltungsort:**

Dieses Feld beinhaltet die genaue Angabe, wo der Vortrag stattfindet; üblicherweise ist dies ein Hörsaal oder Seminarraum.

**Organisator (*optional*):**

Hier kann der Benutzer den Veranstalter angeben, z. B. die Fachschaft der Fakultät.

**Vortragsreihe:**

Es stehen verschiedene Vortragsreihen zur Auswahl, z. B. das Informatik-Kolloquium oder eine Disputation.

**Vortragender:**

Dies ist ein Personenfeld – hier kann der Nachname des Sprechers angegeben werden. Wie bei einer Konferenz oder Veröffentlichung wird versucht, die Person automatisch zu ermitteln (s. Abschnitt 4.1.3).

**4.1.7 Programmfehler**

Auch dieses System verfügt über eine Reihe von Unschönheiten und Programmierfehlern. Trifft man auf solche, kann man diesen einfachen Artikeltyp verwenden, um dem Programmierer „bescheid“ zu sagen.

Das Datenformular entspricht dem eines allgemeinen Artikels, d. h. es existieren nur die Textfelder für Titel, Untertitel und Beschreibung des Fehlers.

**4.1.8 Person**

Ein Artikel des Typs Person dient dazu, dem System Daten über eine Person mitzuteilen – dies sind in erster Linie Name, Mail-Adresse und Homepage. Weiterhin können auch die Arbeitsstelle sowie deren Homepage angegeben werden. Einzige zwingend erforderliche Daten sind Vor- und Nachname und Geschlecht.

*InfoNews* nutzt diese Daten, um bei der Präsentation eines Artikeltextes automatisch Hyperlinks zu generieren; dazu muss lediglich der volle Name (Vorname und Nachname) im Text auftauchen. Je nach den bekannten Daten wird dann ein Link zur Homepage und/oder E-Mail-Adresse erstellt.

Einige Artikeltypen verfügen über ein Personenfeld, in dem der Nachname einer Person eingegeben werden kann – falls diese Person dem System bekannt und eindeutig ist, wird diese Information verwendet. Kann der Nachname nicht eindeutig zugeordnet werden, erscheint eine Auswahlliste aller Personen mit diesem Nachnamen; wird kein passender Eintrag gefunden, wird automatisch das Formular zur Eingabe der Daten zu dieser Person angezeigt.

**4.1.9 Benutzerdaten**

Der Artikeltyp „Benutzerdaten“ dient zum einen dazu, sich bei *InfoNews* zu registrieren, sodass man selbst Artikel einreichen kann, zum anderen kann man zu einem späteren Zeitpunkt die eigenen Benutzerdaten ändern (außer der Login-Kennung).

Ist man noch nicht eingeloggt, nimmt *InfoNews* an, dass man sich registrieren möchte, andernfalls werden sämtliche Daten angezeigt, die man gegebenenfalls ändern kann (mit Ausnahme der Login-Kennung). Die Anmeldung am System und der Login-Vorgang werden in den beiden folgenden Abschnitten genauer beschrieben.

## 4.2 Benutzerregistrierung

Um auch selbst Artikel schreiben und nicht nur auf Artikel zugreifen zu können, die für „Gäste“ lesbar sind, muss man sich bei *InfoNews* anmelden. Diese Registrierung besteht lediglich in der Angabe der für das System notwendigen Daten zur Benutzerverwaltung.

Zwingend erforderlich sind Vor- und Nachname, Geschlecht, E-Mail-Adresse, Login-Kennung, Passwort (dieses muss zur Sicherheit zweimal eingegeben werden), die eigene Benutzergruppe und das Benachrichtigungsintervall, mit dem das System per Mail über neu eingegangene Artikel unterrichten soll. Optional können außerdem die eigene Homepage, die Arbeitsstelle und deren Homepage angegeben werden.

Sind alle angegebenen Daten gültig (z. B. muss die gewählte Login-Kennung natürlich systemweit eindeutig sein), wird ein neuer Benutzeraccount erstellt, über den man sich zu einem späteren Zeitpunkt am System einloggen kann. Nach erfolgreicher Anmeldung ist man automatisch eingeloggt.

## 4.3 Login

Loggt sich ein Benutzer am System an, z. B. um einen neuen Artikel einzureichen oder die Artikel zu lesen, die für seine Benutzergruppe sichtbar sind, so überprüft das System die eingegebene Login-Kennung und das Passwort mit den in der Datenbank gespeicherten Daten. Nur bei Übereinstimmung erhält der Benutzer Zugriff auf seine Benutzerdaten und die Möglichkeit, einen neuen Artikel einzugeben.

## 4.4 Anzeige der neuesten Artikel

Ruft man die Homepage von *InfoNews* auf oder loggt sich ein Benutzer ein, wird neben einem Inhaltsverzeichnis auf der linken Seite eine Übersicht über die aktuellsten Artikel angezeigt.

Die Artikel werden dabei jeweils mit Datum, Titel und Autor aufgeführt, die drei neuesten zusätzlich, sofern vorhanden, mit einem verkleinerten Bild und den ersten Zeilen des (formatierten) Textes.

Abbildung 2 zeigt eine exemplarische Startseite.

## Willkommen bei InfoNews

---



**Fr, 10.03.00: RoboCup 2000** ([Arne Guicking](#) )

Auch im Jahr 2000 findet wieder ein RoboCup statt - diesmal in Zusammenhang mit der "Sixth Pacific Rim International Conference on Artificial Intelligence (PRICIA-2000)" in Melbourne, Australien. In diesem Jahr auch dabei: die SONY Legged Robot League - Roboterhunde versuchen Fußball zu spielen. [\[Mehr...\]](#)

---

**Fr, 10.03.00: Neue Person: Andrew Jennings** ([Arne Guicking](#) )

[\[Mehr...\]](#)

---

**Fr, 10.03.00: Benutzerdaten von Arne Guicking** ([Arne Guicking](#) )

[\[Mehr...\]](#)

---

[Ältere Artikel \(Archiv\)](#)

---

Automatisch generiert von InfoNews am 11. März 2000 um 12:53

Abbildung 2: Die Startseite von *InfoNews*.

## 4.5 Anzeige eines einzelnen Artikels

Auf die Einzelansicht eines Artikels gelangt man über einen Verweis des betreffenden Titels, entweder von der Startseite, der Archivansicht oder der Seite mit den Suchergebnissen.

Diese Ansicht dient der eigentlichen Darstellung der Informationen – hier findet man sämtliche Daten eines Artikels, insbesondere auch diejenigen, die für die einzelnen Artikeltypen charakteristisch sind (s. Abschnitt 4.1 auf Seite 19 für eine genaue Beschreibung der verschiedenen Artikeltypen).

Abbildung 3 zeigt eine Beispielseite für eine Tagung, den RoboCup 2000 in Melbourne.

## RoboCup 2000

### Der RoboCup etabliert sich immer mehr als Herausforderung für Robotiker und KIler.

Autor: [Arne Guicking](#) , 10. März 2000 um 09:38, 3. Zugriff

Diese Tagung findet statt vom **26. August 2000** bis zum **2. September 2000**.  
Veranstaltungsort: **Melbourne Exhibition Centre, Melbourne, Australien**  
Veranstalter: **RoboCup Organisation**

Auch im Jahr 2000 findet wieder ein RoboCup statt - diesmal in Zusammenhang mit der "Sixth Pacific Rim International Conference on Artificial Intelligence (PRICIA-2000)" in Melbourne, Australien.

In diesem Jahr auch dabei: die SONY Legged Robot League - Roboterhunde versuchen Fußball zu spielen. Eingesetzt werden dabei Aibos, die von Sony entwickelten Tamagochi-Nachfolger in Hundform.



Im Rahmen des Würzburger physikalischen Kolloquiums berichtete Anfang des Jahres Prof. Burghardt aus Berlin anschaulich mit Wort, Bild und Ton von der doch noch recht eigenwilligen Verhaltensweise dieser Roboter. Probleme entstanden vor allem bei so grundsätzlichen Vorgängen wie "Gehen, wieder stehen bleiben und dabei nicht umfallen".

Man darf aber gespannt sein, inwieweit sich die Entwicklung der Metallhunde weiterentwickelt hat.

Und für wen die Entfernung zu groß ist: Ende Mai findet der RoboCup Euro 2000 in Amsterdam statt.

Meine Meinung zum  [RoboCup 2000](#): Ein Muss für jeden Robotik-Interessierten.

Weitere Informationen finden sie auf der Homepage <http://www.robocup2k.rmit.edu.au/> oder wenden Sie sich direkt an [Arne Guicking](#) .

---

Sie können auf diesen Artikel [antworten](#).

---

[Vorheriger Artikel](#) [Diese Ausgabe](#)

---

Automatisch generiert von InfoNews am 11. März 2000 um 12:48

Abbildung 3: Ausführliche Ansicht eines Artikels.

Unterhalb des Textes kann man über einen Link auf den dargestellten Artikel antworten; bei einem solchen Antwortartikel wird dann zusätzlich ein Verweis zum Bezugsartikel generiert, über den man sich diesen direkt in der Einzelansicht anschauen kann.

Am Seitenende kann (sofern vorhanden) zum nächst aktuelleren bzw. vorigen Artikel geblättert oder wieder die Startseite mit den aktuellsten Artikeln angezeigt werden.

Die Einzelansicht wird bei jedem Aufruf neu aus den vom Autor eingegebenen Rohdaten generiert, da das System versucht, möglichst aussagekräftige Hyperlinks in den Text einzufügen. Damit wird gewährleistet, dass stets die aktuellen Daten z. B. von Personen referenziert werden. Je nach Artikeltyp werden die Informationen unterschiedlich präsentiert, allen liegt aber ein einheitliches Layout zugrunde (Blocksatz, Verweise fett in dunkelrot etc.).<sup>1</sup> Wurde beim Einreichen des Artikels ein Bild auf den Server geladen, wird es im Text in passender Größe eingefügt.

Damit *InfoNews* eine Person erkennt, muss der komplette Name im Text auftauchen, je nach vorhandenen Daten wird dabei ein Link zu seiner E-Mail-Adresse und Homepage erzeugt, ferner ein Link zur Suche nach allen Artikeln, die diese Person eingereicht hat.

Artikel werden anhand des vollen Titels erkannt und in einen Link übersetzt.

Das Übersetzen des Rohtextes ist nachsichtig, d. h. es sucht lediglich nach den eben beschriebenen Wörtern und Phrasen, lässt den Text aber ansonsten unverändert. Auf diese Weise kann der Benutzer Links zu externen Internetseiten sowie komplexere Objekte wie Tabellen und zusätzliche Grafiken in seinen Artikel eingeben.

## 4.6 Anzeige aller Artikel in der Archivansicht

Über die Verweise „Archiv“ im Inhaltsverzeichnis oder „Ältere Artikel“ am Ende der Startseite gelangt man zur Archivansicht. Hier werden alle Artikel der aktuellen Ausgabe (d. h. des aktuellen Monats) aufgeführt, die für den jeweiligen Benutzer sichtbar sind, also die von der Benutzergruppe gelesen werden dürfen, zu der der Benutzer gehört oder die er selbst eingereicht hat.

Wie in Abbildung 4 zu erkennen, wird am Seitenanfang ein kleines „QuickSearch“-Formular zur Suche angezeigt, mit dessen Hilfe auf einfache Weise nach bestimmten Begriffen oder Autoren gesucht werden kann.

---

<sup>1</sup>Diese Angaben werden in einem zentralen Cascading Stylesheet vereinbart und können so auf einfache Weise angepasst werden.



Abbildung 4: Die Archivansicht von *InfoNews*.

Über den Link „Optionen...“ wird das komplette Suchformular angezeigt, in dem man zusätzliche Angaben zur Eingrenzung der Suche vornehmen kann; dies wird im nächsten Abschnitt genauer erläutert.

Mit den Verweisen am Seitenende kann, sofern vorhanden, zur nächsten und vorigen Ausgabe geblättert oder wieder die aktuelle Ausgabe angezeigt werden.

## 4.7 Suchen im Archiv

Eine zusätzliche Funktion von *InfoNews* ist die Möglichkeit, das Archiv nach bestimmten Informationen zu durchsuchen. Dabei stehen auch nicht registrierten Benutzern eine Reihe von Optionen und Parametern zur Verfügung, mit denen die Suche z. B. auf bestimmte Artikeltypen oder Ausgaben beschränkt werden kann.

Im „QuickSearch“-Formular auf einer Archivseite oder der Ergebnisseite einer Suchanfrage kann lediglich die zu suchende Zeichenkette eingegeben und die Kategorie (Suche nach Autoren oder Artikeln) ausgewählt werden; die anderen Parameter sind auf folgende „großzügige“ Defaultwerte eingestellt:

- Es werden alle Ausgaben durchsucht.
- Die Suche wird nicht auf bestimmte Artikeltypen beschränkt.

- Es wird weder zwischen Groß- und Kleinschreibung unterschieden noch nach ganzen Wörtern gesucht.

Im vollständigen Suchformular, das über den Punkt „Suche“ im Inhaltsframe oder über „Optionen...“ im QuickSearch-Formular angezeigt werden kann und in Abbildung 5 dargestellt ist, kann der Anwender diese Defaultwerte anpassen.

Abbildung 5: Das Suchformular von *InfoNews*.

Der Benutzer kann angeben, ob nur in einer bestimmten Ausgabe oder im gesamten Archiv gesucht werden soll, er kann außerdem die Suche auf bestimmte Artikeltypen beschränken. Ferner kann er eine beliebige Kombination der drei folgenden Suchoptionen auswählen, die standardmäßig alle deaktiviert sind:

**Berücksichtigung von Groß- und Kleinschreibung:**

Ist diese Option aktiviert, wird zwischen Groß- und Kleinschreibung unterschieden, d. h. „Puppe“ ist ungleich „puppe“.

**Nur ganze Wörter:**

Mit dieser Option kann angegeben werden, ob zusätzlich auf Wortgrenzen vor und hinter dem Suchstring getestet werden soll. Z. B. werden in diesem Fall bei der Suche nach „ein“ nicht Worte wie „eine“, „einziger“ etc. gefunden.

**Suchergebnis negieren:**

Durch Auswählen dieser Option kann das Suchergebnis invertiert werden, d. h. die Liste der gefundenen Artikel sind gerade diejenigen, die

den angegebenen Suchstring nicht enthalten. Diese Option bezieht sich nicht auf die durchsuchten Ausgaben und Artikeltypen. Die hier gemachten Einstellungen werden also *nicht* invertiert.

## 4.8 Einreichen eines neuen Artikels

Um einen Artikel zu schreiben, wählt man den Punkt „Artikel schreiben“ im Inhaltsverzeichnis auf der linken Seite des Browserfensters. Das endgültige Einreichen von Artikeln zum System ist registrierten Benutzern vorbehalten, jedoch können auch andere Benutzer bis einschließlich zu einer Vorschauseite einen Artikel demonstrationshalber eingeben, um zu sehen, wie ein Artikel aussehen würde.

Die Registrierung (s. Abschnitt 4.2) dient lediglich dazu, das Bereitstellen von Informationen über *InfoNews* etwas beobachten zu können. Des Weiteren hilft sie bei Zitierungen in Artikeltexten neben den Verweisen zur Homepage und E-Mail-Adresse der betreffenden Person auch einen Link zur Suche nach von ihm geschriebenen Artikeln erstellen zu können.

Das Einreichen eines Artikels gliedert sich in einzelne Schritte, zwischen denen man über Buttons „< Zurück“ und „Weiter >“ vor- und zurückblättern kann. Der Button „Eingaben löschen“ setzt sämtliche Daten des aktuellen Formulars auf ihre Anfangswerte zurück. Bevor der Artikel gespeichert wird, kann man auf einer Vorschauansicht das Layout des Artikels begutachten und gegebenenfalls zu den vorigen Seiten zurückgehen um Änderungen vorzunehmen.

Das Schreiben gliedert sich in die folgenden Einzelschritte:

### **Auswahl des Artikeltyps:**

Hier wählt der Benutzer den gewünschten Typ seines Artikels aus (vgl. Abschnitt 4.1 auf Seite 19).

### **Dateneingabe:**

In diesem Formular werden die Daten und der Inhalt des Artikels eingegeben. Optionale Felder sind *kursiv* hervorgehoben. In erster Linie sind diese Daten Überschrift, Untertitel, der eigentliche Text und die Benutzergruppen, die auf diesen Artikel später zugreifen dürfen.

Ausgehend von dem im vorigen Schritt gewählten Artikeltyp können weitere Daten wie eine Internetseite mit zusätzlichen Hinweisen angegeben werden.

### **Grafik hinzufügen:**

Mithilfe dieses Formulars kann der Benutzer zusätzlich zum Artikeltext eine Grafik (im JPEG-Format) auf den Server laden, die vom Sys-

tem automatisch an optisch günstiger Stelle im Text (in der Einzelansicht) platziert wird. Auch auf der Startseite mit den neuesten Artikeln wird dieses Bild – entsprechend verkleinert – angezeigt.

Die Angabe einer Grafik ist optional; möchte man keine Grafik hinzufügen (oder nicht auf den Server laden), kann man das Textfeld einfach leer lassen und zur nächsten Seite gehen.

**Vorschauansicht:**

Auf dieser letzten Seite sieht man die vom System generierte Webseite mit dem kompletten Artikel, wie sie nach dem endgültigen Einreichen bei Aufruf der Einzelansicht generiert wird (vgl. Abschnitt 4.5).

**Login und Speicherung:**

Ist der Benutzer noch nicht am System eingeloggt, so muss er dies an dieser Stelle nachholen, damit der Artikel in der Datenbank gespeichert werden kann. Nach erfolgreichem Login wird der Artikel automatisch übernommen und steht ab sofort an oberster Stelle der Startseite.

Treten im Laufe des Einreichvorgangs Fehler auf, z. B. wenn im Datenformular zwingend notwendige Daten nicht eingegeben wurden oder eine ausgewählte Grafik nicht das richtige Format hat, wird eine entsprechende Fehlermeldung und dieselbe Seite noch einmal ausgegeben. Im Datenformular bleiben dabei bereits gemachte Eingaben erhalten.

## 4.9 Logout

Ein eingeloggter Benutzer kann sich bewusst wieder vom System abmelden, die Daten seiner aktuellen Sitzung werden damit gelöscht und er hat wieder den Benutzerstatus „Gast“. Meldet er sich nicht explizit ab, so wird er aus Sicherheits- und Effizienzgründen nach einer bestimmten Zeitspanne automatisch abgemeldet. Diese Aufgabe wird vom Webserver übernommen, auf dem das System installiert ist; das Zeitlimit liegt defaultmäßig bei 30 Minuten.<sup>2</sup>

---

<sup>2</sup>Dies wird nicht von allen Webservern unterstützt; die meisten verfügen aber über dieses Feature, u. a. Apache, JRun und Tomcat.

## 5 Architektur von *InfoNews*

Die grobe Architektur von *InfoNews* ist in der folgenden Abbildung 6 veranschaulicht.

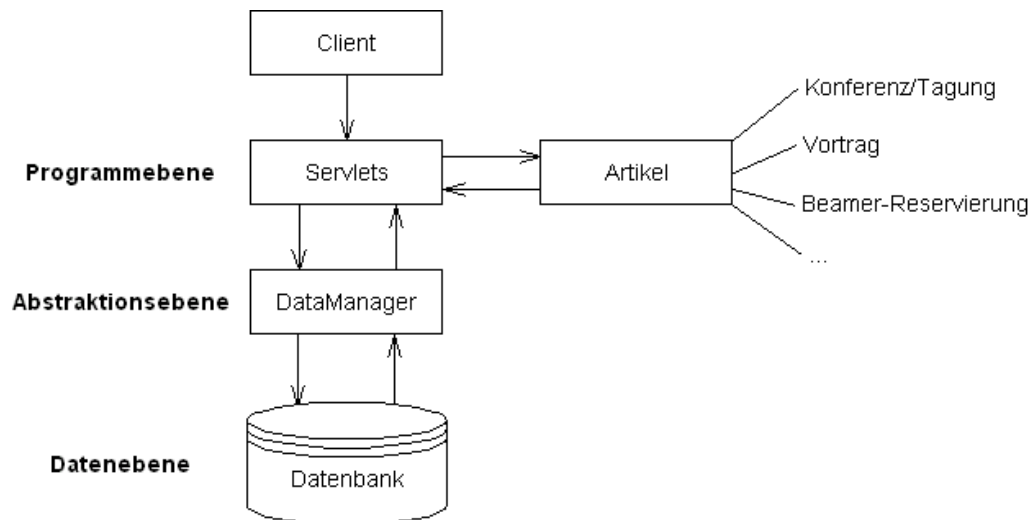


Abbildung 6: Grobarchitektur von *InfoNews*.

Die oberste Schicht des Systems besteht aus den verschiedenen Java-Servlets und Hilfsklassen, um die Daten und Artikel für den Benutzer aufzubereiten und die HTML-Seiten zu generieren. Auf dieser Ebene findet auch die Interaktion mit den einzelnen Artikelobjekten statt.

Darunter liegt der `DataManager`, der den Zugriff auf die einzelnen Daten in der Datenbank kapselt. Dadurch ist es möglich, dass die Servlets die genaue Organisation der Datenbank nicht zu kennen brauchen, sondern über die verschiedenen Methoden des `DataManagers` die gewünschten Daten aus der Datenbank abrufen können.

Die unterste Schicht besteht aus der Datenbank, in der die einzelnen Artikel und Benutzer gespeichert sind. Verwendet wird das objektorientierte Datenbanksystem POET, mit dem die Java-Objekte (Artikel, Benutzer etc.) direkt in der Datenbank gespeichert werden können.

### 5.1 Die Artikel

Wie bereits erwähnt, unterstützt *InfoNews* verschiedene Artikeltypen, um für den jeweiligen Typ passende Zusatzinformationen bereitstellen zu können. Das System ist in bezug auf diese Artikeltypen modular aufge-

baut und das Hinzufügen weiterer Artikeltypen ist verhältnismäßig einfach.

Die Kommunikation zwischen den Artikeltypen und *InfoNews* findet über *Exceptions* statt; auf diese Weise kann dem *ArticleSubmitter* auf einfache Weise mitgeteilt werden, ob z. B. noch nicht alle erforderlichen Daten eingegeben wurden oder Fehler bei der Datenkonvertierung auftraten.

Jeder Artikeltyp ist eine eigenständige Java-Klasse, die von der Klasse *Article* abgeleitet ist. Damit verfügt jeder Typ automatisch über die Datenfelder *Titel*, *Untertitel*, *Text* und *Sichtbarkeit* für die einzelnen Benutzergruppen.

Um konkret einen neuen Artikeltyp zu definieren, muss also eine Java-Klasse erzeugt werden, die von *Article* erbt. Die wichtigsten Methoden dieser Klasse (bzw. deren Unterklassen) sind:

**showSubmitDataForm:**

Generiert das Datenformular im Rahmen des *ArticleSubmitters*.

**processSubmitDataForm:**

Überprüft die im Datenformular eingegebenen Daten auf Vollständigkeit und Korrektheit. Gegebenenfalls wird eine entsprechende Ausnahme ausgelöst, die vom *ArticleSubmitter* verarbeitet wird.

**convertFormData:**

Überprüft z. B. den in einem Personenfeld eingegebenen Nachnamen; falls in der Datenbank keine Person dieses Namens gefunden wurde oder der Name nicht eindeutig zuzuordnen ist, wird auch hier eine Ausnahme ausgelöst.

**writeArticlePage:**

Generiert die Einzelansicht eines Artikels.

## 5.2 Die Programmebene

Die Servlets können als Kommunikationsschnittstelle zwischen dem Anwender und der Datenbank aufgefasst werden. Sie ermöglichen im Rahmen der in Kapitel 3 beschriebenen Möglichkeiten die Kommunikation zwischen dem Client, also dem Benutzer bzw. dessen Webbrowser und dem Webserver, auf dem das System installiert ist.

Die Programmebene besteht aus einer Reihe von Servlets und Hilfsklassen, die die verschiedenen bereits in Kapitel 4 beschriebenen Aufgaben erfüllen. Im folgenden werden die wichtigsten dieser Klassen und ihre Aufgaben beschrieben.

Abbildung 7 zeigt die Programmschicht mit den einzelnen Servlets und wichtigsten Hilfsklassen, deren Aufgaben im folgenden kurz beschrieben werden.

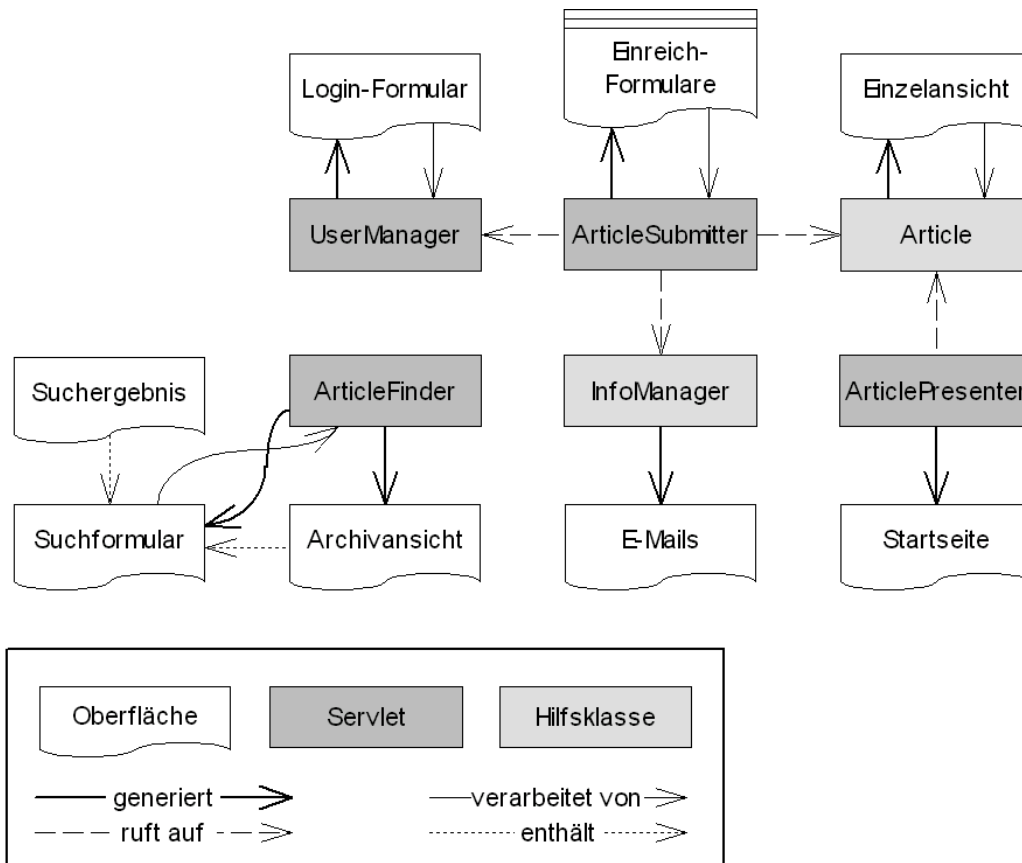


Abbildung 7: Die Programmebene von *InfoNews*.

### 5.2.1 ArticleSubmitter

Das Servlet `ArticleSubmitter` verarbeitet einen neu eingereichten Artikel und veranlasst den `DataManager`, diesen in der Datenbank zu speichern.

Der `ArticleSubmitter` beantwortet sowohl Anforderungen über das HTTP-Kommando **GET** als auch über **POST**. Die eingegebenen Daten werden aber wie zur Auswertung von HTML-Formularen üblich über das **POST**-Kommando verarbeitet, um eventuelle Beschränkungen zu umgehen (die Länge des bei **GET** übertragenen Querystrings ist bei älteren

Webbrowsern auf 255 Zeichen beschränkt – der Text eines Artikels ist aber in der Regel wesentlich länger).

Das Schreiben eines Artikels gliedert sich in einzelne Schritte, in denen jeweils eine Formularseite im Webbrowser des Benutzers angezeigt wird (s. Abschnitt 4.8 auf Seite 31). Die prinzipielle Vorgehensweise gliedert sich dabei für jede dieser Seiten in die Auswertung der Daten und je nach aufgetretenen Fehlern Anzeige des gleichen oder nächsten Formulars.

Nachdem der Benutzer im ersten Schritt einen Artikeltyp ausgewählt hat, wird ein entsprechendes Objekt dieses Typs erzeugt. Zur Eingabe der eigentlichen Artikeldaten (Überschriften, Text und ggf. weitere typabhängige Informationen) ruft der `ArticleSubmitter` die Methode `showSubmitDataForm` des im ersten Schritt erzeugten Artikelobjekts auf.

Anschließend werden diese Daten ausgewertet – auch dies übernimmt der Artikel selbst durch die Methode `processSubmitDataForm`. Tritt bei dieser Auswertung ein Fehler auf, wird eine entsprechende Ausnahme an den `ArticleSubmitter` zurückgegeben, sodass in diesem Fall eine Fehlermeldung und die selbe Seite nochmal angezeigt werden, um dem Benutzer die Möglichkeit zu bieten, seine falschen oder fehlenden Daten zu korrigieren.

Auf der folgenden Seite kann der Benutzer optional eine JPEG-Grafik angeben, die auf den Server geladen und später bei Anzeige des Artikels an optisch günstiger Stelle angezeigt wird. Dazu wird ein `MultipartRequest`-Objekt<sup>1</sup> erzeugt, das Formulardaten des MIME-Typs `multipart/form-data` verarbeitet und die Grafikdatei auf den Server kopiert.

Auf der nächsten Seite wird eine Vorschau des Artikels präsentiert. Der `ArticleSubmitter` ruft dazu die Methode `writeArticlePage` des Artikeltyps auf, diese wiederum den `ArticleProcessor`, der den Text z. B. nach Personen und Titeln anderer Artikel durchsucht, um automatische Verweise zu generieren (s. 5.2.2). Dadurch sind die Artikeltypen selbst für ihr „Aussehen“ verantwortlich und können bei Bedarf die standardmäßige Darstellung von *InfoNews* überschreiben.

Nachdem der Benutzer diese Seite bestätigt, wird überprüft, ob er bereits eingeloggt ist; falls nicht, wird die Anfrage zur Speicherung des Artikels an den `ArticleSubmitter` zunächst zum Login-Formular des `UserManagers` umgelenkt (s. Abschnitt 5.2.5 auf Seite 40). Nach erfolg-

---

<sup>1</sup>`Package com.oreilly.servlet,`  
URL: <http://www.servlets.com/resources/com.oreilly.servlet/index.html>

reichem Login wird der Artikel schließlich an den `DataManager` weitergereicht, der ihn in der Datenbank speichert.

Gespeichert wird dabei der rohe Text, d.h. er wird in keinem internen Format gespeichert, sondern so, wie der Benutzer ihn eingegeben hat. Wird der Artikel zu einem späteren Zeitpunkt angezeigt, wird der Text erneut geparkt, sodass also stets die aktuellen Daten von Personen und Benutzern angezeigt werden.

Bei erfolgreicher Speicherung in der Datenbank wird außerdem ein entsprechender Eintrag ins Logfile geschrieben; die gespeicherten Daten sind Datum, Uhrzeit, Hostname des Clients und ID-Nummer des Artikels.

Zum Schluss des Einreichens wird nach erfolgreicher Speicherung der `InfoManager` aufgefordert, eventuell anstehende Benachrichtigungen an die registrierten Benutzer über neu eingegangene Artikel zu verschicken (s. Abschnitt 5.2.6 auf Seite 42).

## 5.2.2 ArticleProcessor

Der `ArticleProcessor` ist eine Hilfsklasse zur Konvertierung der Rohdaten eines Artikels in HTML-Code. Dazu ruft das anzuzeigende Artikelobjekt die Methode `processArticleText` auf. Ferner greift der `ArticlePresenter` zur Darstellung der Startseite auf die Methode `processArticleOverview` zu, um die ersten Zeilen des Textes in kompakterer Form darzustellen.

Diese beiden Methoden unterscheiden sich darin, dass bei letzterer nur die ersten Sätze bzw. Zeilen geparkt und angezeigt werden. Beide Methoden haben als Parameter den rohen Text des Artikels, wie der Autor ihn eingegeben hat und liefern ein `String`-Objekt zurück, das den geparkten Text enthält.

Der `ArticleProcessor` führt nacheinander eine Reihe von Ersetzungen im Artikeltext durch, die jeweils nur für die vom Client angeforderte Ansicht generiert wird:

- Um dem Benutzer die Gliederung des Textes zu erleichtern, werden bei Leerzeilen jeweils am Beginn der nächsten (nichtleeren) Zeile eine Absatzmarkierung (HTML-Tag `<P>`) eingefügt.
- Erscheinen im Text Personen mit vollem Namen, die in der Datenbank gespeichert sind, werden (sofern vorhanden) Hyperlinks zu ihrer Homepage, E-Mail-Adresse und zu einer Suchanfrage nach Artikeln dieser Person erzeugt. Letztere natürlich nur bei Namen von registrierten Benutzern.

- Entsprechend wird bei Artikeln, die mit vollem Titel zitiert werden, ein Link zur Einzelansicht dieses Artikels eingefügt (s. Abschnitt 5.2.3).

Da HTML-Tags im Text nicht herausgefiltert werden, hat der Anwender die Möglichkeit, komplexere Objekte wie Tabellen oder zusätzliche externe Grafiken in seinen Artikel einzubinden. Desweiteren hat er damit auch die Möglichkeit, das Standardlayout (Schriftfarbe und -größe, Hintergrundfarbe etc.) von *InfoNews* zu „überschreiben“, das durch ein systemweites Stylesheet festgelegt ist. Dies ist allerdings nur in sehr begrenztem Umfang empfehlenswert, da dadurch die Einheitlichkeit der Oberfläche gestört wird.

Links zu externen Seiten im WWW muss der Autor über das von HTML her bekannte Tag `<A HREF="URL">name</A>` eingeben.

### 5.2.3 ArticlePresenter

Der `ArticlePresenter` ist für die dynamische Generierung der Artikel-daten als HTML-Seiten zuständig. Auch dieses Servlet verarbeitet Requests über `GET` und `POST` gleichermaßen.

#### **action=issue**

Diese Anfrage zeigt die Startseite mit den für den jeweiligen Benutzer sichtbaren neuesten Artikeln an (s. Abschnitt 4.4 auf Seite 25). Dafür werden die neuesten Artikelobjekte aus der Datenbank geladen und anschließend für die neuesten drei Artikel die Methode `processArticleOverview` vom `ArticleProcessor` aufgerufen um automatisch Verweise zu Personen und anderen Artikeln zu erzeugen (s. Abschnitt 5.2.2).

#### **action=article&article=<id>**

Dieser Request generiert eine ausführliche Ansicht des Artikels, inklusive aller typspezifischen Daten (s. Abschnitt 4.5 auf Seite 26). Um diese Daten angemessen präsentieren zu können, wird zunächst das über die *id* eindeutig referenzierte Artikelobjekt aus der Datenbank geladen und anschließend dessen Methode `writeArticlePage` aufgerufen.

Diese ruft wiederum die Methoden `writeArticlePageHead` und `writeArticlePageFoot` der Klasse `Article` auf, die den für alle Artikel identischen Kopf- bzw. Fußteil generieren.

Hat der Autor des Artikels beim Einreichen eine Grafik angegeben, die auf dem Server gespeichert wurde, so wird zusätzlich die Methode `wri-`

`teArticleImage` von `Article` aufgerufen, um das Bild in der richtigen Größe darzustellen.

Nachdem die generierte Webseite vollständig zum Client übertragen wurde, wird ein entsprechender Eintrag ins Logfile geschrieben; die gespeicherten Daten sind Datum, Uhrzeit, Hostname des Clients, Login-Kennung des Benutzers (bzw. „Gast“) und die ID-Nummer des Artikels.

### 5.2.4 ArticleFinder

Das Servlet `ArticleFinder` verarbeitet Suchanfragen an die Datenbank und die Archivansicht. Aufgerufen wird das Servlet über **GET**-Requests und je nach dem Wert des übergebenen Request-Parameters **action** werden verschiedene Methoden aufgerufen:

#### **action=search**

Dieser Request zeigt das Suchformular an, in dem der Benutzer neben der zu suchenden Zeichenkette auch alle weiteren Suchparameter eingeben kann (s. Abschnitt 4.7 auf Seite 29). Aufgerufen wird dafür die Methode `writeSearchPage`, die das überwiegend statische HTML-Formular generiert.

#### **action=archive**

Diese Anfrage ruft die Funktion `writeArchivePage` auf, die zunächst alle für den Benutzer sichtbaren Artikel der aktuellen Ausgabe (d. h. des aktuellen Monats) lädt. Ausgegeben werden anschließend ein kleines „QuickSearch“-Formular und die Tabelle der gefundenen Artikel (s. Abschnitt 4.6).

#### **action=find**

Bei diesem Request wertet der `ArticleFinder` eine Suchanfrage aus, indem er über den `DataManager` in der Datenbank nach den Artikeln sucht, auf die die angegebenen Suchparameter zutreffen. Zur genaueren Spezifizierung der Suche tauchen im Querystring weitere Parameter auf:

- **query=<text>**  
Die zu suchende Zeichenkette.
- **category={articles|authors|all}**  
Gibt die Suchkategorie an (Artikel, Autoren oder beides).

- **{articletypes=<classname>}**\*  
Gibt an, auf welche Artikeltypen die Suche beschränkt werden soll.
- **month={all|0|..|11}**  
Gibt den Monat der zu durchsuchenden Ausgabe an.
- **year={all|2000|..}**  
Gibt das Jahr der zu durchsuchenden Ausgabe an.
- **options={case,words,negate}**  
Gibt zusätzliche Suchoptionen an: Berücksichtigung der Groß- und Kleinschreibung, Suche nur nach ganzen Wörtern bzw. Suchergebnis umkehren.

Bei den Suchoptionen wird so vorgegangen, dass im Falle der Nichtberücksichtigung von Groß- und Kleinschreibung zunächst der Suchstring als auch die jeweils zu durchsuchenden Textdaten in Kleinbuchstaben umgewandelt werden.

Soll nur nach Artikeln gesucht werden, die den angegebenen Suchbegriff als ganzes Wort enthalten, wird zusätzlich auf Wortgrenzen vor und nach dem Vorkommen im durchsuchten `String` getestet, d. h. es wird geprüft, ob das Zeichen davor und danach (sofern es existiert) Whitespace oder eines der Zeichen `. , : ; - ! ? ( ) [ ] { }` ist. Beachtet wird natürlich auch der Fall, dass ein Wort am Anfang oder Ende der durchsuchten Zeichenkette steht.

Soll das Suchergebnis negiert werden, werden die Artikel, die den Suchstring nicht enthalten als Ergebnis präsentiert – die anderen Suchparameter wie Artikeltyp und durchsuchte Ausgaben sind von dieser Negation *nicht* betroffen.

Zur eigentlichen Auswertung wird die Methode `writeResultPage` aufgerufen, die zunächst je nach ausgewählter Suchkategorie (Artikel oder Autoren) die jeweilige Methode aufruft, um ein `Vector`-Objekt aller passenden Artikel vom `DataManager` zu erhalten. Analog zur Archivansicht wird diese Liste schließlich von der Methode `writeQueryArticleResult` zusammen mit dem „QuickSearch“-Formular tabellarisch ausgegeben.

### 5.2.5 UserManager

Der `UserManager` verwaltet die Benutzerdaten von *InfoNews*. Da auch ein Benutzer nichts anderes ist als ein von `Article` abgeleiteter Artikeltyp, hat dieses Servlet lediglich die Aufgabe, das Einloggen der Benutzer

und ihre Sessions zu verwalten (s. Abschnitt 4.3 auf Seite 25) – die Registrierung wird vom `ArticleSubmitter` verarbeitet.

Der `UserManager` verarbeitet wie auch die anderen Servlets Anfragen über **GET** und **POST**. Das Loginformular mit Benutzerkennung und Passwort wird aber natürlich über **POST** übertragen, da sonst das Passwort direkt im Browserfenster abzulesen ist.

#### **action=login**

Durch diesen Request wird die Methode `showLoginForm` des `UserManager` aufgerufen, die das an sich statische Login-Formular zum Client-Browser schickt.

#### **action=logincheck**

Beim Abschicken des Login-Formulars werden die Daten mit diesem Request mitgeschickt – die aufgerufene Methode `processLoginForm` überprüft zunächst, ob überhaupt Daten eingegeben wurden und wenn ja, wird über den `DataManager` das Benutzerobjekt anhand der eingegebenen Login-Kennung geladen und das übermittelte Passwort mit dem gespeicherten verglichen.

Falls die eingegebenen Daten mit denen in der Datenbank übereinstimmen, wird das Benutzerobjekt in der `HttpSession` gespeichert. Daran können die anderen Servlets und Hilfsklassen feststellen, ob der Benutzer eingeloggt ist – insbesondere benutzt der `DataManager` diese Information, um die Zugriffskontrolle auf gespeicherte Artikel in der Datenbank zu regeln.

Bei erfolgreichem Login wird außerdem ein entsprechender Eintrag ins Logfile geschrieben; die gespeicherten Daten sind Datum, Uhrzeit, Hostname des Clients und Login-Kennung des Benutzers.

#### **action=logout**

Ein angemeldeter Benutzer kann sich bewusst vom System abmelden, seine im Speicher des Webservers gehaltenen Daten werden dann gelöscht. Nachdem sich ein Benutzer erfolgreich bei *InfoNews* eingeloggt hat, wird im entsprechenden `HttpSession`-Objekt das beim Loginvorgang geladene `User`-Objekt gespeichert. Über dieses Objekt können die verschiedenen Servlets und Hilfsklassen feststellen, ob der Absender eines Requests eingeloggt ist oder nicht und dementsprechend passende Informationen bereitstellen.

Aus Sicherheitsgründen verfügen die meisten Webserver außerdem über ein Limit der Lebensdauer einer Session – der Default liegt meist bei

30 Minuten (Apache, Tomcat, JRun). Ein Benutzer, der sich nicht explizit vom System abmeldet wird automatisch nach dieser Frist ausgeloggt.

Möchte ein Benutzer das System nach dem Logout nutzen, muss er sich erneut einloggen – andernfalls wird er wieder als „Fremder“ behandelt und hat lediglich Zugriff auf die Artikel, die auch für die Benutzergruppe „Gast“ lesbar sind.

### 5.2.6 InfoManager

Der `InfoManager` ist eine Hilfsklasse, die für die Benachrichtigung der registrierten Benutzer via E-Mail zuständig ist, sobald neue Artikel bei *InfoNews* eingegangen sind.

Angestoßen wird die Methode `checkInfoMails` nach erfolgreicher Speicherung eines neuen Artikels durch den `ArticleSubmitter` (s. Abschnitt 5.2.1 auf Seite 35). Über den `DataManager` wird eine Liste aller registrierten Benutzer geladen, die jeweils auf ihre „Benachrichtigungswünsche“ überprüft werden, d. h. ob und wenn ja bei wieviel („*n*“) neuen Artikeln sie benachrichtigt werden wollen.

Falls seit dem letzten Login eines Benutzers bereits *n* neue Artikel eingegangen sind, wird die Methode `sendInfoMail` aufgerufen, die über ein `MailMessage`-Objekt<sup>2</sup> eine entsprechende E-Mail generiert. Dabei werden jedoch nur Datum, Titel, Autor und die ersten Zeilen des Rohtextes der Artikel übermittelt.

Der Mailserver und Account des Absenders wird in der allgemeinen Konfigurationsdatei `infonews.conf` angegeben (s. Anhang A auf Seite 55).

## 5.3 Die Abstraktionsebene

Der `DataManager` dient, wie bereits erwähnt, der Kapselung der Datenbankzugriffe. Die schichtenförmige Architektur von *InfoNews* ermöglicht es, die zugrundeliegende Datenbank auszutauschen oder zu verbessern, ohne dass die einzelnen Servlets angepasst werden müssen.<sup>3</sup>

Da der `DataManager` sämtliche Datenbankzugriffe ausführt, ist er auch für die Anbindung an die Datenbank zuständig. Um die Zugriffe möglichst effizient und überschaubar zu implementieren, ist der `DataManager` als Singleton implementiert, d. h. es existiert zu jedem Zeitpunkt

---

<sup>2</sup>`Package com.oreilly.servlet,`  
URL: <http://www.servlets.com/resources/com.oreilly.servlet/index.html>

<sup>3</sup>Zumindest im Idealfall ; - )

maximal eine Instanz dieser Klasse. Existiert noch keine Instanz, wird bei Ausführung des Konstruktors die Datenbank für die restliche Lebensdauer der Instanz geöffnet.

In der Regel bestehen die Methoden des `DataManager` aus einer Transaktion mit nur lesendem Zugriff auf die Datenbank. Dadurch wird die Sperrung der Objekte, auf die die Transaktion zugreift, auf ein Minimum reduziert.

Die wichtigsten Funktionen des `DataManagers` werden im folgenden kurz beschrieben.

#### **`accessGranted(user, article)`**

Zur Überprüfung der Zugriffsrechte wird diese interne Methode verwendet – der Rückgabewert ist vom Typ `boolean`.

#### **`getArticles(type, startDate, endDate, user)`**

In einer Transaktion werden in dieser Methode alle Artikel vom Typ `type` geladen, die vom Benutzer `user` gelesen werden dürfen und deren Einreichdatum zwischen `startDate` und `endDate` liegt. Rückgabewert ist ein `Vector`-Objekt, das die Artikel absteigend sortiert nach dem Eingangsdatum enthält.

#### **`getUsers()`**

Auch diese Methode liefert einen `Vector` zurück, darin enthalten sind sämtliche registrierte Benutzer. Diese Methode ruft z. B. der `InfoManager` auf, um festzustellen, welcher Benutzer eine Benachrichtigung über neue Artikel erhalten soll (s. Abschnitt 5.2.6).

#### **`store(article, name)`**

Wird als Parameter `name` ein `String` angegeben, so wird der Artikel `article` als sog. „benanntes Objekt“ in der Datenbank gespeichert. Dies ist nur für Benutzerobjekte der Fall, der Name ist dabei stets die (eindeutige) Loginkennung. Anhand des Namens kann direkt auf ein bestimmtes Objekt in der Datenbank zugegriffen werden, was zur Folge hat, dass beim Login direkt über die eingegebene Login-Kennung nur dieses Objekt geladen werden muss.

Alle anderen Artikel werden unbenannt gespeichert (`name` ist `null`); auch diese Objekte sind natürlich eindeutig über ihre Objekt-ID identifizierbar – häufig werden aber mehrere Artikel auf einmal geladen (mithilfe der Methode `getArticles`), sodass der zusätzlich benötigte Speicherplatz für einen Namen jedes Artikelobjekts sich nicht rentieren würde.

**updateUserData(user)**

Dies ist eine der Methoden mit Schreibzugriff auf die Datenbank – sie aktualisiert ein bereits in der Datenbank gespeichertes Benutzerobjekt. Aufgerufen wird diese Methode ausschließlich vom `ArticleSubmitter`, wenn ein Benutzer bereits am System eingeloggt ist und anschließend einen Artikel vom Typ „Benutzer“ einreicht.

**updateArticleCounter(article)**

Auch diese Methode aktualisiert ein Artikelobjekt, und zwar das Attribut `accessCount`, über das jedes `Article`-Objekt verfügt und das die bisherigen Zugriffe auf die Einzelansicht dieses Artikels zählt. Zugegriffen wird auf diese Methode vom `ArticlePresenter`, sobald die entsprechende HTML-Seite an den Client übertragen wurde.

## 5.4 Die Datenebene

Die Datenebene von *InfoNews* besteht aus einer einzigen Datenbank. Da auch die Klassen `Person` und `User` vom Typ `Article` abgeleitet sind, enthält sie auch gleichzeitig sämtliche Daten über die registrierten Benutzer.

Jeder Artikel verfügt über folgende Attribute, die in der Datenbank gespeichert sind:

- Name des Typs (`String`)
- Titel (`String`)
- Untertitel (`String`)
- Text im „Rohformat“ (`String`)
- Autor (`Person`)
- Eingangsdatum (`Date`)
- Flag, ob Benutzer eingeloggt sein muss, um einen Artikel dieses Typs einzureichen (`boolean`)
- URL des Bildes (`String`)
- Liste der Benutzergruppen, die den Artikel lesen dürfen (`String`)
- Objekt-ID des Bezugsartikels (`long`)

- Liste der Folgeartikel (`String`)
- Anzahl der Zugriffe (Einzelansicht) auf diesen Artikel (`int`)

Je nach Artikeltyp werden noch weitere Daten gespeichert (vgl. Abschnitt 4.1 für eine ausführliche Beschreibung der einzelnen Artikeltypen).

## 6 Erweiterungsmöglichkeiten

Im Rahmen dieser Studienarbeit wurde versucht, die technischen Grundlagen für ein intelligentes Informationssystem zu schaffen, mit dem die gezielte Suche nach Informationen aus einem – relativ eng umgrenzten – Bereich ermöglicht werden soll. Beim Entwurf des vorgestellten Systems stand die modulare Struktur im Vordergrund, sodass es leicht möglich ist, spätere Erweiterungen auf einzelne Bereiche anzuwenden (vgl. Kapitel 5).

Im folgenden werden dazu die verschiedenen Ansätze erläutert, mit denen eine gezieltere Suche nach interessierenden Daten ermöglicht wird – was mit dem vorhandenen System nur in geringem Umfang möglich ist.

Neben technischen Ergänzungen, die überwiegend die Handhabung des gesamten Systems verbessern und komfortabler gestalten, sind dabei in erster Linie zwei Ansätze zu nennen, die bereits in anderen Informationssystemen eingesetzt werden: Ontologien und persönliche Benutzerprofile (s. Abschnitte 6.2 und 6.3).

### 6.1 Technische Ergänzungen

Unter technischen Ergänzungen werden in diesem Zusammenhang Erweiterungen von *InfoNews* verstanden, die in erster Linie die Bedienung des Programms komfortabler gestalten und die Funktionalität des Systems erweitern.

#### **Administrator-Account**

Einer der momentanen Schwachpunkte von *InfoNews* ist die Verteilung der Zugriffsrechte auf die einzelnen Artikel. Der Benutzer wählt bei der Registrierung selbst eine Benutzergruppe aus, zu der er gehört. Anhand dieser Auswahl erkennt das System, welche Artikel ein Benutzer lesen darf und welche nicht.

Damit ergeben sich automatisch zwei Nachteile: Diese Methode ist unsicher, da das System nicht überprüfen kann, ob der Benutzer tatsächlich beispielsweise Professor, in Wirklichkeit aber ein Student ist. Der andere Nachteil ist, dass ein Benutzer nur einer Benutzergruppe angehören kann.

Jedes einigermaßen sichere System verfügt darum über einen Administrator-Account, über den diese Einstellungen von einer einzelnen Person vorgenommen werden. Dadurch ist sichergestellt, dass ein Benutzer nicht eigenhändig seine Benutzergruppe ändern kann. Indem die interne Speicherung der Benutzergruppe angepasst wird, kann man außerdem erreichen, dass man zu mehr als einer Gruppe zählen kann, wie man es von

Multiuser-Betriebssystemen gewohnt ist.

Der Administrator ist außerdem für die Wartung und Pflege des Systems zuständig, indem er beispielsweise aufgetretene Fehlermeldungen analysiert und an den Programmierer weiterreicht (sofern er das nicht selbst ist).

Über ein Konfigurationsprogramm (evtl. mit grafischer Benutzeroberfläche, z. B. ein Java-Applet) sollte der Administrator außerdem die Möglichkeit besitzen, weitere Einstellungen des Systems vorzunehmen und bei Bedarf neue Artikeltypen hinzuzufügen.

### **Zusätzliche Benutzerdaten**

Um den Informationsaustausch zwischen den Benutzern zu erhöhen, können weitere Benutzerdaten angegeben werden, anhand derer nach Personen mit bestimmten Eigenschaften gesucht werden kann.

Man denke beispielsweise an einen russischen Informatiker, der einem Interessierten seine brandneue Veröffentlichung zuschickt – auf russisch. Sofern man nicht selbst dieser Sprache mächtig ist, stellt sich die Frage, wer bei der Übersetzung helfen kann. *InfoNews* ist prinzipiell hervorragend dafür geeignet, in so einer Situation weiterzuhelfen, indem ein Benutzer bei seiner Registrierung in einer zusätzlichen Kategorie sein Arbeitsgebiet, Spezialkenntnisse, Hobbies etc. anhand von Stichworten beschreibt.

Bei der Suche nach solchen Stichworten werden dann die passenden Personen aufgelistet.

### **Verbesserte Multimedia-Unterstützung**

Der Einsatz von Multimedia-Elementen bei *InfoNews* ist noch recht beschränkt, durch Erweiterung auf andere Grafikformate wie GIF, PNG oder auch PostScript kann der Bedienungskomfort beim Einreichen eines neuen Artikels noch erhöht werden.

Wird beim Schreiben eines Artikels kein Bild ausgewählt, so kann auf ein Foto des Autors zurückgegriffen werden, um das Erscheinungsbild des Artikels interessanter zu machen.

Autoren von Artikeln sollten außerdem in der Lage sein, mehrere Bilder je Artikel auf den Server laden zu können, um nicht auf extern über das WWW zugängliche Grafiken zurückgreifen zu müssen. Ferner ist es zweckmäßig, wenn bereits auf dem Server befindliche Grafiken wieder verwendet werden können; das setzt voraus, dass beim Einreichen eine Vorschau aller vorhandenen Grafiken angezeigt wird („Thumbnails“), die dann per Mausklick selektiert werden können.

Für diese Funktionalität ist außerdem erforderlich, dass aus den tatsächlich gespeicherten Bildern eine solche Thumbnail-Grafik erzeugt wird, die deutlich kleiner ist. Der Aufbau der Seiten im Browserfenster erfolgt dann wesentlich schneller, da anstatt der kompletten großen Grafik, die im HTML-Code auf die richtige Größe gesetzt wird, nur diese kleine Grafik übertragen werden muss.

Um sich von einer gedruckten Zeitung zu unterscheiden und die Vorteile des World Wide Web voll auszuschöpfen, sollte auch die Unterstützung von anderen Multimedia-Objekten wie Animationen (AVI, MPEG) und evtl. auch Audiodateien (WAV, MP3) realisiert werden.

### Thesauri

Ein wichtiger Punkt auch in Zusammenhang mit der intelligenteren Artikelsuche (vgl. Kapitel 6.3) ist der Einsatz von Thesauri. Anhand von Stichworten kann ein Autor den Inhalt seines Artikels beschreiben, sodass man nach bestimmten Stichworten suchen kann. Die Ergebnisliste enthält dann, sortiert nach Übereinstimmung, Artikel mit diesem oder verwandten Stichworten.

Als Grundlage für die Stichworte kann beispielsweise der 1998 von der ACM<sup>1</sup> veröffentlichte Thesaurus dienen. Jedoch ist es unbedingt notwendig, dass diese Auflistung editierbar ist, d. h. falls ein Autor nicht das gewünschte Stichwort findet bzw. die Klassifizierung zu grob ist, kann er weitere Einträge hinzufügen.

Um auch auf andere Systeme und Fachgebiete übertragbar zu sein, sollte das System über eine Importmöglichkeit verfügen, mit der auch andere Thesauri und Schlagwortkataloge in das System integriert werden können. Denkbar sind z. B. Anwendungen in der Akustik, für die ein sehr ausgefeiltes Klassifikationsschema existiert<sup>2</sup>.

### Datenverwaltung

*InfoNews* verwaltet die Artikeldaten in der objektorientierten Datenbank POET<sup>3</sup>. Dieses Datenbank-Managementsystem (DBMS) ist kommerziell, jedoch für Testzwecke und im Rahmen einer Diplomarbeit frei erhältlich. Ein weiterer Nachteil besteht darin, dass das System nur für die Plattformen 32Bit-Windows und Solaris erhältlich ist – eine Tatsache, die der Plattformunabhängigkeit von Java im Wege steht.

<sup>1</sup>URL: <http://www.acm.org/class/1998/>

<sup>2</sup>URL: <http://www.aip.org/pacs/pacs99/pacs99app.html>

<sup>3</sup>URL: <http://www.poet.de/>

Aus diesem Grund wurde bereits während der Implementierung des vorgestellten Systems darauf geachtet, die Servlets durch die dazwischenliegende Datenabstraktionsebene (s. Abschnitt 5.3) von der darunter liegenden Datenbank zu trennen. Durch diese schichtenförmige Architektur ist es möglich, das verwendete DBMS auszutauschen, ohne die Servlets grundlegend anpassen zu müssen.

Setzt man als Datenbank ein relationales System ein, so ist zwar der Aufwand, den `DataManager` für dieses System zu implementieren, zunächst recht hoch, ein Vorteil wurde aber bereits genannt – die Plattformunabhängigkeit, sodass *InfoNews* auch unter Betriebssystemen eingesetzt werden kann, die von POET nicht unterstützt werden.

Ein weiterer eher technischer Vorteil ist, dass die Probleme während der Entwicklungsphase beim Zusammenspiel zwischen den drei Komponenten Entwicklungsumgebung, Servlets und DBMS deutlich reduziert werden, da das Java API über ein umfangreiches Sortiment an Klassen und Methoden zur Anbindung an relationale Datenbanksysteme verfügt.

### Übergeordnetes Servlet zur Organisation

Ein dem gesamten System übergeordnetes Servlet kann beispielsweise folgende Aufgaben übernehmen:

- Ein situationsabhängiges Inhaltsverzeichnis anzeigen (z. B. macht der Eintrag „Logout“ nur dann Sinn, wenn ein Benutzer bereits eingeloggt ist).
- Externe Hyperlinks auf Artikel in der Datenbank an den `ArticlePresenter` weiterleiten und diese ebenfalls mit einem passenden Inhaltsverzeichnis anzuzeigen.
- Falls ein Benutzer dies in seinen Benutzerdaten aktiviert hat, kann er automatisch vom System kurz vor Beginn einer Veranstaltung per E-Mail oder ICQ daran erinnert werden.

### Interne Artikelverarbeitung des `ArticleSubmitters`

Auch die interne Verarbeitung der Artikelobjekte kann verbessert werden. Zur Zeit wird der Artikeltyp `Person` teilweise gesondert behandelt, um die Überprüfung der Nachnamen in einem Personenfeld eines Artikels zu ermöglichen (z. B. der Erstautor einer Veröffentlichung).

Mithilfe des `Reflection-Packages`, das Teil des Java API ist, kann diese Sonderrolle behoben werden und unabhängig vom Typ `Person` jeder andere Artikeltyp – pro Feld dann auch mehrere hintereinander – referenziert werden.

### Verbesserte Suche

Die zur Zeit in *InfoNews* implementierte Suchfunktion deckt nur die nötigsten Bereiche ab; neben einer Suche nach verwandten Stichworten kann auch eine Suche nach regulären Ausdrücken hilfreich sein. Auf die Erweiterung der Suchfunktion wird im nächsten Abschnitt noch näher eingegangen.

## 6.2 Ontologien

Jede Wissensbasis und jedes wissensbasierte System basiert auf einer Konzeptualisierung, d. h. einer formalen Repräsentation der Objekte und anderer Entitäten aus einem bestimmten Themenbereich. Unter Ontologien versteht man in diesem Zusammenhang eine explizite Beschreibung dieser formalen Zusammenhänge [4].

Eingesetzt werden Ontologien, um zu ermöglichen, dass mehrere (Software)-Agenten auf ein gemeinsames Vokabular zur Repräsentation von Wissen eines bestimmten Gebiets zugreifen können und um die Struktur von komplexen Objekten auf eindeutige Weise zu formalisieren. Aus diesem Grund eignen sich Ontologien besonders gut für die Beschreibung von heterogenen, verteilten Informationsquellen.

Als Beispiel sei hier zunächst das bereits in Abschnitt 2.4 erwähnte System *On2Broker*<sup>4</sup> angeführt, das Ontologien einsetzt, um die Semantik von Webseiten-Inhalten zu beschreiben. Das System ist in die folgenden Einzelkomponenten untergliedert [3]:

- Eine *Anfrage-Schnittstelle* zur Verarbeitung von Suchanfragen; diese werden in entsprechende Anfragen an die Datenbank übersetzt.
- Ein *Info-Agent*, der für das Sammeln von Faktenwissen zuständig ist und die Semantik der gefundenen Seiten anhand von Metaangaben in den HTML-Tags oder direkt aus XML-Tags ableitet und in der Datenbank speichert.
- Ein *Inferenzagent*, um aus Fakten mithilfe von Ontologien zusätzliche Fakten abzuleiten, die ebenfalls in der Datenbank gespeichert werden. Damit wird also auch implizites Wissen gespeichert.
- Ein *Datenbankmanager*, der für die eigentliche Datenverwaltung zuständig ist – er bekommt Fakten des Info-Agenten, tauscht mit dem

---

<sup>4</sup>URL: <http://www.aifb.uni-karlsruhe.de/www-broker>

Inferenzagenten Daten als Input und Output und liefert Fakten an die Suchschnittstelle.

Ein anderes System, das den Einsatz von Ontologien mit dem eines Online-Informationssystems kombiniert, ist Planet-Onto<sup>5</sup>, eine Erweiterung von Kmi-Planet [2].

Hier sind in der Regel die Autoren der eingereichten Artikel selbst für die Beschreibung der Semantik zuständig. Dazu wird ein formularbasiertes Eingabewerkzeug bereitgestellt, mit dem auch Personen arbeiten können, die über kein ausgereiftes Wissen bezüglich Wissensakquisition verfügen [1].

Bei der Erweiterung von *InfoNews* um den Einsatz von Ontologien können diese Systeme als prinzipielle Anregung dienen. In einem zusätzlichen Schritt kann der Autor beispielsweise – ähnlich wie bei Planet-Onto – Hinweise auf die Semantik des Inhalts geben, die automatisch vom System ausgewertet werden und die Datenbank aktualisieren.

### 6.3 Persönliche Benutzerprofile

Neben den im vorigen Abschnitt erläuterten Möglichkeiten, das System in Bezug auf die Suchtechnik zu verbessern, ist der Einsatz von persönlichen Benutzerprofilen ein Ansatz, das System als Ganzes intelligenter zu machen.

Dabei wird das Verhalten der einzelnen Anwender bei der Nutzung des Systems mitprotokolliert und ausgewertet, um sie jeweils in erster Linie nur mit den Informationen zu versorgen, an denen sie besonders interessiert sind.

Dazu ein Beispiel: Angenommen das System wird im gesamten Institut für Informatik eingesetzt und ein Benutzer informiert sich regelmäßig über Tagungen, Vorträge und Ereignisse aus dem Bereich der künstlichen Intelligenz – jedoch nie oder nur äußerst selten über andere Themen, wie beispielsweise der Netzwerktechnik. Anhand eines persönlichen Benutzerprofils ist es nun möglich, dieses Verhalten aufzuzeichnen und dem Benutzer bei weiteren Sitzungen zunächst nur die Artikel zu präsentieren, die aus dem Bereich der künstlichen Intelligenz stammen, während Artikel zum Thema Vernetzung automatisch „verschwiegen“ werden. Über entsprechende Verweise hat der Benutzer aber natürlich trotzdem Zugriff auf diese ausgeblendeten Artikel.

<sup>5</sup>URL: <http://kmi.open.ac.uk/planet/index.html>

Technisch kann die Erstellung und Aktualisierung eines Benutzerprofils über verschiedene Methoden realisiert werden.

Zum einen kann der Benutzer selbst bei Angabe seiner Benutzerdaten während der Registrierung Themenbereiche anhand von Schlagworten angeben, die ihn überwiegend interessieren. Anhand dieser Daten kann bereits ein erstes grobes Profil erstellt werden.

Zum anderen kann die Aufenthaltsdauer des Benutzers auf einzelnen Artikelseiten dazu verwendet werden, sein Interesse an einzelnen Sachgebieten festzustellen und seine angegebenen Themenbereiche zu verifizieren und zu aktualisieren. Dazu kann der bereits in Abschnitt 6.1 erwähnte Thesaurus herangezogen werden.

Im Laufe der weiteren Nutzung lernt das System allmählich, dieses Profil zu verfeinern. Diese Verfeinerung hat beispielsweise auf die folgenden Punkte Einfluss:

- Es werden nur bestimmte Artikeltypen präsentiert.
- Je nach dem, wie oft er auf ältere Artikel im Archiv zugreift, werden überwiegend neuere oder auch ältere Artikel bei Präsentation von Suchergebnissen höher bewertet.
- Die Zahl der präsentierten Verweise in einem Artikeltext wird dem Benutzer angepasst, indem das System das Folgen von Hyperlinks protokolliert.
- Durch Kombination mit der Suchfunktion des Systems kann ein anwenderspezifisches Ranking der Suchergebnisse erfolgen.
- Die Benachrichtigung neu eingegangener Artikel kann sich auf Artikel beschränken, die den Anwender interessieren.

## 7 Zusammenfassung und Ausblick

In dieser Arbeit wurde *InfoNews* vorgestellt, eine Online-Zeitung im World Wide Web, die in erster Linie am Lehrstuhl für Informatik VI eingesetzt werden soll. Aufgrund der modularen Programmierung ist es jedoch möglich, mit geringem Aufwand das System in andere Bereiche zu übertragen, sodass auch dort die Vorzüge einer Online-Zeitung zum Einsatz kommen können.

Das System dient vor allem dem schnellen Informationsaustausch zwischen einzelnen Personen und Benutzergruppen, indem Artikel geschrieben werden können, die nach Aufnahme in die Datenbank sofort für andere Benutzer sichtbar sind. Die Artikel sind monatsweise in einzelnen Ausgaben zusammengefasst, wobei die neuesten Artikel auf der Eingangsseite im Überblick dargestellt sind, ältere Artikel und Ausgaben können über entsprechende Verweise in der Archivansicht abgerufen werden.

Um das Einreichen der Artikel etwas überschauen zu können, erfordert *InfoNews* eine Registrierung, durch die man sich später am System einloggen kann, um z. B. neue Artikel zu schreiben.

Beim Entwurf des Systems wurde auf eine möglichst intuitiv klare Bedienbarkeit und die Einbettung ins WWW geachtet. So wird bei der Anzeige eines Artikels automatisch versucht, Hyperlinks zu anderen Artikeln in der Datenbank sowie zu Personen bzw. deren Homepage und E-Mail-Adresse zu generieren, um dem Leser die weitere Informationsbeschaffung im Internet zu erleichtern.

Durch diesen bisher nur prinzipiellen Ansatz, der in zukünftigen Versionen jedoch noch weiter ausgebaut werden kann, wird versucht, das Informationsangebot im WWW auf interessante Seiten zu beschränken. Verantwortlich für die Güte der angebotenen Links ist der Autor des jeweiligen Artikels, da dieser allein für den Inhalt zuständig ist.

Durch die Möglichkeit, dass der Autor je nach Art der Information, die er bereitstellen möchte, aus verschiedenen Artikeltypen auswählen kann, können außer dem eigentlichen Artikeltext noch weitere spezifische Daten angegeben werden, die dem Leser ebenfalls helfen, schneller die ihn interessierenden Artikel zu finden.

Um das Archiv aller Artikel nach bestimmten Informationen zu durchsuchen, kann der Benutzer über ein Suchformular außer einer Zeichenkette auch die Artikeltypen sowie die Monats-Ausgabe selektieren, um seine Suche auf bestimmte Bereiche des Archivs zu beschränken.

Einsatzgebiete für *InfoNews* gibt es mehrere; außer als Informationssystem am Lehrstuhl kann es dort insbesondere für die Kommunikation der Mitglieder des Arbeitskreises Robotik verwendet werden. Der Arbeitskreis besteht aus mehreren Studenten, die – so ist es geplant – eine Steuerungssoftware für mehrere autonome mobile LEGO Mindstorms-Roboter entwickeln wollen. Mithilfe von *InfoNews* kann die Koordination und Aufgabenverteilung innerhalb des Arbeitskreises enorm vereinfacht und effizienter gestaltet werden.

Eine weitere Möglichkeit, das hier vorgestellte System praktisch einzusetzen ist eine umfangreiche Bibliographie zu einem Teilgebiet der Akustik, der „aktiven Lärm- und Schwingungsbekämpfung“. Diese Bibliographie umfasst inzwischen weit über 10 000 Einträge und könnte mit dieser Software auf einfache Weise im Internet präsentiert werden und so weltweit den Forschern auf diesem Gebiet eine einfache und effiziente Möglichkeit der Informationsbeschaffung zur Verfügung stellen.

# A Installation und Konfiguration

## A.1 Installation

*InfoNews* ist als Java-Package `infonews` implementiert. Dies bedeutet, dass der Webserver die `class`-Files stets in einem Verzeichnis `infonews` unterhalb der in der folgenden Aufstellung angegebenen Verzeichnisse sucht.

1. Installation eines Webserver, der mindestens das Java Servlet API Version 2.1 unterstützt. Getestet wurde das System mit JRun 2.3<sup>1</sup>, Apache 1.3.9<sup>2</sup> und Tomcat 3.0<sup>3</sup>.
2. Installation von POET in ein beliebiges Verzeichnis.
3. Anpassung der Webserver-Konfiguration, sodass die POET-Bibliotheken und die POET-JAR-Files gefunden werden.
4. Kopieren der statischen HTML-Dateien von *InfoNews* in ein gewünschtes Verzeichnis unterhalb des „Document-Root“-Verzeichnisses des Webserver.
5. Anpassen des Servernamens und -ports in `index.html` und `toc.html`, damit die Servlets gefunden werden.
6. Kopieren der Servlet-Classfiles (`ArticleSubmitter`, `ArticleFinder`, `ArticlePresenter` und `UserManager`) in das Servlet-Verzeichnis unterhalb des Installationsverzeichnisses des Webserver (häufig `/servlet` oder `/servlets`).
7. Kopieren aller anderen `class`-Files in ein Verzeichnis, das im `CLASSPATH` des Webserver aufgeführt ist.
8. Kopieren der Dateien `poet.cfg` und `infonews.conf` in das Installationsverzeichnis des Webserver bzw., falls von diesem verschieden, in das „Document-Root“-Verzeichnis des Servers. Bei JRun ist dies das Verzeichnis `jsm-default` im Installationsverzeichnis, bei Tomcat und Apache das Installationsverzeichnis selbst.
9. Kopieren des gesamten Datenbank-Verzeichnisses `databases` in ein beliebiges Verzeichnis auf dem Server-Rechner.

---

<sup>1</sup>URL: <http://www.livesoftware.com/products/jrun/>

<sup>2</sup>URL: <http://www.apache.org/>

<sup>3</sup>URL: <http://jakarta.apache.org/tomcat/>

10. Anpassen der POET-Konfigurationsdatei `poet.cfg`, in dem die Verzeichnisse für die Datenbank selbst und das Datenbankschema gemäß dem im vorigen Schritt gewählten Verzeichnis eingetragen werden müssen.
11. Anpassen der InfoNews-Konfigurationsdatei `infonews.conf`, wie im nächsten Abschnitt beschrieben.

## A.2 Konfiguration

*InfoNews* wird über die zentrale Konfigurationsdatei `infonews.conf` im Installationsverzeichnis des Webservers an die jeweilige Umgebung angepasst.

Im folgenden werden sämtliche Einträge dieser Datei aufgeführt und kurz erläutert. Kommentare innerhalb der Datei werden wie von C++ und Java bekannt in `/* . . . */` eingeschlossen oder hinter `//` angegeben. Zwischen Groß- und Kleinschreibung wird unterschieden.

### Server:

Dies ist der Name des Servers im Netz mit Portnummer, auf der Webserver Requests an Servlets verarbeitet.

**Default:** `localhost:8080`

### ServletPath:

Dies ist das relative Verzeichnis zum Installationsverzeichnis des Webservers, in dem die `class`-Files der Servlets liegen.

**Default:** `/servlet`

### Homepage:

Dies ist die Homepage von InfoNews.

**Default:** `/infonews/index.html`

### ImagePath:

Dies ist das relative Verzeichnis zum Installationsverzeichnis des Webservers, in dem die auf den Server geladenen Bilder zu den Artikeln gespeichert werden.

**Default:** `/infonews/images`

### IconPath:

Dies ist das relative Verzeichnis zum Installationsverzeichnis des Webservers, in dem die Icons für Links zu Homepage etc. liegen.

**Default:** `/infonews/icons`

**MaxImageSize:**

Dies ist die maximale Größe einer auf den Server geladenen Grafik in Bytes. Größere Grafiken werden abgelehnt.

**Default: 1 048 576 (1 MB)**

**ArticleTypes:**

Dies ist eine Liste aller Artikeltypen, die das System verwenden soll. Angegeben werden muss jeweils der komplette Klassenname (also mit Package-Name). Trennung der einzelnen Einträge mit `,` `.` oder `:`. Blanks (Whitespace) werden entfernt.

**Default: —**

**UserGroups:**

Dies ist eine Liste aller Benutzergruppen, die dem System bekannt sind (die Gruppe „Admin“ wird automatisch geladen, zur Zeit aber vom System noch nicht unterstützt). Trennung auch hier mit `,` `.` oder `:`. Blanks (Whitespace) werden ebenfalls entfernt.

**Default: —**

**PageHeader:**

Dies ist der absolute Pfad der Datei mit dem Seitenkopf, der für alle dynamisch generierten Seiten identisch ist. Zu beachten sind die doppelten Backslashes.

**Default: —**

**LogFile:**

Dies ist der absolute Pfad der Log-Datei. Hier werden Daten über die einzelnen Servlet-Zugriffe protokolliert.

**Default: <server\_root>/infonews.log**

**ErrorLogFile:**

Dies ist der absolute Pfad der Fehlerlog-Datei. Hier werden aufgetretene Fehler und Ausnahmen protokolliert.

**Default: <server\_root>/errors.log**

**MailServer:**

Dies ist der Hostname des Mailservers, den der InfoManager verwendet, um Benachrichtigungsmails an die Benutzer zu verschicken.

**Default: localhost**

**MailAccount:**

Dies ist der Mail-Account, der als Absender verwendet wird.

**Default: infonews**

## Literatur

- [1] DOMINGUE, J., AND MOTTA, E. A knowledge-based news server supporting ontology-driven story enrichment and knowledge retrieval. In *11th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW '99), Dagstuhl, Germany, May 26–29, 1999* (1999).  
URL: [http://kmi.open.ac.uk/~john/papers/planet\\_onto.pdf](http://kmi.open.ac.uk/~john/papers/planet_onto.pdf).
- [2] DOMINGUE, J., AND SCOTT, P. Kmi planet: A web based news server. In *Asia Pacific Computer Human Interaction Conference (APCHI '98), Shonan Village Center, Hayama-machi, Kanagawa, Japan, July 15–17, 1998* (1998).
- [3] FENSEL, D., ANGELE, J., DECKER, S., ERDMANN, M., SCHNURR, H.-P., STAAB, S., STUDER, R., AND WITT, A. On2broker: Semantic-based access to information sources at the www. In *Proceedings of the World Conference on the WWW and Internet (WebNet 99), Honolulu, Hawaii, USA, October 25-30, 1999* (1999).  
URL: <ftp://ftp.aifb.uni-karlsruhe.de/pub/mike/dfe/paper/webnet.pdf>.
- [4] GRUBER, T. R. What is an ontology?  
URL: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>.
- [5] HUNTER, J., AND CRAWFORD, W. *JAVA Servlet Programming*. O'Reilly & Associates, Sebastopol, CA, 1998.
- [6] MÜNZ, S. SelfHTML: HTML-Dateien selbst erstellen.  
URL: <http://www.teamone.de/selfhtml/>.
- [7] NEUBERGER, C. Die deutschen Tageszeitungen im World Wide Web. Redaktionen, Nutzer, Angebote. *Media Perspektiven* 12 (1997), 652–662.  
URL: <http://www.ku-eichstaett.de/SLF/JOUR/projekt/ergebnis/main.htm>.