# Vom Weblog lernen... Community, Peer-to-Peer und Eigenständigkeit als ein Modell für zukünftige Wissenssammlungen*

Jörg Kantel
Der Schockwellenreiter
www.schockwellenreiter.de

20. Mai 2003

**Zusammenfassung**

Das Internet erscheint vielen als unübersichtlicher, anarchischer Raum, in dem zwar alles zu finden ist, aber nicht das, wonach man sucht (*Lost in Cyberspace*). Die bisherigen Lösungsansätze bestanden in der Regel darin, daß man versuchte, die Informationen zentral zu sammeln, zu bündeln und sie „geordnet“ wieder zur Verfügung zu stellen. Demgegenüber sind in den letzten Jahre mit den *Weblogs* und um die Weblogs herum Strategien entstanden, wie verteilte Informationen behandelt und dennoch zugänglich gemacht werden können. Dieser Artikel zeigt auf, was für verteilte Informationssammlungen spricht, wie Weblogs mit über das gesamte Netz verstreuten Informationen umgehen, um dann zu untersuchen, wie die dabei entstandenen Techniken auch auf andere Wissenssammlungen im Internet angewandt werden können. Beispielhaft wird das an der Implementierung einer verteilten Musiknoten-Sammlung aufgezeigt.

*Keynote speach given at BlogTalk - A European Conference On Weblogs: Web-based publishing, communication and collaboration tools for professional and private use, Vienna, May 23 - 24, 2003

# 1 Motivation

## 1.1 Weblogs und persönliche Wissenssammlungen

### 1.1.1 Was sind Weblogs

Auch wenn manchmal etwas anderes behauptet wird, werden Weblogs als Medienereignis frühestens seit 1999 (in Deutschland nicht vor 2000) von der Öffentlichkeit wahrgenommen[1]. Im Gegensatz zu der schon vor 1999 existierenden Tagebuchszene (die ihre Webseiten noch liebvoll mit „handgestricktem“ HTML pflegte), sind Weblogs ohne die dazugehörende Software, wie z.B. Blogger (`www.blogger.com`), Radio UserLand (`radio.uslerland.com`) oder Movable Type (`www.movabletype.org`) nicht zu denken. Erst diese Software ermöglichte es, ständig aktualisierte Webseiten zu erstellen und zu pflegen, ohne allzutief in die Abgründe der HTML-Codierung eintauchen zu müssen. Der Weblogger (kurz: Blogger) konnte sich so ganz dem Inhalt (neudeutsch auch *content* genannt) seiner Seiten widmen.

Und genau das sind Weblogs[2]: Ständig aktualisierte, persönliche Webseiten mit kurzen, oder auch längeren Beiträgen, die umgekehrt zeitlich geordnet sind, d.h. der jüngste Beitrag steht immer ganz oben an erster Stelle. Ältere Beiträge rutschen immer weiter nach unten, bis sie irgendwann endgültig im Archiv verschwunden sind.

Der Fama nach sind Weblogs aus dem Brauch entstanden, sich Hinweise auf interessante Webseiten per Email zuzuschicken[3] und was lag dann näher, statt der Email die persönliche Homepage für diese **häufig, bis zu mehrmals täglich aktualisierten, kommentierten Linklisten** (so ein weiterer Definitionsversuch) zu benutzen.

Über die Bedeutung von **Links** in einem Weblog habe ich schon an anderer Stelle geschrieben[4], daher hier nur noch einmal so viel:

- Weblogbeiträge verweisen in der Tat häufig auf eine oder mehrere externe Quellen (*verlinken* auf diese Quelle), mehr noch: Der Hinweis auf diese Seite(n) ist oft der (einzige) Grund für diesen Weblogbeitrag.

[1] Der Streit darüber, wer das erste Weblog führte und ob nicht etwa schon das Tage- oder Logbuch von XX ein Weblog gewesen sei, erinnert mich immer ein wenig an den in den 70er Jahren erbittert durchgeführten Versuch, Science-Fiction-Literatur für die Hochkultur zu legitimieren, indem man nachträglich die Utopien von Platon, Morus, Campanella und Swift zur Science-Fiction-Literatur erklärte. Und so bringt es auch nichts, Slashdot (`www.slashdot.org`) nachträglich zu einem Weblog zu ernennen.

[2] Auch wenn natürlich niemand die Definitionshoheit über den Begriff *Weblog* besitzt.

[3] Allerdings habe ich bis heute noch keinen Beleg für diese vielkolportierte Behauptung gefunden.

[4] Vgl. [Kantel 2002]

- Die Kommentare zu diesem Links sind persönlich und oft sehr sarkastisch. Ausgewogenheit und *political correctness* darf man von einem Blogger nicht erwarten[5].

Die Weblogsoftware – in der Regel sind das kleine Content-Management-Systeme (CMS) – erlaubt ein schnelles und einfaches Publizieren der Weblogbeiträge. Dabei kann man drei verschiedene Typen von Weblog-Software unterscheiden[6]:

- Weblogsoftware, die auf einem zentralen Server läuft und dessen Server die Weblogs hostet. (Blogger, Manila, Antville.)
- Weblogsoftware, die als CGI-Script auf dem – meist bei einem Internet Service Provider (ISP) gemieteten – Webspace des Bloggers läuft[7]. (Movable Type, pMachine, Sunlog, Blosxom und viele andere, meist PHP-basierte Weblog-Tools.)
- Weblogsoftware, die auf dem Rechner des Benutzers installiert ist und statische Seiten erzeugt, die via FTP auf dem – ebenfalls meist angemieteten – Webspace bei einem ISP heruntergeladen werden. Diese Variante hat den Vorteil, daß beim ISP keine weitere Software installiert sein muß. (Radio UserLand)

Auf Basis dieser Grundfunktionalität (Beitrag schreiben und Publizieren) entstanden im Laufe der letzten zwei bis drei Jahre zahlreiche zusätzliche Funktionalitäten und Techniken, die das Publizieren erleichterten und erweiterten, ohne die grundsätzliche Einfachheit des Publizierens eines Weblogs aufzugeben.

Und genau darum geht es hier in diesem Beitrag: Ich möchte untersuchen, ob und wieweit diese Grund- und Zusatzfunktionen auch genutzt werden können, um Wissenssammlungen, insbesondere verteilte Wissenssammlungen, ebenso einfach wie Weblogs zu erstellen und zu pflegen.

### 1.1.2 Persönliche Wissenssammlungen

Weblog-Software wurde sehr schnell zu mehr genutzt, als nur kommentierte Linksammlungen anzulegen. Da die meisten der Weblog-Tools auch die

[5]Dies gilt natürlich in besonderem Maße für mein eigenes Weblog, den Schockwellenreiter.

[6]Vgl. [Doctorow et al. 2002]

[7]Dies setzt Zugriff auf das `cgi-bin`-Verzeichnis des Servers und die Installation mindestens einer Scriptsprache (Perl, PHP, Python) dort voraus, ein Service, den nicht jeder ISP bietet und der in der Regel auch mehr Geld kostet.

Möglichkeit boten, neben den täglichen Nachrichten auch sogenannte „Stories“, das sind längere Beiträge, die unter einer eigenen URL abgelegt werden können, anzulegen, entstanden sehr schnell auch „Features“, längere Beiträge zu einem Thema.

Desweiteren gibt es in den meisten Weblog-Tools die Möglichkeit, Beiträge „Kategorien“ (Schlag- und/oder Stichworte) zuzuordnen und sie unter diesen Kategorien abzulegen.

Außerdem bot zumindest zeitweise mindestens ein Weblog-Tool (Radio-UserLand zusammen mit Manila) die Möglichkeit, sich Directories (das sind Linksammlungen im Yahoo-Stil) anzulegen.

So wuchsen viele der ursprünglich einfachen Weblogs sehr schnell zu kleinen, oft liebevoll gepflegten persönlichen Wissenssammlungen zusammen.

### 1.1.3 Weblogs und Wikis

Wikis[8] liegen mir ihrem Verlinkungsmechanismus noch am nächsten zur ursprünglichen Hypertext-Idee. Obgleich Wikis eigentlich für alle offene Webseiten sind, die jeder editieren und so Texte ändern und/oder hinzufügen kann, sind in der Realität viele Wikis eher persönliche Informations- und Wissenssammlungen, die von einer Person oder wenigen Personen gepflegt werden – *Personal Information Management Software (PIM).* (Was ich für eine durchaus legitime Anwendung von Wikis halte.)

Wikis stehen nicht in Konkurrenz zu Weblogs, können aber in vielen Fällen eine sinnvolle Ergänzung zu einem Weblog als „Persönliche Wissenssammlung“ sein. Wie oben schon angedeutet, gehören auch längere Beiträge in eine Wissenssammlung. Frontier (`frontier.userland.com`), die Software, auf der Manila basiert, bot schon sehr früh mit dem automatischen Anlegen von *Shortcuts* für Beiträge Wiki-ähnliche Möglichkeiten der Verlinkung, noch weiter in die Richtung „längere Beiträge mit einem Weblog kombinieren“ ging dann Radio Userland[9] mit dem „eingebauten“ CMS. Außerdem gibt es Wiki-Systeme, die es auch erlaubten, ein Weblog zu führen und dafür auch genutzt werden, das in REBOL (`www.rebol.org`) von Christian Langreiter (`www.langreiter.com`, meines Wissens das erste deutschsprachige Weblog) programmierte Vanilla (`www.langreiter.com/space/Vanilla`) und den in Java programmierten Vanilla-Klon Snip-Snap (`http://snipsnap.org/`). Manchmal wird aber auch einfach ein Wiki parallel zu einem Weblog betrieben.

[8] Den ultimativen Text zu Wikis liefert [Mattison 2003]

[9] Zu Radio UserLand und die Möglichkeiten, die diese Software über das Führen eines Weblogs hinaus bietet, vgl. [Kantel 2002a].

### 1.1.4 Dokumentenmanagement-Systeme

Eine weitere Möglicxhkeit, verteilte Wissenssammlungen anzulegen, bieten Dokumentenmanagement-Systeme, wie zum Beispiel BSCW (*Basic Support for Cooperative Work*)[10], das am Institut für Angewandte Informationstechnik der Gesellschaft für Mathematik und Datenverarbeitung (GMD, jetzt Fraunhofer Gesellschaft, daher FhG FIT) entwickelt wurde und nun von OribTeam-Software (`www.orbiteam.de`), einem *spin off* der FhG FIT weiterentwickelt und vertrieben wird.

Allerdings legen Dokumentenmangement-Systeme ihren Schwerpunkt auf die gemeinsame Arbeit und Kooperation an Dokumenten, weniger auf die **Präsentation** von Dokumenten und/oder Ergebnissen. Mir geht es aber um Wissenssammlungen und die Präsentation von Wissen, von Ergebnissen wissenschaftlicher Arbeit, daher werden Dokumentenmanagement-Systeme in diesem Aufsatz nicht weiter berücksichtigt[11].

## 1.2 (Wissenschaftliche) Sammlungen und Publikationen im Internet

Es folgen einige Beispiele von dem, was ich unter „Wissenssammlungen" verstehe. Die Beispiele wurden auch unter dem Gesichtspunkt ausgewählt, wie und ob sie sich für eine verteilte Wissenssammlung eignen.

Das viele der Beispiele aus dem Umfeld des Max-Planck-Instituts für Wissenschaftsgeschichte (MPIWG) (`www.mpiwg-berlin.mpg.de`) kommen, ist sicher kein Zufall. Zum einen hängt es mit meiner beruflichen Situation zusammen, zum anderen liegt es aber auch sicher daran, daß das MPIWG bei der Digitalisierung und der Publikation historischer Dokumente und Objekte im World Wide Web in vielen Fällen eine Vorreiterrolle einnimmt.

Ich behandle in diesem Papier nur einen von vielen möglichen Typen von Wissenssammlungen. Es geht in der Hauptsache um das Sammeln von Artefakten, seien es Schriften, Bilder, Noten, Filme etc., nicht um das Sammeln von Informationen in lexikon-ähnlichen Strukturen (wie z.B. der Wikipedia - `www.wikipedia.org`).

[10] Eine Übersicht über die Funktionen von BSCW bietet [FIT und OrbiTeam 2001], einen auf 10 MB Datenvolumen begrenzten Probeaccount kann man bei der GMD unter `bscw.gmd.de/pub/bscw.cgi` nutzen.

[11] Ich habe allerdings intensiv mit BSCW „gespielt" und Überlegungen und Ergebnisse, die aus dem Experimentieren mit und dem Ausprobieren von BSCW entstanden sind, beeinflußten sehr wohl diese Arbeit.

### 1.2.1 Das Projekt Gutenberg

Das Projekt Gutenberg-DE (`gutenberg.spiegel.de`) ist eines der ambitioniertesten Digitalisierungsprojekte im deutschsprachigen Internet. Es enthält Texte in deutscher Sprache von mehr als 400 Klassikern und über achtzig heutigen Autoren. Das Archiv enthält zur Zeit ungefähr 50.000 HTML-Dateien (250.000 Textseiten), 850 MB an Text- und Bildmaterial, 10.800 Gedichte, 1.300 Märchen, 1.000 Fabeln, 3.000 Sagen, 1.200 vollständige Romane, Erzählungen, Novellen und 7.300 Bilder. Es wurde 1994 als Freizeitprojekt begonnen, als es nur wenige deutschsprachige Texte im Internet gab. Bis heute sind mehrere zehntausend Arbeitsstunden für das Projekt vom Gutenberg-Team aufgebracht worden. Außerdem haben viele weitere Freiwillige Text- und Bildmaterial für dieses Archiv beigesteuert. Es ist heute einer der ersten Adressen, wenn es um die Quellen deutschsprachiger Literatur im Internet geht. Das belegen die ca. 1,5 Millionen Pageviews und ca. 4,5 Millionen abgerufenen Dokumente im Monat. Laut Eigenaussage betrug das abgerufene Datenvolumen seit 1977 mehr als ein Terrabyte.

Und hier liegt eines der Probleme: Im Jahre 2001 wurde das Bereitstellen des Servers für das Projekt dem bisherigen Sponsor, der AOL, zu teuer und es wurde kurzerhand abgeschaltet. Zwar fand sich nach kurzer Zeit der Spiegel-Verlag bereit, daß Projekt weiterzuführen, doch unter der URL des Spiegel-Verlags. Das kam einer mittleren Katastrophe gleich. Zehntausende von Links, die auf die Seiten des Projekts Gutenberg verwiesen, zeigten auf einmal ins Leere.

Diese *Katastrophe* war Auslöser für die hier angestellten Überlegungen.

### 1.2.2 Perseus-Projekt

Während das Gutenberg-Projekt im Prinzip nur mit einer Art von Daten abreitet, nämlich mit eingescannten und via Optical Character Recogignaten (OCR) maschinenlesbar gemachten Daten, ist das Perseus-Projekt (URL?) bedeutend umfangreicher. Es hat sich nicht mehr und nicht weniger vorgenommen, als die gesamte Kultur des antiken Griechenlands online zur Verfügung zu stellen. Das Perseus-Projekt bietet Texte der griechischen Klassiker, Abbildungen von antiken Kunstwerken, Landkarten etc. Dafür hat es in seiner langen Geschichte schon einmal das Medium wechseln müssen. Damals als gigantischer HyperCard-Stack[12] angefangen, wechselte es mit dem Aufkommen des WWW dorthin und wurde unter einem gewaltigen Aufwand

[12]Es wäre interessant, einmal zu erforschen, wie viele der heute noch existierenden Web-Projekte ihren Urspruch in HyperCard-Stacks hatten. Ich glaube, daß es ziemlich viele sind.

an Perl-Skripten ins Internet gestellt.

Neben den eigentlichen Materialien stellt das Perseus-Project auch Hilfsmittel zur Erschließung dieses Materials zur Verfügung: Es gibt eine große Anzahl von Wörterbüchern, Übersetzungshilfen etc., die mit Hilfe eines morpholgischen Analyzers auch online genutzt werden können.

### 1.2.3 The Virtual Laboratory of Physiology (VLP)

Das 1997 von Sven Dierig und mir konzipierte „Virtuelle Laboratorium für Physiologie“ (`vlp.mpiwg-berlin.mpg.de`) ist ein Projekt, das einen begrenzteren Radius besitzt, als die vorherigen Projekte. Es erforsch die „Wurzeln“ der Physiologie von etwa 1870 bis 1930. Ähnlich wie das Projekt Gutenberg begann es als kleines, eher in den Bereich „private Wissenssammlung“ gehörendes Projekt, wuchs aber – auch bedingt durch die von außen kommende, positive Resonanz – sehr schnell darüber hinaus. Im Gegensatz zum Projekt Gutenberg war das VLP aber von vorneherein darauf ausgelegt, mit heterogenen Medien zu arbeiten. Neben der „klassischen“ Bibliothek, die aus gescannten Images der Literatur besteht, waren von Anfang an Zeichnungen von Experimenten, Lebensläufe der Wissenschaftler, Beschreibungen der Labore etc. Bestandteil der Sammlung. Zur Zeit wird das VLP im Rahmen des von der VW-Stiftung geförderten Projektes „Experimentalisierung des Lebens“ am Berliner Max-Planck-Institut für Wissenschaftsgeschichte weitergeführt und -entwickelt.

Das VLP ist unter anderem deshalb für die hier angestellten Überlegungen interessant, weil es mindestgens zwei „Communities“ bedient. Da ist zum einen die Bibliothek des VLP, die bis heute etwa 1000 Titel (vom Artikel bis zum Buch) zur Verfügung stellt und in der Regel eher den „klassischen“ (Medizin-) Historiker interessiert. Daneben findet aber insbesondere die Instrumentensammlung das Interesse der Sammler wissenschaftlicher Instrumente. Instrumentensammler sind auch heute schon eine im Netz sehr rege und aktive Community, die über die Mailingliste *Rete* (URL?) miteienander kommuniziert.

Bei einer Realisierung der hier vorgestellten Ideen wären vom VLP also mindestens zwei „Community-Server“ (siehe Abschnitt xxx) zu bedienen.

### 1.2.4 Das eDoc-Projekt des ZIM

> **Projekt Vision:** eDoc wird die Forschungsergebnisse sichtbarer machen und die Verbreitung und das Management von wissenschaftlicher Information in digitaler Form für Wissenschaftler und

Angestellte der Max-Planck-Gesellschaft vereinfachen[13].

Der eDoc-Server[14] des *Heinz Nixdorf Zentrums für Informationsmanagement (ZIM) in der Max-Planck-Gesellschaft* ist als *institutional repository*, das die gesamten Forschungsergebnisse der Max-Planck-Gesellschaft (MPG) und ihrer Institute sammelt, archiviert und präsentiert. Es soll als Schaufenster für die wissenschaftlichen Leistungen der MPG dienen. Darüber ist eDoc als technologische Plattform konzipiert, um das Internet und seine ganzen Möglichkeiten für die wissenschaftliche Kommunikation und Information den Max-Planck-Instituten nutzbar zu machen. Es sollen Dokumente und Informationen sowohl institutsintern als auch in Forschungskooperationen gemeinsam verwaltet werden können. (Dokumenten-Server)

Die grundätzliche Organisation von Dokumenten und Material erfolgt auf eDoc in sigenannten *Collections* (Sammlungen), die von Mitarbeitern von Max-Planck-Instituten angelegt, gepflegt und verwaltet werden.

Ich halte des eDoc-Projekt für ein sehr spannendes Projekt, weil es in seiner Grundidee meiner Vision von einer verteilten Wissenssammlung am nächsten kommt. Zwar wird das eDoc-Material (noch) auf einem zentralen Server gespeichert, für das Einspeisen und Pflegen sind jedoch die Wissenschaftler an den beteiligten Instituten zuständig.

Außerdem haben die Mitarbeiter des ZIM ausführlich zu den Problemen der Metadaten von Sammlungen Überlegungen angestellt. Ich komme darauf in dem Abschnitt über Metadaten noch einmal zurück.

### 1.2.5 Cuneiform Digital Library Initiative

Das CDLI ist ein funktionierendes Beispiel, das zeigt, das digitale Wissenssammlungen tatsächliche mehr leisten können als herkömmliche Sammlungen. Keilschrifttafeln „schmoren" in der Regel in den *Asservatenkammern* der Museen, nur wenige ausgewählte „Schaustücke" werden gezeigt. Ein Buch (oder besser: eine Buchreihe) die all die tausende von Tafeln abphotographiert, wäre von kaum einer Bibliothek zu bezahlen. Hier kann das WWW tatsächlich mehr leisten, als es die konventionellen Medien können. Zwar ist der Aufwand, der betrieben wird und betrieben werden muß, immer noch beachtlich, aber, wie die mittlerweile auf mehrere zehntausend Tafeln angewachsene Sammlung der CDLI zeigt, machbar.

Daneben zeigt die Cooperation der vielen, an der CDLI beteiligten Museen und Institute, daß eine Community durchaus auch im wissenschaftlichen

[13] Präambel zu [ZIM 2003]

[14] Dieser Abschnitt folgt weitgehend der Einleitung aus [ZIM 2003].

Bereich entsteht, wenn sich alle Beteiligten davon einen Mehrwert versprechen.

### 1.2.6 European Cultural Heritage Online (ECHO)

[...]

Das ECHO-Projekt ist aus meiner Sicht vor allem deshalb ein spannendes Projekt, weil es die Notwendigkeit einer verteilten Wissenssammlung aufzeigt: Dutzende von Institutionen in Dutzenden von Europäischen Ländern mit unterschiedlichen Vorstellungen und Zielsetzungen, mit unterschiedlichen kulturellen Hintergründen sind *unter einen Hut* zu bringen.

Das kann nur gelingen, wenn man allen Beteiligten die größtmögliche Eigenständigkeit läßt, sowohl in der Gestaltung als auch in der Auswahl der zu präsentierenden Wissensobjekte und nur versucht, eine Infrastruktur bereitzustellen, die es den beteiligten Wissenschaftlern und Institutionen ermöglicht, mit einfachen Mitteln, ohne großen zusätzlichen Aufwand (auch ohne großen zusätzlichen Lernwaufwand) an diesem Projekt teilzunehmen.

## 1.3 Das Client-Server-Dilemma

### 1.3.1 Das Manila-Syndrom

Manila (`manila.userland.com`) ist eine Weblog-Server-Software, die auf den Application-Server Frontier (`frontier.userland.com`) aufsetzt. Beides, Manila wie auch Frontier stammen aus dem Hause Userland (`www.userland.com`). Um seine Software zu promoten, unterhielt Userland einen kostenlosen Hosting-Service für Manila-Weblogs auf einem (oder sogar mehreren) seiner Server. Als der Traffic so überhand nahm, daß ein vernünftiges Bloggen nicht mehr möglich war, veranstalte Dave Winer (CEO von UserLand) mal wieder einen seiner berüchtigten *Cornerturns* und erklärte das zentrale Hosten von Weblogs für erledigt. Sein neues Produkt hieß Radio UserLand (`radio.userland.com`), war „Weblogging on your desktop“ und erlaubte es, statische Seiten auf einem Server der Wahl abzulegen. Nur die notwendigsten Funktionen wurden noch zentral verwaltet, die da wären

- der „Ping“ an Weblogs.com
- der „Radio Community Server“ (RCS)
- Syndication mittels RSS

### 1.3.2 Antville

Zur Zeit leidet ein weiterer kostenloser Hoster, das österreichische, auf den *Helma Application Server* basierende Antville (`www.antville.org`) unter dem *Manila-Syndrom*. Dies wurde kurzfristig mittels einer vielbeachteten Spendenaktion behoben, bei der die User knappe Euro 3.500,– aufbrachten, um einen neuen, leistungsfähigeren Server zu finanzieren. Langfristig ist dies m.E. jedoch keine Lösung – jede Client-Server-Architektur wird über kurz oder lang unter dem ständig wachsenden Traffic zusammenbrechen.

Bei Antville kommt m.E. verschärfend hinzu, daß – soweit ich informiert bin – Java-Servlets die Grundlage der Applikation bilden. Und Servlet-basierte Architekturen litten schon immer unter einem massiven Ressourcenhunger.

Beide Beispiele zeigen, daß die klassische Client-Server-Architektur aus der Welt der begrenzten, lokalen und überschaubaren Netzwerke sehr schnell ihre Grenzen erreicht, wenn sie diese lokale Umgebung verläßt und auf die Gegebenheiten des World-Wide-Web übertragen wird.

### 1.3.3 One Point of Failure

Ein von henso (`www.henso.com`) geäußertes Argument (URL?), daß ja auch große Hoster alles auf einen Server setzen und dies dann doch funktioniert, zieht meines Erachtens nicht. Das Beispiel der beiden großen deutschen Hoster zeigt dies: Während Strato (`www.strato.de`) alles auf eine Karte, sprich einen Server setzt, setzt Puretec (1&1, `www.1und1.de`) bewußt eine Serverfarm von mindestens 1.200 Linux-Servern ein (Zahlen überprüfen und den entsprechenden Heise-Artikel zitieren). Und auch Strato verfügt meines Wissens über mehrere Sun-Enterprise-Maschinen, die so redundant ausgelegt sind, daß so ziemlich jedes Teil bei laufendem Betrieb ausgewechselt werden kann, ohne daß irgendein Verlust auftritt. Trotzdem fallen die Ausfälle bei Strato häufiger auf, da bei einem tatsächlichen Serverstillstand alle Kunden betroffen sind. Fällt dagegen bei 1&1 ein Server aus, sind nur etwa 1/1200 Teil der Kunden betroffen.

## 1.4 Konformität vs. Individualität

Es gibt etliche Literatursammlungen im Internet, viele werden von großen Institutionen getragen. Andere entstehen aber auch aus privater Initiative, wie z.B. die großartige Quellensammlung zur Biologiegeschichte von Kurt Stüber (`www.mpiz-koeln.mpg.de/~stueber/stueber_library.html`).

Viele dieser Sammlungen haben ein liebevolles und individuell gestaltetes

Design, daß von den Betreibern sicher nur ungern zu Gunsten einer zentralen, konformen Sammlung aufgegeben wird. Paradebeispiel hierfür sind für mich die überall im Internet gepflegten Sammlungen mit den Texten der Songs der beliebtesten Pop-Artisten (übrigens auch ein gutes Beispiel für verteilte Sammlungen – ich komme später noch einmal darauf zurück).

Daher scheitern viele, auch ambitionierte Web-Projekte oft daran, daß die Macher ihrer bestehenden Projekte aufgeben und sie zentral auf einem Server neu installieren müßten. Das bedeutet, alle bisherigen Projekte werden vereinheitlicht, bekommen ein einheitliches Layout und die Individualität der verschiedenen Macher verschwindet darin. Neben Fragen des persönlichen Geschmacks spielen hier z.B. auch Fragen der *Corporate Identity* der Organisation (auch Wissenschaftsorganisationen haben eine oder sollten eine haben) oder des Zuwendungsgebers eine Rolle und können im Zweifelsfalle solch ein Gemeinschaftsprojekt scheitern lassen.

Auch hier wäre es sinnvoller, Bedingungen zu entwerfen, unter denen die Projekte wie gewohnt weiterlaufen könnten und trotzdem ein gemeinsames Projekt dabei herauskommt. Oder, wie Jürgen Renn schrieb:

> **Information will become a patchwork-like structure.** The forms of scientific representation will radically change as a consequnece of the information revolution. There is no reason why future scientific representations should take the forms of journals or books, forms that are largely determined by the print medium and the respective agents. Instead they may take any form suitable as a contribution to a global scientific information network and its structure.[15]

Und konsequent weitergedacht, besteht auch kein Grund mehr, daß Informationen und (digitalisierte) Materialen zentral verwaltet und gelagert werden müssen.

## 1.5 Die „Agora-Lösung“

Das alles verlangt nach einer Lösung, die die Interessen der einzelnen Informationslieferanten ebenso berücksicht wie das übergeordnete Interesse an einem für alle zugänglichen und durchsuchbaren Zugang zu den im Netz vorhandenen Wissenssammlungen und Informationen. Diese Lösung bezeichnete Jürgen Renn in Anlehnung an den „Marktplatz der Meinungen“ in der Polis

[15] [Renn 2000], Seite 3

des antiken Griechenlands, die „Agora-Lösung“[16]:

> Imagine that every scholarly project in Europe, every archive, museum, or library could join a network of digital libraries by makeing resources available on the Web with only a minimum effort, using a set of standard formats and the corresponding tools allowing their implementation. Imagine a universal electronic working environment for these resources and imagine that it would provide you, first of all, with an overview of what is presently available within the distributed network...

Die Ironie ist, daß einige der von Jürgen Renn im Abschnitt *The Implementation of the Agora Solution*[17] angemahnten Tools zu einem großen Teil bereits existieren, ebenso wie die sozialen Erfahrungen in der Arbeit mit diesen Tools. Nur daß sie in einem Bereich des Internets entwickelt wurden und angewandt werden, die dem Wissenschaftsbetrieb (noch) fremd sind. Die meisten Geisteswissenschaftler sind eben noch keine Weblogger.

# 2 Die Werkzeuge der Weblogger

Weblogger neigen dazu, eine Community, einen sozialen Organismus[18] zu bilden. Warum das so ist, ist bisher noch nicht untersucht werden. Ohne Zweifel tragen aber die oben erwähnten Werkzeuge dazu bei. Über Weblogs.com und den Community-Server werden die einzelnen Blogger informiert, welche Blogs neuen Content ins Netz gestellt haben und welche Blogs zu *ihrer Community* gehören, mittels *Syndication* werden die Updates der abonnierten Blogs frei Haus geliefert.

## 2.1 CMS und Template-Mechanismen

Template-Mechanismen stammen aus den Vorbildern der Weblogtools, den „großen“ *Content Management Systemen (CMS)*. Sie ermöglichen im Wesentlichen die logische Trennung von Bearbeitungs- und Präsentationsschicht, von Inhalt (Content) und Design.

[16] Jürgen Renn: *ECHO – An Infrastructure to Bring European Cultural Heritage Online.* A keynote speach given at the Workshop *Humanities, Research, and Cultural Heritage in Europe*, Brussels, October 24, 2001, in [Renn 2002], p. 63 - 69

[17] ebda., p. 68-69

[18] Vgl. [Kantel 2002]

## 2.2 XML

Weblog-Tools sind *Content Management Systeme* und wie fast alle CMS basieren sie auf XML. Das klingt trivial, daraus folgt aber eine wichtige Konsequenz: Zumindest potentiell liegen alle Daten als reine Nutzdaten vor, befreit vom Ballast der Layout-Informationen[19].

Nur dadurch ist es möglich, daß die Inhalte von anderen Clienten plattform- und sprachenunabhängig „abgegriffen“ und in anderen Applikationen weiterverarbeitet werden können.

Das bekannteste Beispiel der Wiederverwendung dieser XML-Daten ist RSS (siehe dort).

XML wurde ursprünglich als „abgespeckte“ Version für das von vielen als zu unhandlich und zu kompliziert empfundene SGML für das Web entwickelt. Leider droht XML an seinem Erfolg zu ersticken. Mit wachsender Popularität entdeckten immer mehr das Format und begannen, ihre eigenen Erweiterungen zu entwickeln. Mit dem Effekt, daß XML heute von vielen als noch komplizierter empfunden wird als SGML. Daher ist bei XML-basierten Entwicklungen darauf zu achten, daß der Einfachheit den Vorzug gegeben wird. Die meisten Weblogtools halten sich daran. Soweit ich weiß, gibt es für kaum eine RSS-Version eine DTD, sondern nur verbale Beschreibungen. Weblog-Tools nutzen XML-RPC (URL?) (siehe nächsten Abschnitt) statt dem bedeutend komplizierteren SOAP (URL?).

### 2.2.1 XML-RPC

Die meisten der Bloggertools kommunizieren über XML-RPC miteinander. XML-RPC steht für *Remote Procedure Call auf Basis XML* und ist ein einfaches, aber trotzdem effektives Protokoll, um Prozeduren über Betriebssystem- und Sprachgrenzen hinweg aufzurufen.

[=> Weiter ausführen!]

[19] Einige Weblog-Tools benutzen ein anderes – aber ebenfalls strukturiertes Format – zur Speicherung der einzelnen Artikel. Jedoch ist eine Wandlung nach XML aus diesen Formaten in der Regel trivial.

### 2.2.2 Weblog API

### 2.2.3 Meta Weblog API

### 2.2.4 OPML

### 2.2.5 Und was ist mit SOAP?

## 2.3 Der „Ping“

Der *Ping* an Weblogs.com ist der zentrale Punkt des oben angesprochenen *cornerturns* von Dave Winer. Schon vorher hatten sich die Weblogs, die das wollten (und das waren fast alle) bei Weblogs.com eingetragen und Weblogs.com selber fragte im Stundenrythmus die eingetragenen Weblogs ab, ob es ein *Update* gibt. Schon nach wenigen Monaten hatten sich soviele Weblogs eingetragen, daß der Server mit der stündlichen Nachfrage nicht mehr mitkam, die Abfrage war noch nicht beendet, aber die Stunde schon vorbei und der Server mußte mit einer neuen Abfrage beginnen.

Die Lösung war, daß nicht mehr der Server die Clients – also die Weblogs – abfragt, sondern die Clients dem Server mitteilen, wann es bei ihnen etwas Neues gibt. Der *Ping* war geboren. Er ist eine einfache XML-RPC-Prozedur, die dem Server nur mitteilt, daß es bei diesem Weblog etwas Neues gibt.

Die Umsetzung innerhalb der Community erfolgte erstaunlich schnell. Obwohl UserLand selber nur seine eigene, proprietäre Scriptsprache *UserTalk* unterstützte, waren schon nach wenigen Tagen XML-RPC-Pings für alle gängigen und weniger gängigen Programmier- und Scriptsprachen vorhanden. Es gibt heute meines Wissens kein Weblog-Tool, das den Ping an Weblogs.com *nicht* unterstützt.

### 2.3.1 Andere „Pings“

Auch von anderen wurde das Prinzip des „Anpingens“ übernommen. Während blo.gs (URL?) eine schlichte Kopie des Weblos.com-Pings ist, haben andere tools das Ping-Prinzip verfeinert:

### 2.3.2 Trackback

### 2.3.3 Blogrolling

### 2.3.4 Erweiterungen der bestehenden Werkzeuge.

Der Ping an Weblogs.com war eine geniale Idee. Immer mehr Werkzeuge entstehen, die auf diesem Ping aufbauen. So z.B. Blogdex.com mit dem Versuch,

ein erweitertes *Ranking* einzuführen und Technorati.com, die versuchen, so etwas wie einen *Blogger-Cosmos* aufzuzeigen. (URLs?)

Eine dritte Entwicklung, die aber direkt nicht von Weblogs.com beeinflußt wurde, trotzdem für meine Argumentation wichtig ist, nämlich die *Community- oder Kollektiv-Blogs* werden in einem späteren Abschnitt behandelt.

## 2.4 Communities und der „Community Server“

Der „Community-Server“ hat eine ähnliche Funktion wie Weblogs.com. Nur daß er nicht die Updates *aller* Weblogs anzeigt, sondern nur die Updates aller Weblogs, die zu einer *Community* gehören. Grob läßt sich der Begriff *Community* erst einmal damit übersetzen, daß zu dieser Community alle Weblogs gehören, die vor dem *cornerturn* auf einem Server zentral untergebracht waren. Das dies die Community-Bildung fördert, sieht man heute noch an den Antville-Blogs auf `www.antville.org` und `www.twoday.net`. [=> Erläutern!]

[Exkurs: Community/Communities auf den Antville-Server(n)]

Erst einmal hat dieser Server tatsächlich nur die Funktionen, die eine klassische serverbasierte Weblog-Software anbietet: Rankings, Last Update, Most Read Sites etc. Doch zwei weitere Entwicklungen beginnen sich abzuzeichnen.

[weiter?]

Hier ist/sind die derzeitigen Implementierungen noch inkonsequent. Statt daß der *Ping* an den Community-Server geht, der ihn dann an Weblogs.com durchreicht, geht ein Ping an Weblogs.com und über einen zweiten XML-RPC-Mechanismus wird der Community-Server über ein Update informiert. Hier wäre folgende Hierarchie anzustreben, die auch wieder Weblogs.com entlasten würde:

[Zeichnung]

### 2.4.1 Communities, die durch gemeinsame Interessen gebildet wurden.

Pionier im deutschsprachigen Raum war in diesem Falle sicher das *Bloghaus* (`www.blogworld.de`), das sich als *Gemeinschaft der deutschsprachigen Weblogs* verstand. Während das einzige gemeinsame Interesse bisheriger Communities in der Nutzung der gleichen Software bestand, entstand hier etwas völlig Neues, eine Community, die sich – emanzipiert von der Software – durch andere, gemeinsame Interessen definiert.

Interessant dabei ist, daß diese Communities im Prinzip keine neuen Tools benötigen. Der Weblog-Monitor des Bloghauses bedient sich bei Web-

logs.com, nur daß er eben nur die Updates der im Bloghaus angemeldeten Weblogs zeigt.

### 2.4.2 Community-Blogs

Gibt es überhaupt ein Potential unter den Netzbenutzern, die bereit sind, an gemeinsamen Projekten wie einer gemeinsamen, verteilten Wissenssammlung mitzuwirken? Auch hier gibt die Weblogwelt eine Antwort:

[Slashdot, Metafilter, Rollberg, Kollektivblogs]

## 2.5 Syndication mittels RSS

### 2.5.1 Push vs. Pull

Bei dem oben erwähnte Ping an Weblogs.com erfährt der Benutzer – wenn er sich auf die Seite von Weblogs.com begibt – zwar, daß es auf einer Seite resp. in einem Weblog etwas Neues zu lesen gibt, aber nicht, was es Neues zu lesen gibt. Immer noch muß der Benutzer, um etwas über den Inhalt der Seite zu erfahren, diese aktiv ansurfen und sich den Inhalt auf seinen Rechner laden (*to pull*).

Ein anderer Weg, der – natürlich mit unterschiedlichen Techniken und Formaten, wie sollte es anders sein - schon vor einigen Jahren von Netscape und Microsoft beschritten wurde, ist der, daß der Anbieter selber informiert, wenn sich seine Seite geändert hat. Er stellt eine Datei ins Netz, die der Webbenutzer via http (dem Standardprotokoll der Browser) abrufen kann und die ihn informiert, ob sich die Seiten geändert haben. Als Format dafür bot sich XML an. Spezielle Webserver (Aggregatoren) sammeln diese Informationen und stellen sie dem Benutzer (konfigurierbar, d.h. aus den gesammelten Informationen stellt sich der Surfer seine Auswahl zusammen) zur Verfügung. Aus historischen und marketing-technischen Gründen fangen diese Server meist mit *my* an, wie z.B. myNetscape.com oder myUserLand.com. Der Benutzer bekommt die Daten gewissermaßen geliefert (*to push*) und muß nicht mehr selber nach den aktuellen Seiten suchen. Die einzelnen Seiten, die auf den Aggregatoren zusammengefaßt werden, nennt man *channel*.

Syndication-Formate sollten ursprünglich nur die Headlines und ggf. einen *Abstract* der Nachrichten zeigen. (Der User soll schließlich - nachdem er sich überzeugt hat, daß die Information ihn interessieren könnte, auf die eigenen Seiten gelockt werden.) Die meisten heutigen RSS-Feeds leisten mehr. So liefert z.B. Radio UserLand von Anfang an den kompletten Artikel incl. aller Bilder und Links im RSS-Feed mit[20]. Das erlaubte einmal den Bezieher des

[20]Diese Art des RSS-Feeds ist allerdings umstritten [=> erläutern]

Feeds, diesen Channel vollständig in seiner Website zu integrieren.

### 2.5.2 RSS

Damit nun die Aggregatoren die Dateien überhaupt lesen konnten, mußte man sich auf ein gemeinsames Format einigen - und dieses Format ist RSS (Rich Site Summary)[21].

Die spannendere Geschichte ist aber, daß dieses Format es dann ermöglicht, den Inhalt pur auch auf der Website des Client für weitere Informationen, Spider und ähnliches abzulegen. So kann z.B. daß oben erwähnte Blogrolling nicht nur dadurch funktionieren, daß der Client sich die Informationen über Weblogs, die ein Update erfahren haben, bei Weblogs.com oder einem anderen Server abholt, sondern auch dadurch, daß er eine Liste der „Lieblingsblogs“ mit RSS-Feed verwaltet und regelmäßig bei dort nachschaut, ob dieser Feed ein Update erfahren hat. Auch dies ist eine P2P-Technik (siehe Abschnitt Peer-to-Peer-Netzwerke weiter unten), die zentrale Server wie Weblogs.com entlastet, wenn in Zukunft nicht sogar überflüssig macht.

Es sind noch viele weitere Anwendungen von RSS denkbar (und werden zum Teil auch praktiziert). So kann man sich z.B. in seine eigene Webseite die akutellen Headlines der Lieblingsblogs in einem Fenster einblenden lassen oder man kann Seiten auf seinem Server haben, die die kompletten aktuellen Meldungen der Lieblingsblogs spiegeln.

## 2.6 (re)Structured Text

## 2.7 Metadaten

Eine Diskussion über Metadaten hat in der Weblog-Welt bisher kaum stattgefunden, obwohl sie an einigen Stellen – wenn auch zaghaft – angestoßen wurde. Dabei existieren durchaus schon rudimentäre Metadaten in den Headern der RSS-Feeds. Mit der weiteren Verbreitung und auch Spezailisierung von Weblogs ist damit zu rechnen, daß diese Diskussion bald verstärkt in Gang kommen wird.

### 2.7.1 Wozu Metadaten?

### 2.7.2 Dublin Core

> In the diverse World of the Internet, Dublin Core can be seen as a „metadata pidgin for digital tourists“: easily grasped, but not

[21] Zu Aufbau und Funktion einer RSS-Datei vgl. auch [Kantel 2001]

necessarily up to the task of expressing complex relationssships or concepts.[22] “

### 2.7.3 RDF und das „Semantische Web“

# 3 Szenario einer virtuellen, verteilten Sammlung

Betrachten wir nun einmal eine Sammlung wie das bereits oben erwähnte *Projekt Gutenberg-DE*, nur daß wir uns vorstellen, daß diese Sammlung verteilt angelegt ist. Jeder Sammler legt die Dateien auf einem Server seiner Wahl ab, zusammen mit einem Satz Metadaten. Diese Metadaten bestehen aus einem Kern, der möglichst klein und unkompliziert zu halten ist, damit die freiwilligen (sic!) Mitarbeiter nicht die Lust verlieren, diesen Metadatensatz anzulegen. Neben diesem Kern steht es jedem Sammler und jeder Gruppe natürlich frei, weitere Metadaten (in einem anderen *Namespace*) anzulegen, die sie für ihre spezielle Sammlung oder für ihr spezielles Gruppenprojekt benötigen.

Nach Fertigstellung erfolgt ein Ping an einen oder mehrere „Community-Server“. Wenn mehrere Community-Server vorhanden sind – was aus Gründen der Ausfallsicherheit zu empfehlen ist – sollten sich diese Server untereinander synchronisieren. Dies kann ebenfalls mittels eines XML-RPC-Pings geschehen. Der oder die angepingten Server holen sich vom Client den Metadatensatz und fügen ihn in ihren Datenbestand ein.

Anfragen an das Projekt gehen immer an einen der Community-Server, die mit Hilfe dieses Datenbestandes ein *Redirect* vornehmen und den Anfragenden auf den Server und die Seiten verweisen, auf denen die Informationen tatsächlich liegen.

Gleichzeitig bieten die einzelnen Clients einen aus den Metadaten und (XML-) Datensätzen automatisch generierten RSS-Feed an, den einmal die Server zum Generieren von Inhaltsverzeichnissen nutzen können, als auch von den Clients innerhalb und außerhalb der „Community“ als Update-Information bezogen werden kann.

## 3.1 Anforderungen an einen Community-Server

[...]

[22] Discussion Paper der International Working Group ECHO, European Cultural Heritage Online, in renn02, p. 59

Natürlich muß ein Community-Server nicht nur die Standardfunktionen abdecken. Es ist ebensogut möglich, daß er z.B. (XML-RPC-) Schnittstellen zu einer morphologischen Analyse mit anschließender Übersetzung anbietet, als Newsserver fungiert oder ein Forum oder eine Mailingliste betreibt. Wichtig ist nur, daß dies alles auf Basis von *Open Source* und *Open Protocol* erfolgt:

### 3.1.1 XML-RPC

Nach all dem bisher gesagten dürfte klar sein, daß ein Community-Server die Möglichkeit bieten muß, daß mit ihm via XML-RPC kommuniziert werden kann. Das heißt, ein Community-Server muß auch ein XML-RPC-Server sein. Das ist heute keine Einschränkung mehr, für die meisten gängigen Scriptsprachen existieren Bibliotheken, die einen XML-RPC-Server implementieren und nicht zuletzt „spricht“ das Open-Source Web-Application-Framework Zope (`www.zope.org`) ebenfalls XML-RPC.

### 3.1.2 Open Source und Open Protocol

Die Struktur eines Community-Servers muß allen Beteiligten klar sein und alle Beteiligten müssen die Möglichkeit haben, ebenfalls einen Community-Server hochzuziehen, sei es, um einen Spiegel anzulegen oder sei es, um einen Community-Server hochzuziehen, weil der bisherige Betreiber eines solchen Servers ihn nicht mehr betreiben will oder kann. Daraus folgt einleuchtend, daß nur ein Open-Source-Produkt als Community-Server fungieren kann.

Noch wichtiger als *Open Source* ist für einen Community-Server aber, daß alle Protokolle, die auf ihm laufen, auch *Offene Protokolle* sind. Da der begriff *Open Protocol* nicht so geläufig ist, folgt hier eine kurze Erläuterung[23]:

Protokolle sind die technische Sprache, die es verschiednenen Systemen erlauben, miteinander zu kommunizieren. Eines der bekanntesten Protokolle ist das *HyperText Transfer Protocel (http)*, das die Art und Weise beschreibt, wie ein Webbrowser mit einem Webserver kommuniziert. Genauso wie es offene und geschlossene/proprietäre Software gibt, gibt es offene und geschlossene/proprietäre Protokolle. Proprietäre Protokolle werden von Firmen definiert und dienen oftmals dzu, eine bestehende Marktmacht zu festigen oder eine Anwendung zu monopolisieren. Offene Protokolle hingegen sind Versuche, solch wesentlichen Kommunikationsmittel in einer offenen und einigermaßen demokratischen Weise zu entwickeln und zu beschreiben. Offene Protokolle stehen allen Programmierern frei zur Verfügung und erlauben es, eigene Programme auf dieser Basis zu entwickeln.

[23]Die Erläuterung folgt weitgehend [Schatten 2003], Seite 5 - 6

Aus diesen Gründen ist natürlich unabdingbar, daß ein Community-Server nur über offene Protokolle mit seinen Clients kommuniziert[24].

## 3.2 Was ist für eine Implementierung notwendig?

Wie sieht nun solch eine verteilte Wissenssammlung – in unserem Beispiel eine Notensammlung – aus? Jeder „Sammler“ legt sich seine Sammlung gescannter Noten so ab, wie er es für sinnvoll hält. Je nach Ziel der Sammlung und Vorliebe können das in HTML eingebettete JPEGs sein oder sie können als downloadbare PDFs vorliegen[25] (zu den Clients vergl. das nächste Kapitel). Im Wurzelverzeichnis der Sammlung liegt eine XML-Datei, die im einfachsten Fall wie folgt aufgebaut sein könnte.

```
<xml>
...
</xml>
```

Nach jedem Update schickt der Client – entweder automatisch ausgelöst oder von Hand gestartet – einen Ping an einen oder mehrere Community-Server. Diese Server lesen dann die im Ping angegebene XML-Datei aus und überprüfen anhand ihres Datenbestandes, ob tatsächlich ein Update vorliegt.

Wenn ein Update vorliegt, werden die Liste der *last updates* und das Inhaltsverzeichnis akutalisiert. Gleichzeitig bietet der Community-Server ein Update seines RSS-Feeds an und pingt auf ggf. vorhandene Spiegelserver, damit diese ebenfalls auf dem aktuellen Stand sind. Zusätzlich ist es noch denkbar, *Masterserver*[26] auch noch anzupingen, damit auch diese ihre Up-

[24]Der Client wiederum muß nicht unbedingt Open Source sein, auch wenn dies wünschenswert wäre. Wenn ein Mitglied der Community z.B. mit Microsoft Word und Visual Basic einen Client entwickelt, weil dies in sein Konzept und in seine Arbeitsweise paßt, so ist das ohne weiteres zu akzeptieren. Nirgendwo steht geschrieben, daß offene Protokolle nur von Open-Source-Produkten verwendet werden dürfen.

[25]Denkbar wäre natürlich auch das Bereitstellen von MIDI-Dateien oder anderen proprietären Notenformaten, dies sollte jedoch nur *zusätzlich* erfolgen, da entweder diese Formate nicht auf allen Betriebssystemen gelesen resp. weiterverarbeitet werden können oder diese Formate zusätzliche, proprietäre (und im Falle von MIDI meist auch extrem teure) Softwäre voraussetzt, um sie lesen zu können.

[26]Ein **Masterserver** ist ein Server, der mehrere *untergeordnete* Sammlungen zusammenfaßt. So ist es im Falle der Notensammlung z.B. denkbar, daß eine Community existiert, die Blues- und Ragtime Noten (URL?) sammelt, eine andere populäre Musik aus dem 19. Jahrhundert, eine dritte dann Kammermusik der Barockzeit und eine vierte vielleicht Werke der Spätromatik usw. usf.

Eine mögliche Aufteilung wäre, daß auf der ersten übergeordneten (über den eigentlichen Community-Servern) Ebene z.B. die Noten der populären Musik in einem Inhaltsverzeichnis erfaßt werden (Beispiele 1 und 2) und auf einem anderen Masterserver die Notensammlungen von Komponisten der Klassik (Beispiele 3 und 4). Ein letzter übergeordneter

datelisten und ihren RSS-Feed aktualisieren.

Der oder die Community-Server wiederum stellen ein Interface bereit, mit dem die Suche und das Finden von Content (in unserem Fall von Noten), den die angeschlossenen Community-Mitglieder bereitstellen, vereinfacht oder überhaupt erst ermöglicht wird. Er leitet Anfragen, die an ihn – z.B. nach den Noten für Scott Joplins Ragtime Waltz Bethina – gestellt werden, reihum an die Server weiter, die diese Noten bereithalten. So findet auf simple Weise ein einfaches *load balancing* statt[27]. Trotzdem sollte der Server auch auf die Seiten der anderen Anbieter hinweisen, die die gesuchten Noten ebenfalls bereithalten. Denn viele Sites werden neben den reinen Noten zusätzliche Informationen z.B. zu Komponisten, Interpreten oder Aufführungen bieten, an die der Sucher ebenfalls interessiert sein könnte. Im einfachsten Fall stellt der Community-Server das Resultat auf einer Ergebnis-Seite bereit und läßt die Treffer nach jeder Suche rotieren. Erfahrungsgemäß klickt die Masse der User immer zuerst auf das oberste Suchergebnis.

Weitere minimale Anforderungen an einen Community-Server sind – wie bereits teilweise erwähnt:

- eine Liste der an der Community beteiligten Institutionen, Personen oder Sites,
- eine Liste der zuletzt aktualisierten Sites,
- eine Liste der am häufigsten besuchten Sites (*ranking*) und
- die Bereitstellung eines RSS-Feeds.

Die Forderung nach einem *ranking* mag nicht unmittelbar einleuchten, doch zeigt die Erfahrung aus der Weblog-Welt, das dieses *ranking* ein wesentliches Element der Community-Bildung ist und den Wettbewerb und die Motivation der einzelnen Community-Mitglieder ungemein fördert.

---

Server wiederum nimmt nur von diesen Servern der Ebene 2 Pings an. Dieses Verfahren ist durchaus sinnvoll. So ist es einem Klassik-Freund nicht unbedingt zuzumuten, Informationen in *seinen* Metadaten mitzuschleppen, die nur für Jazzfreunde von Interesse sind und *vice versa*. Daher findet von Ebene zu Ebene noch einmal eine Reduzierung der Metadaten auf das absolut Notwendige statt. Diese Bündelung reduziert den Datenverkehr und konzentriert die Informationen von Ebene zu Ebene.

[27]Dieses *load balancing* geht von der Vermutung aus, daß die Schwerpunkte der Interessen der Anbieter mit den Schwerpunkten der Interessen der Nachfrager in quantitativer Hinsicht ähnlich sind. Es wird sicher mehr Interesse an Joplin auf beiden Seiten geben, als an anderen, eher unbekannten Ragtime-Komponisten. Daher wird es auch mehr *Anbieter* geben, die Joplin auf ihren Webseiten haben, so daß die Nachfragen auf mehrere Anbieter aufgeteilt werden können.

Aus den gleichen Gründen (Community-Bildung) könnte auch ein Trackback-Mechanismus für eine Wissenssammlung wie die hier skizzierte Notensammlung durchaus sinnvoll sein, doch ist solch ein Mechanismus Aufgabe des Clients und nicht des Community-Servers.

## 3.3 Der Client

Grundsätzlich kann ein Client für verteilte Wissenssammlungen – analog zu den Weblogs – seinen Anteil an der Sammlung auf zwei unterschiedliche Weisen ins Netz stellen: Entweder liefert er statische Seiten, die der Client per Script generiert und die per FTP auf den Server übertragen werden oder es werden nur die „reinen“ Bilddateien und die Metadaten auf den Server übertragen und der Server oder ein CGI-Script liefert auf Anfrage eine dynamisch generierte Seite[28]. Beide Versionen haben ihre Vor- und Nachteile:

### 3.3.1 Statische Seiten

Der größte Vorteil von statischen Seiten ist, daß sie die geringsten Anforderungen an den Webserver stellen. Nahezu jeder Webserver bei nahezu jedem Internet Soervice Provider (ISP) kann statische Seite liefern[29]. Und darüberhinaus: Das Ausliefern von statischen Webseiten ist die Hauptaufgabe eines „klassischen“ Webservers – wie z.B. Apache – und dafür ist er entwickelt worden, das kann er am schnellsten und am besten[30].

Außerdem können Wissenssammlungen, die auf statischen HTML-Seiten basieren, auf Wunsch auf einen Datenträger (z.B. CD-ROM) gepackt werden und ohne Server – nur mit Hilfe eines Webbrowser – gelesen werden. Dies ist

[28] Die dritte (gehostete) Variante aus der Weblogwelt scheint mir für verteilte Wissenssammlungen nicht unbedingt sinnvoll und wird hier daher nicht berücksichtigt.

[29] Ausnahme sind lediglich die spezialisierten ISP, die ihren Hosting-Service auf einen bestimmten Web-Application-Server (z.B. Manila oder Zope) beschränken.

[30] Das Ausführen von CGIs und die Verwaltung von Sessions beherrschen die meisten Webserver nicht so gut. Auch der Apache – als weltweit am meisten verbreitete Webserver – tut sich da ohne eincompilierte Zusatzmodule (z.B. `mod_perl`, `mod_python` oder `fast_cgi`) sehr schwer, da er bei jedem Aufruf eine Instanz des entsprechenden Interpreters starten muß und das zwingt die Performance ziemlich in die Knie. Noch schwerer fallen den klassischen Webservern die Verwaltung von Sessions. Das ist kein Fehler der Webserver, sondern liegt am *stateless mode* des zugrundeliegenden Protokolls (http). CGI und Sessions waren von den „Erfindern“ des WWW ursprünglich einfach nicht vorgesehen.

Daher basieren fast alle Web-Application-Server (wie z.B. Zope oder Frontier) auf einen eigenen Server-Mechanismus und einem eigenen Framework, die für die Generierung von dynamischen Content und die Verwaltung von Sessions besser optimiert sind, als es ein klassischer „nur“-http-Server ist.

z.B. im Hinblick auf eine geplante Buchproduktion mit beigelegter CD-ROM ein nicht unwesentlicher Vorteil.

Last but not least können statische Seiten problemlos von Suchmaschinen gefunden und indiziert werden. Jedoch ist dieses – noch vor ein bis zwei Jahren wichtige – Argument mittlerweile nahezu bedeutungslos geworden. Einerseits haben die Suchmaschinen gelernt, mit dynmischen generiertem Inhalt umzugehen, zum anderen sind fast alle heutigen auf dem Markt erhältlichen CMS und Web-Application-Server in der Lage, ihren Inhalt so zu produzieren, daß er von den Suchmaschinen berücksichtigt wird[31].

Der größte Nachteil statischer Seiten ist der Aufwand, der erforderlich ist, um diese zu pflegen. Nehmen wir nur eine Site die – was durchaus wünschenswert ist und auch den gängigen Usability-Richtlinien entspricht – ein Inhaltsverzeichnis in der linken Spalte hat. Wenn nun Inhalt hinzugefügt wird, sagen wir – um bei unserer Notensammlung zu bleiben – Scott Joplins Bethina, muß die gesamte Site neu herausgerendert werden, da sich im Zweifelsfalle auf jeder Seite das Inhaltsverzeichnis geändert hat. Allein das kann – ohne Berücksichtigung des zusätzlichen Verwaltungsaufwandes – je nach Umfang bei einer größeren Site schon einmal mehrere Stunden Rechnerzeit in Anspruch nehmen[32] [33].

**Exkurs:** Dieser Umstand ist auch in der Weblog-Welt nicht unbekannt. Der größte Kritikpunkt an MovableType (`www.movabletype.org`), einem Weblog-Tool, das versucht, das beste aus beiden Welten, den statisch generierten Websites und den dynamisch generierten Inhalten, zu liefern, ist die Wartezeit, wenn ein Neurendern (bei Movable Type *rebuild* genannt), notwendig ist. Auch Radio UserLand kennt dieses Problem, jedoch ist es hier dadurch entschärft, daß der Render-Prozess in einem eigenen Thread im Hintergrund läuft und weitergearbeitet werden kann, während Movable Type während des *rebuild* einfach blockiert ist – und das unter Umständen über Stunden.

Den anderen Nachteil einer statischen Lösung, nämlich das Fehlen von dynamisch generierten Inhalten, halte ich im Zusammenhang dieser Arbeit für eher nebensächlich. Die meisten dynamischen Inhalte (aktueller Wetter-

[31] Daß daher einige der größten Zeitungen und Zeitschriften in ihrer Online-Ausgabe immer noch Monster-URLs liefern, die von Suchmaschinen (speziell von Google) nicht indiziert werden, ist mir unverständlich. Aus technischer Sicht gibt es für solche Monster-URLs heutzutage keine Notwendigkeit mehr.

[32] Nein, Frames sind in meinen Augen *keine* Alternative zur Lösung dieses Problems.

[33] Und genau dieser Umstand war ausschlaggebend für den Wechsel des VLP von einer mit Hilfe des *static site tools* von Frontier gerenderten statischen Lösung zu einer Zope-basierten, dynamisch generierten Version.

bericht oder ähnliches) sind eher Gimmicks und für die Wissenssammlungen, um die es in diesem Artikel geht, vernachlässigbar. Und Kommentarfunktionen kann man – wenn gewünscht – als CGI nachrüsten oder man implementiert gleich eine entsprechende Foren-Software.

### 3.3.2 Dynamisch generierte Webseiten

Der größte Vorteil dynamisch generierter Webseiten kann[34] in dem Umstand liegen, daß es für den Betreiber der Wissenssammlung ausreicht, daß er seine Bilddateien und seine Metadaten einfach per FTP auf den Webserver hochlädt. Weder muß er ein Skript anstoßen, das HTML-Dateien generiert, noch muß er sich um die Verwaltung oder um Web-spzifische Eigenheiten kümmern. Um wieder auf unsere Notensammlung zurückzukommen: Im Idealfall legt er die Scans der Notenblätter zusammen mit einer XML-Datei mit den Metadaten einfach auf seinem Server ab – um die Generierung der HTML-Seiten, der Inhaltsverzeichnisse, der RDF-Dateien und des RSS-Feeds (und was es sonst noch alles gibt), kümmert sich der Server[35].

Vor- und Nachteile dieser Lösung halten sich in etwa die Waage. Das Einrichten und die Wartung eines (Zope oder Quixote oder...) Servers ist sicher erst einmal aufwendiger und bei kleinen Sammlungen unverhältnismäßig, als die Erstellung eines Skriptes zur Generierung statischer HTML-Seiten. Aber ab einer gewissen Größe der Sammlung schlägt das Pendel zurück – der (einmalige) Aufwand für den Server wird geringer als der bei jedem Update fällige Aufwand für das ständige Neugenerieren und anschließende Hochladen der statischen Seiten.

Doch egal, welcher Lösung man den Vorzug gibt, der von mir skizzierte Community-Server sollte mit beiden Varianten zurechtkommen.

# 4 Fazit

Das von mir aufgezeigte Szenario ist ein Projekt, das im Sinne einer *Informationslogistik* einen universellen Zugriff auf heterogene Wissensressourcen ermöglicht. Für den Content sind nicht die Betreiber des *Community-Servers*, sondern nach wie vor die einzelnen, teilnehmenden (Unter-) Projekte selber verantwortlich. Eine redaktionelle Vor- oder Nachbearbeitung liegt ebenfalls

[34] muß aber nicht, es gibt auch Applikation-Server, die dieses in meinen Augen für Wissenssammlungen essentielle Feature, nicht unterstützen.

[35] Noch idealer (Gibt es das? Idealer als ideal?) ist natürlich die Radio-UserLand-Lösung des *upstreaming*. Die Scans und die Metadaten kommen einfach in das *upstream*-Verzeichnis auf dem lokalen Rechner des Nutzers und um den Upload auf den Server und die anderen Dinge kümmert sich ein im Hintergrund laufender Thread.

in der Hand der einzelnen Projekte. Technische Bedingung zur Teilnahme an dieser Community ist einzig und allein die Bereitschaft, den Community-Server anzupingen, einen einheitlichen Metadatensatz und einen RSS-Feed anzubieten. Auf der anderen Seite ist es mehr als eine *Metasuchmaschine* mit einem einheitlichem Interface für Suche und Präsentation, wie z.B. das vom kunstgeschichtlichem Seminar der Berliner Humboldt-Universität initierte Projekt *Prometheus* [Bredekamp 2002].

Die Frage ist aber: Werden Wissenschaftler eine ähnliche Community bilder oder bilden wollen, wie sie in der Bloggerwelt zu beobachten ist? Ich denke, sie werden es nicht nur wollen, sondern müssen. Das Netz wächst mit rasender Geschwindigkeit, immer mehr Wissenssammlungen entstehen und werden via Internet zugänglich gemacht. Es wird sinnlos werden, einige Teilsammlungen wieder und wieder ins Netz zu stellen, nur weil sie zum eigenen Projekt gehören, wenn sie doch schon in hinreichender Qualität im Netz existieren und zugänglich sind. Kein Mittelgeber wird in Zukunft für solch ein Vorhaben Geld herausrücken. Hier ist Vernetzung und Verlinkung angesagt. Auf der anderen Seite wächst der (Kosten-) Druck von Seiten der Verlage und es droht die ständige Verschärfung der Urheberrechtsgesetze aus wirtschaftlichem Interesse, die bestimmte wissenschaftliche Arbeiten entweder erschweren oder gar unmöglich machen.

Die Alternative kann daher nur heißen, Wissenssammlungen ins Netz zu stellen und der (nicht nur wissenschaftlichen) Öffentlichkeit zugänglich zu machen und zu kooperieren, wo es nur geht. Um die dabei notwendige Koordination zu schaffen, müssen sich Communities innerhalb der wissenschaftlichen Welt bilden. Das ECHO-Projekt ist in meinem Augen ein großer Schritt in diese Richtung.

Denn nur so kann Wissenschaft der vielbeschworenen Informationsflut Herr werden und von den Fortschritten des *Informationszeitalters* profitieren. Wissenschaft, die sich dem entzieht oder Wissenschaftler, die dies ignorieren, werden sich in Zukunft zwangsläufig aus der Wissensgesellschaft verabschieden.

# Literatur

[Bredekamp 2002] Horst Bredekamp, Ingeborg Reichle: *Prometheus*. Das verteilte digitale Bildarchiv für Forschung und Lehre, in: Humboldt-Spektrum, 9. Jahrgang, Heft 4/2002, S. 48-53

[Charlier 2003] Michael Charlier: *Pingpong mit Pingback*. Lass mich Deine Suchmaschine sein: Webseiten finden neue Wege der Vernetzung, Frank-

furter Rundschau vom 04.04.2003

[chromatic, Aker und Krieger 2002] chromatic, Brian Aker & Dave Krieger: *Running Weblogs with Slash*, Sebastopol (O'Reilly) 2002

[Dierig, Kantel und Schmidgen 2000] Sven Dierig, Jörg Kantel und Henning Schmidgen: *The Virtual Laboratory for Physiology*, A Project in Digitalising the History of Experimentalisation of Nineteenth-Century Life Sciences *with an enclosed CD-ROM*, Berlin (Max Planck Institute for the History of Science, Preprint No. 140) 2000, (`www.mpiwg-berlin.mpg.de/Preprints/P140.PDF`), PDF, 6 MB

[Doctorow et al. 2002] Cory Doctorow, Rael Dornfest, J. Scott Johnson, Shelley Powers, Benjamin Trott & Mena G. Trott: *Essential Blogging*. Selecting and Using Weblog Tools, Sebastopol (O'Reilly) 2002

[FIT und OrbiTeam 2001] FIT und OrbiTeam: *BSCW - Basic Support for Cooperative Work, Version 4.0 – Handbuch*, Sankt Augustin 2001

[Graßhoff, Liess, Nickelsen 2001] Gerd Graßhoff, Hans-Christoph Liess, Karin Nickelsen: *COMPAGO – der systematische bildvergleich*. Handbuch, Bern Studies in the History and Philosophie of Sciences (Bern Studies, Educational Materials; 3), 2001

[Grötker 2003] Ralf Grötker: *Dosenware*. DRM aus Leipzig: „Copy Hall" setzt Bibliotheken unter Druck, Telepolis 2003 (`www.heise.de/tp/deutsch/special/copy/14044/1.html`), zuletzt besucht am 19. Februar 2003

[Hammersley 2003] Ben Hammersley: *Content Syndication with RSS*, Sebastopol (O'Reilly) 2003

[Ianella 1998] Renato Ianella: *An Idiot's Guide to the resource Description Framework*, 1999 `archive.dstc.edu.au/RDU/reports/RDF-Idiot/`, zuletzt besucht am 31. März 2003

[Kantel und Dierig 1998] Jörg Kantel, Sven Dierig: *„Hypermediale Wissensanordnung und collagierender Umgang mit dem Material?" - Digitales „Bildersammeln" am Beispiel des VIPP (Virtual Institute of Physiology Project)*, Vortrag, gehalten auf der HyperKult VII am 18. Juli 1998, Universität Lüneburg, (`derschockwellenreiter.editthispage.com/vlp/hypermedia.html`), zuletzt besucht am 27. April 2003

[Kantel und Dierig 1999] Jörg Kantel, Sven Dierig: *Zur Verwaltung großer Datenmengen im WWW am Beispiel des VIPP (Virtual Institute for Physiology Project)*, in: Friedbert Kaspar, Hans-Ulrich Zimmermann (Hrsg.): *Internet- und Intranet-Technologien in der wissenschaftlichen Datenverarbeitung, 15. DV-Treffen der Max-Planck-Institut 18. - 20. November 1998 in Göttingen*, Göttingen (GWDG-Bericht Nr. 53) 1999, auch online zu lesen unter `derschockwellenreiter.editthispage.com/vlp/datenmengen.html`, zuletzt besucht am 27. April 2003

[Kantel 2001] Jörg Kantel: *Was ist RSS?*. Eine Einführung in Content Syndication, (`www.schockwellenreiter.de/webdesign/rss.html`), zuletzt besucht am 13. Februar 2003

[Kantel 2002] Jörg Kantel: *Archäologie des Bloggens*. Ein erster und unvollständiger Versuch, dem „Phänomen Weblog“ mittels einer Geschichte seiner einzelnen Teile näher zu kommen (`www.tzw.biz/www/home/article.php?p_id=2028`), in Christian Eigner (Hg.): *Blogging und die neue Medienkultur des Netzes*, 2002, (`www.tzw.biz/www/home/article.php?p_id=2034`), zuletzt besucht am 13. Februar 2003.

[Kantel 2002a] Jörg Kantel: *Radio UserLand*, Bloghaus, März 2002 (`www.blogworld.de/tm_article.php?article_id=5`), zuletzt besucht am 27. April 2003

[Langham 2003] Matthew Langham: *Kleines Format ganz groß.* RSS – Eine Einführung in Content Syndication mit XML, in: *Linux Enterprise 3.03*, S. 81 - 84

[Laurent, Johnston & Dumbill 2001] Simon St. Laurent, Joe Johnston & Edd Dumbill: *Programming Web Services with XML-RPC*m Foreword by Dave Winer, Sebastopol (O'Reilly) 2001

[Mattison 2003] David Mattison: *Quickwiki, Swiki, Twiki, Zwiki and the Plone Wars* – Wiki as a PIM and Collaborative Content Tool, April 2003 (`www.infotoday.com/searcher/apr03/mattison.shtml`), zuletzt besucht am 8. April 2003

[Mortensen & Walker 2002] Torill Mortensen & Jill Walker: *Blogging thoughts: personal publication as an online research tool*, in: Andrew Morrison (ed.): *Researching ICTs in Context* InterMedia Report 3/2002, University of Oslo, 2002, p. 249 - 279 (`www.intermedia.uio.no`

/konferanser/skikt-02/skikt-research-conferance.html), zuletzt besucht am 28. April 2003

[Neuburg 1998] Matt Neuburg: *Frontier. The Definitive Guide*, Sebastopol (O'Reilly) 1998

[Renn 2000] Jürgen Renn: *Challenges of the Information Revolution for the Max Planck Society*, Berlin (Max-Planck-Institut für Wissenschaftsgeschichte, Preprint 151) 2000 (www.mpiwg-berlin.mpg.de/Preprints/P151.PDF), PDF, 512 KB

[Renn 2001] Jürgen Renn: *Erwirb es um es zu besitzen: Kulturelles Erbe im Zeitalter der Informationsrevolution*, Berlin (Max–Planck-Institut für Wissenschaftsgeschichte, Preprint 176) 2001 (www.mpiwg-berlin.mpg.de/Preprints/P176.PDF), PDF, 64 KB

[Renn 2002] Jürgen Renn [ed.]: *ECHO - an infrastructure to bring Eurpean Cultural Heritage Online*, Berlin (Max-Planck-Institut für Wissenschaftsgeschichte, Preprint 191) 2002 (www.mpiwg-berlin.mpg.de/Preprints/P191.PDF), PDF, 1,4 MB

[Schatten 2003] Alexander Schatten: *Die offene Wissensgesellschaft und ihre Feinde* (www.schatten.info/), Version vom 27. Februar 2003

[Snell, Tidwell & Kulchenko 2002] James Snell, Doug Tidwell & Pavel Kulchenko: *Programming Webservices with SOAP*, Sbastopol (O'Reilly) 2002

[Udell 1999] Jon Udell: *Practical Internet Groupware*, Sebastopol (O'Reilly) 1999

[ZIM 2003] Heinz Nixdorf Zentrum für Informationsmanagement (ZIM) in der Max-Planck-Gesellschaft: *eDoc Handbuch* (edoc.mpg.de/3564), Fassung vom 10. März 2003