Webarchivierung

Eine Studie der KOST

Inhalt

1 Einleitung

Schon relativ kurz nach dem Entstehen des *World Wide Web* wurde klar, dass Inhalte und Informationen in diesem WWW sehr volatil und unbeständig sind: Man findet und liest etwas, und wenn man es tags darauf zitieren möchte, ist es schon nicht mehr da. Darum entstanden bald Lösungen, um von Webseiten nicht nur via Bookmarks die URL, sondern den ganzen Inhalt offline speichern und wenn möglich gleich auch noch den ganzen Kontext dazu, also den Webauftritt, lokal festhalten zu können[1]. Um auf einer institutionellen Ebene den mit der Volatilität des WWW verbundenen Verlust des gesellschaftlichen Gedächtnisses zu bekämpfen, wurde schon 1996 das *Internet Archive* in San Francisco gegründet; andere Institutionen folgten.

Mit ihrer Ausrichtung auf das Internet wurde von allen Protagonisten der ersten Stunde die Webarchivierung immer nahe an der Web-Technologie gedacht und möglichst als vollständige Kopie des Webservers im Archiv umgesetzt. Authentizität bei der Darstellung - das wurde bei der rasanten Entwicklung im Browserbereich schnell klar - ist dabei nur mit Browseremulation zu erreichen. Damit positionieren sich diese Institutionen grundlegend anders als die Archive mit einer Vergangenheit in der Papierwelt. Aus dieser traditionellen Sichtweise stellen sich andere Fragen, zum Beispiel, wie die chaotische Flut der Webwelt durch Bewertung zu bändigen wäre oder wie der technologische Wandel, der beinahe im Jahrestakt eine neue Browsergeneration hervorbringt, im Archiv entschleunigt werden kann.

Diese Studie zur Webarchivierung versucht diese und andere Fragen, die sich bei der Webarchivierung stellen, möglichst allgemeinverständlich zu beantworten.

[1] Eine *Webseite* (Webpage) ist gemeinhin das, was der Browser beim Aufruf einer URL als DOM-Objekt aufbaut und als Seite darstellt. Es ist darauf hinzuweisen, dass dieses Objekt temporär ist und bei verschiedenen Benutzern zu verschiedenen Zeitpunkten unterschiedlich aussehen kann. Den Rahmen einer Webseite bildet ein HTML-Dokument, das weitere Webressourcen einbinden kann. Bei der Verlinkung mit anderen Dokumenten spricht man von Hypertextdokument. Die Verlinkung erlaubt das Navigieren im Netz der Webseiten.

Ein *Webauftritt* oder Internetauftritt (Website) umfasst die Menge aller Webseiten, die miteinander verlinkt thematisch eine Einheit bilden. In der Regel, aber nicht zwingend, sind die Seiten in einer bestimmten Domain zusammengefasst.

2 Hypertext

Die Diskussion um Hypertextdokumente oder Dokumentsammlungen als Alternative zu linearen Texten und Textsammlungen hat eine längere Vorgeschichte als die digitale Informationsverarbeitung[2]. Das erste kommerzielle Hypertextsystem war HyperCard von Apple 1987 (eng verwandt mit dem Konzept des Netzwerkdatenbankmodells, auch bekannt unter dem Namen „CODASYL Datenbankmodell“). Aber erst mit der Erfindung des Internet und der Hypertext Markup Language HTML begann Mitte der 90er Jahre die Entwicklung und Verbreitung von Hypertextdokumenten oder -dokumentsammlungen.

Hypertextinformationen sind nicht seriell mit andern Texten verknüpft, wie wir das aus bekannten Textklassen wie Kapiteln in Büchern oder Dokumenten in Dossiers kennen, sondern ihre Verknüpfung bildet ein Netzwerk, in dem der Lesende von Text zu Text navigieren kann. (Ein klassisch anschauliches Beispiel für eine Hypertext-Dokumentsammlung ist Wikipedia: Jedes (Wiki-)Wort kann zu einem Link auf eine neue Wiki-Seite werden.) Wir haben damit eine qualitativ andere Informationsaufbereitung als bei seriellen Texten und aus Sicht der Archivierung auch eine andere Problematik, weil die Texte nicht hintereinander im Sinne von Laufmetern auf dem Gestell abgelegt werden können, ohne dass der spezifische Hypertext-Informationsgehalt verloren geht.

Schnell hat sich mit dem Erscheinen von HTML – eigentlich einer Sprache zum Auszeichnen von Texten zur Lektüre am Bildschirm und mit der Möglichkeit zum Setzen von Hypertextlinks – gezeigt, dass das nicht-lineare Konzept auch innerhalb eines einzelnen Dokuments zum Einsatz kommen kann und ungeahnte Möglichkeiten bietet.

Ein Hypertextdokument ist heute also nicht mehr nur ein Dokument, das auf nicht-lineare Art mit weiteren Dokumenten verbunden ist, sondern auch ein Dokument, das in seinem inneren Aufbau die Idee des Verweises auf externe Informationen reflektiert. Erst im Browser, dem Rendering Agent, entsteht daraus wieder ein narratives lineares Dokument, und zwar entsteht das Dokument bei jedem Lesen neu und kann damit auch jedes Mal eine andere Form annehmen.

3 Auszeichnungssprachen und HTML

3.1 HTML als einfache Auszeichnungssprache

Mittels einer Auszeichnungssprache (*Markup Language*) ist es möglich, typische Elemente eines textorientierten Dokuments zu kennzeichnen. Die logischen Bestandteile eines digitalen Textes werden dabei mit Anmerkungen versehen und damit der eigentliche textliche Inhalt mit semantischen Informationen angereichert. Beispielsweise werden Titel, Überschriften, Textabsätze, Listen, Tabellen oder Abbildungen als solche ausgezeichnet. Zudem können auch Unterelemente von Textabsätzen weiter verfeinert werden, so z.B. betonte Textpassagen fett markiert, Aufzählungslisten aus einzelnen Listenpunkten erstellt oder Tabellen in einzelne Tabellenzellen gegliedert werden.

Auszeichnungssprachen werden heute grundsätzlich von nahezu jedem Textverarbeitungsprogramm verwendet. Damit Bearbeitung, Formatierung und Darstellung digitaler Texte von Maschinen ausgeführt werden können, ist es unabdingbar, dass diese Auszeichnung in einem maschinenlesbaren Format und nach bestimmten Regeln erfolgt. Das World Wide Web Consortium (W3C) hat HTML als Standard-Auszeichnungssprache des WWW entwickelt und definiert. Die Auszeichnung erfolgt durch die Einbettung der Textbausteine in sogenannte Tag-Paare. Der Starttag enthält den Elementnamen und optional die dazugehörigen Attribute. Die Elemente werden durch den Endtag geschlossen und lassen sich auch ineinander verschachteln. Diese Auszeichnungen in Form von Tags werden von den Webbrowsern entsprechend interpretiert und aufgelöst. Darauf werden die Elemente der HTML-Dateien optisch ansprechend am Bildschirm dargestellt.

[2] Siehe dazu die Ausführungen in https://en.wikipedia.org/wiki/History_of_hypertext.

3.2 HTML als eine Art virtueller Container

Neben der Strukturierung von Texten besteht mit HTML die Möglichkeit, Dateien in Form einer Referenz in den Text zu integrieren (z.B. Bilder, Filme, Audiodateien). Sofern der Browser über die entsprechenden Plugins verfügt, können diese Dateien korrekt dargestellt werden. Über Schnittstellen für Erweiterungssprachen wie Stylesheets oder JavaScript wird definiert, wie audiovisueller Inhalt und Text in einem Webbrowser dargestellt werden. Weiter können Formulare in den Text eingebunden werden und ermöglichen damit eine Interaktion mit den Benutzenden. Eines der wichtigsten Elemente von HTML ist die Möglichkeit, Hyperlinks zu Adressen im WWW, aber auch zu Internet-Adressen, die nicht Teil des Webs sind, zu setzen. Sämtliche Komponenten werden in HTML ausgezeichnet, und die Applikationen sind in der Lage, diese zu interpretieren, die entsprechenden Dateien oder Verweise aufzurufen und gemäss Definition wiederzugeben.

Dabei ist ausschlaggebend, dass die referenzierten Objekte selbst nicht im HTML-File enthalten sind, sondern lediglich als externe Ressourcen referenziert werden. Damit diese Referenzen entsprechend in die Seite eingebettet werden können, ist es notwendig, dass sämtliche im HTML-File referenzierten Objekte der Applikation (z.B. dem Browser) zur Verfügung stehen.

3.3 Trennung von Form und Inhalt

Grundsätzlich beinhaltet HTML auch Auszeichnungselemente zur gestalterischen Darstellung des Inhalts. Durch die Kombination von darstellerischen und inhaltlichen Informationen in ein und derselben Sprache wurde HTML immer komplizierter. Webentwickler werden deshalb angehalten, auf gestalterische Elemente in HTML-Dateien zu verzichten, um eine strikte Trennung von Form und Inhalt zu erreichen.

Hierfür wurde Cascading Stylesheets (CSS) als unmittelbare Ergänzungssprache zu HTML entwickelt, die sich nahtlos in HTML einbinden lässt. Mit der Verwendung von Stylesheets können HTML-Elemente beliebig formatiert werden. Weiter erlauben Stylesheets das punktgenaue Platzieren von Elementen am Bildschirm oder für andere Ausgabemedien (z.B. Drucker). Mit CSS besteht die Möglichkeit, Formate zentral zu definieren, diese in eine externe "Style-Datei" auszulagern und damit für die Einbindung in unzählige HTML-Dateien verfügbar zu machen. Stylesheets erleichtern so die Arbeit mit einheitlichen Formatvorgaben, und der HTML-Code wird von unnötigem Ballast befreit.

4 Web2.0, DeepWeb, DarkNet

4.1 Web 2.0, ein Schlagwort

Der Begriff Web 2.0 suggeriert zwar eine technologische Neuerung bzw. eine neue Version des Webs, ist aber in Wirklichkeit ein Schlagwort, das verschiedene Neuerungen im Bereich der Webtechnologien zusammenfasst. Ihre Gemeinsamkeit ist, dass das Web keine statische Sammlung von Hypertextdokumenten mehr sein soll, sondern ein Konglomerat von dynamischen Inhalten, die unter anderem auch von den Web-Nutzern stammen können (*Social Network*).

4.2 Abgrenzung Webseite - Webapplikationen

Bereits seit Mitte der 90er Jahre können Browser nicht nur HTML-Seiten anzeigen, sondern beherrschen auch eine ganze Menge anderer Sprachen und Dokumentformate (JavaScript, Flash, Java, aber auch JPEG, GIFF, SVG etc.). Damit können unterschiedliche Dokumenttypen angezeigt werden. Unter Ausnützung der Fähigkeit des Browser, Programmsprachen zu interpretieren, können aber auch ganze Applikationen, also Softwareanwendungen, im Browserfenster ausgeführt werden.

Bei Applikationen, die im Browserfenster laufen, ist zu unterscheiden zwischen rein browserbasierten Applikationen, welche die Fähigkeit des Browsers nutzen, eine Programmiersprache zu interpretieren, und Anwendungen, die auf einem sogenannten Applikationsserver, einem spezialisierten Webserver laufen.

Im ersten Fall wird ein Programm als Grundlage der Applikation von einem (Web-)Server geladen und autonom im Browser ausgeführt. Dazu gehören Java-Applikationen, aber auch die meisten Apps auf Mobilgeräten. Die Abgrenzung zu Webseiten ist relativ einfach.

Die zweite Klasse der Applikationen nutzt die Fähigkeit des Browsers, Formulare zu verarbeiten und deren Inhalt an einen Applikationsserver zu schicken. Damit wird das Browserfenster zur Eingabemaske einer Applikation; auch Datenausgaben/Reports können so erfolgen. Hier läuft die Applikation auf einem Applikationsserver; dem Browser kommt nur die Rolle zu, Eingabe und Ausgabe anzuzeigen (man spricht dabei von *Thin Client*). Obwohl das, was vom Browser dargestellt wird, in der Regel HTML-basiert ist, ist es doch kein Hypertextdokument. Die Abgrenzung ist in diesem Fall aber schwierig, weil wir es auf der Browserseite mit dem Hypertextformat HTML zu tun haben, und weil heute in der Regel auch Hypertextdokumente nicht im klassischen Sinn Dokumente sind, die auf einem Webserver liegen, sondern von einer Applikation auf dem Webserver oder hinter dem Webserver jeweils erst bei einer Anfrage generiert werden.

4.3 Dynamische Webseiten

Dynamic HTHL oder DHTML brachte schon in den 90er Jahren die Möglichkeit, HTML-Seiten aufgrund einer Benutzeraktion dynamisch zu verändern. Ein beliebtes Beispiel dafür sind die ausklappbaren Menus. Die dynamischen Inhalte sind aber bereits Teil der ursprünglich vom Webserver ausgelieferten HTML-Seite, sie werden in der Regel mit JavaScript im Browser erst unterschiedlich dargestellt.

Für die Archivierung von Bedeutung ist, dass eine DHTML-Seite, so wie sie vom Webserver ausgeliefert wird, schon sämtliche Darstellungsmöglichkeiten enthält.

4.4 Web 2.0 im engeren Sinne

Web-2.0-Lösungen im engeren Sinne setzen eine dynamische Interaktion zwischen der HTML-Seite im Browser und einem Server voraus. Mit JavaScript werden unterschiedliche Teile der HTML-Seite vom Server geladen und je nach Bedürfnis oder Benutzerinteraktion die Browserseite teilweise neu geschrieben.

Im Gegensatz zu DHTML müssen nicht alle Informationen bereits beim Anfordern der Seite vom Webserver zum Browser übertragen werden, sondern werden erst bei Bedarf nachgeladen. Andererseits ist es auch nicht mehr nötig, bei einer *Thin-Client*-Lösung für jedes Formular oder jede Ausgabeseite eine neue HTML-Seite vom Server anzufordern. Das schwerfällige Anfordern und Darstellen ganzer Webseiten der ersten Webserver-basierten Applikationen entfällt.

Für diese Technik sind eigene spezielle Frameworks entstanden, der Oberbegriff dafür ist Ajax (ein Akronym für *Asynchronous JavaScript and XML*); es bezeichnet das Konzept der asynchronen Datenübertragung zwischen einem Browser und einem Server.

Die Problematik für die Webarchivierung ist hier, dass der Informationsgehalt solcher Ajax-basierter Seiten nie oder schwer vollständig zu erfassen ist, weder im Browser noch auf dem Webserver.

4.5 JavaScript

JavaScript wurde Anfang der 90er Jahre als Browser-Programmiersprache entwickelt und hat nichts mit Java zu tun. Ursprünglich verpönt als Möglichkeit auf Clientseite, also im Browser, Programme ausführen zu können, hat sich JavaScript zu einem der drei Pfeiler des Web 2.0 entwickelt (neben HTML und CSS).

Für die Webarchivierung ist ein wichtiger Aspekt, dass es mit JavaScript möglich ist, die Webseite nach dem Laden im Browser zu verändern, dass also der vom Webserver heruntergeladene HTML-Datenstrom nicht unbedingt jeweils zur gleichen Darstellung im Browser führen muss.

4.6 DeepWeb, PrivateWeb und DarkNet

Drei weitere Begriffe tauchen oft in Zusammenhang mit Web 2.0 auf und sollen deshalb hier kurz erläutert werden:

Deep Web meint in der Regel datenbankgestützte Webapplikationen, die aufgrund einer Datenbankabfrage nahezu beliebig viele Webseiten generieren können. Ein Crawlen oder Archivieren ist aufgrund der grossen Anzahl Seiten in der Regel nicht möglich oder sinnvoll. Beispiele sind ein Telefonbuch (= grosse Zahl von Seiten) oder die Fahrplanauskunft (=beliebig viele Kombinationen möglich). *Deep Web* verwendet heute oft dieselben Techniken, die wir von Web 2.0 kennen.

Private Web bezeichnet einen Webauftritt, der nicht von aussen zugänglich ist. Das kann ein passwortgeschützter Bereich eines Webauftritts sein oder aber eine Intranet-Domäne, die vom Internet aus gar nicht zugänglich ist.

Dark Net schlussendlich bezeichnet den Bereich des Internets, der insofern nicht öffentlich zugänglich ist, als dass es keine Verzeichnisdienste für diesen Bereich gibt und mögliche Nutzer nur individuell Zugang erhalten können. Die Daten werden in der Regel auch verschlüsselt, womit *Dark-Net*-Netzwerke einen hohen Grad an Anonymität und Diskretion bieten.

5 Objekt der Archivierung

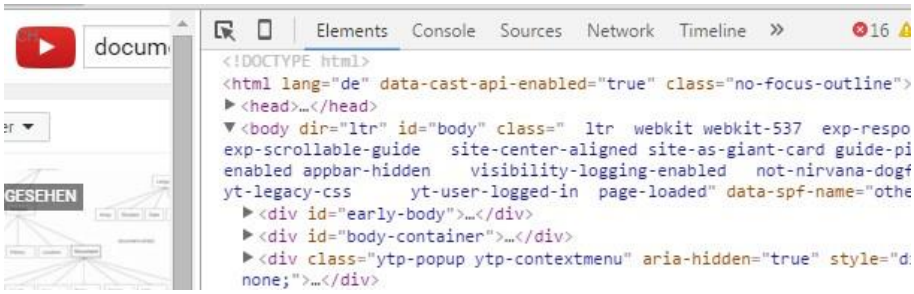
Bei der Archivierung von Webauftritten stellt sich die Frage, was und wo im Web denn eigentlich das Objekt der Archivierung ist:

- auf dem Applikationsserver im CMS?
- im Webserver, wo die HTML-Seiten entstehen und abgeschickt werden?
- im Browser, wo verschiedene Ressourcen, gesteuert durch ein HTML-Dokument, ein DOM-Objekt *(Document Object Model)* aufbauen?
- oder erst bei der Darstellung als eigentliches Dokument im Browserfenster?

Können das *Performance Model* des *National Archives of Australia* (Source – Process – Performance)[3], aber auch die Unterscheidung in OAIS zwischen *Object* und *Representation Information* hier als Erklärungsmodell dienen und die Realität im Web abbilden?

5.1 Document Object Model

Das *Document Object Model* (eigentlich trifft *Document Object Map* besser zu) bildet eine Internetseite in einer hierarchischen Struktur ab.



Firefox *Document Object Model* Ansicht

Es wird vom Browser gebildet und ist eine Bedingung für die Anzeige der kompletten Website. Es ist abrufbar mit der F12-Taste, Tab „Elements“.

Der Browser lädt neben dem HTML weitere zur Anzeige nötige Ressourcen (*Representation Information*) ins DOM, z.B. Listenfunktionen, Stylesheets, Animationen, und fügt auch diese als Objekte der Struktur hinzu.

Dank der Standardisierung des DOM können mit einer Programmiersprache wie JavaScript Änderungen direkt am DOM vorgenommen und damit dynamische Seiteninhalte erzeugt werden.

5.2 Fazit

Um zu einem Archivformat zu gelangen, reicht es also nicht, das HTML zu archivieren und die Files herunterzuladen, sondern es muss mit einem Crawler das ganze DOM abgegriffen werden.

[3] Siehe Helen Heslop, Simon Davis, Andrew Wilson, *An Approach to the Preservation of Digital Records*. December 2002. http://www.naa.gov.au/Images/An-approach-Green-Paper_tcm16-47161.pdf.

6 Significant Properties

Das Konzept der *Significant Properties* (als Synonym wird auch der Begriff *Significant Characteristics* verwendet) wurde im Zusammenhang mit den folgenden Thesen eingeführt:

- Um die Authentizität eines digitalen Objektes prüfen und belegen zu können, ist es unabdingbar, diejenigen Eigenschaften des Objektes, von denen die Authentizität abhängt, zu identifizieren und messbar zu machen. Dieselben Voraussetzungen gelten für die Durchführung von Erhaltungsmassnahmen.
- Sämtliche Eigenschaften eines Objektes zu erhalten, ist, wenn überhaupt möglich, teurer, komplizierter und mit mehr Risiko behaftet als die ausschliessliche Erhaltung der *Significant Properties*.

Significant Properties sind in diesem Sinn

> *the characteristics of digital objects that must be preserved over time in order to ensure the continued accessibility, usability, and meaning of the objects.*[4]

Während über die Bedeutung des Begriffs grosse Einigkeit besteht, sind zwei wichtige andere Punkte in der Diskussion problematisch und umstritten:

- Die Benennung von *Significant Properties* für eine Formatklasse ist an sich schon schwierig. Sie wird zusätzlich erschwert durch die Erkenntnis, dass es nicht möglich ist, eine Formatklasse mit einer Liste von fixen Attributen zu versehen, da Attribute durch das sich laufend ändernde Benutzerumfeld einem kontinuierlichen Wandel unterliegen. Verschiedene Projekte, unter anderen *InSPECT*[5] und *nestor*[6], haben deswegen versucht, die Einordnung gewisser Properties als *significant* von der geplanten Benutzung oder von den voraussichtlichen Benutzungsgruppen abhängig zu machen. Durch Betrachtung der Aufgabe eines Objektes im Zusammenspiel mit dessen Benutzern kann die Funktionalität des Objektes und daraus folgend diejenigen Eigenschaften ermittelt werden, die notwendig sind, die langfristige Verwendbarkeit des Objektes zu garantieren.
- Die Forschung zum Thema *Significant Properties* zielt in erster Linie auf eine Automatisierung von Preservation-Prozessen ab, welche durch Messung und Bewertung der *Significant Properties* wesentlich begünstigt werden könnte. Nun ist jedoch gerade die Bewahrung der wichtigsten Property eines digitalen Dokuments, nämlich des eigentlichen Inhalts, nur schwer maschinell zu verifizieren. Es besteht deshalb die Gefahr, die Anstrengungen nur auf die maschinell messbaren S*ignificant Properties* zu beschränken und somit am Ziel vorbeizugehen.

Leider sind Hypertextdaten in der bisherigen Forschung zu *Significant Properties* kaum untersucht worden. Da eine solche Untersuchung den Rahmen dieser Studie sprengen würde, folgen hier – quasi als Startpunkt für die mögliche Weiterarbeit – einige Ideen für Kandidaten von *Significant Properties* bei Hypertextdaten:

- Eine Kombination der Attribute aller Teile bzw. Unterformate eines Hypertextes, falls vorhanden (z.B. bei Webseiten): Text, Rasterbilder, Audio, …
- Verlinkungen, zum Beispiel in Form eines gerichteten Graphen (Links müssen gegebenenfalls angepasst werden, damit sie noch funktionieren).
- Seitenaufbau, falls das Objekt aus verschiedenen Teilen besteht, deren Reihenfolge relevant ist aber nicht anderweitig festgehalten wird (z.B. auf einer Webseite die Anordnung von Text, Bild, Text wenn entsprechende Verweise im Text auf die Bilder fehlen).

[4] Andrew Wilson, *Significant Properties Report*, InSPECT Project 2007, p.8. http://www.significantproperties.org.uk/wp22_significant_properties.pdf

[5] http://www.significantproperties.org.uk/index.html

[6] Siehe nestor-Arbeitsgruppe Digitale Bestandserhaltung, *Leitfaden zur digitalen Bestandserhaltung. Vorgehensmodell und Umsetzung, Version 2.* Nestor Materialien 15. Frankfurt am Main 2012. http://nbn-resolving.de/urn:nbn:de:0008-2012092400

7 Bewertung und Übernahme von Webseiten

7.1 Einleitung

Eine Reihe von Fragen müssen bei der Webarchivierung von der Bewertung beantwortet werden. Als Erstes aber muss entscheiden werden, ob die im Webauftritt vermittelten Inhalte originär nur auf dem Web verfügbar sind, oder ob der Webauftritt nur ein sekundäres Abbild einer anderen Repräsentationsform darstellt, ob die Inhalte also nicht eigentlich primär als Dokumente in einem GEVER-System geführt werden und von dort zum Archiv gelangen.

7.2 Erscheinungsform des Webauftritts

Webauftritte können heute unter der gleichen URL unterschiedliche Erscheinungsformen haben, z.B. mobile Seite, Desktop-Seite, Reader-Ansicht, oder Drucker-Version. Die Ansichten werden durch verschiedene Parameter gesteuert (Browsertyp, Seitenbreite etc.). Bei der Bewertung muss auch festgelegt werden, welche Erscheinungsform archiviert werden soll.

7.3 URL und Domäne

Ein Webauftritt wird in der Regel durch eine gemeinsame Domäne (*Domain*) oder Subdomäne als Teil der URL definiert (Domäne: www.kost-ceco.ch, Subdomäne: www.bar.admin.ch). Nicht mehr sehr gebräuchlich ist die Möglichkeit, dass eine Teil-URL einen Webauftritt festlegt (www.yahoo.com/myweb/~mustermann). Das Ergebnis der Bewertung ist demzufolge eine Liste von Domänen oder URLs und/oder eine Anweisung, welche eine Liste von URLs generiert, zusammen mit einer Angabe zur Periodizität, in der die Archivierung durchgeführt werden soll. Der Planung der Periodizität kommt bei der Webarchivierung deshalb eine besondere Bedeutung zu, weil es nach der Bewertung keine Ablieferung im klassischen Sinn gibt: Ein Webauftritt ist nach der Archivierung genauso vorhanden wie vorher.

7.4 Verschiedene Methoden der Webarchivierung

Es gibt drei mehr oder weniger verbreitete Methoden, wie ein Webauftritt archiviert werden kann. Jede dieser Formen wird von unterschiedlichen Tools unterstützt.

7.4.1 Die klassische Webarchivierung

Die Idee bei der klassischen Methode ist, einen Webauftritt zu einem bestimmten Zeitpunkt möglichst vollständig zu erfassen. Die Bewertung erfolgt also allein auf der Aussage, dass ein gesamter Webauftritt archivwürdig ist.

Die klassische Methode ist das Crawlen eines Webauftritts. Dabei werden ausgehend von einer Start-URL (die nicht notwendigerweise die Domänen-URL sein muss) alle Links verfolgt und die gefundenen Webseiten aufgezeichnet. Um zirkuläres Crawlen oder Crawlen in unrealistische Tiefe zu unterbinden, wird festgelegt, wie tief gecrawlt werden soll, d.h. wie manchem Link (von der Start-URL aus gezählt) gefolgt werden soll. Es können auch mehrere Start-URL verwendet werden. Im Prinzip entsteht beim Crawlen eine Baustruktur aller besuchten Webseiten, ähnlich einem hierarchischen Verzeichnisbaum, deren Wurzel die Start-URL und die Länge derer Äste durch die Crawltiefe bestimmt ist.

Zu beachten ist, dass bei diesem Vorgehen ein Webauftritt selten wirklich vollständig erfasst werden kann, weil immer auch Formulare oder dynamische Elemente eingebaut sein können, und dass nur ein gewissermassen unbestimmter Zeitstand erfasst wird. Mutationen können gleichzeitig mit dem Crawlen stattfinden.

Diese Methode führt zu einer weitgehend dokumentarischen Archivierung

7.4.2 Die kuratierte Webarchivierung

Die Idee bei der Methode mit manueller Bewertung ist die, dass eine ausgewählte Liste von URLs archiviert wird. Die Liste kann entweder von Hand, das heisst durch Bewerten jeder einzelnen Seite, durch Analyse der URL oder durch Suche ermittelt werden. Der Vorteil ist hier, dass die Menge der zu archivierenden Seiten klein bleibt und es quasi eine kuratierte Archivierung ist; der Nachteil, dass bei relativ häufigen Änderungen im Webauftritt viele URLs in der Liste ungültig werden oder auf einen anderen Inhalt zeigen. Durch die relativ bescheidene Zahl der ausgewählten Seiten kann der Zeitpunkt der Archivierung genauer festgelegt werden.

Diese Methode entspricht am ehesten der gezielten Archivierung bewerteter Quellen.

7.4.3 Die Compliance-Archvierung

Ziel dieser Archivierung ist es, alle Seiten oder ausgewählte Seiten eines Webauftritts über einen bestimmten Zeitraum festzuhalten und dabei auch sämtliche Veränderungen in diesem Zeitraum zu dokumentieren. Wir nennen sie Compliance-Methode, weil damit der Nachweis für einer Veröffentlichungspflicht oder Veröffentlichungsabsicht erbracht werden kann.

Sämtliche Änderungen über einen bestimmten Zeitraum in einem Webauftritt zu verfolgen, ist technisch recht anspruchsvoll. Entweder werden Änderungen an der Quelle verfolgt, also beim Arbeiten mit dem Webautorensystem bzw. Content Management System festgehalten. Oder es wird der Webserver beim Publizieren der Webseiten verfolgt und jede ausgelieferte Webseite mit den bereits bekannten Seiten verglichen und so Änderungen entdeckt und festgehalten.

Diese Methode ist mehr eine Überwachung der Webquelle als ein Festhalten oder Dokumentieren eines Zustandes

7.5 Typologie des Webauftritts

Die Typologie des Webauftritts legt fest, welche Methoden bei der Bewertung konkret angewendet werden müssen und können und wie die Übernahme anschliessend ausgeführt wird.

7.5.1 Hierarchische URL

Ursprünglich war ein Webauftritt eine Abbildung einer Dateiablage, und die URL der einzelnen Seiten repräsentierten die Verzeichnisse in der Dateiablage, welche inhaltlich unterschiedliche Bereiche auf dem Webserver enthielten. Die Bezeichnung ist heute nicht mehr ganz korrekt, da in der Regel keine Dateiablage mehr vorhanden ist. Repräsentiert die ganze URL einen Verzeichnisbaum, spricht man deshalb von einem REST (*Representational State Transfer*) Server. REST fordert, dass eine URI (Adresse) genau einen Seiteninhalt repräsentiert. Für unsere Belange ist entscheidend, dass die URL der Seiten gewisse statische Teile enthält, mit Hilfe derer wir die Seiten inhaltlich gruppieren können.

Beispiel:
https://www.bar.admin.ch/bar/de/home/ueber-uns/Medienmitteilungen.html. Seiten vom Typ https://www.bar.admin.ch/bar/de/home/ueber-uns/Medienmitteilungen* sind inhaltlich dem Navigationsreiter "Medienmitteilungen" untergeordnet, müssen also archiviert werden, wenn wir alle Informationen zu diesem Thema als archivwürdig betrachten.

7.5.2 URL mit Parameter

Die URLs, welche Webseiten eines Webauftritts bezeichnen, können auch völlig dynamisch generiert werden. Die URL ist dann ohne jede Aussage über den Inhalt der Webseite, und es gibt keine Möglichkeit, die Seiten nach Inhalt auszuwählen, ausser sie einzeln anzuwählen oder zu suchen (z.B. mit Google Site Search).

Beispiel: http://kost-ceco.ch/cms/index.php?id=104,125,0,0,1,0. Die URL ist parametrisiert, der Webserver erkennt am mitgegebenen Parameter, welche Seite gewünscht wird, die Basis-URL bleibt immer die gleiche, nämlich http://kost-ceco.ch/cms/index.php.

7.5.3 Dynamisch generierte URL

Diese URLs werden durch Benutzereingabe via Formular oder einfach durch Aufruf der Seite jedes Mal verändert. In gewissen Fällen kann die URL auch nach einer gewissen Zeit ungültig werden (*Secure URL*). Ein Auffinden der Seiten ist also nur durch manuelles Eingeben und Ausfüllen der Formulare möglich, ein Crawlen oder Suchen bringt nicht den gewünschten Erfolg. Bei *Secure URLs* ist ein Archivieren sogar nur im Augenblick möglich, es kann keine Liste mit den bewerteten URLs angelegt werden.

Beispiele: http://www.stopmasern.ch/de-ch/masern-risiko-check.html?start=1&absenden=Hier+gehts+zum+Risiko-Check. Diese Seite navigiert auf Grund der eingegebenen Daten.

https://www.egate.admin.ch/irj/portal?NavigationTarget=navurl://adcbaa8c2457b05725c3d78b56ed3e5b&ExecuteLocally=true. Diese Seite verfällt nach einer gewissen Zeit.

7.6 Bewertung und Übernahme als Matrix

Die gewünschte Archivierungsmethode und die Typologie des Webauftritts verbinden sich zu einer Matrix möglicher Kombinationen. Jede Kombination verlangt bei der Bewertung, aber auch bei der technischen Umsetzung der Übernahme ein anderes spezifisches Vorgehen, wobei gewisse Fälle zusammenfallen können.

	Klassische Webarchivierung	Kuratierte Webarchivierung	Compliance-Archivierung
Hierarchische URL	(1) (a)	(2) (b)	(2) (e)
URL mit Parameter	(1) (a)	(3) (c)	(1) (f)
Dynamisch generierte URL	--	(4) (d1)	(1) (d2)

7.7 Bewertung

Beschreiben wir jetzt das Vorgehen der Bewertung in den einzelnen Fällen im Detail (1 – 4).

7.7.1 (1) Bewertung der Domäne

In diesem Fall können nur die Domäne und der damit verbundene Webauftritt als Ganzes als archivwürdig oder nicht archivwürdig bewertet werden. Zudem kann oder muss eine Aussage über die Periodizität der Archivierung gemacht werden. Im Falle der Compliance-Archivierung bestimmt die Periodizität die Dauer der Beobachtung.

Beispiel:
Start-URL: http://www.bar.admin.ch/
Periodizität: halbjährlich

7.7.2 (2) Bewertung einer URL

Die Bewertung besteht hier aus einer Liste von URLs, welche die archivwürdigen Seiten bezeichnen. Die Liste basiert auf einer Struktur des Webauftrittes, vergleichbar mit einer Dateiablage, und enthält Platzhalter. Die Struktur des Webauftrittes ist oft recht einfach aus der Navigationsstruktur auslesbar.

Bei der wiederholten Archivierung muss wenn möglich automatisch überprüft werden, ob die Struktur noch unverändert ist.

Beispiel:
Domäne: http://www.bar.admin.ch/
Selektierte URLs: http://www.bar.admin.ch/dienstleistungen/* http://www.bar.admin.ch/aktuell/*
Periodizität: halbjährlich

7.7.3 (3) Bewertung durch Suchen

Die Bewertung besteht hier aus einer dynamisch generierten Liste der archivwürdigen Webseiten. Eine Suchabfrage zusammen mit der Domäne (*site search*) ergibt jeweils einen Teil dieser Liste von URLs. Die Bewertung legt fest, wie viele Treffer maximal pro Suchabfrage verwendet werden sollen. Die Suchabfragen werden erst zum Zeitpunkt der Archivierung ausgeführt. Veränderungen im Aufbau der URL zwischen Bewertung und Archivierung können so aufgelöst werden.

Beispiel:
Domäne: http://www.bar.admin.ch/
Suchbegriffe: Dienstleistungen, Beratung, Weiterbildung Medienmitteilungen, News, Aktuelles z.B. *Google site search*
https://www.google.ch/search?q=site:bar.admin.ch+Dienstleistungen
https://www.google.ch/search?q=site:bar.admin.ch+Beratung
Anzahl Treffer maximal: 100
Periodizität: halbjährlich

7.7.4 (4) Bewerten durch Klicken

Hier ist es für die Bewertung notwendig, jede Seite mit dem Browser anzusteuern, allenfalls Formulare auszufüllen und zu entscheiden, ob die sichtbare Seite archivwürdig ist oder nicht.

Es gibt Tools, die entweder den gesamten Browserverlauf aufzeichnen, einzelne Seiten im Browserverlauf taggen und so festhalten (ähnlich wie beim Speichern von Bookmarks) oder von jeder besuchten Seite einen Screenshot anlegen.

Beispiel:
Start-URL: http://www.stopmasern.ch/de-ch/masern-risiko-check.html?start=1&absenden=Hier+gehts+zum+Risiko-Check
Methode: Screenshot in PDF von jeder Seite
Periodizität: halbjährlich

7.8 Übernahme

Nun beschreiben wir das eigentliche Vorgehen bei der Übernahme, soweit das für das Gesamtverständnis notwendig ist (a - f):

Im Prinzip kann die eigentliche Übernahme vom Anbieter des Webauftritts oder vom Hosting-Dienstleister oder vom Archiv selber durchgeführt werden. Abhängig ist das von den institutionellen Vorgaben und den technischen und personellen Voraussetzungen. Bedingt durch die Internettechnologie ist der Ort der Archivierung nicht an den Ort gebunden, wo der

Webauftritt entsteht (Webserver), ausser es handelt sich um einen Webauftritt in einem abgeschlossenen Intranet.

Bei der Übernahme können wir zwei Schritte unterscheiden: das Harvesting (das eigentliche Einsammeln der Webseiten) und das Umwandeln und Ablegen in einem archivtauglichen Format. Die beiden Schritte können softwaretechnisch in einer Lösung zusammengefasst werden. Wir unterscheiden sie aber, weil der zweite Schritt unabhängig von der Bewertung erfolgt und alleine bestimmt ist durch den Grundsatzentscheid, in welchem Format Webauftritte archiviert werden sollen. Nach heutigem Stand der Erkenntnis kommen dafür WARC oder PDF/A-2 in Frage. Die Ausgestaltung der Archivierungsfälle (a – f) ist davon nicht betroffen.

7.8.1 (a) Webauftritte crawlen

Das *Crawlen* ist mit Sicherheit der bekannteste Weg der Webarchivierung. Eine *Crawler*-Software holt, beginnend mit einer von der Bewertung festgelegten Start-URL, sämtliche Webseiten, auf welche Links von der aktuellen Seite verweisen. Gespeichert und weiterverfolgt werden nur Seiten, welche der vorgegebenen Domäne oder Subdomäne entsprechen. In der Regel werden Bilder, Scripts, CSS etc. auch von andern Domänen geholt. Angedockt an den *Crawler* ist der *Harvester*, der sämtliche gelesenen Dateien in geeigneter Form speichert. Im Falle der Speicherung in PDF ist als dritte Komponente noch ein *Rendering Agent* (im Prinzip ein Browser) notwendig, der die gelesenen Dateien als Webseite aufbereitet und in PDF abspeichern kann.

Für diese Art der Webarchivierung gibt es sehr viele gängige Softwarelösungen am Markt. Im Prinzip ist jeder Browser ein Harvester und Rendering Agent, Microsoft Internet Explorer kann auch crawlen (Funktion: Ausdrucken aller verbundenen Seiten). Die fortlaufende Entwicklung der Webtechnologien macht es aber notwendig, dass der Crawler technisch immer auf dem neusten Stand ist.

7.8.2 (b) URL-Liste einsammeln

Definiert die Bewertung eine Liste von URLs ohne Platzhalter (*Wildcards*), findet nur ein Harvesting der Webseiten statt. Technisch ist das mit einem Crawlen mit der Tiefe 1 gleichzusetzen, siehe deshalb die Bemerkungen zu (a). Enthält die URL-Liste Wildcards wie in unserem Beispiel, ist ein Crawlen wie in (a) notwendig, die Liste mit den Wildcards dient als Start-URL. Von allen aufgefunden URLs werden aber nur diejenigen weiterverfolgt und archiviert, die dem Wildcard-Muster entsprechen. Zu beachten ist, dass das Ergebnis ein unvollständiges Abbild des Webauftritts darstellt und nicht alle Links in den Webseiten innerhalb der archivierten Daten aufgelöst werden können. Diese Links verweisen dann aus dem archivierten PDF/A oder WARC-File auf den ursprünglichen Webserver.

7.8.3 (c) Treffer der Websuche einsammeln

Hier ist eine etwas innovativere Lösung erforderlich. Verschiedene Anbieter von Suchmaschinen stellen aber die notwendigen Schnittstellen für die gezielte Suche in einer Domäne zur Verfügung (siehe z.B. *Google Site Search*). Das Ausführen der Suchabfragen führt zu einer Liste von URLs, die nächsten Schritte sind analog zu (b).

7.8.4 (d) Einsammeln beim Klicken

Hier wird zur Bewertung durch den Webauftritt navigiert und beim Klicken von einer Seite zur nächsten, die besuchten Seiten eingesammelt. (d1) und (d2) unterscheiden sich im Ort, wo das Klicken beobachtet wird und wo die Aufzeichnung stattfindet.

Bei (d1) wird das Navigieren auf der Client-Seite aufgezeichnet, also bei der Bewertung selber. Es gibt dafür entsprechende Softwaremodule für den Browser, die dieses Tracking erlauben. Aufgezeichnet werden nur die Seiten, die bei der Bewertung besucht worden sind.

Bei (d2) wird das Navigieren beliebiger Benutzer auf der Serverseite über eine bestimmte Periode aufgezeichnet. Der beliebige Benutzer kann auch die Bewertung sein. Dazu ist ein spezielles Softwaremodul auf der Webserver- oder Proxy-Seite notwendig, das alle ausgehenden Seiten aufzeichnet. Alternativ kann auch das sog. *Webserver Access Log*, das alle Anfragen an den Server aufzeichnet, zeitversetzt so ausgewertet werden, dass eine Liste von URLs zur Verfügung steht, die dann nach (b) abgearbeitet wird.

7.8.5 (e) URL-Liste beobachten

Hier wird auf der Webserver-Seite eine kuratierte Liste von URLs nach der Bewertungsregel (2) in einer bestimmten Periode beobachtet. Damit eine Vollständigkeit erreicht wird, muss zu Beginn der Beobachtungsperiode ein Crawlen

der Webseite nach (b) stattfinden, da nicht mit Sicherheit davon ausgegangen werden kann, dass alle vorgängig von der Bewertung ausgewählten oder durch Wildcard-Muster definierten URLs auch jemals abgerufen werden. Ein Harvesting nach (b) ist nicht nötig, da alle durch das Crawlen ausgelösten Auslieferungen von Webseiten auf der Serverseite festgehalten werden. Nach dem initialen Crawlen beginnt das Aufzeichnen aller neuen Abfragen, festgehalten werden auf der Serverseite jeweils nur veränderte Webseiten.

7.8.6 (f) Domäne beobachten

Es handelt sich um eine Variante zu (e) ohne kuratierte Liste von URLs. Da das primäre Interesse dabei die öffentlich wahrgenommenen Webseiten sind, wird wohl in der Regel auf ein initiales Crawlen verzichtet und werden nur die zugegriffenen Webseiten aufgezeichnet. Auch hier wird bei jeder neu abgerufenen Seite jeweils geprüft, ob eine identische Seite in der beobachteten Periode bereits archiviert worden ist.

8 Crawlen und Harvesting

Um zu verstehen, wie ein Crawler funktioniert, ist es hilfreich, einige Grundlagen zur Funktion von Client, Server und dem Protokoll http zu kennen.

Bei einem Abruf einer Webseite passiert grob beschrieben folgendes:

Der Browser (in diesem Fall der Client) fragt beim Webserver eine Webseite an. Dazu schickt er eine Anfrage (request) mit der entsprechenden URL (z.B. http://www.bar.admin.ch/bar/de/home.html) an den Webserver. Die URL besteht aus den Teilen Protokoll (http), Hostname (www.bar.admin.ch) und Pfad (/bar/de/home.html). Zusätzlich können auch noch Port, Query String und Anker vorhanden sein, worauf hier aber nicht eingegangen wird. Ebenso ausgelassen wird hier die Methode der Anfrage (GET, POST und weitere).

Mit Hilfe dieser drei Angaben weiss der Webserver, was er dem Client zurückliefern soll und auf welche Art. Geliefert werden kann alles Mögliche (HTML-Webseite, CSS-File, Bild, JavaScript-File, Video, ausführbare Programme, Viren, ...).

Anhand des vom Webserver mitgelieferten MIME-Types weiss der Browser nun, wie er die Antwort interpretieren muss (Rendern von HTML, Anzeigen mittels Plugin, abspeichern).

Im Falle einer Webseite (HTML) interpretiert der Browser diese, um sie darstellen zu können. Dabei werden üblicherweise im Code der Seite weitere Ressourcen gelistet, die vom Browser geholt werden sollen (CSS, JavaScript, Bilder und weiteres). Bevor der Browser die Seite vollständig darstellen kann, muss er für jede dieser Ressourcen den beschriebenen Prozess separat noch einmal durchlaufen.

Ressourcen, welche der Benutzer als Links sieht, sind hiervon jedoch ausgenommen. Im Normalfall holt der Browser diese erst, wenn der Benutzer darauf klickt.

Beim Holen und Anzeigen einer Webseite werden normalerweise also Dutzende von einzelnen Requests vom Browser an den Server geschickt und anschliessend zusammengesetzt. Dabei muss nicht jeder Request an den Ursprungsserver gehen. Es kann auch externer Content in eine Seite eingebettet sein, ohne dass der Benutzer etwas davon merkt. Gängige JavaScript-Bibliotheken, CSS oder Fonts werden zudem oft von sogenannten *Content Delivery Networks* geholt. So muss ein Browser nur auf der allerersten Seite eine solche Bibliothek holen, und hat sie für alle weiteren Webseiten bereits im Cache.

Crawler und Harvester arbeiten nach dem gleichen Prinzip wie ein Browser und gehen ebenfalls nach dem oben beschriebenen Ablauf vor. Sie zeigen am Schluss die geholten Webseiten aber nicht an, sondern sie prüfen, katalogisieren, normalisieren oder speichern sie. Im Unterschied zu einem Harvester, dessen Aufgabe es ist, einen definierten Bereich des Webs zu holen und abzuspeichern, wird ein Crawler dazu verwendet, Informationen über einen Bereich des Webs einzuholen. Er kann beispielsweise eine Sitemap erstellen oder auf eine effiziente Weise prüfen, ob die Ziele von Links noch existieren, oder nachfragen, wann eine Seite das letzte Mal geändert wurde. Für die letzteren zwei Aufgaben reicht es, wenn ein Crawler nur den Header einer Webseite (also gewissermassen die Metadaten) und nicht die ganze Webseite holt. Dies kann er mit einem HEAD request tun (anstatt mit den oben genannten Methoden GET und POST).

Beim "normalen" Browsen mit einem Webbrowser kann (und soll) jeder nicht-passwort-geschützte Bereich einer Webseite angesehen werden können. Beim maschinellen Abfragen ist dies aber nicht immer gewünscht. Um Maschinen (also z.B. Crawlern oder Harvestern) mitzuteilen, dass sie gewisse Bereiche einer Webseite nicht abfragen sollen, existiert der *Robots exclusion standard*[7]. Dabei wird in einem Textfile mit Namen `robots.txt`, welches jeweils zuoberst in einer Webseitenhierarchie liegt, aufgelistet, welche Bereiche für einen Crawler oder Harvester erlaubt und welche verboten sind. Diese Auflistung ist zwar lediglich eine Richtlinie (die "verbotenen" Bereiche sind technisch nicht geschützt, da sie ja von "normalen" Webbesuchern angesehen werden können sollen), trotzdem halten sich aber Crawler und Harvester wie z.B. die Indizierer von Suchmaschinen an diese Vorgaben. Crawler und Harvester, deren Aufgabe nun aber nicht die Indizierung für Suchmaschinen ist, sondern die Archivierung von Webseiten, sollten im Normalfall die Anweisungen aus dem `robots.txt`-File ignorieren. Allenfalls ist hierzu aber auch eine Absprache zwischen Seitenbetreiber und Bewertung notwendig.

Bei der Verwendung von Crawlern und Harvestern für die Archivierung treten folgende Problemfelder auf:

- Bei der Festlegung des Bereichs, welche ein Crawler oder Harvester abarbeiten soll, können die folgenden Angaben kombiniert werden: Einstiegspunkt oder Liste von Einstiegspunkten (beides in Form von Links), Filter (White- oder Blacklists) auf Hostnamen, Pfad, Methode oder MIME-Type sowie Linktiefe bei einer rekursiven Abarbeitung. Mittels dieser Angaben den für das Harvesten gewünschten Bereich einzuschränken, kann unter Umständen schwierig oder sogar unmöglich sein, ist im besten Fall aber aufwändig und nur durch wiederholtes Probieren umsetzbar.
- Die Abbildung einer klassischen Bewertung der Archivwürdigkeit auf Webseiten kann problematisch sein, wenn sich die Bewertungselemente durch obige Methoden nicht klar trennen lassen.
- Mit obigen Methoden ist es nicht immer möglich, unerwünschte Elemente ganz vom Harvesten auszuschliessen. Problematisch können beispielsweise Navigationsbalken sein, oder auch Werbung und Social-Media-Buttons.
- Intervallfestlegung: Crawlen oder Harvesten ist grundsätzlich intervallgesteuert. Wenn also zwischen zwei Abfragen mehrere Änderungen an einer Webseite vorgenommen wurden, erkennt ein Crawler bzw. Harvester dies nur als eine einzige, übergreifende Änderung. Je kürzer das Abfrageintervall ist, desto granularer wird die Erkennung von Änderungen und desto mehr Ressourcen (Client-seitig, Server-seitig und Netzwerk-seitig) werden benötigt.

[7] Siehe https://en.wikipedia.org/wiki/Robots_exclusion_standard.

9 Anhang

9.1 Toolliste Webarchivierung

Es gibt mittlerweile eine Reihe von Werkzeugen, die teils als Open-Source-Komponenten zur Verfügung stehen. Die nachfolgende Liste wurde nach bestem Wissen und Gewissen erstellt und ist nicht vollständig. Die Beurteilung basiert teilweise auf Drittmeinungen.

Die Liste enthält nebst dem Namen mit Link auch die Herausgeber/Ersteller sowie die wichtigsten Eigenschaften.

Crawler, Harvester & Co.

Heritrix
Internet Archive
Open-Source-Crawler und -Harvester, welcher in WARC und ARC speichert.

NetarchiveSuite
Entwickelt durch die Nationalbibliotheken aus DK, FR und A
Komplettes Open-Source-Softwarepaket (Crawler, Harvester und Viewer), welches Heritrix und Wayback verwendet.

WCT - Web Curator Tool
British Library und National Library of New Zealand, von IIPC initialisiert
WCT ist ein Open-Source-Crawler und –Harvester, welcher auch für Bibliothekare handhabbar sein soll. Nutzt Heritrix und speichert in WARC oder ARC

GNU Wget
GNU
Open-Source-Harvester, welcher auch einfache Crawls durchführen kann. Bestens geeignet für http; WARC ist noch ausbaufähig.

HTTrack
Xavier Roche
HTTrack ist ein kostenloses Tool, welches speziell für das Harvesting von ganzen Webseiten geeignet ist.

HTTrack2ARC
Portuguese Web Archive
Open-Source-Konverter, welcher den HTTrack-Output in ARC konvertiert.

WARCreate
Mat Kelly
WARCreate ist eine kostenlose Google Chrome Extension und konvertiert nur eine einzelne Seite in WARC.

Scrapy
Scrapinghub
Open-Source-Harvester speziell zum Herunterladen von Daten aus Webseiten.

DeepArc
Bibliothèque nationale française und XQuark
Open-Source-Harvester zum Herunterladen von Datenbankinhalten in ein XML-Dokument

Nutch – Apache Nutch
Apache Software Foundation
Nutch ist ein Open-Source-Java-Framework für Internet-Suchmaschinen (Crawler)

Viewer

Wayback
Internet Archive
Wayback ist die Open-Source-Java-Implementierung der Internet Archive Wayback Machine.

OpenWayback
IIPC
OpenWayback ist die neue Open-Source-Version von Wayback.

NetarchiveSuite
Entwickelt durch die Nationalbibliotheken aus DK, FR und A
Komplettes Open-Source-Softwarepaket (Crawler, Harvester und Viewer), welches Heritrix und Wayback verwendet.

WebArchivePlayer
Ilya Kreymer
WebArchivePlayer ist ein einfacher Desktop-Viewer, welcher ein oder mehrere WARC und ARC-Dateien anzeigen kann.

PDF

Qumram
Qumram
Die kommerzielle Lösung von Qumram ist spezialisiert auf die rechtsichere Archivierung von Webseiten. Nebst PDF/A ist auch WARC als Output-Format möglich.

Acrobat Pro
Adobe
Der kostenpflichtige Acrobat Pro kann auch Webseiten in PDF konvertieren. Das Look&Feel ist wie bei Qumram vergleichbar mit der Webseite.

3-Heights™ Document Converter
PDF-Tools
Der kommerzielle Konverter konvertiert die Druckansicht von URLs nach PDF/A. Entsprechend gehen die Links verloren.

7-PDF Website Converter
7-pdf
Der kommerzielle Konverter konvertiert jeweils eine URL nach PDF.

PDFmyURL
PDFmyURL
Der kommerzielle Konverter konvertiert die URLs nach PDF. Links zeigen ins Internet und nicht auf die entsprechende Seite im PDF.

9.2 Archivierungsbeispiel

Als Beispiel zur Illustration der erwähnten Prinzipien soll der Webauftritt der Wasserschutzpolizei der Stadt Zürich dienen. Dies ist nebst der Hauptseite https://www.stadt-zuerich.ch/pd/de/index/stadtpolizei_zuerich/gewaesser/wasserschutzpolizei1.html jede andere Seite der Wasserschutzpolizei inkl. Anhänge soweit möglich.

9.2.1 PDF/A

Die Archivierung eines Webauftrittes in PDF/A erfolgt meist in mehreren Schritten. Zuerst wird ein PDF erstellt, dieses wird bei Bedarf noch ergänzt und am Schluss erfolgt eine Konvertierung in PDF/A.

Im Beispiel wurden die Tools Acrobat Pro von Adobe[8] und PDF/A-Manager von PDFTron[9] verwendet.

Der schnellste Weg zur Erstellung der kompletten PDF-Abbildung ist es, zuerst die Hauptseite zu konvertieren. Dabei wurde das Format A3 Hochformat gewählt, da dies die Breite des Bildschirms am besten wiedergibt und Seiten nicht unnötig zerstückelt werden.

Sobald die PDF-Seite existiert, können die zusätzlichen Seiten mit dem Befehl „An Dokument anhängen“ (rechte Maustaste auf den Link) angehängt werden. Dies erscheint zwar im ersten Moment mühsam, ist aber verglichen mit der Erstellung des WARC sehr schnell. Die ganze Domäne kann nicht ausgewählt werden, weil dies die Stadt Zürich insgesamt ist. Zu beachten ist, dass in einem PDF-Dokument anschliessend keine Seiten gelöscht werden dürfen, da sonst Links nicht mehr funktionieren.

Nachdem alle Seiten und Anhänge hinzugefügt wurden, können Fehler korrigiert werden. Im Beispiel wurde verständlicherweise ein Video nicht geladen, und der entsprechende Link fehlte. Dieser Link wurde manuell bei „Play / Pause“ hinterlegt, so dass sichtbar wurde, was fehlt.

Ganz am Schluss wurde noch das PDF in ein PDF/A konvertiert. In der Regel funktioniert dies nicht mit Adobe. In unserem Beispiel wurde dazu die Testversion von PDF/A-Manager verwendet (zusätzliches Wasserzeichen). Durch die Konvertierung wurde z.B. die Formularfunktion (Suche oben rechts) deaktiviert.

9.2.2 WARC

Verwendet man für die Erstellung der WARC-Datei ein professionelles Tool, ist die Handhabung nach der Installation um einiges einfacher. Zur Veranschaulichung wurde jedoch von jeder Webseite mit WARCreate[10] via Google Chrome eine WARC-Datei erstellt und diese einzelnen Dateien mit CAT zusammengefügt respektive aneinandergehängt. Dadurch entstanden viele Doppelspurigkeiten, von denen die grössten gelöscht wurden.

Diese WARC-Datei kann mit WebArchive-Player[11] angeschaut werden. Damit die Historisierungsfunktion getestet kann, wurden zusätzlich von der Webseite http://www.bistrot-bern.ch/ die Tagesangebote unter http://www.bistrot-bern.ch/offer an zwei unterschiedlichen Tagen in WARC konvertiert. Leider zeigt der „WebArchivePlayer“ dies nicht an (eine entsprechende Issue wurde gemeldet). Die Historisierungsfunktion ist jedoch ersichtlich, wenn die Wayback Machine[12] verwendet wird. Da die Wasserschutzpolizei teilweise auch durch das Internet Archive gesichert wurde, kann diese Funktionalität direkt in der Wayback Machine angeschaut werden[13].

9.2.3 Möglichkeiten der Versionierung

Zur Illustration verschiedener Möglichkeiten der Versionierung wurde der Webauftritt http://www.bistrot-bern.ch verwendet und in PDF/A-2 archiviert.

[8] https://acrobat.adobe.com/ch/de/acrobat.html.

[9] https://www.pdftron.com/pdfamanager/.

[10] http://warcreate.com/.

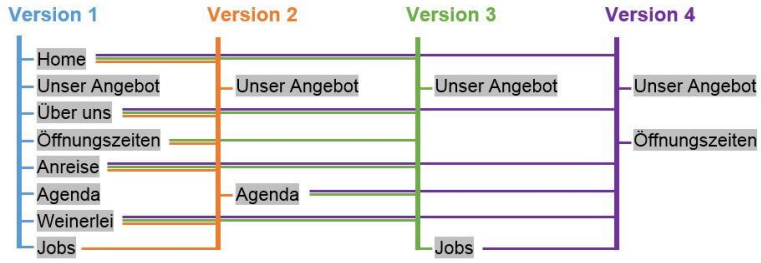
[11] https://github.com/ikreymer/webarchiveplayer.

[12] http://archive.org/web/

[13] https://web.archive.org/web/20151026203459/https://www.stadt-zuerich.ch/pd/de/index/stadtpolizei_zuerich/gewaesser/wasserschutzpolizei1.html.

Die Archivierung des Webauftritts mit Adobe Acrobat Pro in PDF respektive PDF/A ergibt für jede Version eine eigene, komplette Archivdatei. Siehe Kurzanleitung und Archivdateien in der Beilage.

Bei der Archivierung mit Qumram wird der Webauftritt in eine Serie einzelner PDF/A-Dateien gesichert. Dies hat den Vorteil, dass der Hauptbestandteil in einer neueren Version wiederverwendet werden kann, indem diese auf die bereits archivierten Webseiten verweisen.



9.3 Wer archiviert das Internet?

Als Bibliotheken mit einem gesetzlichen Archivierungsauftrag sehen die Nationalbibliothek und die Kantonsbibliotheken Webseiten grundsätzlich als (elektronische) Publikationen an und daher als aufbewahrungswürdig. In Zusammenarbeit mit den Kantonsbibliotheken stellt die Nationalbibliothek mit dem Webarchiv Schweiz[14] eine Sammlung zur Verfügung, die eine Selektion landeskundlich relevanter Websites beinhaltet. Harvesting-Anfragen von Seiten der Nationalbibliothek haben die Vorstellung erweckt, die Frage der Webarchivierung sei gelöst.

Man kann sich als Archiv nun die Frage stellen, ob es ausreichend ist, wenn die Nationalbibliothek den Webauftritt der jeweiligen kantonalen oder kommunalen Verwaltung im Sinn eines elektronischen Printwerk, archiviert und ob damit dem Anspruch auf die Nachvollziehbarkeit staatlichen Handelns Genüge getragen wird.

- Soll man die Webauftritte öffentlicher Verwaltungen als graue Literatur betrachten und die Archivierung den Kantonsbibliotheken bzw. der Nationalbibliothek überlassen?
- Soll man Webseiten der Verwaltungsorgane als elektronische Amtsruckschriften behandeln?
- Werden auf amtlichen Webseiten Inhalte präsentiert, die in den Archiven eine lange Überlieferungstradition aufweisen können und daher in die klassische Zuständigkeit des Archivs gehören?
- Wie sind verwaltungsinterne Kommunikation und Publikationen (Intranet) in diesem Zusammenhang zu bewerten?
- Ist der Internetauftritt als völlig neue Art der Information und unabhängig von bestehenden Systemen anzusehen?

Wenn sich ein Archiv dann dafür entschieden hat, eine Webseite als archivwürdig zu bewerten, wie soll das Objekt der Archivierung in das Archiv gelangen?

- Sollte es aufbereitet werden und als Ablieferungspaket (SIP) ans Archiv übergeben werden?
- Wäre es allenfalls ressourcenschonender, wenn ein Archiv (oder mehrere Archive gemeinsam) die Hilfe eines Dienstanbieters in Anspruch nimmt und eine Webseite "professionell" archivieren lassen würde?
- Soll das Archiv sich die gewünschten Objekte selber holen, das es ja den Zugriff ohnehin selber hat?
- Inwiefern handelt es sich um Unikate (die beim Aktenbildner traditionell nicht mehr vorhanden sind)?
- Gibt es überhaupt ein finales Dokument?
- Wird man künftig lediglich noch Stichtage definieren, an denen Webseiten ins Archiv gelangen?
- Inhaltliche Veränderungen einer statischen Webseite seit ihrer Erstveröffentlichung werfen schon Probleme bei der Archivierung auf, wie ist dies bei dynamischen Websites zu bewerten? Diese sind bereits bei der Erstveröffentlichung nicht abgeschlossen, sondern werden laufend aktualisiert. Das Internet bietet für die Interaktion von Bürger/innen und Verwaltung völlig neue Konzepte.

[14] http://www.nb.admin.ch/nb_professionnel/01693/index.html?lang=de.

– Wie sind die interaktiven Funktionalitäten einer Webseite zu werten? Inwieweit kann/soll die Webarchivierung diese Probleme überhaupt lösen?

Fazit

Zum heutigen Stand fehlt eine archivischen Richtlinie zur Archivierung von Webseiten. Die Ausarbeitung von Leitlinien und Prinzipien im Umgang mit Webseiten aus archivischer Sicht wäre wünschenswert.