Universität zu Köln
Philosophische Fakultät
Informationsverarbeitung

Sind die kanonischen Zitierweisen der Geisteswissenschaften als nachhaltige Komponenten digitaler Repositorien geeignet?

Magisterarbeit zur Erlangung des Magister Artium durch den Fachbereich Historische Kulturwissenschaftliche Informationsverarbeitung bei Prof. Dr. M. Thaller.

vorgelegt von:
Bernhard Assmann
Im Leimfeld 3
51065 Köln
as@ba.tuxomania.net
(0 22 1) 6 95 09 46

Inhaltsverzeichnis

1. Einleitung

Primäre Ressourcen der Geisteswissenschaften (Archivalien, Editionen, Bibliotheksbestände) werden, neben der Veröffentlichung in Buchform oder CD-ROM, immer häufiger über das World Wide Web (WWW) zugänglich gemacht. Die einfache Benutzbarkeit und die sofortige internationale Erreichbarkeit zählen zu den Hauptargumenten für diese Form der Veröffentlichung. In einer ersten Phase der Mediennutzung wurde viel mit den neuen Möglichkeiten experimentiert, zum einen, um die Machbarkeit zu demonstrieren, zum anderen, um mögliche Implikationen auf die Methoden der Aufbereitung und der Präsentation der Ressourcen zu untersuchen. Diese Phase ist zwar noch längst nicht abgeschlossen, dennoch hat sich die Ansicht durchgesetzt, dass das WWW zukünftig als bevorzugter Ort der Veröffentlichungen von Primärdaten anzusehen ist. Mit dieser Ansicht steigen aber auch die Ansprüche an die jeweiligen Angebote, und die Unterschiede zwischen den Veröffentlichungsmedien werden deutlicher sichtbar.

Ein wichtiger Aspekt bei der Publikation geisteswissenschaftlicher Ressourcen ist deren Zitierfähigkeit. Bei einem Buch ist die Zitierfähigkeit durch die bibliographischen Angaben gegeben. Die Verzeichnungssysteme der Bibliothekare garantieren die Auflösung dieser Angaben. Es ist auch selbstverständlich, dass die Angaben, die die Zitierfähigkeit garantieren, vom Autor, Herausgeber oder den Bibliotheken bereits sorgfältig ausgewählt werden. Die Angaben unterliegen zudem einem etablierten Schema, dass von den beteiligten Organen eingehalten wird.

Das Gegenstück zu einer Zitation in einem Buch ist in der Regel die URL (Uniform Resource Locator). Bei den beiden Hauptaspekten der Zitierfähigkeit, garantierte Auflösung und Sorgfalt der Auswahl, ist in den heutigen Online-Angeboten noch nicht dieselbe Qualität gegeben wie es die Nutzer vom Buch her gewöhnt sind. Dies liegt aber hauptsächlich in der mangelnden Sorgfalt bei der Erstellung der Angebote begründet. Denn auch eine URL kann und will sorgsam gewählt werden. Bei den heutigen Online-Angeboten ist dies aber oft nicht der Fall, denn dort wird noch zu sehr auf das Funktionieren Wert gelegt. Die garantierte und sorgfältig vorbereitete Zitierfähigkeit durch die URL steht nicht im zentra-

len Blickfeld dieser Angebote. Die URL wird meist unreflektiert den technischen Rahmenbedingungen unterworfen, was die Möglichkeit, sie als Quellenangabe zu verwenden, stark einschränkt.

Weitere Aspekte, in denen sich die Veröffentlichung von Daten in gedruckter Form und in Online-Angeboten unterscheiden, sind einmal die vorhandenen Methoden und Traditionen der Geisteswissenschaften in Bezug auf die Daten, zum anderen die Einheitlichkeit der Art und Weise, wie die Daten aufbereitet werden. Die Ressourcen, die in Online-Angeboten präsentiert werden, sind meistens für die jeweiligen Wissenschaften nicht grundlegend neu. Ob es sich um die Retro-Digitalisierung einer Editionsreihe handelt oder aber um eine Erstveröffentlichung, es existieren Methoden und Traditionen, mit diesem Material referenzierend umzugehen. In den meisten Fällen hat sich auch für die Zitation des Materials schon lange ein System in der jeweiligen Wissenschaft etabliert: die sogenannten kanonischen Zitierweisen. Der Wissenschaftler geht mit diesen Zitierweisen täglich um, und sie treten ihm in Form von Referenzen auf Lexika, Standardwerke oder wissenschaftlichen Zeitschriften entgegen. Sie werden als so grundlegend erachtet, dass jeder Student dieser Wissenschaft Listen mit Standardwerken und Zeitschriften sowie deren Abkürzungen auswendig zu lernen hat. Untersucht man nun, inwieweit diese Referenzen in den schon bestehenden Online-Angeboten erscheinen, ergibt sich ein disparates Bild: Einige Angebote übernehmen sie, bei anderen fehlen sie gänzlich.

Bezüglich der Einheitlichkeit der Aufbereitung besteht ebenfalls noch großer Handlungsbedarf. Jedes neue Angebot implementiert seine eigene Sicht auf die jeweiligen Daten. Dies hat zur Konsequenz, dass die Steuerung eines Angebotes im Vergleich zu anderen variiert, obwohl die Daten, die präsentiert werden, gleichartig sein können. Auch hier ist der Benutzer vom Medium Buch anderes gewohnt. Urkundeneditionen besitzen beispielsweise einen gleichartigen Aufbau, und jeder Benutzer wird sich in den verschiedenen Werken schnell zurechtfinden[1].

[1] Die Einheitlichkeit bei dem Beispiel Urkundenedition geht sogar so weit, dass ein deutscher Fachmann sich selbst in Ausgaben georgischer Urkunden grob orientieren kann, ohne auch nur ein Zeichen lesen zu können, vgl. K'art'uli istoriuli sabut'ebis korpusi. 445 K'art'uli istoriuli sabut'ebi IX - XIII ss, hg. v. T'ina Enuk'ije, (= Sak'art'velos istoriis cqaroebi, Bd. 30), T'bilisi 1984.

Neben den eben erwähnten Mängeln, die den Betreibern der Online-Angebote anzulasten sind, ist die URL an sich mit einem gewissen Makel der Unbeständigkeit behaftet. Jeder Benutzer des WWW ist schon einmal auf die berühmte HTTP-Fehlermeldung 404 (File not found) gestoßen. Das Problem der Unbeständigkeit ist genau so alt wie das WWW selbst. Deswegen wird seit geraumer Zeit nach Lösungsmöglichkeiten gesucht. Gerade in den Geisteswissenschaften ist die Nachhaltigkeit von Publikationen eines der Kriterien, das erfüllt sein muss, wenn ernsthaft in Erwägung gezogen wird, den Schritt zu einem Online-Angebot zu machen. Es wird also ein Weg, ein Konzept gesucht, das nachhaltig stabile, den Traditionen des Faches verpflichtete URLs liefert. Diese Arbeit will einen Beitrag dazu leisten.

Im ersten Teil der Arbeit werden die Bedingungen untersucht, unter denen kanonische Zitierweisen vorkommen. Dies geschieht zunächst für den traditionellen Raum, das gedruckte Buch, dann für den Online-Raum, die Web-Applikation. Weiterhin wird ein Konzept vorgestellt, wie anhand eines semantischen Netzes Werke, die über kanonische Namen referenziert werden, sowie die Verbindungen innerhalb und zwischen diesen Werken, modelliert und gespeichert werden können. Daneben werden die Funktionsweisen eines solchen Netzes und die Möglichkeiten, die es bieten kann, aufgezeigt.

Der zweite Teil handelt von den nachhaltigen Komponenten digitaler Repositorien. Es werden die schon bestehenden Konzepte und Bemühungen untersucht, die versuchen, die Stabilitätsprobleme, die den URLs anhaften, zu beseitigen. Dabei steht die Frage im Vordergrund, wie gut sich die kanonischen Namen in die jeweiligen Systeme integrieren lassen. Abschließend wird ein neues Konzept der Adressierung von geisteswissenschaftlichen Objekten vorgestellt, das mehr auf die eigentliche Referenzierung der Objekte im Rahmen der Traditionen und Gepflogenheiten der jeweiligen geisteswissenschaftlichen Disziplinen abzielt.

2. Die kanonischen Zitierweisen der Geisteswissenschaften

2.1. Die Zitierweisen im Raum der gedruckten Medien

Für die gedruckte Monographie, den Aufsatz ebenso wie für Quelleneditionen und Nachschlagewerke haben sich im Laufe der Wissenschaftsgeschichte feste Konventionen etabliert, die eine eindeutige und klare Referenzierung eines Werkes erlauben. Die Referenzierung erfolgt in der Regel über zentrale bibliographische Angaben wie Autor oder Herausgeber, Titel oder Reihenzugehörigkeit. Diese werden - von Fachwissenschaft zu Fachwissenschaft variierend - in ein Schema gebracht, das auflösbar und konsistent sein muss. Zu den Zitierkonventionen zählen darüber hinaus auch verbindliche Kurzbezeichnungen bestimmter Werke. Die kanonischen Zitierweisen sind mehr als bequeme Abkürzungen von langen Buchtiteln, sie sind traditionsreiches Wissen.

Zu den Werken, die mit einem kanonischen Namen versehen sind, gehören zum einen die zentralen Werke eines Fachs wie etwa Lexika und Nachschlagewerke, Editionen und Ausgaben von „Texten“[2] sowie Verzeichnisse aller Art. Zum anderen werden die Werke mit kanonischen Namen bezeichnet, die von besonderer Relevanz bzw. häufig gebraucht werden, wie etwa wissenschaftliche Zeitschriften. Aber auch außerhalb des Wissenschaftsbetriebs finden sich etablierte Abkürzungssysteme. Ein Beispiel hierfür ist die Bibel. Auf die „Bücher“ der Bibel wird mittels einer Abkürzung des „Buches“ verwiesen, gefolgt von Kapitelnummern und Versangaben (z.B. Mt 12,4). Der Nutzen liegt auf der Hand: Jeder, der sich mit dem Werk beschäftigt, ob wissenschaftlich oder nicht, kann, sobald er die Konventionen kennt, das Referenzsystem erfolgreich anwenden. Die Bibel ist allerdings ein Sonderfall, was die Popularität und die Verbreitung des Referenzsystems angeht. Andere Referenzen, die im wissenschaftlichen Bereich existieren, sind hingegen kaum einem breiten Publikum bekannt, ja, es wäre zum Teil sogar

[2] Der Textbegriff ist hier als sehr weit anzusehen, da darunter auch Ausgaben von musikalischen Werken fallen.

verwunderlich, wenn ein Forscher aus einer Geisteswissenschaft alle Zitierweisen der Nachbardisziplin auflösen könnte, da diese häufig fachspezifisch sind.

Ein Grund für die Entwicklung der kanonischen Abkürzungen ist in der Notwendigkeit zu sehen, Platz einzusparen. Der Raum, der für die bibliographische Angabe zur Verfügung steht, etwa innerhalb einer Fußnote, ist in der Regel begrenzt, und soll daher nicht mit unnötigen „Zeichen“ überfrachten werden. Des weiteren können innerhalb eines Beitrages sehr viele Anmerkungen enthalten sein, was den Referenzapparat zusätzlich aufbläht. In welcher Form und in welchem Maße abgekürzt wird, liegt in der Verantwortung des Autors. Das wichtigste Ziel, nämlich das angegebene Werk exakt und schnell aufzufinden, darf aber nicht aus den Augen verloren werden. Eine kanonische Zitierweise für einen Zeitschriftenartikel lautet beispielsweise: Geldner, Das Problem der vierzehn Slavenkirchen Karls des Großen, in: DA 42, 1986, 192–205. Im Gegensatz zur vollständigen bibliographischen Angabe[3] lassen sich bei diesem Beispiel durch die Kürzung des Titels 119 und durch die kanonische Referenz 54 „Zeichen“ einsparen.

Neben dem Zwang zur Kürze besteht auch die Notwendigkeit der Entschlüsselbarkeit und der logischen Verortung einer kanonischen Referenz. In den geisteswissenschaftlichen Fächern werden häufig große Reihen aufgelegt, um das grundlegende Material, mit dem sich das Fach beschäftigt, zu publizieren. Für den Bereich der mittelalterlichen Geschichte hat diese Aufgabe u.a. die Monumenta Germaniae Historica (MGH) übernommen. Die Editionsreihe, die von dieser Institution herausgegeben wird, ist in mehrere Reihen gegliedert, die sich jeweils einem speziellen Bereich widmen. Für jede Reihe und für jede Unterabteilung einer Reihe wird eine eigene Sigle verwendet. Sie dient einerseits als Referenz für einen Band innerhalb der Reihe bzw. Abteilung, andererseits ermöglicht sie die Zugehörigkeit und die Einordnung in das Gesamtsystem[4]. Die Referenzierung bewegt sich nicht nur auf der Band-Ebene, sondern wird auch innerhalb eines Bandes an-

[3] Geldner, Ferdinand: Das Problem der vierzehn Slavenkirchen Karls des Großen im Lichte der bisher unbeachteten Dorsalvermerke der Urkunden Ludwigs des Deutschen (845) und Arnolfs (889), in: Deutsches Archiv für Erforschung des Mittelalters Bd. 42, 1986, S. 192–205.

[4] Beispielsweise werden mit der Referenz MGH Ldl die Streitschriften des sog. Investiturstreites bezeichnet. Sie werden innerhalb der Scriptores-Reihe der MGH geführt. Dies sollte man vorher wissen, denn sonst ergeben sich Schwierigkeiten, einen Band dieser Abteilung in einer Bibliothek aufzufinden.

gewendet. Besonders wird dies bei der Diplomata-Reihe der MGH deutlich, in der die Königs- und Kaiserurkunden publiziert werden. Eine übliche Referenz lautet: MGH D HIV 325. Die Bandangabe erfolgt durch die Bezeichnung des entsprechenden Herrschers (hier Heinrich IV.), die darauf folgende Nummer (325) verweist auf das einzelne Stück. Da es sich innerhalb der Geschichtswissenschaften etabliert hat, in Urkundeneditionen die einzelnen Stücke mit einer fortlaufenden Nummer zu versehen, ersetzt sie die sonst zu erwartende Seitennummer.

Ein Problem im Umgang mit den kanonischen Zitierweisen ist die Uneinheitlichkeit. Häufig können für einen Verweis mehrere Systeme angewendet werden, die alle zu dem selben Ergebnis führen. Das im vorherigen Absatz gewählte Beispiel kann auch in folgender Weise angegeben werden: MGH DD Abt. 4, Bd. 6, 325 oder äquivalent MGH DDRegImpGerm Bd. 6, 325. Diese Angaben betonen stärker die Verortung innerhalb des Reihenschemas der MGH, denn die Urkunde ist im sechsten Band („Die Urkunden Heinrichs IV.“) der vierten Abteilung („Die Urkunden der deutschen Könige und Kaiser“) der Diplomata-Reihe erschienen. Welche der Alternativen im konkreten Fall verwendet wird, entscheidet der Autor[5].

2.2. Die Zitierweisen im Raum der digitalen Medien

2.2.1. Rahmenbedingungen

Das Gegenstück zu einer traditionellen Quellenangabe, die ein gedrucktes Werk referenziert, ist im Raum der Online-Angebote die URL. Sie kann grob in zwei Teile gegliedert werden. Der erste Teil besteht aus dem verwendeten Protokoll[6] und dem Rechnernamen. Im zweiten Teil steht die Anforderung an den HTTP-Server. Beide Teile können unter gewissen Einschränkungen frei gewählt werden. Diese freie Wählbarkeit bietet die Möglichkeit, kanonische Namen dort zu integrieren.

[5] Innerhalb der geschichtswissenschaftlichen Mediävistik ist die Tendenz zu beobachten, dem ersten Beispiel (mit Herrschername) den Vorzug zu geben.

[6] Hier wird nur das HTTP-Protokoll beachtet.

Der Rechnername ist, wie das Wort schon andeutet, nur ein Name, der zusätzlich eingeführt wurde, um Menschen den Umgang mit den numerischen Adressen der beteiligten Rechner zu erleichtern. Technisch gesehen macht diese Zwischenstufe keinen Sinn, da hier ein zusätzlicher Auflösungsmechanismus eingeführt werden musste, damit zwischen den Namen und den technischen Adressen vermittelt werden kann. Das Domain Name System (DNS) bietet die Grundlage für diesen Auflösungsmechanismus. Bei der Vergabe eines DNS-Namens sind zwei Einschränkungen zu beachten: Zum einen steht das Ende eines solchen Namens mehr oder weniger „fest“, da nur eine begrenzte Anzahl von Top Level Domains (TLDs) existieren, und zum anderen sind die einmal vergebenen Namen nicht mehr verfügbar, da diejenige Person oder Institution, die sich eine entsprechende Domain gesichert hat, diese dauerhaft nutzen darf[7]. Leider haben es die geisteswissenschaftlichen Fächer versäumt, sich in großem Stil entsprechende Namen zu sichern. So kommt beispielsweise das Lexikon des Mittelalters zu spät: die Domains lma.de und lexma.de, die den kanonischen Namen nahe kommen würden, sind beide schon vergeben[8].

Die nachträgliche Sicherung von Domain-Namen durch die Anbieter geisteswissenschaftlichen Materials, wie etwa die herausgebenden Institute, Bibliotheken oder Archive, ist ein Problem, da es keine konsistente Domain-Struktur gibt, die die kanonischen Namen berücksichtigt. Ein Weg wäre die Einführung einer neuen Top Level Domain. Diese Absicht ist langfristig zu verfolgen, aber es ist fraglich, ob dies wirklich zu einem befriedigenden Erfolg führen würde. Ein anderer Weg wäre die Einführung einer neuen Unterdomain im nationalen Bereich, wie dies etwa in Großbritannien üblich ist[9]. Es bleibt ungewiss, ob die kanonischen Namen in die Domain-Struktur Einzug halten werden. Angebote sollten dies berücksichtigen und es bedarf einer zusätzlichen Konstruktion, um das Pro-

[7] Auf die juristischen Feinheiten, wann eine Domain zurückgegeben werden muss, bzw. einer anderen Person oder Institution übergeben werden muss, soll hier nicht eingegangen werden.

[8] Die Domain lma.de führt zu einem Hersteller von Larynxmasken, die während einer Narkose gebraucht werden können, lexma.de zu einem Full-Service-Anbieter in den Medienbereichen Radiowerbung und Internet-Präsentation.

[9] Dort wurde für die Domainnamen der akademische Institutionen der Unterbereich „ac“ eingeführt. Die Bodleian Library der Universität von Oxford erreicht man z.B. unter dem Namen: www.bodley.ox.ac.uk.

blem der kanonischen Referenzierung zu lösen[10]. Das später vorgestellte URL-Konzept wird sich dieser Herausforderung stellen.

Die Anforderung an den HTTP-Server wird meist direkt auf das dem Server zugrunde liegende Dateisystem abgebildet. Diese statische Auslieferung einer Ressource tritt bei den hier zu behandelnden Systemen aber in den Hintergrund. In den Online-Angeboten, die Primärdaten präsentieren, werden oft dynamische Verfahren eingesetzt. In diesem Bereich der HTTP-Server kommt der Vorteil zum Tragen, dass die Form der Anforderung, also die für den Benutzer sichtbare URL, sehr frei gestaltet werden kann. Es sind nur wenige einfache Einstellungen in der Konfiguration eines HTTP-Servers nötig, und schon kann sich ein URL-Gestalter ans Werk machen. Das liegt daran, dass solche Server ein weit reichendes Instrumentarium anbieten[11], mit denen die öffentlichen Namen (also die URLs) auf interne Ressourcen (statische Dateien, Programme) abgebildet werden können.

Die gängigste und einfachste Möglichkeit, HTML-Seiten dynamisch durch ein Programm generieren zu lassen, bedient sich des Common Gateway Interfaces (CGI)[12]. Das CGI definiert die Art und Weise, wie der HTTP-Server mit externen Programmen umzugehen hat und welche Informationen ihnen zur Verfügung stehen sollen. Des weiteren ist in CGI definiert, wie mit Formulardaten umgegangen wird, die vom Client kommen. Es gibt zwei Möglichkeiten des Zugriffes für das externe Programm: über die Umgebungsvariable QUERY_STRING (GET) oder mittels der Standardeingabe (POST). Die POST-Methode kann hier vernachlässigt werden, da es damit nicht möglich ist, URLs anzubieten, die zitabel sind, da sie nicht alle notwendigen Informationen aufnehmen können. Bei der GET-Methode sind diese Informationen Teil der URL und sind somit zur Zitierung geeignet. Zwar ist das Common Gateway Interface nur eine Vereinbarung unter den Entwicklern der HTTP-Server, die nie offiziell standardisiert wurde, doch kann es durch seine weite Verbreitung und Akzeptanz als De-facto-Standard angesehen werden.

---

[10] Siehe dazu auch Thaller, Handschriftenbibliothek, S. 35.

[11] Bei dem HTTP-Server Apache kann dafür das Modul mod_rewrite benutzt werden. Es stellt vielfältige Möglichkeiten zur Manipulation von URLs zur Verfügung, selbst das Verknüpfen von öffentlichen Namen mit Inhalten, die von anderen Servern generiert wurden, stellt kein Problem dar.

[12] Vgl. die CGI-Spezifikation, http://hoohoo.ncsa.uiuc.edu/cgi/interface.html.

Anhand der Konfiguration des HTTP-Servers Apache[13] soll kurz veranschaulicht werden, wie dem Server mitgeteilt wird, dass ein externes Programm über die CGI-Schnittstelle für den Inhalt einer Ressource verantwortlich ist, und welche Möglichkeiten in Bezug auf die verwendeten Namen existieren. Grundsätzlich bedarf es nur zweier Angaben. Man braucht erstens den öffentlichen Namen, also die Ressource, die von einem WWW-Browser angefordert wird, und zweitens eine Angabe über das lokale Programm, welches extern (aus Sicht des HTTP-Servers) gestartet wird. Dazu ein Beispiel:

```
ScriptAlias /cgi-bin/ /web/cgi-bin/
```

Diese Direktive in der Konfiguration des Apache-Servers führt dazu, dass alle Programme, die unter dem lokalen Verzeichnis /web/cgi-bin/ vorhanden sind, vom Apache-Server als CGI-Programm angesehen werden (Direktive ScriptAlias) und wenn sie angefordert werden, ausgeführt werden. Der öffentliche Name für dieses Verzeichnis ist /cgi-bin/. Dies bedeutet, dass falls die Ressource /cgi-bin/test.pl vom Server gefordert wird, versucht wird, /web/cgi-bin/test.pl als CGI-Programm auszuführen, und die Ausgabe dieses Programms das Ergebnis der Anfrage ist.

Neben der Deklaration eines Verzeichnisses ist in diesem Rahmen noch eine weitere Vorgehensweise von Interesse, nämlich die Verbindung eines öffentlichen Namens mit einem einzigen Programm.

```
ScriptAlias /test/ /web/test/test.pl
```

Mit dieser Direktive wird der öffentliche Name /test/ direkt und ausschließlich mit dem Programm /web/test/test.pl verbunden. Dies hat den entscheidenen Vorteil, dass das Programm hinter dem externen Namen versteckt wird und die URL noch weiter gehende Angaben aufnehmen kann. Durch das CGI sind diverse Umgebungsvariablen garantiert und stehen dem vom Server gestarteten Programm zur Verfügung. Dieses kann die Variableninhalte dann auswerten und entsprechend bei der Ausgabe reagieren. Eine der Umgebungsvariablen ist PATH_INFO.

[13] Vgl. http://httpd.apache.org/.

Der Wert dieser Variable ist der Bestandteil einer URL, der nach dem öffentlichen Namen eines Programmes beginnt und bis zu den Formular-Daten (nach der GET-Methode) reicht. Durch diese Eigenschaft ist die Variable ideal für eine Web-Applikation, bei der man den Aufbau der URL selbst bestimmen möchte. Dies soll an einem weiteren Beispiel demonstriert werden; gegeben sei folgende Direktive:

```
ScriptAlias /diplome/ /web/diplome/diplome.cgi
```

Wird nun /diplome/heinrichIV/325/ gefordert, interpretiert der Web-Server die URL und erkennt, dass /diplome/ mit dem Programm /web/diplome/diplome.cgi verbunden ist. Die restlichen Angaben der URL sind die Extra-Pfadangaben, die in der Umgebungsvariable PATH_INFO durch den HTTP-Server bereit gestellt werden. Das Programm diplome.cgi kann auf den Wert (hier also /heinrichIV/325/) zurückgreifen und diesen ebenfalls interpretieren. Damit ist eine Möglichkeit geschaffen, die URL zur Steuerung einer Web-Applikation zu benutzen. Denn wie die Bestandteile der URL nach dem externen Namen aussehen, liegt ganz in der Hand des URL-Designers. Er entscheidet, welchen Aufbau die URLs besitzen und was deren jeweilige Bedeutung ist. Bei der Entwicklung einer URL hat es sich eingebürgert, den Schrägstrich (/) als Trennzeichen zu benutzen, was allerdings nur eine Konvention und grundsätzlich nicht erforderlich ist.

Zum Schluss dieses kleinen Exkurses in die Konfiguration des Apache-Servers soll noch auf folgende Einstellungsmöglichkeit hingewiesen werden.

```
ScriptAlias / /web/web-site.pl
```

Diese Direktive veranlasst, dass alle öffentlichen Namen des Servers frei gewählt werden können, da die komplette Web-Site von einem Programm (web-site.pl) generiert wird. Diese Eigenschaft ist gleichzeitig der größte Nachteil dieser Möglichkeit, denn das zuständige Programm muss eine relativ hohe Komplexität besitzen, um allen praktischen Aufgaben bei der Bereitstellung von Informationen, die täglich anfallen können, gerecht zu werden.

Zusammenfassend lässt sich sagen, dass es aufgrund der ungelösten Domain-Problematik eher schwierig ist, kanonische Namen in die Rechneradresse zu über-

nehmen. Die Integration der Namen in die restliche URL ab dem Rechnernamen stellt aber keine großen Anforderungen an die entsprechende Technik. Durch die Möglichkeiten, die HTTP-Server den Entwicklern von entsprechenden Web-Applikationen bieten, in Verbindung mit dem Common Gateway Interface, sind der freien Gestaltung des lokalen Bestandteils einer URL keine Grenzen gesetzt.

2.2.2. Beobachtungen

Seitdem das World Wide Web (WWW) seinen Siegeszug als beherrschende Anwendung des Internets zur Veröffentlichung von Texten, Bildern und Daten angetreten hat, sind zahlreiche Websites entstanden, die geisteswissenschaftliches Material anbieten. Nachdem die erste Phase des experimentellen Umgangs mit den neuen Medium vorüber ist, wird das WWW mehr und mehr zum Veröffentlichungsort von Primärdaten der einzelnen Fächer. Ein zentrales Verzeichnis aller Online-Angebote der Geisteswissenschaften existiert jedoch nicht. Für die einzelnen Fachbereiche muss auf viele verstreute Portale zurückgegriffen werden, die jedoch keinen Anspruch auf Vollständigkeit erheben, da sie über den Charakter von „Linklisten“ selten hinauskommen[14]. Die Notwendigkeit für ein zentrales, systematisch zusammenfassendes Verzeichnis ist daher gegeben. Im Bereich der digitalisierten Bibliotheksbestände ist dies bereits erkannt worden, und es wird an entsprechenden Projekten gearbeitet[15].

In einer kurzen stichprobenartigen Schau soll untersucht werden, wie aus technischer Sicht schon bestehende Angebote mit der freien Gestaltungsmöglichkeit der URL umgehen, und ob sie die kanonischen Namen hier integrieren. Zu diesem Zweck werden die URLs der kleinsten zitierfähigen Einheiten analysiert, wie

[14] Dazu einige (willkürlich gewählte) Beispiele: Geschichtswissenschaft, http://www.lehre.historicum.net/links/histquell1.html; Germanistik, http://www.phil.uni-erlangen.de/~p2gerlw/ressourc/eltext.html; Anglistik, http://www.ub.ruhr-uni-bochum.de/DigiBib/Volltext/Anglistik_Vol.htm.

[15] Siehe beispielsweise die Übersicht über die Retrodigitalisierungs-Projektserver des DFG-Förderschwerpunktes Retrospektive Digitalisierung von Bibliotheksbeständen bei der Historisch Kulturwissenschaftlichen Informationsverarbeitung Köln (http://www.hki.uni-koeln.de/retrodig/), oder das entstehende Portal „Zentrales Verzeichnis Digitalisierte Drucke“ an der Staats- und Universitätsbibliothek Göttingen, das einen systematischen Zugang zu allen online erreichbaren gedruckten Werken ermöglichen will (Projekt-Homepage: http://www.zvdd.de/, schon im Projekt erfasste digitale Sammlungen: http://www.zvdd.de/projekte.html).

etwa Seiten bei Handschriften, Kapitel bei längeren Texten oder eine Einheit bei Angeboten, die auf ein Nummerierungssystem zurückgreifen. Es wurden nur solche URLs einbezogen, die von dem jeweiligen Projekt angeboten werden und durch Verknüpfungen erreichbar sind. Falls die URL einer Einheit nicht sofort ersichtlich war, wie dies etwa bei Frame-basierten Angeboten der Fall sein kann[16], wurde sie extrahiert, was meistens zur Folge hatte, dass die umgebende Navigation verloren ging. Dies ist aber vernachlässigbar, da die Navigation in diesem Rahmen nicht relevant ist, sondern nur, wie auf einzelne Objekte zugegriffen werden kann. Dieser notwendige Zwischenschritt demonstriert einen deutlichen Nachteil von Frame-basierten Angeboten. Es ist die Frage, ob die praktischen Vorteile bei der Erstellung des Angebotes diese Nachteile aufwiegen.

Die Analyse untersucht außerdem, ob die URLs denselben Ansprüchen genügen, wie sie an traditionelle Zitierformen von Büchern angelegt werden. Dadurch, dass die folgenden Angebote in diese Untersuchung aufgenommen wurden, genügen sie schon dem Kriterium der Exaktheit, da alle vorgestellten URLs ein Objekt direkt und eindeutig adressieren können. Folgende Projekte bzw. Angebote wurden untersucht:

- die Perseus Digital Library
- das Angebot der Regesta Imperii
- die Codices Electronici Ecclesiae Coloniensis
- die Online-Versionen verschiedener Bibel-Ausgaben der Deutschen Bibelgesellschaft.

Perseus Digital Library

Die Perseus Digital Library[17] gehört zu den großen Textangeboten des WWWs. Neben dem Bereich des griechischen und lateinischen Altertums bietet sie weite-

[16] Frame-basiert meint hier die Art und Weise, wie die Seiten in HTML erstellt wurden. Werden HTML-Frames verwendet so ändert sich für den Benutzer die angezeigte URL nicht, wenn er der Verknüpfung zu einem Objekt folgt. Die direkte Adressierbarkeit ist zwar gegeben, sie wird für den Benutzer jedoch nicht offensichtlich.

[17] Vgl. http://www.perseus.tufts.edu.

re Sammlungen wie etwa die Werke von William Shakespeare. Daneben werden Hilfsmittel zum Umgang mit den Inhalten der Texte angeboten, so etwa Wörterbücher und Atlanten. Hier interessieren vor allem die angebotenen Texte zur Antike.

Nach Eingabe der URL[18] [...]/cgi-bin/ptext?doc=Perseus:text:1999.02.0002:book=2:chapter=2 erhält der Betrachter als Ergebnis das zweite Kapitel des zweiten Buches von Cäsars *De Bello Gallico*. Die Ausgabe wird dynamisch generiert. Das verantwortliche Programm heißt ptext und liegt an dem Ort, der dem HTTP-Server mit cgi-bin angegeben wurde. Da ptext nicht hinter einem (sinnvollen) Namen versteckt wurde, wird die Nachhaltigkeit der angebotenen URLs erschwert. Zukünftige Versionen des Ausgabeprogrammes müssten weiterhin mit /cgi-bin/ptext angesprochen werden, will man nicht riskieren, dass alle bisherigen URLs ungültig werden. Die Weiterführung des Programmnamens stellt vom technischen Standpunkt her zwar kein Problem dar, doch wenn schon ein arbiträrer Name eingeführt wird, warum dann nicht gleich ein sprechender?

Die Übergabe des einzigen Parameters von ptext (doc) erfolgt über die GET-Schnittstelle. Der Wert von doc hat seinen eigenen Aufbau und nimmt die Angabe des verwendeten Textes (text) und der benötigten Stelle (book und chapter) auf. Der Doppelpunkt und das Gleichheitszeichen stellen in diesem Zusammenhang Sonderzeichen dar und werden im Angebot entsprechend maskiert[19]. Mit der Maskierung sollen Probleme mit älteren Browsern vermieden werden, während moderne Browser diese Zeichen intern umsetzen und die Anfrage an den HTTP-Server mit den aufgelösten Sonderzeichen gestellt wird. Die Angabe ohne die Maskierung erfolgte hier aus Gründen der Übersichtlichkeit. Durch den Aufbau der URL kann sie intuitiv genutzt werden, um selbstständig innerhalb des Textes zu navigieren. Über den Text hinaus zu gehen, wird durch die Kodierung des Werkes mittels einer Nummer (1999.02.0002) erschwert, da nicht bekannt sein dürfte, welche Nummer einem Werk zugewiesen ist.

[18] Um die Darstellung ein wenig zu verkürzen, wird im Folgenden darauf verzichtet, den Rechnernamen und das verwendete Protokoll (http) mit anzugeben. Um die Beispiele nachvollziehen zu können, muss noch der Rechnername, der bei der kurzen Vorstellung des Projektes angegeben wurde, vorangestellt werden.

[19] Der Doppelpunkt mit %3A und das Gleichheitszeichen mit %3D.

In der Perseus Digital Library existieren aber mehrere Wege, die zu Cäsars Text führen. Alternativ bietet das System die Anforderung [...]/cgi-bin/ptext?lookup =Caes.+Gal.+2.2.1 an, um dieselbe Stelle zu referenzieren. Die alternative Form beinhaltet, als Variablenwert von lookup, die kanonische Zitierweise (Caes.+Gal. +2.2.1) dieser Stelle in den Altertumswissenschaften. Allein die üblichen Leerräume zwischen den einzelnen Bestandteilen wurden durch das Plus-Zeichen ersetzt, da auch das Leerzeichen innerhalb einer URL zu der Klasse der Sonderzeichen gehört, das, wenn es denn gebraucht wird, maskiert wird (und zwar als %20). Leider gibt es einen Unterschied in der Ausgabe der beiden Alternativen: Während die erste Form den lateinischen Text ausgibt, wird bei der zweiten Form eine englische Übertragung angezeigt. Dies liegt wohl daran, dass aus Sicht der Projektverantwortlichen der englische Text die standardmäßige Version des Textes darstellt. Die Vermutung wird von den verwendeten internen Nummern der Textversionen gestützt: der englische Text wird unter der Nummer 1999.02.0001 geführt, während der lateinische die Nummer 1999.02.0002 hat. Damit wird der Vorteil, der die Referenzierung durch den kanonischen Namen bietet, nicht genutzt, da in den Altertumswissenschaften sicherlich der lateinische Text als primäres Ziel des Verweises angesehen würde und nicht eine englische Übertragung.

Regesta Imperii

Die Deutsche Kommission für die Bearbeitung der Regesta Imperii e.V. bei der Akademie der Wissenschaften und der Literatur Mainz gibt ein Inventar aller urkundlichen und historiographischen Quellen der römisch-deutschen Könige von den Karolingern bis zu Maximilian I. sowie der Päpste des frühen und hohen Mittelalters heraus, das damit zu den großen Quellenwerken der deutschen und europäischen Geschichte gehört. Neben der traditionellen Veröffentlichung in Buchform existiert auch ein Online-Angebot[20], welches zum größten Teil die Volltexte der Regesten zur Verfügung stellt. Daneben kann zu allen Bänden ein Bild-Digitalisat der schon in Buchform erschienen Seiten abgerufen werden.

[20] Vgl. http://regesta-imperii.uni-giessen.de.

Die Bereitstellung der Bildseiten und die der Volltexte wird von unterschiedlichen Servern geleistet. Während die Volltexte auf dem eigentlichen Heimatserver der Regesta Imperii residieren, werden die Digitalisate der Buchseiten über einen Server des Münchener Digitalisierungszentrums (MDZ)[21] erreicht. Beide Angebote sind Frame-basiert, was zur Folge hat, dass die eigentlichen Referenzen auf ein Objekt nicht direkt erkennbar sind.

Um eine Bildseite zu erreichen, ist folgende URL nötig: [...]/digbib/urkunden1/rimages/ri08/ri08_hub1877/@Generic__BookTextView/215; cs=default;ts=default;pt=209;lang=de#X. Dies ist ein Beispiel für eine hochgradig unschöne URL. Der Einwand, dass hier die Funktion im Vordergrund steht, und diese Funktion, nämlich eine Seite anzuzeigen, hervorragend umgesetzt ist, wird dadurch überstimmt, dass die andere geforderte Funktion, nämlich ein Objekt auf einfache, transparente Art und Weise zu referenzieren, nur mangelhaft umgesetzt ist. Hier beherrscht die verwendete Software eindeutig die URL, die Technik und nicht der Inhalt bestimmt die Form. Bei den Bild-Digitalisaten wird das Programm DynaWeb verwendet, ein Softwarepaket, mit dem sich auf einfache Weise SGML und XML basierte Inhalte im WWW veröffentlichen lassen. Dieses Paket wird vom MDZ eingesetzt, um die digitalisierten Buchseiten seiner zahlreichen Projekte zu veröffentlichen. Der Wunsch, ein Standardpaket zur Veröffentlichung der Inhalte zu verwenden, kann unter praktischen Gesichtspunkten gut nachvollzogen werden, doch sollte die gewählte Software dann auch in höherem Maße konfigurierbar sein, damit sich nicht der Inhalt des Angebotes der Software anpassen muss (wie es hier der Fall ist), sondern der umgekehrte Weg beschritten werden kann.

Eine DynaWeb-URL lässt sich in drei Teile gliedern. Nach dem öffentlichen Namen für das Programm (digbib), folgt eine Pfadangabe zur aktuell angezeigten Kollektion. Danach wird das Layout-Template aufgeführt, dessen Namen mit dem @-Zeichen beginnt (@Generic__BookTextView) und schließlich folgen noch einige Anzeigeparameter[22]. Damit ist die URL ein Beispiel für die Werteübergabe

[21] Vgl. http://mdz.bib-bvb.de:80.

[22] Der ursprüngliche Hersteller Isno Co. existiert nicht mehr, und die Software wird nun von der Firma Enigma betreut. Die Funktionsweise der URL wurde anhand einer „ergoogleten“ (Suchwort dynaweb) Dokumentation ermittelt und findet sich u.a. unter: http://fobe.itaw.hu-

mittels PATH_INFO. Doch leider ist der Aufbau nicht besonders intuitiv: Es ist weder ersichtlich, um welches Objekt es sich handelt, noch wird dem Benutzer klar, wie diese Schnittstelle wohl funktionieren könnte, um beispielsweise ein ähnliches Objekt zu referenzieren. Beides sind aber Eigenschaften, die eine URL haben sollte.

Im Gegensatz zur Anzeige einer Bildseite ist die URL für eine Textseite erfreulich kurz. Es genügt die Angabe von [. . .]/regesten/regshow.php?pk=45878, um ein entsprechendes Ergebnis zu erhalten. Für die Auslieferung ist das Skript regshow.php verantwortlich, welches im Verzeichnis regesten liegt. Das Skript erwartet einen Parameter über die GET-Schnittstelle mit dem Namen pk. Hinter dieser Variable steckt wohl eine interne Identifikationsnummer für ein Regest. Leider hat diese Nummer nichts mit dem entsprechenden kanonischen Zitat des Regestes zu tun. So ist es unmöglich, selbstständig andere Regesten aufzuschlagen. Es liegt der Verdacht nahe, dass alle Regesten, die in das Angebot integriert wurden, durchnummeriert worden sind. Aus Sicht des Erstellers eines solchen Angebots ist dies verständlich, da es die einfachste Methode ist, die gestellte Aufgabe zu bewältigen. Doch wiederum muss bemängelt werden, dass dies nicht die hauptsächliche Funktion der URL ist.

Die Verwendung einer URL, die stark mit den technischen Gegebenheiten verzahnt ist, birgt auch Probleme in Bezug auf die Zukunftssicherheit der URLs. Da sich die dem Angebot zugrunde liegende Technik höchstwahrscheinlich ändern wird, können Schwierigkeiten auftreten, die URLs zu migrieren. Es ist sicherlich machbar, die Identifikationsnummer beizubehalten, doch der Aufwand ist eigentlich unnötig, da bereits ein eindeutiger, etablierter und zukunftssicherer Name existiert, nämlich der kanonische.

Codices Electronici Ecclesiae Coloniensis

Die Codices Electronici Ecclesiae Coloniensis (CEEC)[23] beinhalten die mittelalterlichen Kodizes der Erzbischöflichen Diözesan- und Dombibliothek Köln in di-

berlin.de/dynaweb/client/client/@Generic__BookTextView/197;cs=default;ts=default;pt=197/*#X.

[23] Vgl. http://www.ceec.uni-koeln.de.

gitaler Form. Die Digitalisierung wurden im Rahmen des DFG-Programms „Retrospektive Digitalisierung von Bibliotheksbeständen“ gefördert. Das Angebot umfasst die kompletten Kodizes mit Einband und allen Seiten. Neben den Digitalisaten werden auch noch Beschreibungsdaten über die Handschriften aus gedruckten Katalogen angeboten. Eine Besonderheit dabei ist, dass die unterschiedlichen Meinungen, zu denen die einzelnen Bearbeiter gekommen sind, nebeneinander stehen.

Wie schon die Online-Fassung der Regesta Imperii ist auch dieses Angebot Frame-basiert. Das gilt jedoch nur für die Beschreibungsdaten und die kleinste Auflösung der Handschriftenseiten. Für die Anzeige der größeren Auflösungen wird ein neues Browser-Fenster geöffnet, welches aus dem Frame-Verbund ausschert. Hier ist dann die URL direkt erkennbar.

Die Anzeige einer Seite aus einer Handschrift (Nr. 199, 7r) erfolgt über die URL [. . .]/ceec-cgi/kleioc/0010/exec/pagemed/"kn28-0199_014.jpg". Das zuständige Programm hört auf den Namen kleioc und liegt im Verzeichnis mit dem Namen ceec-cgi. Die restlichen Bestandteile der URL beinhalten die Extra-Pfadangaben, auf die über die Umgebungsvariable PATH_INFO zugegriffen werden kann. Die ersten beiden Werte (0010 und exec) sind technischer Natur und haben nichts mit dem Objekt zu tun. Der nächste Bestandteil pagemed steht für die Anzeigefunktion. Das System unterstützt vier verschiedene Größen, in denen die Handschriftenseiten angezeigt werden können. Pagemed steht für eine solche. Nach der Anzeigefunktion folgt die eigentliche Angabe, um welches Objekt es sich handelt. Warum diese Angabe in Anführungszeichen (hexadezimal mit %22 kodiert) eingerahmt wird, ist nicht ersichtlich. Der erste Bestandteil der Objektangabe kn28, bezeichnet die offizielle Signatur der Erzbischöflichen Diözesan- und Dombibliothek Köln im deutschen Bibliotheksverbund. Danach folgt die Signatur der Handschrift im System der Bibliothek und die laufende Nummer im System der Digitalisierung. Es ist zu vermuten, dass die gesamte Objektangabe einen Dateinamen bezeichnet, was durch das Ende der Angabe (.jpg) noch verstärkt wird. Leider wird durch die Maskierung der Anführungszeichen erschwert, dass dem Benutzer ein Dateiname angeboten werden kann, den er bei der eventuellen Speicherung einer Handschriftenseite nur noch übernehmen müs-

ste[24]. Es fällt weiterhin auf, dass eine Handschriftenseite nur in Verbindung mit einer Auflösungsgröße referenziert werden kann und dadurch ein Objekt nicht alleine stehen kann. In dieser URL ist die Angabe der Auflösung ein integraler Bestandteil der Objektadresse und nicht ein optionales Element, das was die Referenzierung einer Handschriftenseite angeht, eigentlich vernachlässigbar ist.

Für die Referenzierung von Handschriftenseiten hat sich in den Geisteswissenschaften die Recto-, Verso-Zählung etabliert, welche auch von dem System unterstützt wird. Mit der Eingabe der URL [...]/ceec-cgi/kleioc/0010/exec/pagemed/"|kn28-0199_7r" kommt der Benutzer zu demselben Ergebnis wie mit der oben genannten URL. Der einzige Unterschied ist im Objektnamen zu finden. Nach einem einleitenden Vertikalstrich tritt an Stelle der Nummer im Digitalisierungsprozess (014), der kanonische Name (7r). Leider ist diese Version der Referenzierung nicht die standardmäßige Form der benutzen Verweise. Alle im Online-Angebot auftretenden Verweise auf Bildseiten sind in der ersten Form gehalten, die den kanonischen Namen nicht enthält. Die zweite Form, die, wie das Beispiel zeigt, einwandfrei funktioniert, wurde der Literatur entnommen[25].

Der zweite Bestandteil des Angebotes beinhaltet die Beschreibungsdaten zu einer Handschrift. Eine solche Seite wird durch die URL [...]/ceec-cgi/kleioc/0010/exec/katm/"kn28-0199" erreicht. Bis zu der Funktionsangabe unterscheidet sie sich nicht von der URL, die für die Bildanzeige nötig ist. Das Angebot der CEEC umfasst drei Arten der Meta-Angaben, die sich u.a. in ihrer Ausführlichkeit unterscheiden. Mit katm wird die mittlere Größe bezeichnet. Die fehlende Angabe der Handschrift, die nach demselben Muster wie bei den Handschriftenseiten aufgebaut ist, folgt auf die Funktionsangabe.

Die Gleichförmigkeit der URLs unterstützt sehr gut die intuitive Benutzung des Systems unabhängig von der Navigation. Vorausgesetzt der Benutzer kennt die entsprechenden Parameter (z.B. pagemed für eine Bildseite mittlerer Auflösung, oder katm für die ausführlicheren Beschreibungsdaten), dann ist es nicht schwierig, zwischen den verschiedenen Funktionen wie Bildseite und Beschreibung ei-

[24] Der WWW-Browser Firefox beispielsweise schlägt als Speichername "kn28-0199_014.jpg" .html vor, der Browser Opera kapituliert ganz und schlägt nur .html.html vor.
[25] Vgl. Thaller, Handschriftenbibliothek, S. 38.

ner Handschrift hin und her zu schalten. Die intuitive Benutzung beschränkt sich dabei nicht nur auf eine Handschrift, sondern gelingt auch für die weiteren Digitalisate des Angebotes. Es müssen nur entsprechende, schon etablierte und bekannte Signaturen der Handschriften ausgetauscht werden.

Deutsche Bibelgesellschaft

Die Deutsche Bibelgesellschaft gibt im Auftrag der Evangelischen Kirche in Deutschland u.a. die Lutherbibel heraus. Neben den Versionen in Buchform existiert auch ein Online-Angebot, das den Text der verschiedenen Bibeln zugänglich macht[26]. Neben der Lutherbibel kann auch noch auf den Text der „Gute Nachricht Bibel“ zugegriffen werden.

Wenn man den vorhandenen Verweisen der Website folgt, ergibt sich folgende URL für eine Bibelstelle: [...]/channel.php?channel=36&SELECT=gnb&INPUTREF=1003000. Die Ausgabe wird durch das Programm channel.php erzeugt, welches über die GET-Schnittstelle mehrere Parameter erwartet. Die Parameter channel und SELECT beziehen sich beide auf die Version der Bibel (hier „Gute Nachricht Bibel“), mit INPUTREF wird die eigentliche Stelle bezeichnet (hier 1Mose 3). Wie schon bei den anderen Angeboten beobachtet, stehen auch hier die technischen Rahmenbedingungen bei der Gestaltung der URL im Vordergrund.

Alternativ kann mittels einer Suchmaske eine Stelle aufgeschlagen werden. Da nicht erwartet werden kann, dass Benutzer die internen Parameter kennen, ist es dort möglich, das übliche Referenzierungssystem der Bibel zu verwenden[27]. Nach Eingabe der entsprechenden Stelle ändert sich auch die URL. Wiederum über die GET-Schnittstelle ist es möglich, auf eine weitere Form der Referenzierung zurückzugreifen. Diese alternative Form lautet [...]/channel.php?channel=36&SELECT=gnb&INPUT=1+Mose+3. Die ersten Bestandteile unterscheiden sich nicht von der ersten vorgestellten Version. Doch an Stelle von INPUTREF tritt nun der Parameter INPUT. Der Wert von INPUT beinhaltet den kanonischen Namen.

[26] Vgl. http:/www.dbg.de.

[27] Die Referenzierungssysteme der Bibel sind je nach Konfession, Ausgabe und Sprache unterschiedlich. So wird das erste Buch Mose in einer deutschen katholischen Ausgabe Genesis genannt (Abkürzung Gen). Das Online-Angebot der Bibelgesellschaft kann aber mit beiden deutschen Systemen (1Mose 3 bzw. Gen 3) umgehen.

Die auftretenden Leerzeichen wurden durch das Pluszeichen maskiert. Mit dieser URL, die nur über den Umweg der Suchmaske erreicht werden kann, ist es dann auch möglich, intuitiv auf das Angebot zuzugreifen.

2.2.3. Fazit

Als Ergebnis der Beobachtungen ist festzuhalten, dass zum größten Teil die technischen Rahmenbedingungen, unter denen ein Angebot existiert, die Form der Objektreferenzierung diktiert. Dies ist unnötig. Es hat sich noch nicht die Erkenntnis durchgesetzt, dass mit den URLs Objekte in zitierbare Form gebracht werden können. Manche Projekte haben die Notwendigkeit der Übernahme der etablierten Zitierweise erkannt und setzen sie ein. Doch keines der untersuchten Angebote schafft es, eine URL zu bieten, bei der die kanonische Zitierweise im Vordergrund und die technischen Bedingungen im Hintergrund stehen. Oft liegt auch der Verdacht nahe, dass den praktischen Gesichtspunkten bei der Erstellung des Online-Angebotes der Vorrang gegenüber einer systematischen Lösung gegeben wurde.

Bis auf das CEEC-Projekt beschränken sich alle untersuchten Angebote darauf, nur eine Darstellungsfunktion zu unterstützen, nämlich die Anzeige eines Objektes. Zusätzlich dazu werden bei den Kölner Kodizes Meta-Informationen angeboten. Obwohl eine große Übereinstimmung bei der eigentlichen Funktion herrscht, implementieren alle Angebote eine eigene Sicht, ein eigenes Schema, wie diese Funktion ausgeführt wird. Dies ist unnötig und führt dazu, dass die Interoperabilität zwischen den Angeboten erschwert wird.

Bei den verwendeten Schnittstellen zum HTTP-Server vertrauen alle auf das Common Gateway Interface und dabei genauer auf die GET- (Perseus, Regesta Textseite, Bibelgesellschaft) und die PATH_INFO-Variante (CEEC, Regesta Buchseite). Dies ist nicht weiter verwunderlich, da dies der einfachste Weg ist, einzelne Objekte direkt durch eine URL anzusprechen. Dass die PATH_INFO-Variante klare Vorteile hat, ist offensichtlich. So beinhaltet eine URL der CEEC eine kanonische Referenz, ein Ausgabeformat und beinahe einen sprechenden Dateinamen. All diese Eigenschaften in einer GET-basierten URL unterbringen zu wollen, wäre zwar nicht unmöglich, doch darunter würde die Übersichtlich-

keit stark leiden. So müsste für jeden Wert auch ein Parametername eingeführt werden, der dann noch mit dem Ampersand getrennt werden müsste - von dem Gleichheitszeichen zwischen Parametername und Wert ganz zu schweigen. Bei der intuitiven Benutzung haben die meisten Projekte Schwierigkeiten, am ehesten ermöglicht dies noch das CEEC-Projekt. Aber selbst hier besteht der erste Teil der URL unnötiger Weise aus technisch statt inhaltlich determinierten Bestandteilen.

2.3. Die Zitierweisen als semantisches Netz

2.3.1. Semantische Netze - Topic Maps

Die Nutzung des Wissens um die geisteswissenschaftlichen Zitierweisen im informationstechnischen Umfeld erfolgt generell in zwei Schritten. Zunächst muss das Wissen kodiert werden, dann kann eine Aufbereitung der Daten erfolgen, sodass ein sinnvoller Umgang mit ihnen möglich ist. Die Ansprüche an die Kodierung sind vergleichbar mit den Ansprüchen, die auch bei anderen Konzepten der geisteswissenschaftlichen Informationstechnologien angelegt werden. Zu nennen sind hier hauptsächlich Plattformunabhängigkeit, Langlebigkeit und die Möglichkeit, die zuweilen uneindeutige Natur der Daten berücksichtigen zu können. Diese Ansprüche implizieren, dass mit den kodierten Daten an sich nicht viel anzufangen ist, außer das Wissen und das Konzept, welches hinter dem Wissen steht, zu speichern. Dies ist auch gut so, denn die Trennung von Speicherung und Aufbereitung bzw. Präsentation der Daten hat Vorteile. Durch die Trennung wird die Dauerhaftigkeit der Daten vergrößert, da Präsentationsmethoden meist kurzlebiger sind als die Konzepte selbst und das Wissen um diese Konzepte.

Eine Möglichkeit, die geisteswissenschaftlichen Zitierweisen in der Informationsverarbeitung zu realisieren, ist die Verwendung eines semantischen Netzes. Ein semantisches Netz ist ein formales Modell der Informationstechnologie, um das Wissen und die Beziehungen, die zwischen den Elementen des Wissens bestehen, zu repräsentieren. Dieser allgemeinen Ansatz ist schon in verschiedenen Modellen konkretisiert worden, darunter auch Topic Maps[28].

[28] Weitere Modelle sind beispielsweise das Resource Description Framework (RDF) und die Web Ontology Language (OWL).

Topic Maps existieren in zwei formalen Beschreibungen. Im Herbst 1999 verabschiedete die International Organization for Standardization (ISO) den Standard Nummer 13250[29]. Ein gutes Jahr später im Dezember 2000 erfolgte die Freigabe der XML Topic Maps (XTM) durch die TopicMaps.Org Authoring Group[30]. XTM fußt auf dem ISO-Standard. Der hauptsächliche Unterschied liegt in der verwendeten Markup-Metasprache begründet. Während ISO 13250 die Standard Generalized Markup Language (SGML) verwendet, ist XTM, wie der Name schon andeutet, in der Extensible Markup Language (XML) formuliert. Beide Möglichkeiten genügen den oben genannten Ansprüchen an die Kodierung der Daten, und zwar hauptsächlich durch die Verwendung von Auszeichnungssprachen.

In dieser Arbeit wird XTM verwendet, um die Zitierweisen der Geisteswissenschaften abzubilden. Die Abbildung erfolgt durch Topics (Themen), Occurences (Vorkommensangaben) und Associations (Assoziationen)[31]. Des weiteren sind Subject-Identities (Subjekt-Identitäten) und Scopes (Gültigkeitsbereiche) von Bedeutung.

Topics, Typen und Occurences

Topics bilden das Grundgerüst für eine Topic Map. Sie stellen die kleinsten und elementarsten Bausteine einer Topic Map dar. Die Frage, was genau zu einem Topic wird, hängt stark von der Domäne ab, deren Wissen modelliert wird. In dem konkreten Fall der später vorgestellten Kanon XTM stellen beispielsweise die Werke, die durch eine Referenz angesprochen werden sollen, Topics dar.

Daneben werden durch Themen Typen deklariert. Typen werden gebraucht, um an verschiedenen Stellen einer Topic Map Dinge aus der wirklichen Welt einem Typ zuzuweisen. Typisierte Themen stehen dann nicht mehr für sich allein, son-

[29] Vgl. ISO/IEC 13250:2000 Topic Maps. Information Technology - Document Description and Markup Languages, hg. v. Michel Biezunski, Martin Bryan, Steven R. Newcomb.

[30] Vgl. XML Topic Maps (XTM) 1.0, http://www.topicmaps.org/xtm/index.html.

[31] In dieser Arbeit werden die englischen und deutschen Bezeichnungen parallel verwendet. In Klammern wird beim ersten Auftreten des englischen Begriffes eine deutschen Übertragung angegeben. Da die deutschen Bezeichnungen nicht einheitlich benutzt werden, wurden sie hier aus der Übersicht von Eva Lenz entnommen, um die Begriffsvielfalt nicht unnötig zu erweitern. Für die Übersicht vgl. http://coli.lili.uni-bielefeld.de/~eva/Verschiedenes/topic_map_terminologie.html.

dern drücken durch ihren Typ ein Konzept oder eine Hierarchie aus. Wenn es die Möglichkeit der Typisierung nicht gäbe, müssten die Hierarchien umständlich (und fehlerträchtig) durch IDs umschrieben werden.

Topics besitzen immer eine ID, die sie mit keinem anderen Topic teilen. Daneben besitzen sie sinnvollerweise einen sprechenden Namen. In der Syntax von XTM wird ein Topic mit dem `topic`-Element eingeleitet.

```
<topic id="lma">
  <baseName>
     <baseNameString>Lexikon des Mittelalters</baseNameString>
  </baseName>
</topic>
```

Listing 1: Ein einfaches Topic

Listing 1[32] enthält ein einfaches, nicht typisiertes Thema mit einer ID und einem Basisnamen in der Notation von XTM. Der Basisname wird durch die Elemente `baseName` und `baseNameString` gekennzeichnet. Ein Designziel bei XTM war es, soweit wie möglich auf Attribute zu verzichten. Deswegen kommen hier zwei Elemente zum Einsatz, obwohl auch ein Element mit entsprechendem Attribut ausreichend gewesen wäre. Eine Occurence liefert eine nähere Erläuterung zu einem bestimmten Aspekt eines Topics. Ein Topic kann beliebig viele von ihnen enthalten. In den meisten Fällen stellen die Occurences Verbindungen zu externen Ressourcen dar. So kann beispielsweise beim Thema Goethe das Vorkommen einer Biografie angegeben werden. Bei den kanonischen Zitierweisen kommen diese externen Verweise nur selten vor. Statt dessen überwiegt hier die direkte Kodierung durch Wertangaben. Auch Occurences können typisiert sein, es wird dann von einem Occurence-Type (Typ der Vorkommensrolle) gesprochen.

[32] Die Beispiele greifen schon auf folgende Kapitel vor, da sie aus der Kanon XTM entnommen wurden. Sie sollen in diesem Kapitel aber nur die grundlegenden Mechanismen einer Topic Map verdeutlichen. Auf die eigentlichen Konzepte, auf denen die Kanon XTM beruht, wird erst im nächsten Kapitel näher eingegangen.

```
<topic id="lma">
  <baseName>
     <baseNameString>Lexikon des Mittelalters</baseNameString>
  </baseName>
  <occurence>
    <resourceData>lma</resourceData>
  </occurence>
</topic>
```

Listing 2: Ein Thema mit einer Occurence

In Listing 2 wird das vorherige Thema durch eine Vorkommensangabe erweitert, die den Wert lma besitzt. Die Angabe wird durch das `occurence`-Element eingeleitet. Hier folgt darauf das `resourceData`-Element, um den Wert der Occurence aufzunehmen. Wäre der Wert ein Verweis auf eine andere Resource, würde dies durch das `resourceRef`-Element ausgedrückt.

In der obigen Form wird aber nicht deutlich, was es mit dieser Angabe auf sich hat und wofür der Wert lma überhaupt steht. Dasselbe gilt für das Thema, bei dem es wünschenswert ist, eine nähere Bestimmung angeben zu können. Dieser Missstand kann durch die Typisierung des Topics und der Occurence aufgehoben werden.

```
<topic id="lma">
  <instanceOf>
    <topicRef xlink:href="#lexikon"/>
  </instanceOf>
  <baseName>
    <baseNameString>Lexikon des Mittelalters</baseNameString>
  </baseName>
  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#refName"/>
    </instanceOf>
    <resourceData>lma</resourceData>
  </occurrence>
</topic>
```

Listing 3: Ein typisiertes Thema

Wie in Listing 3 zu sehen ist, wird das `instanceOf`-Element benötigt, um die Bestandteile einer Topic Map einem bestimmten Konzept zuzuweisen. So erfahren wir hier, dass das Topic mit der ID lma, eine Instanz des Topics mit der ID lexikon ist, während die Occurence, die den Wert lma besitzt, eine Instanz des Topics mit der ID refName ist. Die Bezugnahme auf andere Topics wird durch das `topicRef`-Element erreicht. Als Ziel des Bezuges werden nur Verweise zugelassen, die einem „Simple Link“ der XLink-Spezifikation entsprechen[33]. Sie müssen nicht zwingend in demselben Dokument definiert sein. Der Verweis kann sich auch auf andere erreichbare Ressourcen beziehen.

Namen, Gültigkeitsbereiche und Subjekt-Identitäten

Die Namen eines Topics können in verschiedenen Gültigkeitsbereichen definiert sein. Dies wird verwendet, um unterschiedliche Namen für dasselbe Topic aufzunehmen. Das klassische Beispiel dafür sind Namen in unterschiedlichen Sprachen, die mit Hilfe von Scopes nebeneinander stehen können. Ein Gültigkeitsbereich wird mit dem `scope`-Element definiert und kann auch typisiert sein. Genau genommen besitzt jeder eingeführte Name einen Scope, nämlich den standardmäßigen, nicht näher qualifizierten Bereich. Das Topic, welches in Listing 4 definiert ist, besitzt drei Namen. Den allgemeinen („Verweis auf ein sekundäres Objekt“) und zwei spezielle Namen mit einem Gültigkeitsbereich („ist Zielpunkt des Verweises von“ und „ist Ausgangspunkt des Verweises zu“).

```
<topic id="verweisSekundaer">
  <baseName>
    <baseNameString>
      Verweis auf ein sekundäres Objekt
    </baseNameString>
  </baseName>
  <baseName>
    <scope>
      <topicRef xlink:href="#verweisZiel"/>
    </scope>
```

[33] Vgl. dazu die Ausführungen ab ’Simple Links’ (Abschnitt 5.2) in der Spezifikation der XML Linking Language (XLink), http://www.w3.org/TR/xlink/#simple-links.

```
    <baseNameString>
      ist Zielpunkt des Verweises von
    </baseNameString>
  </baseName>
  <baseName>
    <scope>
      <topicRef xlink:href="#verweisAusgang"/>
    </scope>
    <baseNameString>
      ist Ausgangspunkt des Verweises zu
    </baseNameString>
  </baseName>
</topic>
```

Listing 4: Gültigkeitsbereiche

Namen haben die Eigenschaft, nicht immer eindeutig zu sein. Es können viele Objekte mit dem Namen „Lexikon des Mittelalter“ existieren, jedoch wird immer nur ein spezielles Objekt gemeint sein. Deswegen sieht XTM vor, Subjekt-Identitäten zu vergeben. Subjekt-Identitäten sind Verweise auf eine externe Ressource, die das Subjekt näher beschreiben und eindeutig bestimmen. Sie werden durch das Element Paar `subjectIdentity` und `subjectIndicatorRef` gekennzeichnet. Letzteres Element nimmt den Verweis auf die externe Ressource auf. Wie auch schon beim `topicRef`-Element gesehen, wird dazu ein einfacher Verweis der XLink-Spezifikation verwendet.

Im Bereich der geisteswissenschaftlichen Literatur existiert leider kein Online-Angebot, das umfassend veröffentlichte Werke verzeichnen würde. Die Zuständigkeit für ein solches Angebot liegt bei Der Deutschen Bibliothek. Dort existiert auch schon eine Möglichkeit, Subjekte näher zu bestimmen. Listing 5 enthält eine Subjekt-Identität des Lexikon des Mittelalters in Form eines Verweises auf den Online-Katalog Der Deutschen Bibliothek. Bei der hier verwendeten Identifikationsnummer (PPN) besteht aber die Gefahr, dass sie in Einzelfällen Änderungen unterworfen sein kann[34]. Deswegen steht sie streng genommen nicht für eine

[34] Persönliche E-Mail von Matthias Templin, Informationstechnik, Die Deutsche Bibliothek, vom 7. Juni 2005.

nachhaltige Lösung zur Verfügung. Abgesehen davon ist nicht jedes Werk im Katalog verzeichnet und besitzt eine solche Nummer. Zudem ist diese Schnittstelle in erster Linie nicht dazu entwickelt worden, Werke eindeutig zu bestimmen. Doch bis Die Deutsche Bibliothek eine solche Schnittstelle anbieten kann, muss man sich mit dieser Lösung behelfen. In der Kanon XTM wird sie dort, wo es möglich ist, schon verwendet.

```
<topic id="lma">
  <instanceOf>
    <topicRef xlink:href="#lexikon"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef xlink:href="http://dispatch.opac.ddb.de/
      DB=4.1/PPN?PPN=550564438">
    </subjectIndicatorRef>
  </subjectIdentity>
  <baseName>
    <baseNameString>Lexikon des Mittelalters</baseNameString>
  </baseName>
</topic>
```

Listing 5: Namen und Identitäten

Assoziationen

Mit Assoziationen werden Beziehungen zwischen einzelnen Topics gespeichert. Eine Assoziation an sich verhält sich neutral gegenüber der Art der Beziehung. Grundsätzlich können beliebig viele Topics an einer Assoziation teilnehmen. Sie bekleiden dann eine Rolle innerhalb der Assoziation.

```
<association id="lma_enthaelt_urkunde>
  <member> <topicRef xlink:href="#lma"/> </member>
  <member> <topicRef xlink:href="#lma-urkunde"/> </member>
</association>
```

Listing 6: Eine einfache Assoziation

Listing 6 definiert eine einfache Assoziation. Es wird eine Beziehung zwischen zwei Themen (lma und lma-urkunde) aufgebaut. Eingeleitet wird sie mit dem `association`-Element. Die Teilnehmer der Beziehung finden sich in den `member`-Elementen. Wie zu sehen ist, bleibt die Art der Beziehung völlig offen. Allein die verwendete ID der Assoziation enthält einen Hinweis. Auch über die Natur der beteiligten Rollen wird keine Aussage gemacht. Wie schon bei Topics und bei den Occurences gesehen, kann dies durch Typisierung geändert werden.

```
<association>
  <instanceOf>
    <topicRef xlink:href="#enthaeltObjekt"/>
  </instanceOf>
  <member>
    <roleSpec>
      <topicRef xlink:href="#liefertObjekt1"/>
    </roleSpec>
    <topicRef xlink:href="#lma"/>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#istUnterObjekt1"/>
    </roleSpec>
    <topicRef xlink:href="#lma-urkunde"/>
  </member>
</association>
```

Listing 7: Eine vollständige Assoziation

Die vollständige Assoziation in Listing 7 ist typisiert. Jede Assoziation kann nur Instanz eines Topics sein, was durch das `instanceOf`-Element ausgedrückt wird. Auch die Rollen wurden spezifiziert, und zwar mit Hilfe des `roleSpec`-Elementes. Die eigentlichen Verweise werden wie üblich mit dem `topicRef`-Element realisiert. Durch die Typisierung wird die Art und Weise der Beziehung deutlich. In natürliche Sprache übersetzt, drückt diese Assoziation den Umstand aus, dass grundsätzlich eine Aussage über den Inhalt von Objekten getroffen wird, dass das Thema lma ein Objekt liefert und dass das Thema lma-urkunde ein entsprechendes Unterobjekt ist.

Linear Topic Map Notation

Die manuelle Notation von Topic Maps kann eine aufwändige und fehleranfällige Sache sein. Durch die Verwendung von Markup-Sprachen wird der Aufwand noch erhöht, da diese Systeme viele Zeichen benötigen, um die eigentlichen Daten zu kodieren. Bei Topic Maps kommt noch hinzu, dass sich das Verhältnis von eigentlicher Information und dem syntaktischen Überbau, den die Standards vorgeben, stark in Richtung Überbau ausrichtet. Ein menschlicher Bearbeiter ist somit mehr damit beschäftigt, Elemente und Attribute zu notieren, als sich auf die eigentlichen Daten zu konzentrieren. Dieses Missverhältnis kann dadurch erklärt werden, dass die manuelle Notation eigentlich nicht vorgesehen ist. In größeren Projekten sollte deswegen eine spezialisierte Eingabe-Software zum Einsatz kommen, die den Bearbeiter entlastet.

Im Rahmen dieser Arbeit wurde keine eigene Eingabe-Software eingesetzt. Statt dessen wurde hier auf die Linear Topic Map Notation (LTM) zurückgegriffen. Die LTM wurde von der Firma Ontopia entwickelt und liegt als Spezifikation vor[35]. Sie versteht sich als einfache und kompakte Notation für Topic Maps und kommt ohne den Überbau der Markup-Sprachen aus, da sie sich bei der Notation auf das Wesentliche konzentriert. Bei einem Thema bedeutet dies, dass die Angabe eines Identifikationsnamens (ID) ausreicht, um ein Thema zu notieren.

```
[einThema]
```

Dies wäre die knappste Form, um ein Thema mit der ID `einThema` zu kreieren. Um aus dieser Form eine gültige Topic Map zu erstellen, ist noch eine Übertragung nötig, die zwischen den Notationsformen vermitteln kann.

```
[lma : lexikon = "Lexikon des Mittelalters"
 @"http://dispatch.opac.ddb.de/DB=4.1/PPN?PPN=550564438"]
{lma, refName, [[lma]]}
```

Dies definiert das Thema lma, welches eine Instanz des Themas lexikon ist. Es folgt die Angabe des Basisnamens, danach wird ein Subjekt-Indikator angegeben. In geschweiften Klammern erscheint dann eine Vorkommensangabe, die dem

[35] Vgl. http://www.ontopia.net/download/ltm.html.

Thema lma zugeordnet ist und eine Instanz des Themas refName ist. Der Wert der Ausprägung wird in doppelten eckigen Klammern angegeben, um ihn von einem Verweis auf eine andere Ressource zu unterscheiden. Der Unterschied zu der eigentlichen Notation wird deutlich, wenn man sich noch einmal vor Augen hält, wie die Entsprechung dieser vier Zeilen in XML aussieht: Listing 8.

```
<topic id="lma">
  <instanceOf>
    <topicRef xlink:href="#lexikon"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef xlink:href="http://dispatch.opac.ddb.de/
      DB=4.1/PPN?PPN=550564438">
    </subjectIndicatorRef>
  </subjectIdentity>
  <baseName>
    <baseNameString>Lexikon des Mittelalters</baseNameString>
  </baseName>
  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#refName"/>
    </instanceOf>
    <resourceData>lma</resourceData>
  </occurrence>
</topic>
```

Listing 8: Umwandlung eines Themas

Assoziationen werden in der LTM notiert, indem der Assoziationstyp angegeben wird und danach in runden Klammern die jeweiligen „Mitspieler“ (getrennt durch ein Komma). Die Mitspieler können wiederum typisiert sein.

```
enthaeltObjekt(lma : liefertObjekt1,
               lma-urkunde : istUnterObjekt1)
```

In Listing 9 wird erneut die Kompaktheit dieser Notation gegenüber der eigentlichen XML-basierten deutlich.

```
<association>
  <instanceOf>
    <topicRef xlink:href="#enthaeltObjekt"/>
  </instanceOf>
  <member>
    <roleSpec>
      <topicRef xlink:href="#liefertObjekt1"/>
    </roleSpec>
    <topicRef xlink:href="#lma"/>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#istUnterObjekt1"/>
    </roleSpec>
    <topicRef xlink:href="#lma-urkunde"/>
  </member>
</association>
```

Listing 9: Umwandlung einer Assoziation

2.3.2. Die Zitierweisen als XML Topic Map - Kanon XTM

In der Kanon XTM wird das Wissen der Domäne der kanonischen Zitierweisen der Geisteswissenschaften gespeichert. Dabei geht es zum einen um die Daten der Werke, die mit Referenzen angesprochen werden, und zum anderen um die Beziehungen, die zwischen den Werken bestehen. Die erste Aufgabe wird durch die entsprechenden Vorkommens-Typen erfüllt. Bei ihnen wurde Wert darauf gelegt, dass sich möglichst einfach der komplette Satz der Referenzen aufnehmen lässt. So besteht eine Edition von Urkunden aus beispielsweise 500 einzelnen Objekten. Die einzelnen Objekte können im Netz natürlich einzeln aufgenommen werden, doch es bestehen auch Vereinfachungen, die es erlauben, einen Bereich anzugeben. Anhand dieses Bereiches kann dann entschieden werden, ob eine bestimmte Referenz gültig ist oder nicht. Falls etwas Besonderes über eine der 500 Urkunden zu vermerken ist, kann diese zusätzlich einzeln aufgenommen werden. Ein weiteres Beispiel sind die Seiten einer Handschrift. Jede einzelne Seite für sich

wird durch eine Referenz angesprochen (17r), aber durch die Vereinfachung einer Umfangsangabe wird der Bearbeiter des Netzes von der Bürde entlastet, jede einzelne Seite speziell aufzunehmen.

Bei den Beziehungen zwischen den Werken gibt es zwei verschiedene Arten: ein Werk enthält ein anderes und ein Werk verweist auf ein anderes. Die Enthält-Beziehung hätte auch durch Vererbung der Themen realisiert werden können. Die dann entstehende Kopplung wurde aber als zu eng erachtet, deswegen ist diese Art von Beziehung als Assoziation konzipiert. Bei den Verweisen auf andere Werke kam nur die Verwendung einer Assoziation in Frage.

Themen

Die folgenden Themen bilden die Grundlage bei der Speicherung der kanonischen Zitierweisen in den Geisteswissenschaften durch die Kanon XTM.

Werk-Typen Die Werk-Typen bilden die Ausgangsbasis für die Topics von Werken, die kanonische Namen enthalten. Die Werke können die entsprechenden Objekte direkt enthalten oder aber das Werk bildet nur eine institutionelle Klammer um die Objekte. Ein Beispiel für den ersten Typ wäre ein Lexikon, das als untergeordnete Objekte Artikel enthält. Für den zweiten Typ käme eine Reihe in Frage, bei der die untergeordneten Objekte eigenständig sind und wiederum Unterobjekte enthalten können. Alle Werk-Typen sind von dem Typ `werk` abgeleitet. Die Topics der Werk-Typen sind im Einzelnen (aufgeführt mit ihren Basisnamen): Verzeichnis, Lexikon, Edition, Bibliotheksbestand, Zeitschrift, Reihe, Textsammlung, Sonstige und als Spezialfall die Bibel.

```
[werk           = "Ein Werk mit kanonischen Referenzen"]
[lexikon : werk = "Lexikon"]
[lma : lexikon  = "Lexikon des Mittelalters"]
```

Das Beispiel gibt die Typ-Hierarchie in der LTM-Notation wider, um das Lexikon des Mittelalters darzustellen.

Objekt-Typen Die Objekt-Typen werden für die Inhalte der Werke verwendet. Je nach Ebene kann zwischen drei verschiedenen Typen gewählt werden. Das folgende Beispiel demonstriert anhand des Artikels „Urkunde“ ein Objekt der ersten Ebene.

```
[lma-urkunde : objekt1 = "Artikel Urkunde"]
```

Assoziations-Typen und Rollen

Hier werden die Themen aufgeführt, die in Assoziationen verwendet werden. Es werden jeweils auch die zugehörigen Rollen angegeben.

enhaeltObjekt Dies ist der grundlegende Assoziations-Typ, um die Beziehungen von einem Werk und dessen Inhalt zu den untergeordneten Werken zu realisieren. Mit Inhalt sind hier die zur Verfügung gestellten Referenzen gemeint. Wie bei den Themen gesehen, kann dies auf verschiedenen Ebenen geschehen. Dieser Typ sagt aber nichts über die Anordnung der Ebenen, was nur über die verwendeten Rollen ausgedrückt wird. Für die verschiedenen Rollen besitzt die `enthaeltObjekt`-Assoziation verschiedene Namen, ausgedrückt durch einen jeweils eigenen Gültigkeitsbereich. So ist für jede sinnvolle Rolle ein Name angegeben. Dies hat nur den Zweck, dass die verarbeitende Software jeweils den passenden Namen angeben kann. Damit ist es möglich, dem Betrachter eine natürlichsprachliche Umschreibung anzubieten. Beispielsweise erscheint bei dem Topic der großen Editions-Reihe der Monumenta Germaniae Historica (MGH) in der Ansicht der enthaltenen Objekte: „Monumenta Germaniae Historica liefert als Objekt1 DH4“. Bei dem Topic DH4 wiederum erscheint „DH4 ist als Objekt1 Teil von Monumenta Germaniae Historica“. Dies ist viel eingängiger und klarer als der allgemeine Name dieser Assoziation: „Objekt-Beziehung“.

Zugehörige Rollen: `liefertObjekt1`, `liefertObjekt2`, `liefertObjekt3`, `istUnterObjekt1`, `istUnterObjekt2` und `istUnterObjekt3`.

verweisObjekt Dieser Typ wird eingesetzt, um einen Verweis auf ein verwandtes Objekt zu klassifizieren. Der Unterschied zu verweisSekundaer besteht darin, dass verweisObjekt auf gleichartige Objekte zielt. So ist es möglich, beispielsweise in einer Briefedition Verweise auf andere (vorherige oder spätere) Briefe zu markieren. Die anderen Briefe sind dabei wieder Objekte der Topic Map, auch wenn sie nicht extra aufgeführt worden sind. Die Verwandtschaft besteht auf der Ebene der Funktion bzw. des Typs des anderen Objektes.

Zugehörige Rollen: `verweisAusgang` und `verweisZiel`.

verweisSekundaer Im Gegensatz zu verweisObjekt wird hier nicht auf ein verwandtes Objekt verwiesen, sondern auf ein Objekt, welches eine andere Funktion hat. In einer Briefedition sollten damit Verweise kodiert werden, die in den Bereich der Sekundärliteratur fallen, also Aufsätze, Monographien usw., und weiterführende Informationen über diesen Brief enthalten. Auch die Ziele der sekundären Verweise könnten ihrerseits wieder Mitglieder der Topic Map sein.

Zugehörige Rollen: `verweisAusgang` und `verweisZiel`

Rollen für enthaeltObjekt Die `liefertObjekt`-Rollen werden für den Objektlieferanten verwendet, während die `istUnterObjekt`-Rollen das enthaltene Objekt kennzeichnen.

Rollen für die Verweise Die beiden Verweis-Rollen typisieren den Ausgang (`verweisAusgang`) und das Ziel (`verweisZiel`) in einer Verweis-Assoziation.

Occurence-Typen

Die Occurence-Typen speichern die eigentlichen Daten der Werke. Die Werte der Daten sind in das Netz mittels des `resourceData`-Elementes zu integrieren. Die einzige Ausnahme ist `istIdentischMit`, bei dem auf ein anderes Thema verwiesen wird. Es bestehen durch die Vorkommensangaben also keine Verweise außerhalb des Netzes.

version Die Versionsangabe der Referenz als Zeichenkette. Sie muss nicht unbedingt eine Jahreszahl repräsentieren, sondern kann einen sprechenden Begriff enthalten. Die verwendete Zeichenkette sollte aber in der Objekt-Hierarchie eindeutig und sinnvoll sein.

bibl Die vollständige bibliographische Angabe des Werkes. Sie wird u.a. dazu verwendet, um nicht eindeutige Referenzen aufzulösen.

refName Der Name der kanonischen Referenz. Mittels der Vorkommensangaben `refNameAlt1,2,3` können noch Alternativen angegeben werden, falls ein Objekt unter mehreren Namen bekannt ist. So besitzt das Lexikon des Mittelalters zwei gängige kanonische Namen: Lma und Lexma.

refCheck Kennzeichnet, dass die Überprüfung einer Referenz nicht durch die Daten der Kanon XTM erfolgen kann. Der einzig gültige Wert „applikation“ steht für eine Überprüfung durch die Web-Applikation. Dies kann erforderlich sein, wenn nicht alle in einem Werk enthaltenen Objekte in der Kanon XTM aufgeführt worden sind und eine einfache Überprüfung mittels der `umfangListe`-Occurences nicht möglich ist.

umfangListe Vereinfachungen die Angabe der gültigen Referenzen eines Werkes betreffend. So kann ein Gesamtumfang der Referenzen angegeben werden (`umfangListeBeginn` und `umfangListeEnde`), aber auch Ausnahmen, die sich aus dieser Angabe ergeben (`umfangListePlus` und `umfangListeMinus`). Bei den Ausnahmen ist bei mehreren Angaben eine Listendarstellung zu wählen, bei der die einzelnen Elemente durch Leerzeichen getrennt werden. Als Beispiel dient wieder eine Edition von Urkunden. So können darin 500 einzelne Urkunden enthalten sein, wobei aber nicht alle Urkunden regelmäßig durchnummeriert worden sind, sondern einzelne Nummern fehlen, bzw. durch ein zusätzliches Ordnungssystem (mittels a und b) neue Nummern hinzugekommen sind. So kann sich folgendes Bild ergeben: `umfangListeBeginn=1`, `umfangListeEnde=498`, `umfangListePlus=45a 77a 77b`, `umfangListeMinus=88`.

istIdentischMit Verweis auf ein anderes Objekt, das auf dieselbe Resource zielt, aber eine grundsätzlich andere Referenz besitzt. Dies kommt beispielsweise bei Handschriften vor, die zu verschiedenen Zeiten zu verschiedenen Sammlungen zugeordnet waren und dadurch unterschiedliche Namen erhalten haben, die sich etablieren konnten.

Möglichkeiten für die Kanon XTM

Die Kanon XTM beinhaltet verschiedene Potentiale. Zum einen speichert sie das Wissen der kanonischen Zitierweisen. Dieses Wissen kann dazu verwendet werden, in dem später vorgestellten Adressierungsschema gültige von ungültigen URLs zu unterscheiden oder aber mit den Daten und den enthaltenen kanonischen Namen der aufgenommenen Werke Listen zu generieren. Durch die Möglichkeit des Mergings von Topic Maps können in allen Fachbereichen selbständige Instanzen einer Kanon XTM erstellt werden. Diese Instanzen könnten dann in eine große Wissensbasis zusammengeführt werden. Mit dieser Wissensbasis wäre es zudem möglich, einen Resolving-Mechanismus für kanonische Namen in den Geisteswissenschaften aufzubauen, der allgemein Namen validiert. Der Resolving-Mechanismus könnte so weit ausgebaut werden, dass über die Validierung hinaus eine Weiterleitung zum entsprechende Projekt angeboten werden könnte, falls das Objekt digital vorliegt.

Die gespeicherten Beziehungen der Objekte untereinander könnten Grundlage weiterführender Forschungen sein. Besonders die Modellierung der Verweise innerhalb der Kanon XTM macht dies deutlich. Da die Kanon XTM grundsätzlich zwei verschiedene Arten von Verweisen zur Verfügung stellt, ergeben sich interessante Optionen. Zum einen kann auf die Beziehungen innerhalb von verwandten Objekten besser zugegriffen werden, zum anderen kann mit den Verweisen auf nicht verwandte Objekte die Beziehung zu der vorhandenen und zitierten Literatur aufgezeigt werden. Die Anzeige schließt neuartige Visualisierungsstrategien, die im Bereich der semantischen Netze erprobt werden, ausdrücklich mit ein. Des weiteren wäre die Erstellung eines Zitations-Index der Geisteswissenschaften durch eine statistische Auswertung denkbar.

3. Die nachhaltigen Komponenten digitaler Repositorien

3.1. Vorhandene Lösungen

Eine Herausforderung im Umgang mit dem digitalen Veröffentlichungsmedium ist die Nachhaltigkeit. Die angebotenen Ressourcen und Objekte sollen über lange Zeit hinweg auf dieselbe Art adressiert werden können, damit die Referenzierfähigkeit dauerhaft erhalten bleibt. Um ein Objekt anzusprechen, wird in der Regel eine URL verwendet. Eine URL ist in erster Linie eine Adresse, die einen Speicherort auf einem entfernten Rechner angibt. Diese Komponente ist aber technischen Begebenheiten unterworfen, so auch der „Gefahr“ von Umstrukturierungen, die den Speicherort einer Ressource ändern können. Dies wäre nicht weiter von Bedeutung. Da aber die Adresse (die URL) den einzigen Zugriffspunkt auf die Ressource darstellt, ist deren nachhaltige Auffindbarkeit nicht garantiert: Der Verweis auf ein Objekt kann fehlschlagen und es kommt zu der berüchtigten HTTP-Fehlermeldung 404 („file not found“). Ein weiteres Szenario ist, dass unter einer Adresse ein ganz anderes Objekt erreicht wird, als dies zu einem früheren Zeitpunkt einmal der Fall gewesen ist.

Eine Lösung für das skizzierte Problem ist die Umdeutung der Funktion einer URL. Anstatt einen Speicherort anzugeben, wird ein zusätzlicher und eindeutiger Name eingeführt. Der Name wird im Rahmen eines Resolving-Mechanismus mit dem eigentlichen aktuellen Speicherort verbunden. Ein Benutzer kann über ihn mit der Ressource verbunden werden. Eine weitere Lösungsstrategie ist der vollständige Verzicht auf URLs und die Einführung eines neuen Namensschemas. Auch hier steht am Ende aus Sicht des Benutzers eine Adresse des elektronischen Objektes, die zum heutigen Zeitpunkt meist mit einer URL ausgedrückt wird.

Die im folgenden vorgestellten Strategien, Schemata und Frameworks versuchen alle, eine persistente und damit dauerhafte Referenzierung eines Objektes zu gewährleisten. Jede Lösung verwendet einen anderen Ansatz und ist unterschiedlich komplex aufgebaut. Dies liegt in der Zielsetzung der einzelnen Ansätze begründet. Vorgestellt werden

- das URN-Schema Der Deutschen Bibliothek
- die Persistent URL
- der Archival Resource Key
- das Handle-System der CNRI
- das Digital Object Identifier System.

Neben Funktionsweise und Aufbau der einzelnen Systeme wird außerdem untersucht, ob die Lösungen mit den kanonischen Zitierweisen umgehen können, denn dies würde sie zum Aufbau von digitalen Repositorien in den Geistenwissenschaften qualifizieren.

3.1.1. Das URN-Schema Der Deutschen Bibliothek

An der Nationalbibliothek von Finnland wurde ein URN-Namensbereich für Nationalbibliotheken entwickelt, die National Bibliographic Number (NBN). Der Namensbereich ist im RFC 3188 spezifiziert[36]. Er ist nur für Nationalbibliotheken zugänglich und wird von der Library of Congress[37] vergeben. Der generelle Aufbau lautet:

```
urn:nbn:de:...
```

Die nationale Zugehörigkeit wird durch den zweistelligen ISO-Landeskode ausgedrückt; so steht im verwendeten Beispiel „de“ für Deutschland. Nach der Länderangabe fällt die Zuständigkeit an die jeweilige Nationalbibliothek, im deutschen Fall also an Die Deutsche Bibliothek (DDB). Wegen der Weitergabe der Zuständigkeit können die verschiedenen URNs in den jeweiligen Ländern unterschiedlich aufgebaut sein. Es ist deswegen nicht sinnvoll, das Schema auf allgemeiner Ebene zu diskutieren. Es wird daher nur die deutsche Variante besprochen.

Die URN-Strategie Der Deutschen Bibliothek wurde im EPICUR Projekt erarbeitet[38]. Es wurden die technischen und organisatorischen Rahmenbedingungen

[36] Vgl. http://www.ietf.org/rfc/rfc3188.
[37] Vgl. www.loc.gov/.
[38] Vgl. http://www.persistent-identifier.de/?link=3351, oder als URN: urn:nbn:de:1111-200312 1811.

festgelegt, unter denen URNs vergeben, verwaltet und aufgelöst werden können. Für jedes Objekt, das über eine Adresse wie eine URL verfügt, kann eine URN vergeben werden. Der Hauptaugenmerk liegt auf einzelnen Monographien, wie etwa online verfügbare Dissertationen. Ein weiterer Aspekt betrifft die langfristige Archivierung der Objekte, die gewährleistet sein muss, um eine URN zu erhalten. Dabei ist es unerheblich, an welchem Ort die Archivierung stattfindet. Die Deutsche Bibliothek archiviert gegenwärtig Online-Hochschulschriften und Online-Publikationen von Verlagen und verlegenden Stellen, zukünftig sollen Netzpublikationen hinzukommen. Soll ein Objekt außerhalb der DDB archiviert werden, muss der Server zertifiziert sein, z.B. durch DINI[39].

Das folgende Beispiel soll den Aufbau einer URN verdeutlichen:

```
urn:nbn:de:gbv:089-3321752945
```

Die ersten Bestandteile wurden schon vorgestellt und weisen auf die deutsche Sektion im Namensbereich der Nationalbibliotheken (`urn:nbn:de:`). Für eine URN aus dem Verantwortungsbereich eines Bibliotheksverbundes folgt dann die offizielle Sigle des Verbundes (`gbv`) und die Sigle der Bibliothek (`089`). Nach dem Spiegelstrich folgt die Produktionsnummer (`332175294`), die auf das eigentliche Objekt verweist. Den Abschluss bildet eine Prüfziffer (`5`), die die Richtigkeit der gesamten URN garantieren soll[40].

Bei einer URN für eine Institution ändert sich der generelle Aufbau nicht. Der einzige Unterschied ist, dass an Stelle des Siglenpaars Verbund / Bibliothek eine eindeutige vierstellige Ziffer (`1111`) tritt, die der ausgebenden Institution zugeordnet ist.

```
urn:nbn:de:1111-2003121811
```

Die Verwaltung der einzelnen URNs kann auf mehreren Wegen erfolgen. Einzelne Meldungen können in Web-Formular eingetragen werde, während für umfangreichere Datensätze die Möglichkeit besteht, eine OAI-Schnittstelle[41] zu verwenden.

---

[39] Deutsche Initiative für Netzwerkinformationen, http://www.dini.de/.

[40] Der dazu gehörige Algorithmus kann unter http://www.persistent-identifier.de/?link=316 eingesehen werden.

[41] Vgl. http://www.openarchives.org/.

Dazu existiert ein Transferformat (EPICUR-XML-Schema), das zur automatisierten Verwaltung genutzt werden kann, die immer zentral über Die Deutsche Bibliothek erfolgt. Die Aktualisierungsgeschwindigkeit soll zukünftig auf Echtzeit-Niveau angehoben werden, im Moment werden die Neuerungen einmal am Tag sichtbar gemacht.

Für das Resolving stellt Die Deutsche Bibliothek einen zentralen Auflösungsdienst zur Verfügung. Dieser nimmt die URN entgegen und leitet den Benutzer per HTTP-Redirect zur entsprechenden URL. Der Dienst ist als CGI-Programm ausgeführt und kann über die GET-Schnittstelle die URN entgegennehmen[42]. Daneben kann eine URN auch direkt in die Adresszeile eines Browsers eingegeben werden, wenn vorher ein Zusatzprogramm installiert wurde.

Das URN-Schema Der Deutschen Bibliothek ist stark auf die Bedürfnisse von Bibliotheken ausgerichtet, was kaum verwunderlich ist. Als einziger Ort für die mögliche Integration der kanonischen Zitierweisen eignet sich die Produktionsnummer. Diese muss nicht nur aus Ziffern bestehen, doch der Zeichensatz ist durch den Prüfalgorhitmus eingeschränkt, so dass eine Referenz nicht gut strukturiert werden kann[43]. Verschiedene Ausgabeformate von einzelnen Objekten sind mit diesem Schema zwar möglich, aber nicht Teil der URN, sondern werden über den Resolving-Mechanismus realisiert.

Die URNs sind aufgrund ihrer Verwaltung und Auflösung stark an Die Deutsche Bibliothek gekoppelt, was nicht unproblematisch ist. Neben dem - eher unwahrscheinlichen - generellen Risiko, dass der Dienst in Zukunft eingestellt werden könnte und damit alle URNs ihre Gültigkeit verlieren würden, besteht die Gefahr bzw. die Einschränkung, dass die Vorgaben Der Deutschen Bibliothek nicht mit den Bedürfnissen der Wissenschaft (und weiterer Nutzergruppen) übereinstimmen könnten. Durch die starke Ausrichtung auf eine Dokumentengruppe (z.B. Online-Monographien) ist eine Unvereinbarkeit mit den Notwendigkeiten der kanonischen Zitierweisen absehbar. Danben wird durch die nationale Veren-

[42] Die Auflösung des ersten Beispiels geschieht über die folgende URL: http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:gbv:089-3321752945.

[43] So dürfen keine Schrägstriche (/), Unterstriche und Punkte (.) vorkommen. Die Groß- und Kleinschreibung wird nicht unterschieden.

gung nicht jedes Werk in den Zuständigkeitsbereich Der Deutschen Bibliothek fallen.

3.1.2. Persistent URL

Die Persistent URL (PURL) wurde 1996 am Online Computer Library Center entwickelt[44]. Eine PURL gleicht in der Funktionsweise und dem Aufbau einer URL, doch anstatt auf direktem Wege eine Ressource anzusprechen, wird ein Resolving-Mechanismus zwischengeschaltet, mit dem die PURL aufgelöst wird. Das Ergebnis des Resolvers ist eine URL, die mittels eines HTTP-Redirect dem Client übergeben wird. Dadurch wird erreicht, dass der Name einer Ressource, ausgedrückt durch die PURL, von der Adresse der Ressource, ausgedrückt durch die zurückgelieferte URL, getrennt wird. Für den Prozess der Auflösung werden nur aktuelle Standard-Internettechniken benutzt. Die Persistenz einer PURL wird dadurch erreicht, dass sie, einmal eingerichtet, für immer im Datenpool der eingeführten Namen verbleibt. Änderungen an dem Ziel einer PURL sind jederzeit möglich, doch die Löschung des Namens ist nicht vorgesehen. Falls das Ziel nicht mehr existieren sollte, gibt das Auflösungssystem dies bekannt und zeigt dem Benutzer dann die Bearbeitungshistorie. Solange der Resolver existiert, solange bleiben auch die Namen gültig.

Eine PURL besteht aus drei Teilen: Protokoll, Adresse des Resolvers und der Name der Ressource.

```
http://purl.oclc.org/OCLC/PURL/FAQ
```

Als Protokoll wird HTTP verwendet, die Ermittlung der Adresse des Resolvers (`purl.oclc.org`) beruht auf dem DNS. Die Software, die für den Aufbau eines Resolvers benötigt wird, ist frei erhältlich und kann auch für die Einrichtung eines eigenen Auflösungsdienstes genutzt werden. Sie enthält neben dem eigentlichen Dienst auch Komponenten, die eine Verwaltung der PURLs erlauben. Der Name der Ressource (`OCLC/PURL/FAQ`) kann frei gewählt werden. Ein Name

[44] Vgl. http://purl.oclc.org/

ist in der Regel immer mit einer Ressource verknüpft[45]. Eine Ausnahme besteht, wie schon erwähnt, in der Löschung des Verweiszieles.

Der Namensbestandteil einer PURL hat die Besonderheit, dass die Groß- und Kleinschreibung nicht, wie sonst bei URLs üblich, unterschieden wird. Weiterhin ist ein Name nur innerhalb eines Resolvers gültig, da es viele verteilte Resolver geben kann, die jeweils eigene Namen verwalten und es somit zu Überschneidungen kommen kann. Ein Name wird in sog. „domains“ unterteilt, die durch den Schrägstrich (/) getrennt werden[46]. Für die Registrierung eines Namens müssen entsprechende Rechte für die Domain oder eine untergeordnete Domain vorhanden sein.

Das Konzept einer PURL besticht durch seine Einfachheit und den möglichen dezentralen Aufbau: „Alles kann, nichts muss.“ Da in dieser Arbeit davon ausgegangen wird, dass eine URL für die Verwendung von kanonischen Referenzen grundsätzlich geeignet ist, gilt dies auch für eine PURL. Die Gliederung in Ebenen („domains“) setzt aber eine starke Hierarchisierung der Referenzen voraus, die nicht immer gegeben sein muss. Die einfache Strukturierung, dass sich genau ein Name und eine Adresse entsprechen, verengt die Möglichkeiten einer PURL auf die reine Objektreferenzierung. Weiterführende Funktionen wie verschiedene Ausgabeformate etc. sind nur schwer zu implementieren.

3.1.3. Archival Resource Key

Der Archival Ressource Key (ARK) wurde von J. Kunze und R. P. C. Rodgers entwickelt und in Form eines Internet-Drafts veröffentlicht[47]. Die Arbeiten daran sind noch nicht abgeschlossen, die letzte Änderung erfolgte im August 2005. ARK ist ein Framework für die dauerhafte Identifizierung von Objekten, mit dem auch Unterobjekte und verschiedene Versionen eines Objektes ausgedrückt werden können. Ein ARK zielt immer auf drei Eigenschaften eines Objektes:

- die Meta-Daten eines Objektes

[45] Das Beispiel wird gegenwärtig durch die URL http://purl.oclc.org/docs/purl_faq.html aufgelöst.

[46] Im Beispiel bezeichnet `OCLC` eine „top level domain“ und `PURL` eine „subdomian“.

[47] Vgl. die Projektseite, http://www.cdlib.org/inside/diglib/ark/. Der Draft ist u.a. über die Adresse http://www.cdlib.org/inside/diglib/ark/arkspec.pdf zu erreichen.

- das Objekt selbst
- eine Angabe, wer sich für die Dauerhaftigkeit verpflichtet hat und in welcher Form („commitment statement“, bzw. „permanence policy“).

Die Verwaltung der ARKs erfolgt durch eine Name Assigning Authority (NAA), die die Erlaubnis hat, eine Nummer auszugeben: die sog. NAAN. Von diesen, im Bibliotheksbereich angesiedelten Ausgabestellen existieren bereits einige[48]. Die Erweiterung um weitere NAAs ist vorgesehen und richtet sich vornehmlich an Nationalbibliotheken, Nationalarchive und Verlage. Den Aufbau eines ARKs soll am folgenden Beispiel verdeutlicht werden:

```
http://ark.cdlib.org/ark:/13030/ft4w10060w
```

Zu Beginn steht die Adresse des Name Mapping Authority Hostport (NMAH), also des Resolvers eines ARKs. Der Resolver ist austauschbar und kann in Zukunft weggelassen werden. Die Voraussetzung dafür ist, dass die Client-Software (z.B. WWW-Browser) das ARK-Schema erkennt und selbständig einen erreichbaren Resolver aufsucht. Dies ist zum heutigen Zeitpunkt nur durch Zusatzprogramme möglich, eine Verankerung über das DNS ist aber angedacht. Dennoch besteht hier die Möglichkeit, einen technisch motivierten Bestandteil des Resolvings theoretisch vernachlässigen zu können und sich ganz auf die eigentliche Referenz zu konzentrieren. Dies impliziert aber auch, dass ein ARK weltweit einmalig und gültig ist.

Die Identifizierung des Objekts beginnt mit der Zeichenkette „ark:/“. Ihr folgt die NAAN, die im Moment aus einer fünfstelligen Zahl besteht. Sollte zukünftig der Adressbereich ausgehen, wird sie auf neun Stellen erweitert. Jede NAAN, die ausgegeben wurde, wird kein zweites Mal verwendet, so dass die Eindeutigkeit garantiert ist. Auf die NAAN folgt der Name der Ressource. Die Namen von Ressourcen werden von den NAAs verwaltet und können beliebig gestaltet werden. Einschränkungen bestehen nur bezüglich der verwendeten Zeichen. So sind die Zeichen „%“, „-“, „.“ und „/“ reserviert und haben jeweils eine spezielle Bedeutung.

[48] Für eine vollständige Liste vgl. http://ark.nlm.nih.gov/etc/natab.

```
ark:/12025/654xz321/s3/f8.05v.tiff
```

Dieser ARK - ohne die Angabe eines Resolvers - wird nach dem eigentlichen Namen (`654xz321`) noch fortgeführt. Es folgen zwei Angaben: ein Unterobjekt (`s3/f8`) und eine Versionsangabe (`05v.tiff`). Unterobjekte werden durch weitere Schrägstriche gekennzeichnet, während die Angaben von Varianten durch Punkte ausgedrückt werden. Welche Bedeutung diese Angaben haben, liegt in der Entscheidung der Ausgabestelle des Namens (NAA).

Ein weiterer Teil des ARK-Frameworks ist das Tiny HTTP URL Mapping Protocoll (THUMP). Damit wird geregelt, wie auf die drei Eigenschaften eines Objektes durch HTTP-Server und deren Benutzer zugegriffen werden können. Die alleinige Angabe eines ARKs führt per HTTP-Redirect zu dem eigentlichen Objekt. Die Meta-Daten erreicht man, indem an den ARK ein Fragezeichen angehängt wird. Für das „commitment statement“ wiederholt sich die Vorgehensweise, nur dass zwei Fragezeichen benötigt werden[49]. Die Ausgabe der beiden Meta-Daten erfolgt im Format der Electronic Resource Citation (ERC). ERC ist ein einfaches Meta-Daten-Format; Einzelheiten dazu finden sich im ARK-Draft.

Der theoretisch dezentrale Aufbau und die relativ freien Gestaltungsmöglichkeiten bei der Vergabe eines Namens erlauben es, dass die kanonischen Zitierweisen in einen ARK integriert werden können. Die Verschlüsselung der ausgebenden Institution mittels einer Ziffernfolge ist ein kleiner Nachteil. Auch die Angabe von verschiedenen Versionen eines Objektes könnte eleganter gelöst werden. Damit wäre auch intuitivere Benutzbarkeit gewährleistet. Die Frage, ob auch andere Institutionen außerhalb des Bibliotheks- und Archivwesens an dem Framework teilnehmen können, scheint noch nicht endgültig geklärt zu sein. Angesichts der dezentralen Ausrichtung des Auflösungsdienstes bleibt abzuwarten, ob sich genügend solcher Dienste finden werden, sodass die Abhängigkeit von einzelnen Institutionen vermieden werden kann.

[49] Ein funktionierendes Beispiel kann über den ARK http://ark.cdlib.org/ark:/13030/ft4w10060w erreicht werden.

3.1.4. Das Handle-System der CNRI

Das Handle-System wurde durch die Corporation for National Research Initiatives (CNRI)[50] entwickelt. Das Handle-System besteht aus drei Teilen:

- einem Namensbereich
- Kommunikations-Protokollen
- einer Referenz-Implementierung

Der Namensbereich liefert Namen für Objekte. So stellt die Zeile ein Beispiel für einen Handle dar[51]:

```
10.1045/january99-bearman
```

Ein Handle beginnt mit einer Zahlenkombination (`10.1045`), die die Handle Naming Authoriy (NA) repräsentiert und die von der Global Handle Registry ausgegeben wird. Die NA kann in Bereiche unterteilt sein, wobei ein Unterbereich (hier `1045`) mit einem Punkt abgetrennt wird. Die Kopplung der Bereiche erfolgt nur auf administrativer Ebene. Es muss kein inhaltlicher Bezug zum übergeordneten Bereich (hier `10`) bestehen. Die Unterbereiche können nur mit Genehmigung des übergeordneten Bereiches entstehen. Auf die NA folgt nach einem Schrägstrich der Handle Local Name, also der eigentliche Name des Objektes. Der lokale Name kann jedes druckbare Zeichen des UCS-2 Zeichensatzes[52] beinhalten. Die Groß- und Kleinschreibung wird berücksichtigt.

Die Auflösung des Handles geschieht unabhängig vom DNS und erfolgt aufgrund von eigenen Protokollen. Die Funktionsweise ist aber mit der des DNS vergleichbar. Der globale Dienst verweist auf den Dienst der zuständigen NA und diese leitet die Anfrage an den eventuell vorhandenen untergeordneten Bereich weiter. Die praktische Auflösung eines Handles, der beispielsweise in einer WWW-Seite eingebettet ist, erfolgt auf zwei Arten: die direkte Auflösung

[50] Vgl. http://handle.net/, die Dokumentation kann unter http://www.handle.net/documentation.html eingesehen werden.

[51] Ein Handle ist ein Name innerhalb des Namensbereiches, der Namensbereich wiederum ist Teil des gesamten Handle-Systems.

[52] Dies ist ein 2-Byte Zeichensatz, der zu Unicode 2.0 kompatibel ist. In der Praxis wird aber UTF-8 verwendet.

durch den Browser - über Zusatzprogramme[53] - oder einen Resolver-Dienst, der als Proxy dazwischen geschaltet wird[54]. Das Ziel einer Auflösung muss nicht immer eine URL sein, sondern es sind auch andere Verweisarten denkbar.

Die Referenz-Implementierung steht auf der Projektseite zur Verfügung und ist dort frei erhältlich[55]. Sie besteht aus einer Server- und einer Client-Distribution und ist in Java implementiert. Mit ihrer Hilfe kann eine lokale NA aufgesetzt werden.

Das Handle-System ist eine technisch ausgereifte Lösung, die mehreren Projekten als Basis dient[56]. Auch hier ist es möglich, die kanonischen Zitierweisen im lokalen Namen zu integrieren. Die Verwendung eines Unicode-Zeichensatzes erlaubt es zudem, ohne Einschränkungen die kanonischen Namen aufzunehmen. Ob die Unabhängigkeit von Standard-Internettechniken (wie dem DNS) einen Vorteil darstellt, soll hier nicht entschieden werden. Auf jeden Fall ist das Handle-System ein Beispiel für ein Konzept, dass ohne die zentral zur Verfügung gestellten Dienste nicht funktioniert. Auch diesbezüglich soll nicht in Abrede gestellt werden, dass das System nicht auf Nachhaltigkeit ausgelegt ist, doch ist diese wesentlich vom dauerhaften Bestand der Institution CNRI abhängig, der kaum über Jahrzehnte hinweg zu garantieren ist.

3.1.5. Digital Object Identifier System

Das Digital Object Identifier System (DOI) wurde von der International DOI Foundation (IDF) entwickelt und existiert seit 1998[57]. Das DOI ist eine Implementierung des Handle-Systems, das es aber erweitert. Es besteht aus vier Komponenten:

- einem Namensschema

[53] Entsprechende Plug-Ins sind über die Projekt-Startseite erhältlich, siehe Fussnote 50.

[54] Vgl. z.B. http://hdl.handle.net/. Um das angegebene Beispiel nachzuvollziehen, muss der Handle noch angefügt werden: http://hdl.handle.net/10.1045/january99-bearman.

[55] Vgl. http://hdl.handle.net/4263537/4003. Die Lizenz erlaubt die freie Nutzung und auch die Veränderung der Software. Eine Einschränkung ist, dass die Software nicht in Staaten angewendet werden darf, die von den USA als sog. „Schurkenstaaten“ angesehen werden.

[56] Vgl. eine Liste mit Projekten, http://hdl.handle.net/4263537/4007.

[57] Vgl. http://www.doi.org/, als Dokumentation steht das DOI-Handbuch zur Verfügung, http://www.doi.org/handbook_2000/.

- einem Resolving Dienst
- einem Meta-Daten Modell
- einem Regelwerk, das beschreibt, wie die einzelnen Operationen ausgeführt werden

Das Namensschema folgt dem Schema eines Handles. Alle DOIs beginnen mit dem Präfix „10“. Dies kennzeichnet die übergeordnete Ebene der Naming Authority des Handle-Systems. Der darunter liegende Bereich wird von der IDF verwaltet. Die Ausgabe der eigentlichen Namen wird jedoch nicht von der IDF verwaltet, sondern geschieht durch sog. Registration Agencies (RAs). Um diesen Status zu erwerben, muss man Mitglied beim IDF werden, womit Kosten verbunden sind[58]. Jede RA hat das Recht, die Namen für einen Unterbereich nach einem beliebigen Geschäfts- und Preismodell auszugeben.

Der Resolving Dienst funktioniert ebenfalls wie beim Handle-System. Darüber hinaus sind mehrere Arten der Auflösung möglich. So können neben der gängigen Auflösung zu einem Objekt auch mehrere Verweisziele angegeben und angesprochen werden.

Alle DOIs müssen mit Meta-Daten ausgezeichnet werden, die demselben Modell folgen. Dadurch steigt die Interoperabilität zwischen den einzelnen DOIs. Für die Meta-Daten wird das indecs Data Dictionary (iDD)[59] verwendet, das neben einem genormten Vokabular auch die Integration anderer Meta-Daten-Formate erlaubt.

Das DOI wird im kommerziellen Bereich stark genutzt. Für die Eignung der kanonischen Zitierweisen gilt dasselbe wie beim Handle-System: Die grundsätzliche Eignung ist vorhanden, doch die zentralistisch angelegte Struktur ist ein Problem. Darüber hinaus steht beim DOI der kommerzielle Aspekt im Vordergrund, was die Eignung für die notorisch finanzschwachen geisteswissenschaftlichen Wissenschaftszweige prinzipiell einschränkt.

[58] Es existieren mehrere Arten der Mitgliedschaft, die Preise liegen zwischen $5.000 und $70.000.
[59] Vgl. http://www.indecs.org/.

3.2. Eine neue Lösung

Die Bereitstellung von Primärdaten der Geisteswissenschaften in Form von nachhaltigen digitalen Repositorien erfordert die Berücksichtigung mehrerer Aspekte. Ein zentraler Punkt ist dabei die Referenzierung eines Objektes durch einen Namen und nicht durch eine Adresse. Die im vorherigen Kapitel vorgestellten Konzepte und Systeme bieten hierfür alle eine Lösung an. Ein genereller Nachteil diser Lösungen ist aber die Tatsache, dass die kanonischen Zitierweisen sich dem jeweiligen Konzept anpassen müssten. Aus dem Entstehungshintergrund der einzelnen Projekte ist dies verständlich. Da die kanonischen Zitierweisen aber bereits etablierte und dauerhafte Namen für Objekte bereitstellen, sollten sie als weitere Lösungsmöglichkeit in Betracht gezogen werden. Aus diesem Grund verwendet das vorzustellende Adressierungs-Schema die kanonischen Namen, um so die Referenzierung eines Objektes über einen Namen zu gewährleisten und sie von der Adresse des Objektes zu entkoppeln.

In einer Web-Applikation, die geisteswissenschaftliches Material präsentiert, sind, neben den Namen, noch weitere Aspekte zu beachten. Es gibt immer wiederkehrende Funktionen, wie etwa die Anzeige eines Objektes oder die Anzeige von Informationen über ein Objekt. Diese Basis-Funktionen sollen ebenfalls in die neue Lösung integriert werden. Ein weiterer Punkt ergibt sich durch die Potentiale der Verarbeitung durch den Rechner. Ist es im Buchdruck nur möglich, *eine* Sicht auf die Daten zu liefern, ergeben sich bei der rechnergestützten Verarbeitung erheblich mehr Möglichkeiten. Das Spektrum reicht dabei von der banalen Unterscheidung des Ausgabeformates (beispielsweise PDF oder HTML), bis zu einer Kodierung von vielfältigen Zuständen und Einstellmöglichkeiten, um die Ausgabe zu steuern[60]. Auch dies ist bei der präsentierten Lösung berücksichtigt.

Die drei Komponenten (Name, Funktion, Format) können in ein einheitliches Schema aufgenommen werden. Das Schema macht sich die Tatsache der freien Gestaltung der URL mittels PATH_INFO zunutze, wie sie in Kapitel 2.2.1 de-

[60] Im Bereich der Digitalen Editionen sind sehr schnell viele solcher Zustände zu erreichen. Bei der Edition des Werkes von Johan Daisne, De trein der traagheid, gibt es theoretisch 80 Trillionen Möglichkeiten, das Material zu präsentieren; siehe das Abstract von Edward Vanhoutte, Editorial theory in crisis. The concepts of base text, edited text, textual apparatus and variant in electronic editions, http://www.ahds.ac.uk/drh2005/viewabstract.php?id=68.

monstriert wurde und versucht, möglichst „Technik-frei“ zu sein. Die eigentlich störenden, aber notwendigen Elemente, die für die reibungslose rechnergestützte Verarbeitung notwendig sind, werden auf ein Minimum reduziert. Durch diese Reduzierung erledigt sich das Problem der Nachhaltigkeit gleich von selbst, da nur auf etablierte, zukunftssichere Elemente zurückgegriffen wird.

3.2.1. Basis-Funktionen

Grundlegend verschiedene Arten des Zugriffs auf geisteswissenschaftliches Material sollten auch verschieden benannt werden. Viele Web-Applikationen konzentrieren sich auf eine Art des Zugriffes, nämlich die Anzeige eines Objektes. Neben der Anzeige existieren noch weitere grundlegende Zugriffsarten, die hier kurz vorgestellt werden sollen. Die vorgeschlagenen Namen sind exemplarisch zu verstehen und in deutscher Sprache gehalten. Die Erweiterung um hier nicht aufgeführte Funktionen ist problemlos möglich.

zeige In allen Angeboten, die auf einzelne Objekten zugreifen wollen, wird diese Funktion benötigt. Sie ist auch für die Referenzierung für ein einzelnes Objekt zuständig und leistet somit einen Beitrag zur Nachhaltigkeit des Gesamtsystems. Anhand dieser Funktion allein ist noch nicht klar, ob nicht auch mehrere Objekte angezeigt werden können. Die Anzeige eines Objektes sollte aber auf jeden Fall möglich sein und auch von der Web-Applikation, wenn sie denn dieses Schema unterstützt, angeboten werden.

meta Die `meta`-Funktion wird verwendet, um Informationen über ein Objekt anzuzeigen. Sie ist stark verwandt mit der `zeige`-Funktion, da sie ebenso auf einzelne Objekte abzielt. Welcher Art die Meta-Informationen sind, hängt ganz von der Ausrichtung des jeweiligen Angebotes ab und soll hier nicht weiter festgelegt werden. Die Bandbreite kann von „echten“ Meta-Informationen, etwa die Bearbeitungshistorie eines Objektes, bis hin zu Daten über Objekte, wie etwa die Katalogdaten zu den Handschriften in dem CEEC-Projekt, reichen.

bestimme Die erwähnte Uneinheitlichkeit im Umgang mit kanonischen Referenzen bringt es mit sich, dass diese Funktion benötigt wird. Sie ist dafür zuständig, dass ein von der festgelegten Grundform abweichendes Zitat, das wegen der Abweichung nicht direkt aufgelöst wird, noch vom System weiterverarbeitet werden kann. So kann sie von den Funktionen `zeige` und `meta` aufgerufen werden, wenn kein direktes Ergebnis auf eine Objektangabe gefunden wurde.

Diese Funktion versucht dann mittels der weiteren Daten, die in der Topic Map gespeichert sind, das Zitat aufzulösen. Wenn dies nicht eindeutig geschehen kann, sollte eine entsprechende Liste mit den möglichen Kandidaten ausgegeben werden. Des weiteren sollte es innerhalb der Web-Applikation möglich sein, ein Zitat sozusagen „auf gut Glück“ einzugeben und auf eine positive Antwort zu hoffen.

index Zu jedem größeren Angebot gehören Indices und listenförmige Aufstellungen von Daten. Dafür existiert die `index`-Funktion. Sie dient zur Abfrage eines Index und liefert eine entsprechende Liste als Ergebnis.

suche Auch die freie Suche ist ein integraler Bestandteil eines Online-Angebotes. Die `suche`-Funktion ist dafür verantwortlich. Da sie sehr stark von den Erfordernissen des jeweils präsentierten Materials abhängt, wird darauf verzichtet, differenziert auf die Parameter einzugehen. Trotzdem wird sie hier aufgenommen, um die Einheitlichkeit und die Interoperabilität zwischen den Angeboten zu fördern. Wenn diese Funktion aufgerufen wird, sollte eine leere Suchmaske generiert werden, die auf das jeweilige Angebot zugeschnitten ist. Die Formular-Daten der Suchmaske werden dann über POST bzw. GET der Web-Applikation übergeben und sind nicht Teil des Adressierungsschemas.

3.2.2. Adressierungsschema

Das Adressierungsschema wird als kontextfreie Grammatik dargestellt, und zwar nach den Syntax-Regeln der Augmented Backus-Naur Form (ABNF), die eine

Vereinfachung der originalen BNF ist[61]. Das Schema versucht sich möglichst auf das Wesentliche zu konzentrieren und technisch bedingte Elemente beiseite zu lassen. Im Anhang C finden sich vollständige Beispiele mit gültigen Adressen dieses Schemas und einer kurzen Beschreibung über die Bedeutung der URL.

```
zeichen     = ( ALPHA / DIGIT ) *( ALPHA / DIGIT / "-" / "_" )
zeichenPlus = ( ALPHA / DIGIT ) *( ALPHA / DIGIT
                                   / "-" / "_" / "." )
objektName  = ( ALPHA / DIGIT ) *( ALPHA / DIGIT
                                   / "-" / "_" / "." / "," )
```

Es existieren mehrere Zeichenklassen. Dies ist notwendig, da in unterschiedlichen Bereichen unterschiedliche Konventionen gelten, welche Zeichen erlaubt sind. Alle Zeichenketten, die hier vorkommen können, beginnen mindestens mit einem Buchstaben oder einer Zahl. Darauf kann eine beliebige Kombination von Zeichen folgen, in der je nach Art unterschiedliche Sonderzeichen enthalten sind.

```
adresse     = rechner "/" funktion "|/" [ "~" format "/" ]
              [ anzeigeName ]
rechner     = "http" ["s"] "://" authority "/" applikation
authority   = zeichenPlus; Vereinfacht
applikation = zeichen
format      = zeichen
anzeigeName = zeichenPlus
```

Eine Adresse dieses Schemas folgt grundsätzlich der Syntax eines Uniform Resource Identifier (URI) mit der Einschränkung, dass als Schema nur das HTTP(S)-Protokoll zugelassen ist. Die Definition des Rechnernamens (`authority`) ist in der realen Anwendung komplexer und wurde hier nur vereinfacht dargestellt. Der genaue Aufbau kann dem entsprechenden Abschnitt der URI-Spezifikation (RFC 3986)[62] entnommen werden. Der Rechnername wird durch den externen

[61] Die Regeln, die für das Verständnis des vorgestellten Schemas nötig sind, lauten: der Schrägstrich (/) steht für eine Alternative (oder), das Semikolon beginnt einen Kommentar, in eckigen Klammern stehen optionale Teile und der Stern bedeutet eine Wiederholung (Null bis Unendlich). ALPHA und DIGIT sind vorgegebene Symbole und umfassen die Zeichen A–Z und a–z (ALPHA) und 0–9 (DIGIT). Die genaue Spezifikation ist aus dem RFC 2234 ersichtlich, http://www.faqs.org/rfcs/rfc2234.html. Die Unterschiede zu den weiteren Formen der BNF listet eine Seite von Pete Jinks auf, http://www.cs.man.ac.uk/~pjj/bnf/ebnf.html.

[62] Vgl. Uniform Resource Identifier (URI), Generic Syntax, http://www.gbiv.com/protocols/uri/rfc/rfc3986.html#authority.

Namen (`applikation`), hinter dem die zuständige Web-Applikation versteckt ist, komplettiert.

Nach diesen technischen Vorbedingungen beginnt die eigentliche Objektreferenzierung. Jede Adresse wird von einer Funktion (`funktion`) eingeleitet, die ihre eigenen Regeln hat. Den Abschluss der Funktion bildet eine feste Zeichenkette (`|/`), die notwendig ist, um die Eindeutigkeit und die Verarbeitbarkeit dieses Schemas zu garantieren. Nach der Angabe der Funktion können noch eine Formatangabe und ein Anzeigenamen folgen, die jedoch beide optional sind. Mit der Formatangabe (`format`) soll unterstützt werden, dass dieselben Objekte unterschiedlich ausgegeben werden können. So ist es sinnvoll, in einer realen Web-Applikation ein Text-Objekt einmal in der HTML-Sicht, aber auch als Druckobjekt im PDF-Format anbieten zu können. Aus diesem Grund wurde die Formatangabe mit in das Schema aufgenommen. Diese beginnt mit einer festen Zeichenkette (~), die wiederum der Eindeutigkeit geschuldet ist. Der Name der Formatangabe kann je nach Applikation angepasst werden. Von einfachen Formatangaben (etwa html oder pdf), über symbolische Ansichtsnamen für verschiedene Benutzergruppen und Ausgaben (etwa einfach, experte oder pagemed), bis hin zu kodierten Zuständen einer komplizierten Benutzeroberfläche mit vielen Anzeigeoptionen[63] ist alles denkbar.

Der Anzeigename (`anzeigeName`) hat eigentlich nichts mit den Objekten zu tun. Er wurde nur deswegen aufgenommen, um den Umgang mit den Objekten durch den Benutzer zu erleichtern. So kann hier ein „Dateiname“ untergebracht werden. Dies hätte Vorteile, falls der Benutzer ein Objekt speichern wollte, denn die Anzeigeprogramme (Browser) interpretieren die letzte Zeichenkette nach einem Schrägstrich als Dateinamen und bieten diesen Namen beim Speichern an. Daneben könnte der Dateiname so gewählt werden, dass er als weiterer Parameter für das Ausgabeformat dienen könnte und somit zwei Felder für das Ausgabeformat zur Verfügung stehen.

[63] So ist es beispielsweise denkbar, in einer Digitalen Edition eine Vielzahl von Optionen anbieten zu wollen. All diese Optionen beziehen sich dabei auf die Ausgabe. Dazu zählen etwa der Grad der Normalisierung, die Nähe zum Original usw. Die Formatangabe kann dann aus einer kodierten Form dieser Optionen bestehen (z.B. ein Bitmuster mit fester Breite (11010011) oder ein entsprechender hexadezimaler String).

Da die Formatangabe und der Anzeigename optionale Elemente sind, kann eine Adresse dieses Schemas schon mit der vollständigen Funktionsangabe beendet sein. Diese Möglichkeit sollte besonders bei der Objekt-Referenzierung gebraucht werden, um die gesamte Adresse möglichst kurz und frei von überflüssigen Elementen zu halten. Das Fehlen der beiden Angaben wird von der Web-Applikation durch eine standardmäßige Aufbereitung des Objektes bei der Ausgabe aufgefangen.

```
funktion    = zeige / meta / bestimme / index / suche
```

Als Funktionen kommen die weiter oben beschriebenen in Betracht, also die Anzeige eines Objektes (`zeige`), die Anzeige von Informationen über ein Objekt (`meta`), die Auflösung einer nicht direkt bekannten kanonischen Referenz durch das System (`bestimme`), die Anzeige eines Indexes (`index`) und schließlich die Durchsuchung des Angebotes mittels einer freien Angabe durch den Benutzer (`suche`). Weitere Funktionen sind im Moment nicht vorgesehen, könnten aber einfach hinzugefügt werden.

```
zeige       = "zeige/" referenz
meta        = "meta/" referenz
```

Die `zeige`-Funktion beginnt mit einer festen Zeichenkette (dem Funktionsnamen), der die eigentliche Referenz auf ein Objekt folgt. Der Aufbau einer Adresse für die Anzeige der Meta-Informationen ist, von der einleitenden Zeichenkette abgesehen, dazu identisch. Beide Funktionen greifen schließlich auf dieselben Objekte zu.

```
referenz    = werk "/" version [ "/" objekt ]
werk        = zeichen
version     = zeichen
objekt      = objekt1 [ "/" objekt2 ] [ "/" objekt3 ]
objekt1     = objektName
objekt2     = objektName
objekt3     = objektName
```

Eine Referenz auf ein Objekt besteht mindestens aus einer Werk- und Versionsangabe. Optional kann sie um weitere Objektangaben erweitert werden. Die Werk-

angabe (`werk`) bezeichnet den Herausgeber, die Institution usw. und korrespondiert mit den grundlegenden Werk-Typen der Kanon XTM (siehe Anhang A). Bei Institutionen, die viele Objekte bereitstellen, wie etwa eine Handschriftenbibliothek oder eine Editionsreihe, sollte die Werkangabe eigentlich in den Rechnernamen integriert sein. Wie schon weiter oben beschrieben, existiert kein einheitliches System für solche Namen. Die nachträgliche Sicherung der Namen wird in den meisten Fällen nicht mehr möglich sein. Deswegen wird die Werkangabe hier aufgeführt, bis die Problematik gelöst worden ist.

Die Versionsangabe (`version`) bezieht sich auf die Werkangabe, falls keine weiteren Angaben folgen. Existiert aber eine nähere Bestimmung des Objektes, bezieht sich `version` auf die jeweils letzte Angabe. Eine Referenz ohne nähere Objektangabe sollte eine listenförmige Ausgabe generieren, die über die enthaltenen Objekte des Werkes Aufschluss gibt.

Das eigentliche Objekt kann auf drei Hierarchie-Ebenen beschrieben werden und ist das Gegenstück zu den grundlegenden Objekt-Typen der Kanon XTM. Anhand eines Beispiels soll die Auflösung dieses Teils der Referenz durch die Web-Applikation demonstriert werden.

.../mgh/retrodig/ss/xx/86|/

Das Beispiel soll auf die Seite 86 des 20. Bandes der Scriptores-Reihe der Monumenta Germaniae Historica verweisen und es wird angenommen, dass dieser Band als Teil der Reihe und die Reihe als Teil des Werkes in die Kanon XTM aufgenommen wurde. Nun wird die Angabe von links nach rechts aufgelöst, wobei die Versionsangabe erst einmal ignoriert wird. Die Applikation sollte nun feststellen, dass die Angabe *86* nicht aufgelöst werden kann. Nun muss überprüft werden, ob sich diese Angabe noch mit den weiteren Informationen der vorherigen Angabe (*xx* in der Version *retrodig*) vereinbaren lässt. Dies geschieht über die Daten, die in den Occurences `refCheck` und `umfangListe` gespeichert worden sind. Ihnen ist zu entnehmen, dass eine Seite 86 in dem Band enthalten ist: die Referenz ist also gültig.

Stellt die Web-Applikation fest, dass die Referenz ungültig ist, wird sie der `bestimme`-Funktion übergeben. Ob dies innerhalb der Web-Applikation, also im selben Ausführungsschritt geschieht, oder aber ein HTTP-Redirect ausgelöst

wird, bleibt der Applikation überlassen. Der internen Weiterverarbeitung sollte aber der Vorrang eingeräumt werden.

Referenzen treten einer Web-Applikation auf zwei Ebenen entgegen. Zum einen in Verweisen, die innerhalb der Applikation selbst erzeugt wurden. Diese sind immer direkt auflösbar und gültig, da sonst ein Fehler bei der Erstellung vorliegen würde. Aufgrund der intuitiven Bedienbarkeit durch den Benutzer existiert aber zusätzlich noch eine zweite Ebene, nämlich die vom Benutzer leicht veränderten oder selbst eingegebenen Verweise. Deswegen sollte die Web-Applikation, gerade bei der Auflösung der tieferen Hierarchie-Ebenen, nicht zu sehr auf exakte Vergleiche setzen, um vermeintlichen Fehlbedienungen entgegenzutreten. Dazu gehört etwa die Gleichbehandlung von Zahlen in römischer und dezimaler Notation, oder Abkürzungen von Bänden (Bd.) und Seiten (S.) und andere mögliche Eingaben. Deswegen sollte die Referenz *.../mgh/retrodig/ss/Bd.20/S.86|/* nicht als ungültig betrachtet werden und zu demselben Ergebnis führen, wie das obige Beispiel.

```
bestimme    = "bestimme/" objektName
```

Die `bestimme`-Funktion besitzt nur ein weiteres Element, nämlich eine Zeichenkette, die das zu suchende Objekt repräsentiert. Sie benutzt die erweiterten Daten aus der Kanon XTM, um ein Objekt auszulösen.

```
index       = "index/" indexName [ "/" indexStelle ]
indexName   = zeichen
indexStelle = zeichen
suche       = "suche/" suchArt "/"
suchArt     = zeichen
```

Die Suchfunktionen sind einfacher strukturiert. Bei der `index`-Funktion wird noch ein Name benötigt, um den Index auszuwählen (z.B. orte für den Ortsindex usw.). Durch das optionale Element (`indexStelle`) kann die Aufschlageseite des Indexes noch beeinflusst werden.

Die `suche`-Funktion hingegen erwartet nur noch eine Zeichenkette, um die Komplexität des Suchformulars anzugeben. Sinnvoll wäre hier etwa „einfach“ oder „experte“.

4. Fazit

Die Frage nach der Eignung der kanonischen Zitierweisen der Geisteswissenschaften als nachhaltige Komponenten in digitalen Repositorien kann mit einem klaren „Ja.“ beantwortet werden. Die Zitierweisen qualifizieren sich besonders dadurch, dass sie ein eigenes, schon etabliertes Referenzsystem darstellen. Eine wichtige Leistung dieses Systems ist die Nachhaltigkeit, da sich die Referenzen von bestehenden Objekten nicht mehr ändern werden und zukünftige Objekte sich ebenfalls in die gängigen Zitierweisen aufnehmen lassen. Mit der Kanon XTM besteht zudem ein Instrumentarium, das die Werke der wissenschaftlichen Literatur speicherbar und informationstechnisch verarbeitbar macht. Daneben kann sie die Verknüpfungen und Beziehungen zwischen Werken sichtbar und für weiterführende Anwendungen nutzbar machen.

Die Erstellung von Web-Angeboten, die mit kanonischen Referenzen umgehen und entsprechende Inhalte anbieten, bereitet aus technischer Sicht keine größeren Schwierigkeiten, zumindest was die Möglichkeit betrifft, mit kanonischen Namen auf sie zu verweisen. Der Aufbau einer URL, die die Referenzierung leistet, ist in den beschriebenen Rahmenbedingungen frei wählbar. Dadurch erlaubt er die problemlose Integration der geisteswissenschaftlichen Zitierweisen. Das vorgestellte Adressierungsschema vereint die kanonischen Namen mit den Anforderungen, die an typische Web-Applikationen geisteswissenschaftlichen Materials gestellt werden, und leistet ein Beitrag zur Entkopplung des Namens von der Adresse eines Objektes. Des weiteren ermöglicht es eine intuitive und transparente Nutzung solcher Angebote.

Die anderen Lösungen, die sich des Problems der nachhaltigen Referenzierung von Objekten annehmen, vermögen nur Teilbereiche der Anforderungen abzudecken und sind deswegen weniger gut für den hier betrachteten Bereich geeignet. Sie können ihre Stärken hingegen im Bereich der Objektreferenzierung ausspielen, dem unter dem Gesichtspunkt der ungelösten Domain-Problematik eine besondere Bedeutung zukommt.

Die Erkenntnis der besonderen Eignung der kanonischen Zitierweisen hat bei den Projektverantwortlichen leider erst in Ansätzen Verbreitung gefunden. Hier

ist durchaus die Notwendigkeit von „Aufklärungsarbeit“ zu konstatieren. Da mit der stärkeren Nutzung der neuen Medien einerseits die Ansprüche an die beteiligten Systeme und die Verantwortlichen steigen, andererseits der Aspekt der Nachhaltigkeit zu den grundlegenden Herausforderungen an das „flüchtige“ Medium Internet wie auch den zentralen Kritikpunkten der Skeptiker zählt, kann mit der Hoffnung abgeschlossen werden, dass die kanonischen Zitierweisen zukünftig in entsprechendem Umfang bei der technischen Realisierung von Digitalisierungs- und Publikationsprojekten berücksichtigt werden.

5. Inhalt der beiliegenden CD

Die beiligende CD enthält die Kanon XTM und ein Anzeige-Programm, das es ermöglicht, allgemein auf Topic Maps innerhalb einer Browser-Oberfläche zuzugreifen. Bei dem Programm handelt es sich um den Ontopia Omnigator der Firma Ontopia. Es ist über das WWW frei erhältlich[64]. Auf der CD befindetsich das Programm im Verzeichnis `Omnigator/`. Der Omnigator ist JAVA-basiert und wird mit einem grafischen Installationsprogramm eingerichtet. Der Start der Installation ist in der Datei `INSTALL` beschrieben, das sich ebenfalls im Verzeichnis `Omnigator/` befindet. Die Einrichtung und Benutzung des Programms kann der beiliegenden Dokumention entnommen werden, die über die Datei `index.html` aus dem gewählten Wurzelverzeichnis der Installation gestartet wird.

Die Kanon XTM befindet sich im gleichnamigen Verzeichnis auf der CD. Sie liegt in zwei äquivalenten Versionen vor: Das Unterverzeichnis `ltm` enthält die Kanon XTM im LTM-Format und das Unterverzeichnis `xtm` im XTM-Format. Die Daten wurden entweder selbst eingegeben oder aus Listen, die im WWW veröffentlicht wurden, konvertiert. In der Kanon XTM sind gut 10.000 Topics gesammelt worden. Mit Hilfe des Omnigators kann in der Topic Map navigiert werden. Als Ausgangspunkt eignet sich beispielsweise der „Topic Type“ Zeitschrift und dann die „Historische Zeitschrift“[65].

Die Daten der Kanon XTM sind weder vollständig noch fehlerfrei. Gerade die automatische Konvertierung birgt viele Fehler, die bestimmt nicht alle erkannt wurden. Dennoch sind die Daten gut genug, um sich eine Vorstellung von den Möglichkeiten der Kanon XTM zu machen.

---

[64] Vgl. http://www.ontopia.net/download/freedownload.html.

[65] Nach erfolgreicher Installation sollte diese Seite unter der URL http://localhost:8080/omnigator/models/topic_complete.jsp?tm=referenzen.ltm&id=hz erreichbar sein.

Literaturverzeichnis

Augmented BNF for Syntax Specifications, RFC 2234, http://www.faqs.org/rfcs/rfc2234.html.

CGI Specification, http://hoohoo.ncsa.uiuc.edu/cgi/interface.html.

DynaWeb Client Guide, Version 4.3, hg. v. Enigma Information Retrieval Systems Inc., http://fobe.itaw.hu-berlin.de/dynaweb/client/client/@Generic__BookTextView/197;cs=default;ts=default;pt=197/*#X.

ISO/IEC 13250:2000 Topic Maps. Information Technology - Document Description and Markup Languages, hg. v. Michel Biezunski, Martin Bryan, Steven R. Newcomb.

Jakobs, Eva-Maria: Textvernetzung in den Wissenschaften. Zitat und Verweis als Ergebnis rezeptiven, reproduktiven und produktiven Handelns, (= Reihe germanistische Linguistik Bd. 210), Tübingen 1999.

Jele, Harald: Wissenschaftliches Arbeiten. Zitieren, München 2003.

Linear Topic Map Notation. Definition and introduction, Version 1.3, hg. v. Lars Marius Garshol, http://www.ontopia.net/download/ltm.html.

Park, Jack (Hg.): XML Topic Maps. Creating and Using Topic Maps for the Web, Boston 2003.

Runkehl, Jens - Siever, Torsten: Das Zitat im Internet. Ein Electronic Style Guide zum Publizieren Bibliografieren und Zitieren, Hannover 2000.

Thaller, Manfred: Die Handschriftenbibliothek des Kölner Doms im Internet, in: Manfred Thaller (Hg.), Codices Electronici Ecclesiae Coloniensis. Eine mittelalterliche Kathedralbibliothek in digitaler Form, Göttingen 2001, 21–39.

Uniform Resource Identifier (URI). Generic Syntax, RFC 3986, hg. v. Tim Berners-Lee, Roy Fielding, Larry Masinter, http://www.gbiv.com/protocols/uri/rfc/rfc3986.html

Widhalm, Richard - Mück, Thomas: Topic Maps. Semantische Suche im Internet, Berlin 2002.

XML Linking Language (XLink), Version 1.0, hg. v. Steve DeRose, Eve Maler,

David Orchard, W3C Recommendation vom 27 Juni 2001, http://www.w3.org/TR/xlink/.

XML Topic Maps (XTM). TopicMaps.Org Specification, Version 1.0, hg. v. Steve Pepper, Graham Moore, http://www.topicmaps.org/xtm/index.html.

A. Die Kanon XTM (grundlegende Themen)

```
#TOPICMAP referenzen
[refTopicMap = "Referenzen von Werken"
 @"#referenzen"]

/* Assoziations-Typen */

[verweisSekundaer = "Verweis auf ein sekundäres Objekt"
                  = "ist Ausgangspunkt des Verweises zu" / verweisAusgang
                  = "ist Zielpunkt des Verweises von" / verweisZiel]
[verweisObjekt    = "Verweis auf auf ein verwandtes Objekt"
                  = "ist Ausgangspunkt des Verweises zu" / verweisAusgang
                  = "ist Zielpunkt des Verweises von" / verweisZiel]
[enthaeltObjekt   = "Objekt-Beziehung"
                  = "liefert als Objekt1" / liefertObjekt1
                  = "ist als Objekt1 Teil von" / istUnterObjekt1
                  = "liefert als Objekt2" / liefertObjekt2
                  = "ist als Objekt2 Teil von" / istUnterObjekt2
                  = "liefert als Objekt3" / liefertObjekt3
                  = "ist als Objekt3 Teil von" / istUnterObjekt3]

/* Assoziations-Rollen */

[liefertObjekt1  = "Objekt1-Lieferant"]
[istUnterObjekt1 = "Unter-Objekt1"]
[liefertObjekt2  = "Objekt2-Lieferant"]
[istUnterObjekt2 = "Unter-Objekt2"]
[liefertObjekt3  = "Objekt3-Lieferant"]
[istUnterObjekt3 = "Unter-Objekt3"]
[verweisAusgang  = "Ausgangspunkt des Verweises"]
[verweisZiel     = "Zielpunkt des Verweises"]

/* Ausprägungs-Typen */

[version           = "Version"]
[bibl              = "Bibliographische Angabe"]
[refName           = "Name der kanonischen Referenz"]
[refNameAlt1       = "Name der kanonischen Referenz (Alternative 1)"]
```

```
[refNameAlt2       = "Name der kanonischen Referenz (Alternative 2)"]
[refNameAlt3       = "Name der kanonischen Referenz (Alternative 3)"]
[refCheck          = "Art des Überprüfungsweges einer Referenz"]
[umfangListeBeginn = "Unterer Wert einer Umfang-Angabe"]
[umfangListeEnde   = "Oberer Wert einer Umfang-Angabe"]
[umfangListePlus   = "Liste mit zusätzlichen gültigen Referenzen"]
[umfangListeMinus  = "Liste mit nicht gültigen Referenzen"]

/* Grundlegende Werktypen */

[verzeichnis = "Verzeichnis"]
[lexikon     = "Lexikon"]
[edition     = "Edition"]
[bibBestand  = "Bibliotheksbestand"]
[bibel       = "Bibel"]
[zeitschrift = "Zeitschrift"]
[reihe       = "Reihe"]
[sonstiges   = "Sonstiges"]

/* Grundlegende Objekttypen */
[objekt1 = "Objekt vom Typ 1 (Angabe auf Ebene1)"]
[objekt2 = "Objekt vom Typ 2 (Angabe auf Ebene2)"]
[objekt3 = "Objekt vom Typ 3 (Angabe auf Ebene3)"]
```

B. ABNF für das Adressierungsschema

```
adresse     = rechner "/" funktion "|/" [ "~" format "/" ] [ anzeigeName ]
rechner     = "http" ["s"] "://" authority "/" applikation
authority   = zeichenPlus; Vereinfacht, siehe Text.
applikation = zeichen
format      = zeichen
anzeigeName = zeichenPlus
funktion    = zeige / meta / bestimme / index / suche
zeige       = "zeige/" referenz
meta        = "meta/" referenz
referenz    = werk "/" version [ "/" objekt ]
werk        = zeichen
version     = zeichen
objekt      = objekt1 [ "/" objekt2 ] [ "/" objekt3 ]
objekt1     = objektName
objekt2     = objektName
objekt3     = objektName
bestimme    = "bestimme/" objektName
index       = "index/" indexName [ "/" indexStelle ]
indexName   = zeichen
indexStelle = zeichen
suche       = "suche/" suchArt "/"
suchArt     = zeichen
zeichen     = ( ALPHA / DIGIT ) *( ALPHA / DIGIT / "-" / "_" )
zeichenPlus = ( ALPHA / DIGIT ) *( ALPHA / DIGIT
                                   / "-" / "_" / "." )
objektName  = ( ALPHA / DIGIT ) *( ALPHA / DIGIT
                                   / "-" / "_" / "." / "," )
```

C. Beispiele für das Adressierungsschema

Protokoll: **http**
Rechnername: **www.test.com**
Öffentlicher Name für die Web-Applikation: **app**

http://www.test.com/app/zeige/mgh/retrodig/dh4/325|/

Das Objekt2 „325“ als Teil vom Objekt1 „dh4“, das im Werk „mgh“ enthalten ist in der Version, die mit „retrodig“ bezeichnet wurde. Dies entspricht dem Objekt, das den kanonischen Namen MGH D HIV 325 besitzt (eine mittelalterliche Urkunde). Da kein Ausgabeformat angegeben wurde, sollte bei der Anzeige des Objektes (Funktion „zeige“) ein standardmäßiges Format gewählt werden. Das Beispiel verdeutlicht die Referenzierung eines Objektes.

…/app/zeige/mgh/retrodig/dh4/325|/~einfach/325.html

Wie das vorherige Beispiel, nur dass hier eine Ausgabeformat angegeben wurde („einfach“). Daneben wurde das Objekt mit einem Speichernamen („325.html“) versehen.

…/app/zeige/mgh/retrodig/dh4|/~narrationes/index.html

Das Objekt1 „dh4“ kann theoretisch mehrere Teilobjekte besitzen. Sie werden hier angesprochen. Mit der Formatangabe („narrationes“) werden sie in einer speziellen Art aufbereitet. Die Idee ist, dass hier beispielsweise bestimmte Teile des Urkundenformulars, nämlich die Narrationes, aller Unterobjekte angezeigt werden.

…/app/zeige/mgh/retrodig/dh4|/~narrationes/10-19_25_123-166.html

Ein weiteres Beispiel für ein spezielles Ausgabeformat. Hier soll die Möglichkeit verdeutlicht werden, dass mit dem Speichernamen ein zweiter zusätzlicher Parameter verfügbar ist, mit dem die Ausgabe gesteuert werden kann. Die Idee ist hier nur bestimmte Objekte in die Narrationes-Liste aus dem vorherigen Beispiel aufzunehmen.

…/app/zeige/mgh/retrodig/dh4/325|/~pdf/325.pdf

Ein weiteres Beispiel für ein Ausgabeformat, mit dem eine PDF-Ausgabe realisiert werden kann.

…/app/zeige/lexma/retrodig/urkunde|/

Das Objekt1 „urkunde“ in der Version „retrodig“.

…/app/zeige/lexma/2005/urkunde|/

Das Objekt1 „urkunde“ in der Version „2005“. Ein Beispiel für eine zweite (neuere) Version eines Objektes. Die andere Version aus dem vorherigen Beispiel sollte weiterhin gültig bleiben.

…/app/meta/kn28/digital/83ii/|/~katm/83ii.html

Ein Beispiel für die Meta-Funktion. Die Meta-Daten des Objektes werden in dem Format „katm“ ausgegeben.

…/app/index/orte/k|/k.html

Diese URL soll den Index, der mit „orte“ bezeichnet wird, an der Stelle „k“ ausgeben.

…/app/suche/einfach|/

Ein Beispiel für die Suche-Funktion. Das präsentierte Formular wird mit „einfach“ bezeichnet.

Erklärung

Hiermit versichere ich, dass ich diese Magisterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Stellen meiner Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht.
Dasselbe gilt sinngemäß für Tabellen, Karten und Abbildungen.

Unterschrift