

# Informationsserver für das Internet oder Die Suche nach der Nadel im Heuhaufen

Manfred A. Jeusfeld, Informatik V, RWTH Aachen  
<http://www-i5.informatik.rwth-aachen.de/mjf/>

Im Oktober 1996 veranstaltet die EMISA ihr Fachgruppentreffen zu dem Thema Informationsserver für das Internet. Als das Thema festgelegt wurde, war 'Internet' schon ein Modewort. Inzwischen hat es gute Chancen zum Wort des Jahres zu werden. Es ist also angebracht, das Thema in das Spektrum der Fachgruppe einzuordnen und darzulegen, welchen besonderen Beitrag die EMISA zu leisten vermag. Ganz nebenbei soll sie dieser Artikel neugierig auf das Fachgruppentreffen machen und hier und dort zur Diskussion oder gar zum Widerspruch anregen.

## 1. Internet und EMISA

Das Internet ist im Grunde nur eine Familie von Datenaustauschprotokollen wie z.B. FTP (file transfer protocol), HTTP (hypertext transfer protocol) und SMTP (small mail transfer protocol). Warum also sollte es Thema für die EMISA sein? Schließlich widmet sich die EMISA den Methoden zur Entwicklung von Informationssystemen und nur am Rande mit Kommunikationsprotokollen! Wenn auch das Internet keine besonders neue Technik ist, so ist sein Zusammenhang mit Informationssystemen zu klären. Bevor wir dies tun, erinnern wir uns an die Entwicklung in den letzten 10 Jahren.

Bevor das HTTP-Protokoll Anfang der 90er Jahre startete, wurden im Internet-Bereich FTP und SMTP zur Übertragung von Informationen genutzt. Das Internet hat den Vorteil, daß jeder Knoten im Rechnernetz einen eindeutigen Identifikator, die sogenannte **Internet-Nummer**, und einen (fast) eindeutigen Namen hat. Dieser wird ähnlich einer Telefonnummer zur Vermittlung von Diensten und zum Austausch von Daten zwischen entfernten Rechnern genutzt. Mit zunehmender Vernetzung wurden immer mehr Informationsquellen für immer größere Nutzerkreise zugreifbar. Voraussetzung war allerdings die Kenntnis der Internet-Nummer des anbietenden Rechners (bzw. seines Internet-Namens) und des Informationsstyps (Text, Binärprogramm für Amiga, Bilddaten, usw.). Dies brachte eine Gruppe um Tim Berners-Lee beim CERN auf die Idee, ein neues Internet-Protokoll HTTP zu kreieren, das dem Nutzer diese Arbeit abnimmt. Das Protokoll fußt auf drei Neuigkeiten:

1. Alle Dokumente werden durch einen URL (Uniform Resource Locator) weltweit eindeutig identifiziert. Der URL besteht aus den Teilen Protokollname, Internet-Name des Rechners und lokaler Dokumentpfad.
2. Ein Informationsserver (Rechner, der Dokumente nach dem HTTP-Protokoll anbietet) typisiert Dokumente, d.h. ein Klient bekommt neben dem eigentlichen Dokument dessen Typ mitgeteilt.

3. Eine strukturierte Dokumentbeschreibungssprache HTML (hypertext markup language) kann URL's anderer Dokumente als Verweise enthalten. Die Verweise nennt man auch Anker. Die Dokumente nennt man verallgemeinernd Hypertexte

Diese drei Ideen waren Grundlage der Entwicklung der WWW-Browser. Vereinfachend sind das Programme, die HTML-Dokumente auf dem Bildschirm darstellen können und die das Verfolgen von Verweise in den Dokumenten komfortabel unterstützen. Die Benutzerfreundlichkeit der WWW-Browser führte zu einem dramatischen Erfolg des HTTP:

- 60% aller im Internet ausgetauschten Daten werden nach HTTP übertragen (Quelle: Martin Koster)
- die Anzahl der Informationsserver stieg von wenigen Dutzend im Jahre 1991 auf mittlerweile über zweihunderttausend (Quelle: Digital Corp., 1996)
- mehr als 40 Millionen Dokumente verschiedenster Art werden angeboten
- der populärste WWW-Browser Netscape wurde für mehr als 38 Millionen Benutzern installiert (Quelle: Nando Times, 1996) und ist damit das am häufigsten installierte Programm der Welt.

Drei Verwendungsarten haben sich im Konsumentenverhalten mit dem WWW herausgebildet: 1) der WWW-Browser als uniforme Benutzerschnittstelle zum Internet (FTP, HTTP, SMTP), 2) der WWW-Browser als Werkzeug zum Nachschlagen von Information und 3) der WWW-Browser als Fernsehersatz (Quelle: Georgia Tech Research Corp., 1995). Dies weist darauf hin, daß die überwiegende Mehrheit das Internet zum Lesen von Information nutzt. Der Nutzungstrend läuft also offenbar in Richtung Massenmedium. Wir wollen uns aber zunächst auf die Aspekte der Entwicklung konzentrieren

Betrachten wir dazu das WWW als eine sehr verteiltes multimediale Informationssystem mit sehr vielen Entwicklern. Wesen dieses Informationssystems ist ihre unkontrollierte Evolution. Keine zentrale Stelle koordiniert das Hinzufügen neuer Dokumente. Jeder Benutzer mit Schreibberechtigung auf einem Informationsserver kann prinzipiell Dokumente (und wie wir später sehen werden auch Dienstprogramme) bereitstellen. Nicht einmal die Liste der erlaubten Dokumenttypen (Datentypen, Schemaklassen) wird zentral verwaltet. Es gibt lediglich eine Liste empfohlener Dokumenttypen. Sie kann für jeden Informationsserver erweitert und verändert werden. Unbestreitbar war und ist die fehlende Kontrolle ein großer Vorteil für das schnelle Wachstum. Aber wie kann man nun das geballte Wissen nutzen?

**Das verteilte Wissen wird verfügbar, wenn man das WWW um die Standardkomponenten eines Informationssystems erweitert. Es fehlen Methoden zur Entwicklung des WWW in Richtung multimedialer, typenreicher Datenbank.**

Wenn diese These richtig ist, so müssen die Standardkomponenten eines Informationssystems übertragbar sein, also u.a. Datenmodellierungssprache, Datenmanipulationssprache, Anfragesprache, Transaktionsverwaltung und Integritätssicherung.

## 2. Datenmodellierung im WWW

Für konventionelle Informationssysteme haben sich graphische Modellierungssprachen (Entity-Relationship-Diagramme, Bachman-Diagramme usw.) bewährt. Auch wenn Unterschiede in der Anzahl der Knoten- und Kantentypen, so ist diesen Sprachen gemeinsam, daß Konzepten (Knoten) untereinander in Beziehung stehen (Kanten). Einerseits wird so die ein Weltausschnitt modelliert, andererseits ist die Aufteilung in Konzepte und Beziehungen bereits nah am vorherrschenden Speichermodell: Daten sind strukturierte Stücke eines uniformen Speichers, die über Adressen (Speicheradressen, Objektidentifikatoren, Schlüssel etc.) aufeinander verweisen können. Die Struktur nennt man auch Datentyp. Der Datentyp ist bekanntlich eng an die Operationen (Anwendungsprogramme) gekoppelt, die auf den Daten arbeiten. So können Compiler bei der Analyse eines Programmes feststellen, ob es typverträglich ist.

Was sind nun die Daten im WWW? Es sind Dokumente, genauer Folgen von Zeichen, für die der Informationsserver nur einen Typbezeichner (z.B. application/x-matrix) übermittelt. In welcher Beziehung ein solcher Typ zu anderen Typen stehen (etwa, ob er ein kartesisches Produkt aus application/x-vector) ist, ist nur bei Dokumenten des Typs text/html explizit: solche Dokumente enthalten beliebig viele Verweise auf andere Dokumente. Wir haben also eine Menge von Knoten und eine Menge von Kanten, über deren Typ allerdings nichts weiter bekannt ist! Der Mangel ist leicht erklärbar: das HTTP-Protokoll ist aus einem Dateitransferprotokoll entwickelt worden. Dateien haben in den meisten Betriebssystemen eine Dateityp, der im Dateinamen kodiert wird). Dieser Dateityp soll anzeigen, welche Anwendungsprogramme eine Datei lesen bzw. schreiben dürfen. Kleinere und größere Einheiten als Dateien werden nicht besonders unterstützt. Wir fordern daher:

**Entwickler für Informationsserver im WWW brauchen eine reichhaltigere Datenmodellierungssprache, in der es insbesondere möglich ist, Verweise zwischen Objekten zu typisieren.**

Die Forderung muß unter der Nebenbedingung erfüllt werden, daß die Dokumenttypen weiterhin dezentral definiert und verändert werden. Ein Informationsserver, der nur lokale Daten verwaltet und sie über eine Schnittstelle dem WWW zugänglich macht, stellt keinen besonderen Unterschied zu konventionellen Informationssystemen dar. Interessant wird es erst, wenn die Objekte eines solchen Informationsservers auf entfernte Objekte (von anderen Informationsservern) verweist.

Ein erfolgversprechender Ansatz zu einer verteilten Datenmodellierung ist das Harvest-Konzept von der Universität Colorado-Boulder: ein Informationsserver sammelt die Typinformationen (Metadaten) über die von ihm bereitgestellten Daten in einem per WWW zugreifbaren Index. Datenmodellierer können dann von außen auf diese Schemadaten zugreifen. Zur Zeit wird dieses Konzept hauptsächlich zum Suchen nach Dokumenten eingesetzt. Das Problem ist, daß es keine Einigkeit über die Datenmodellierungssprache gibt. Offenbar muß man einer Entwicklergruppe, eine Möglichkeit geben, sich spontan auf eine solche Sprache zu einigen. Voraussetzung der Einigung ist eine gemeinsame Terminologie und ein gemeinsames Ziel.

Das Harvest-Konzept kann man auch auf kleinere Einheiten als Informationsserver anwenden. Im Extremfall enthält jedes bereitgestellte Datum seine eigene Typinformation. Diese Idee findet man in Datenaustauschformaten wie OEM (Papakonstantinou et al.) verwirklicht: ein Zahlobjekt wie 123 wird als (Integer,123) repräsentiert. Die erste Komponente ist die Typinformation, die zweite der Wert. Solche Objekte reflektieren die fehlende zentrale Datenmodellierung, ohne auf Typen zu verzichten. Die Herausforderung ist, aus vielen Typbeschreibungen ein sinnvolles Schema zu extrahieren.

### 3. Anfragesprachen im WWW

Die vorherrschende 'Anfragesprache' ist das manuelle Navigieren: ein Informationsserver bieten die Dokumente über eine Schnittstelle an, in der Verweise auf andere Dokumente einfach per Tastendruck verfolgt werden. Auf diese Weise kann eine Benutzerin oder ein Benutzer allerdings nur einen Bruchteil der relevanten Information aufspüren. Zudem eignet sich dieses Verfahren nicht als Anfragesprache in Anwendungsprogrammen.

Als Suchhilfe haben sich von sogenannten WWW-Robotern (Alta Vista, WebCrawler, Lycos etc.) erstellte Indexe bewährt. Sie verfolgen systematisch Verweise von Dokumenten und bauen eine Metadatenbank über die aufgesuchten Dokumente auf. Der Inhalt der Metadatenbank ist meist ein invertierter Index der vorkommenden Schlüsselwörter. Anfragen an die Metadatenbank werden nicht mit klassischen Datenbank-Anfragesprachen gestellt, sondern mit Ausdrücken, welche die Relevanz eines Dokuments bezüglich des Vorkommens von Termen messen. Verfahren aus dem Information Retrieval kommen hier zur Anwendung.

Das Fehlen einer strukturierten Anfragesprache hängt mit der schwachen Datenmodellierungssprache zusammen: wenn man wenig über die Struktur der Daten weiß, so kann man auch nur relativ primitive Anfragen stellen. Insbesondere weiß man wenig über die Struktur der Antwortdokumente. Für die Anwendungsprogrammierung ist dies besonders nachteilig, da vor Ausführung eines Programms die Zugehörigkeit eines Eingabeobjektes zur Menge der erlaubten Objekte (=Typ) zu testen ist. Daher fordern wir:

**In einer Anfragesprache für das WWW sollte die Zugehörigkeit eines Dokuments (Objekts) zu seinem Typ testbar sein. Getypte Beziehungen sollten als solche abfragbar sein.**

Wiederum ist zu beachten, daß die Menge der erlaubten Typen sich jederzeit ändern kann, da wir hier das WWW als verteiltes Informationssystem ohne zentrale Kontrolle begreifen. Sicher macht es keinen Sinn, alle im WWW zugreifbaren Datentypen in einer Anfragesprache zu erlauben. Anfragen dienen einem Zweck, besonders wenn sie in Anwendungsprogrammen vorgefertigt sind. Nur Teilmengen der vorhandenen Typen machen zusammen Sinn. Die Herausforderung ist, diese Teilmengen elegant zu beschreiben und einem Anfrager zugreifbar zu machen. Man braucht anwendungsspezifische Typsysteme, die ad hoc aus den vorhandenen Typbeschreibungen zusammengestellt werden können. Unternehmensintern wird dies

durch zentral verwaltete Data Dictionaries erreicht. Kann man diese für übernehmensübergreifende Anfragen verallgemeinern?

#### 4. Transaktionsverwaltung und Integritätswahrung

HTTP kennt keinen Zustand und keinen Begriff für Transaktionen. Ein Informationsserver, der nach diesem Protokoll arbeitet, bietet also keine Vorkehrungen für Wiederherstellung oder Zurücksetzen von Operationen. Dies reicht allemal für den Einsatz als 'Sender', der Dokumente aller Art einer Vielzahl von Lesern bereitstellt. Die Kopplung mehrerer Prozesse wie die Bestellung einer Ware aus einem elektronischen Geschäft und die Bezahlung über einen Informationsserver einer Kreditkartengesellschaft, ist damit nicht zufriedenstellend möglich. Wie kann man einen Fehler in einem Informationsserver den anderen an einer Transaktion beteiligten Informationsservern mitteilen, um gegebenenfalls einen Abbruch herbeizuführen? Wie kann man überhaupt Geschäftsprozesse im Internet entwerfen, wenn über die Qualität der beteiligten Server wenig bekannt ist?

Integritätswahrung im WWW ist ebenso offen, solange es keinen Transaktionsbegriff gibt. Eine einfache Übertragung bekannter Techniken greift zu kurz, da diese nicht auf die unkontrollierte Evolution der Daten, ihrer Typen und der angebotenen Dienste vorbereitet sind. Sie müssen im Extremfall für jede Transaktion neu bestimmt werden. Kann man Informationsserver dieser Dynamik anpassen?

Ein spezielles Integritätsproblem stellen die replizierte Daten (Kopien) und materialisierte Sichten (abgespeicherte Antworten auf Anfragen) dar. Sie können nämlich veralten und zu falschen Ergebnissen führen. Der durch Suchmaschinen aufgebaute Index ist ein gutes Beispiel: eine Suchmaschine braucht etwa 6 Monate für das Indizieren aller zugreifbaren Dokumente. In dieser Zeit veraltet notwendig ein Teil der gesammelten Information. Der Index ist also immer zu einem gewissen Prozentsatz falsch und liefert daher auch falsche Antworten bei Suchanfragen. Das Kopieren von Daten im WWW ist äußerst verbreitet, um mangelhafte Netzkapazitäten zu kompensieren. Zwar bietet HTTP die Möglichkeit, ein Verfallsdatum für ein Datum zu definieren, aber dies ist kein korrektes Kriterium. Duplizierte Information ist im Datenbankentwurf unerwünscht. Bei verteilten Datenbanken wird sie erlaubt, aber es können genaue Angaben über die Konsistenz der Kopien und der durch sie vermittelten Sicht auf die Gesamtdatenbank gemacht werden. Kann man diese Replikationsprotokolle gewinnbringend übertragen?

#### 5. Zusammenfassung

Durch die Einfachheit des Protokolls ist die im Internet verfügbare Informationsmenge explodiert. Jetzt stehen wir vor der Aufgabe, dieses Chaos zu organisieren, damit intelligente Anwendungen realisiert werden können. Dazu sind Methoden aus dem Entwurf von Informationssystemen vielversprechender als solche, die Information immer nur in der Einheit einer Datei sieht. Anwendungsprogrammierer müssen wissen, welche Art von Information in einer Datei ist.