

# **Das EAD-Kochbuch.**

**(ohne Abschnitte 4 und 5)**

## **Übersetzung des EAD Cook Book**

(Vers. 1)

**von Michael J. Fox**

Übersetzt von Angelika Menne-Haritz

Stand: Sept. 2003

URL der englischen Originalfassung:  
<http://jefferson.village.virginia.edu/ead/cookbookhelp.html>

### **Vorbemerkung zu Übersetzung:**

Das EAD Cook Book von Michael Fox ist eine praktische Anleitung für eine bestimmte Art der Kodierung von Findmitteln in EAD. Es nimmt Bezug auf die *EAD-Tag-Library* und die *Application-Guidelines* der SAA. Dort finden sich ausführlichere Erklärungen der Verwendung der einzelnen Elemente. In Ergänzung dazu wird hier gezeigt, wie sie genutzt werden können. Die beigelegten Stylesheets sind auf das hier beschriebene Verfahren abgestimmt. Alle zugehörige Software ist in der jeweils aktuellsten Version von der Website der Originalfassung des Kochbuchs herunterzuladen.

Zweck der Übersetzung ist es, weitere Information über EAD in deutscher Sprache bereitzustellen. Auch wenn das Kochbuch nicht direkt als Anleitung für die Kodierung von Findbüchern genutzt wird, zeigt es doch anschaulich, wie EAD in der Praxis einsetzbar ist und ergänzt damit die mehr systematischen Darstellungen in den anderen Texten.

Bestimmte Begrifflichkeiten können nicht direkt in die deutsche Terminologie übernommen werden, weil sie einem anderen archivischen Konzept entsprechen. Zum Teil wird deshalb versucht, eine Entsprechung aus der Fachterminologie zu finden, wobei die direkte Übersetzung des englischen Wortes oder der Originalbegriff in Klammern beigelegt wird. In jedem Fall soll aber der Sinn des Originaltextes so genau wie möglich wiedergegeben werden, dabei aber, wo im deutschen Sprachraum nicht verwendete Konzepte die Begrifflichkeit bestimmen, trotzdem verständlich sein.

Weitgehend abweichend von den Konventionen in Deutschland, allerdings dem üblichen amerikanischen Verfahren entsprechend, wird in dem Kochbuch mit Behälter- und Bündelnummern (Box oder Folder, wobei Folder meist eine Unterteilung von Box ist) gearbeitet, die anstelle der Signaturen oder Bestellnummern verwendet werden und mit dem Element <container> erfasst werden. Die Signatur entspricht demgegenüber eher der Identifikation, die mit dem Element <unitid> verwaltet wird. Wenn dieses genutzt wird, müssen aber die Stylesheets darauf eingestellt werden, dass sie den Eintrag in <unitid> als Signatur oder Bestellnummer ausgeben. Die amerikanische Praxis der Verwendung von Behälternummern zeigt sich anschaulich in den EAD-Findbüchern im Internet. Dort ist durchaus auch eine Erschließung ganz ohne Behälternummern oder Signaturen möglich, da die Findbücher teilweise als Inventare genutzt werden, ohne dass sie für die Bestellung des Materials genutzt werden.

Die Übersetzung umfasst vollständig die Abschnitte 1-3 sowie 6-8 und lässt die Abschnitte 4 und 5 aus, in denen die Installation und der Einsatz von Formatvorlagen und Makros für bestimmte Software (XMetaL, Author/Editor und Wordperfect 9) beschrieben wird. Diese Abschnitte können ebenso wie die zugehörigen Formatvorlagen unter der oben angegebenen URL heruntergeladen werden.

Selbstverständlich müssen Element- und Attributnamen, ebenso wie die wählbaren Werte der Attribute in der englischen Originalversion verwendet werden, da sonst die Stylesheets die kodierten Dateien nicht mehr nutzen können.

Angelika Menne-Haritz, Sept. 2003

## Das EAD Kochbuch

Archive und Bibliotheken, die EAD anwenden möchten, stehen vor einer Vielzahl von Wahlmöglichkeiten: Welche Software soll eingesetzt werden? Welche Elemente und Attribute sollen für die Codierung angewendet werden? Wie können am besten gedruckte und über das Internet zugängliche Versionen der Findbücher erstellt werden? Während der Sitzung der EAD Roundtable anlässlich der Jahrestagung der Society of American Archivists von 1999 stimmten die Teilnehmer darin überein, dass ein einfaches, beispielhaftes Kodierungsprotokoll zusammen mit einigen Software-Tools für die redaktionelle Bearbeitung von elektronischen Findbüchern zusammen mit den passenden Stylesheets für ihre Publikation sehr nützlich wäre. Das Kochbuch wurde zu diesem Zweck erarbeitet. Es ist also eine Erweiterung der *EAD-Tag-Library* und der *EAD Application Guidelines*. Um es zu benutzen, wird ein Mindestverständnis für den EAD-Elemente-Satz vorausgesetzt.

Das EAD-Kochbuch schließt ein Kodierungsprotokoll ein, das auf den Empfehlungen der *EAD-Application-Guidelines* und mehrerer EAD-Projekte beruht. Es bietet eine Erklärung der Entscheidungen, die zu den Empfehlungen führten und wird von einer Reihe von Software-Tools begleitet, die eine beispielhafte Markierung in verschiedenen SGML- und XML-Editoren demonstrieren. Es enthält eine schrittweise Einführung in ihre Anwendung. Beigefügt ist ebenfalls ein fiktives Findbuch, das nach diesen Richtlinien kodiert wurde.

Es gibt viele Möglichkeiten, wie ein Findbuch, für das diese Markierungssyntax angewendet wurde, im Internet oder im Druck dargestellt wird. Die Werkzeuge, die dem Kochbuch beigefügt sind, stellen eine einfache Präsentation her, die sich für beide Ausgabeformen eignet. Sein Aussehen lehnt sich an viele andere Beispiele an. Dazu gehören das Format der Minnesota Historical Society und das Format, das bei der Library of Congress für ihre Findbücher eingesetzt wird. Die Arbeit der Minnesota Historical Society an ihrem Findbuchformat wurde von Dennis Meissner im *American Archivist* (Fall 1997) beschrieben. (Auch veröffentlicht unter: *Encoded Archival Description: Context, Theory and Case Studies*.) Die Library of Congress war ein früher und aktiver Partner bei der Entwicklung von EAD. Das Format ihrer Archivfindbücher wurde in der Vergangenheit bereits von vielen Archiven übernommen. Allerdings orientiert sich das in dem Kochbuch vorgestellte Kodierungsprotokoll nicht direkt an der Praxis einer der beiden Institutionen, sondern ist eine zusammengesetzte Form, die aus vielen Quellen schöpft und auch Vorschläge von Gutachtern einschließt. Ein Satz von Stylesheets, geschrieben in XSL, wird beigefügt, damit die Bearbeiter selbst Versionen der Findbüchs generieren können, die für eine Ansicht in einem Webbrowser oder im Druck geeignet sind. Auch diese Stylesheets orientieren sich nicht an einer bestimmten institutionellen Praxis, sondern sind aus vielen Anregungen hervorgegangen. Die Anleitungen führen Schritt für Schritt durch jede Phase der EAD-Anwendung. Bon Appetit!

Michael J. Fox

Juli 2000

## Abschnitt 1: Das Rezept

### 1.1. Mehr über EAD erfahren.

Die drei von der Society of American Archivists herausgegebenen Bände: *Encoded Archival Description: Context, Theory and Case Studies*, die *EAD Application Guidelines* und die *EAD Tag Library* geben einen guten Einstieg. Die Hilfe-Seiten, die von der EAD-Roundtable der Society of American Archivists unterhalten werden, enthalten vielfältige praktische und nützliche Informationen, darunter Beschreibungen vieler Anwendungsprojekte und Listen von verfügbaren Kursen. Diese Seiten finden sich unter:

<http://jefferson.village.virginia.edu/ead/>

### 1.2. Entscheidung, wie die Findbücher im Internet angeboten werden sollen.

Wie werden Benutzer die Findbücher finden? Dafür gibt es viele Möglichkeiten, die einzeln oder gemeinsam genutzt werden können:

- a. Links von anderen Web-Seiten legen.
- b. Links von Einträgen in einem eigenen Online-Katalog anlegen.
- c. Indizierung der Findbücher auf der eigenen Web-Site. Viele Anwender haben inzwischen eigene Software, die zumindest eine Volltextsuche über Dokumente auf der Website durchführt. EAD-kodierte Dokumente, die in HTML umgewandelt wurden, können indexiert und zusammen mit anderen Dateien gefunden werden.
- d. Einem Konsortium beitreten, das eine zentrale Datei für Findbücher bereithält. Solche Datenserver verfügen meist über komplexere Software, die sowohl strukturierte wie textbasierte Suche erlaubt. Eine zunehmende Zahl von staatlichen Projekten stellen solche Dienstleistungen bereit. Das Angebot „Archival Resources“ der Research Libraries Group (RLG) stellt Findbücher von Archiven mehrerer Länder bereit und ist nicht auf Mitgliedsorganisationen beschränkt.
- e. Einen eigenen Datenserver und die entsprechende Software kaufen. Die Auswahl reicht von Freeware bis zu kommerzieller Software für Volltextrecherche.
- f. Ein eigenes Kooperationsprojekt beginnen.

Die Werkzeuge, die dem Kochbuch beigelegt sind, sind für alle diese Varianten nutzbar.

Das Kochbuch liefert ausdrückliche Anweisungen für die Optionen a und b in den Abschnitten 5 bis 8.

### 1.3. Das Datenformat für die Bereitstellung bestimmen.

Das Kochbuch unterstützt das Verfahren, das bisher am breitesten eingesetzt wird, nämlich die Konversion von EAD-Dokumenten in HTML für eine Ansicht in Standard-Browsern. Es umfasst nicht die Verwendung von Browser-Plug-ins oder anderen unterstützenden Dateien, die erforderlich sind, wenn die EAD-Dokumente im klassischen SGML-Format bereitgestellt werden. Die Kosten und Umstände, die damit für Nutzer entfallen, um solche zusätzliche Software zu kaufen und zu

installieren, machen diesen Ansatz in dem Zuge weniger wünschenswert, in dem XML sich verbreitet.

Weitere Hinweise zu diesem Thema liefern die *EAD Application Guidelines*, Kap. 4 und 5 sowie die Darstellung von einzelnen Projekten in dem Band *Encoded Archival Description: Context, Theory and Case Studies*. Ebenso ist es sinnvoll, die Anwendungsverfahren einzelner Institutionen, wie sie auf der EAD-Help-Pages-Website dargestellt sind, zu studieren. Schließlich ist es ebenfalls nützlich, die Diskussionen der EAD-Liste zu verfolgen oder dort eigene Fragen zu stellen.

#### **1.4. Allgemeine Grundsätze und besondere Details in den Abschnitten 2 und 3 studieren.**

Einige Selektionen sind schon vorgenommen worden. Andere müssen selbst bestimmt werden. Diese Abschnitte sollen helfen, die eigene Auswahl vorzunehmen und selbst die erforderlichen Entscheidungen zu treffen. Die oben genannten Publikationen ebenso wie die *EAD Tag Library* geben weitere Informationen.

#### **1.5. Eine der Softwareanwendungen für die Bearbeitung von Findbüchern beschaffen, die vom Kochbuch unterstützt werden.**

Das sind XMetal, Author/Editor und WordPerfect 9.

#### **1.6. Den Schritt 1.4 wiederholen.**

#### **1.7. Installation und Konfiguration der Software (vgl. Abschnitt 4).**

#### **1.8. EAD-Findbücher herstellen (vgl. Abschnitt 4).**

#### **1.9. Eventuell vorhandene SGML-Dateien nach XML konvertieren (vgl. Abschnitt 5)**

#### **1.10. Ein passendes XSL-Stylesheet auswählen (vgl. Abschnitt 6).**

#### **1.11. EAD-Dateien in HTML konvertieren (vgl. Abschnitt 7).**

#### **1.12. Dateien auf den Webserver hochladen (vgl. Abschnitt 8).**

## Abschnitt 2: Die EAD-Kochbuch Grundlagen

EAD eröffnet eine große Flexibilität für die Kodierung von Findbüchern. Die Auswahl, die bei der Erstellung getroffen wird, hat weitreichende Konsequenzen für die spätere Nutzbarkeit der Datei. Das Verfahren, das im EAD Kochbuch vorgeschlagen wird, wurde mit der Absicht entwickelt, den Transport von Daten zwischen Systemen, den Datenaustausch für Verbünde, die Wiederverwendung für andere Zwecke, die Hyper-Navigation innerhalb der Findbücher und die Darstellung der Daten in verschiedenen Umgebungen, etwa im Internet und im Druck, zu vereinfachen. Aus diesen Gründen gelten die folgenden Prinzipien als Grundlage für das beispielhafte Kodierungsprotokoll.

1. EAD-Kodierung ist kein Ersatz für eine gute archivische Erschließung. Bevor ein Dokument kodiert wird, sollte überlegt werden, welche Informationen erfasst werden sollten. Regelwerke wie *Archives, Personal Papers, and Manuscripts* (APPM), die *Rules for Archival Description* (RAD), und der *General International Standard for Archival Description* (ISAD(G)) geben viele interessante Hinweise darauf, welche Informationen in eine gute archivische Erschließung aufgenommen werden sollten und wie diese Angaben strukturiert werden sollten. Findbücher unterscheiden sich nicht grundsätzlich von anderen Werkzeugen, mit denen Benutzer bei der Identifikation und Lokalisierung relevanter Materialien in den Beständen unterstützt werden sollen. Es sollte überlegt werden, ob die Übernahme von Indexbegriffen aus Normvokabularen in den für Recherche wichtigen Elementen die Suche über verschiedene Metadatensysteme hinweg erleichtern kann.
2. Die Erschließung in einem nach EAD kodierten Findbuch zeigt die in verschiedene Ebenen aufgeteilte, hierarchische Struktur des Materials selbst. Die Struktur reflektiert nur die grundsätzlichen und immanenten Eigenschaften des archivischen Materials und seine internen Verbindungen. Das Findbuch hat dabei die Funktion als Surrogat für die Originale und es ist ein Werkzeug für die Entdeckung und Interpretation potentieller Quellen.
3. Die Erschließung geht vom Allgemeinen zum Besonderen, wobei sie die oben beschriebene Hierarchie darstellt. Sie nutzt dabei eine praktische und flexible Lösung für das Problem der intellektuellen und physischen Kontrolle über große Bestände von Material und bietet gleichzeitig die immanente Struktur als Hilfe bei der Quellenentdeckung durch die Benutzer.
4. Die Markierung nutzt die Syntax entweder des klassischen SGML oder von XML. Wenn eine SGML-Datei in HTML umgewandelt werden soll, muss sie zunächst nach XML konvertiert werden.
5. Das im Folgenden genutzte Kodierungsprotokoll geht davon aus, dass das zu verzeichnende Material einen Nachlass oder einen Aktenbestand bildet, der in mehrere Strukturelemente unterteilt ist. Außerdem setzt es voraus, dass der Bestand bereits vollständig bearbeitet und erschlossen ist. Komplexere oder einfachere Bestände, also mit Untergruppen oder Unterbeständen oder solche, die nur aus einer einzigen Gliederungsgruppe bestehen, werden vermutlich Änderungen der beigefügten Stylesheets erfordern.

6. Es wird keine Methode für die Verwaltung größerer Mengen von Images oder anderen zum Findbuch gehörenden, nicht textlichen Dateien vorgeschrieben. Wenn Archive Findbücher mit vielen externen Dateien verlinken möchten, sollten sie die Optionen im Abschnitt 4.4.3 der *EAD Application Guidelines* berücksichtigen. Die einfache Integration eines Institutionenlogos kann mit einem einzelnen „Extended Pointer“-Element oder mit einem der Stylesheets vorgenommen werden, wie sie dem Kochbuch beigelegt sind. (Vgl. Abschnitt 6.2).
7. Es gibt einen wachsenden Konsens unter Archivaren, dass es einen Kern von Elementen gibt, der in jedem Fall in einem Findbuch verwendet werden sollte, um Benutzern zu helfen, die Relevanz eines Bestandes für die jeweilige Untersuchung einzuschätzen und das Material, das verwendet werden sollte, zu identifizieren. Anhang A der *EAD Application Guidelines* nennt vier Elemente, die in jedem Findbuch vorhanden sein sollten: die Erschließungsidentifikation (<did>) der obersten Ebene, die Bestandsgeschichte, zusammenfassende Informationen zu Umfang und Aussage (<scopecontent>) und Stichworte (<controlaccess>). Verwaltungsbezogene Informationen, wie Zugangsbeschränkungen, sollten angegeben werden, wenn sie erforderlich sind. Die besonderen Unterelemente innerhalb dieser Gruppen werden im Abschnitt 3 beschrieben. Außerdem gibt es bestimmte Elemente, die für die Identifikation der elektronischen Datei, die das kodierte Findbuch enthält, erforderlich sind. Einige davon werden von der EAD-DTD verlangt, vor allem im EAD-Header-Element.
8. Das Ablaufprotokoll, das in Abschnitt 3 ausführlich vorgestellt wird, schreibt die ausdrückliche Einbeziehung von bestimmten Daten vor, die für den Transport, für die Darstellung in anderen Systemen, für die interne Hyper-navigation, für die Wiederverwendung der Daten wie etwa bei der Konversion von EAD-Daten in MARC-Datensätze und für die spätere Migration der Daten in neue Systeme verwendet werden können. Da nicht davon ausgegangen werden kann, dass spezifische Protokolle für die Präsentation, die in einem jeweils eigenen Stylesheet formuliert werden, immer die Datei begleiten werden oder für die Darstellung in anderen Systemen und Kontexten verwendet werden können, benutzt das Protokoll Label. Um eine koherente, benutzerfreundliche Darstellung zu erreichen, werden HEAD-Elemente für Textblöcke und Gruppen von verwandten Elementen eingesetzt, LABEL-Attribute auf Standardwerte gesetzt und Kodierungsanalogien (encoding analogs) zu MARC 21 eingefügt. Außerdem umfasst es Beispiele für erklärenden Text, der einem Findbuch hinzugefügt werden kann, um Benutzern zu helfen, die vielleicht nicht auf Anhieb den Zweck einzelner Element, wie der zu Beginn des Findbuchs zusammengestellten Stichworte (<controlaccess>), erkennen können. Um eventuelle Darstellungsprobleme zu vermeiden, sollten keine ganzen Sätze vollständig in Großbuchstaben geschrieben werden.  
  
(Anm. zur Übersetzung: Die mit dem Element <controlaccess> erfassten Stichworte werden in amerikanischen Findbüchern normalerweise dem Findbuchtext als Block vorangestellt. In den EAD-Findbüchern im Internet ist dieses Verfahren oft zu finden.)
9. Wenn Findbücher kodiert werden, sollte auf die Anwendung weiterer einschlägiger Regeln geachtet werden. So sollten die Empfehlungen zur

Verwendung von Leerzeilen und der Interpunktion in den Abschnitten 4.3.5.1 und 4.3.5.2 der *EAD Application Guidelines* und des Abschnittes 3.3.10 dieses Kochbuchs beachtet werden. Interne Links werden im Besonderen mit einem Referenz-Element kodiert. Besondere Identifikationsattribute werden im Kodierungsprotokoll vorgesehen, um interne Hyper-Links zu ermöglichen, besonders zwischen dem Inhaltsverzeichnis und dem Hauptteil des Findbuchs. Die meisten dieser Angaben sind als Standardeinstellungen in den beigefügten Softwareformatvorlagen für die Erstellung neuer Findbücher enthalten und vermehren deshalb nicht die erforderliche Kodierungsarbeit.

10. Zusätzliche Kodierung, wie etwa von Personennamen und Sachen außer in den vorgesehenen Strukturelementen scheint nur dann sinnvoll zu sein, wenn es ein besseres Verständnis oder eine bessere Auffindung ermöglicht. Das Kodierungsprotokoll bietet keine spezielle Unterstützung in Form von Tastaturmakros für die Dateneingabe, oder eine spezielle Ausgabeform über die Stylesheets für Markierungen, wie etwa <persname> innerhalb der Elemente für narrativen Text <p>. Das schließt aber deren Nutzung nicht aus. Diese Frage der Nutzung von Indexbegriffen in narrativen Texten wird in verschiedenen Darstellungen von EAD-Projekten, die in dem Band *Encoded Archival Description: Context, Theory and Case Studies* erschienen sind, diskutiert.
11. Die Markierungsmethode folgt dem „kombinierten“ Modell für die Erschließung von nachgeordneten Strukturbestandteilen. Die dabei erfassten Informationen können entweder in der Reihenfolge, wie sie kodiert wurden, dargestellt werden oder getrennt für einen zusammenfassenden Überblick über den Bestand oder für eine detaillierte Darstellung mit Hilfe eines Stylesheets genutzt werden. Der Gebrauch von Stylesheets für die eine oder die andere Präsentation ausgehend von einem einzigen Satz von Daten erübrigt die Verdoppelung der Eingabe von Daten, die für beide Ansichten gleich sind.
12. Die Reihenfolge der Elemente in dem Kodierungsprotokoll oder den Formatvorlagen hat keine Bedeutung für die Aussagen, außer, wenn EAD eine bestimmte Abfolge von Elementen erfordert. Bei der Anwendung der dem Kochbuch beigefügten Stylesheets, die mit der Transformationssprache XSLT geschrieben sind, ist die Anordnung der Informationen in der Präsentation weitgehend unabhängig von ihrer Anordnung im Quelldokument. Während das Kodierungsprotokoll keine bestimmte Reihenfolge bei der Eingabe voraussetzt, verlangen die Stylesheets eine vordefinierte Reihenfolge für die Ausgabe. Allerdings sind andere Publikationstechniken als XSL-Stylesheets meist nicht so flexibel und sehen geringe oder gar keine Möglichkeiten für eine Umstellung der Elemente vor. In einem solchen Fall kann die Reihenfolge der Eingabe direkt für die Ausgabe und Publikation übernommen werden.



## Abschnitt 3: Das EAD-Kochbuch-Kodierungs-Protokoll

Dieser Abschnitt beschreibt die zentralen Kodierungsempfehlungen des Kochbuchs und ihre Begründung. Die hier vorgenommenen Entscheidungen sind in den folgenden Formatvorlagen und anderen Softwarewerkzeugen integriert, die in Abschnitt 4 beschrieben sind und werden von den Veröffentlichungswerkzeugen aus den Abschnitten 6 und 7 berücksichtigt.

Das Protokoll umfasst jedes zentrale übergeordnete Element, das in EAD verfügbar ist. Wenn in einem Archiv ein bestimmtes Element nicht verwendet werden soll, kann es aus den beigefügten Formatvorlagen gelöscht oder aus einzelnen Dokumenten je nach Bedarf entfernt werden.

Den ID-Attributen werden willkürlich festgelegte Werte zugewiesen, um die Erstellung des Inhaltsverzeichnisses und anderer interne Hyperlinks zu erleichtern. Jede Änderung dieser Werte erfordert ein Umschreiben der Stylesheets. Falls weitere Elemente genutzt werden sollen, etwa <odd>, und sie vom Inhaltsverzeichnis aus verlinkt sein sollen, müssen weitere ID-Attribute zugewiesen werden. Beliebige Werte, die noch nicht verwendet wurden, können dafür eingesetzt werden. Falls ein Element wie <bioghist> rekursiv verwendet wird, sollte der Wert des ID-Attributes, das in diesem Abschnitt festgelegt wurde, nur beim übergeordneten Element angegeben werden. SGML erfordert, dass jedes ID-Attribut einen einmaligen Wert besitzt.

Für bestimmte Elemente wird die Eingabe von Text angenommen, besonders bei Head und EAD-Header-Elementen. Die Sprachkodierung kann natürlich der jeweiligen Praxis angepasst werden. Detaillierte Beschreibung der Form und der Verwendung jedes Elements findet sich in der *EAD Tag Library*.

Kodierungsentsprechungen (encoding analogs) werden für einige Elemente angegeben, damit die EAD-Elemente mit dem MARC 21-Katalogformat in Übereinstimmung gebracht werden können. Sie werden nur angegeben, um die Konversion von EAD in MARC-Datensätze zu erleichtern. Falls es nicht vorgesehen ist, die EAD-Daten für diese Konversion zu nutzen, können diese Attribute ohne Verlust weggelassen oder gelöscht werden. Sie können ebenso durch andere Entsprechungen wie zu ISAD(G) ersetzt werden.

Das Kochbuch setzt das <frontmatter>-Element nicht ein. Es wird weder im Kodierungsprotokoll verwendet, noch nutzen es die Stylesheets. Zahlreiche Institutionen ziehen es vor, alle Informationen, die für die Darstellung einer formellen Titelseite benötigt werden, aus dem <eadheader> zusammenzustellen.

### 3.1 EAD <ead>

Erfasse das RELATEDENCODING-Attribut als "MARC21".

### 3.2 EAD Kopfangaben <eadheader>

Erfasse das AUDIENCE-Attribut als "internal".

Erfasse das LANGENCODING-Attribut als "ISO 639-2".

Erfasse das FINDAIDSTATUS-Attribut als "EDITED-FULL-DRAFT".

Erfasse das ID-Attribut als "a0".

### 3.2.1 EAD Identifikation <eadid>

Erfasse einen Wert, der als Dateiname genutzt werden kann, damit die Dateiverwaltung vereinfacht wird. Stylesheet 3, das diesem Kochbuch beigelegt ist, (vgl. Abschnitt 6.1.3) nutzt den bei <eadid> eingegebenen Dateinamen als Vorlage für mehrere Dateinamen in einem HTML-Frameset.

Erfasse einen kodierten Wert für das Archiv in dem SYSTEMID-Attribut, damit das zugeordnete <eadid> in der Welt der Online-Findbücher nur einmal vorhanden ist. Dazu kann ein MARC-Archivcode, der von der Library of Congress oder der Nationalbibliothek in Kanada ausgegeben wird (*MARC Code List for Organizations*), oder einen Code, der von OCLC, der Research Libraries Group oder jeder anderen nationalen Behörde ausgegeben wurde, verwendet werden.

Erfasse in dem SOURCE-Attribut eine Kodierung, die festhält, welche Stelle den Archivcode im SYSTEMID-Attribut herausgegeben hat. Für die oben aufgelisteten Institutionen wären das die Codes DLC, NLC, OCoLC oder CStRLIN.

Erfasse das ENCODINGANALOG-Attribut als "852".

### 3.2.2 Dateibeschreibung <filedesc>

#### 3.2.2.1 Titel des Findbuchs <titlestmt>

Erfasse den Namen des Erstellers der Unterlagen in dem Bestand, der eventuell anders lautet als der Titel des Findbuchs, im Element <titleproper>.

Erfasse im Element für den Untertitel <subtitle> eine Angabe mit der folgenden Syntax: Ein Findbuch von *Name Charakterisierung des Materials wie im Titel* in *Name des Archivs*.

Beispiel:

<titleproper>William Fonds Nachlass: </titleproper><subtitle>Ein Findbuch über seine Papiere in der Cupcake Corners Historical Society.</subtitle>

Diese Formel sorgt für eine Angabe, die vier Informationselemente umfasst, den Namen des Erstellers der Materialien, den Typ des Dokuments, nämlich Findbuch, die Art der Unterlagen und den Namen des Archivs. Diese Daten werden als Titel-Element in der HTML-Ausgabe verwendet und dienen als Metadaten für Suchmaschinen im Internet.

Erfasse den Namen des Bearbeiters des Findbuchs im Element <author>. Hier werden interessante Angaben für die Nachvollziehung der Geschichte des Findbuchs erfasst.

### 3.2.2.2 Veröffentlichung <publicationstmt>

Erfasse den Namen des veröffentlichenden Archivs im Element <publisher>.

Erfasse das Publikationsdatum im Element <date>.

Erfasse den Ort des Archivs im Element <address>. Für längerfristige Gültigkeit kann es angebracht sein, die genaue Adresse und Telefonnummern etc. in einer externen Datei zu erfassen, die mit einer Entitätenreferenz angesprochen wird oder sie im Stylesheet als Standardangabe festzuhalten.

### **3.2.3 Beschreibung des findbuchprofils <profiledesc>**

#### 3.2.3.1 Erstellung <creation>

Erfasse hier den Namen des für die Kodierung Verantwortlichen. Eingeschlossen werden sollte das Datum mit einem Element <date>.

#### 3.2.3.2 Verwendete Sprache <language>

Erfasse eine Angabe über die Sprache des Findbuchs in dem Element <language>.

Dabei ist zu beachten, dass die Angabe der jeweiligen Sprachen im Unterelement <language> erfasst wird.

## **3.3 Archivische Erschließung <archdesc>**

Erfasse die organisatorische Ebene des ganzen Bestandes mit dem LEVEL-Attribut, soweit angebracht.

Erfasse die Sprachen, die in dem Bestand Verwendung finden, in dem LANGMATERIAL-Attribut, soweit erforderlich. Der Standardwert in den beigefügten Formatvorlagen ist "eng".

Erfasse das TYPE-Attribut als "inventory".

Erfasse Angaben zu wenigstens den folgenden vier Elementen: Erschließungsidentifikation <did>, Biographie oder Geschichte <bioghist>, Umfang und Aussage <scopecontent> und Stichwörter <controllaccess>, wie es in Anhang A der *EAD Application Guidelines* empfohlen wird. Erfasse Angaben zu Benutzungsbeschränkungen, soweit erforderlich. Die beigefügten Formatvorlagen unterstützen die Eingabe weiterer Elemente der obersten Ebene. Die EAD-DTD erfordert, dass die Erschließungsidentifikation <did> als erstes Element verwendet wird. Die weitere Anordnung ist beliebig. Allerdings erstellen die beigefügten Stylesheets bei der Transformation der EAD-Datei von XML nach HTML eine

Ausgabe der Elemente in einer vordefinierten Ordnung. In Abschnitt 6 wird dieser Prozess ausführlicher erläutert.

### **3.3.1 Erschließungsidentifikation <did>**

Erfasse das ID-Attribut als "a1".

Erfasse ein Head-Element zur genaueren Charakterisierung des Elementes <did> und seiner Unterelemente. Die beigefügten Formatvorlagen benutzen die Bezeichnungen "Overview of the Records" oder "Overview of the Collection".

Erfasse den Namen des Archivs in dem Element <repository>, auch wenn seine Ausgabe nicht vorgesehen ist.

Erfasse das ENCODINGANALOG-Attribut als "852\$a".

Erfasse das LABEL-Attribut als "Repository:".

Erfasse den Ort des Archivs in <addressline> so dass er dem ISAD(G) Standard entspricht.

Erfasse den Namen des Erstellers des Materials in dem Element <origination>. Füge ein Element für Korporationen, Personen oder Familien ein, soweit sinnvoll.

Erfasse das ENCODINGANALOG-Attribut für das Unterlement entweder als "100" oder "110", je nachdem, welches die entsprechende MARC –Kategorie ist.

Erfasse das LABEL-Attribut als "Creator:".

Erfasse den Titel des Materials als <unittitle>.

Erfasse das ENCODINGANALOG-Attribut als "245\$a".

Erfasse das LABEL-Attribut als "Title:".

Benutze keine Trennungszeichen zwischen <unittitle> und <unitdate>, da die Stylesheets sie auf getrennten Zeilen präsentieren.

Erfasse die Laufzeit des Materials in <unitdate>. Die beigefügten Stylesheets unterstützen die Kodierung der Daten entweder als separate Elemente oder als Unterelement unter <unittitle>.

Das Element kann wiederholt werden, wenn sowohl eine umfassende Laufzeit wie Zeiträume angegeben werden sollen.

Erfasse die Art des Datums in dem TYPE-Attribut.

Erfasse das ENCODINGANALOG-Attribut als "245\$f" oder "260\$c" je nach den benutzten MARC-Konventionen.

Erfasse das LABEL-Attribut als "Dates:".

Erfasse eine äußere Beschreibung in <physdesc>.

Erfasse das ENCODINGANALOG-Attribut als "300\$a".

Erfasse das LABEL-Attribut als "Quantity:".

Erfasse eine Zusammenfassung für den Inhalt des Materials in <abstract>.

Erfasse das ENCODINGANALOG-Attribut als "520\$a.

Erfasse das LABEL-Attribut als "Abstract:".

Erfasse in <unitid> einen nur einmalig vorhandenen Namen oder eine eindeutige Identifikation für das beschriebene Material, wie es nach ISAD(G) erforderlich ist.

Erfasse das ENCODINGANALOG-Attribut als "099".

Erfasse das LABEL-Attribut als "Identification:".

Erfasse das COUNTRYCODE-Attribut so wie es in ISO 3166 vorgeben ist. Der Standard-Wert in den Formatvorlagen ist "US".

Erfasse im REPOSITORYCODE-Attribut den Wert aus dem SYSTEMID-Attribut in <eadid>.

### **3.3.2 Biographie oder Geschichte <bioghist>**

Erfasse das ENCODINGANALOG-Attribut als "545".

Erfasse das ID-Attribut als "a2".

Erfasse ein Element <head>, das den Inhalt des Elementes beschreibt.

Kodiere den Text des Elementes als Textabsätze mit dem Element <p> und eine chronologische Liste als <chronlist> oder als beides.

Nachgeordnete <bioghist>-Elemente innerhalb des übergeordneten Elementes (Rekursion) können hilfreich sein, wenn der Text in mehrere Abschnitte gegliedert werden soll. Diese Technik wird von den beigefügten Stylesheets unterstützt.

Rekursive <bioghist>-Elemente müssen nicht denselben ID-Attribut-Wert haben wie das übergeordnete <bioghist>-Element, da SGML für jedes ID-Attribut einen nur einmal vorkommenden Wert verlangt.

### **3.3.3 Umfang und Aussage <scopecontent>**

Erfasse das ENCODINGANALOG-Attribut als "520".

Erfasse das ID-Attribut als "a3".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodiere den Inhalt des Elements als Text mit dem Element <p>.

Nachgeordnete <scopecontent>-Elemente innerhalb des übergeordneten Elements (Rekursion) können hilfreich sein, um den Text in separate Abschnitte zu unterteilen. Rekursive <scopecontent>-Elemente müssen nicht denselben ID-Attribut-Wert haben wie das übergeordnete <scopecontent>-Element, da SGML für jedes ID-Attribut einen nur einmal vorhandenen Wert verlangt.

Falls Sie einen Hyperlink von einem Abschnitt des Textes im <scopecontent>-Element zu einem Teil der Verzeichnung von nachgeordneten Strukturbestandteilen im <dsc>-Element herstellen wollen, müssen diese Links ausdrücklich im Text unter Nutzung des Referenzelementes <ref> eingetragen werden. (Vgl. Abschnitt 3.5)

Angaben über die Strukturierung des Materials können hier integriert werden, statt sie als separate Elemente zu erfassen.

### **3.3.4 Struktur <organization>**

Erfasse das ENCODINGANALOG-Attribut als "351\$a".

Erfasse das ID-Attribut als "a4".

Erfasse ein <head>-Element, das den Inhalt dieses Elementes beschreibt.

Kodierte den Inhalt des Elementes als Text mit dem Element <p>.

Es ist auch möglich, die Ordnung des Materials dadurch zu beschreiben, dass man eine Liste der Strukturbestandteile angibt. In diesem Fall sollte der Liste eine kurze Erläuterung vorangestellt werden. Die beigefügten Stylesheets sind so geschrieben, dass sie automatisch Hyperlinks zwischen den hier angegebenen Überschriften und der detaillierten Beschreibung generieren. Die Links müssen hier nicht ausdrücklich kodiert werden. Allerdings müssen dazu die Überschriften in der Liste in der gleichen Reihenfolge erscheinen wie in der Verzeichnung. Die Generierung der Links geschieht unabhängig davon, ob das <organization>-Element separat oder innerhalb von <scopecontent> verwendet wird.

### **3.3.5 Anordnung <arrangement>**

Erfasse das ENCODINGANALOG-Attribut als "351\$b".

Erfasse das ID-Attribut als "a5".

Erfasse ein <head>-Element, das den Inhalt dieses Elementes beschreibt.

Kodierte den Inhalt des Elementes als Text mit den Elementen für Textabschnitt <p>, als Liste oder als Tabelle.

### **3.3.6 Sonstige Verzeichnungsangaben Data <add>**

Die Unterelemente von <add> sind unten beschrieben. Zur Erhöhung der Flexibilität bei der Bearbeitung wird empfohlen, jedes die Elemente in ein eigenes

<add>-Element einzusetzen, statt mehrere Unterelemente in einem übergeordneten <add>-Element zu erfassen.

#### 3.3.6.1 Verwandtes Material <relatedmaterial>

Erfasse das ENCODINGANALOG-Attribut als "544 1".

Erfasse das ID-Attribut als "a6".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt dieses Elements als Text mit Absätzen oder als eine Liste mit Hilfe des Elements <list>.

#### 3.3.6.2 Separiertes Material <separatedmaterial>

Erfasse das ENCODINGANALOG-Attribut als "544 0".

Erfasse das ID-Attribut als "a7".

Erfasse ein <head>-Element, das den Inhalt des Elementes beschreibt.

Kodierte den Inhalt dieses Elements als Text mit Absätzen oder als eine Liste mit Element <list>.

#### 3.3.6.3 Andere Findmittel <otherfindaid>

Erfasse das ENCODINGANALOG-Attribut als "555".

Erfasse das ID-Attribut als "a8".

Erfasse ein <head>-Element das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt dieses Elements als Text mit Absätzen mit dem Element <p>.

#### 3.3.6.4 Index <index>

Hier kommt kein ENCODINGANALOG-Attribut zur Anwendung.

Erfasse das ID-Attribut als "a9".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt dieses Elements als Text mit Absätzen und mit dem Element für Index-Einträge.

#### 3.3.6.5 Bibliographie <bibliography>

Hier kommt kein ENCODINGANALOG-Attribut zur Anwendung.

Erfasse das ID-Attribut als "a10".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt dieses Elements als Text mit Absätzen oder als Element für bibliographische Referenzen.

### 3.3.7 Andere Erschließungsangaben <odd>

Erfasse das ENCODINGANALOG-Attribut als "500".

Erfasse das ID-Attribut als "a11".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt dieses Elements als Text mit Absätzen.

Dieses Element sollte nur verwendet werden, wenn kein präziseres Element verwendet werden kann.

### 3.3.8 Stichworte <controlaccess>

Erfasse das ID-Attribut als "a12".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt. Darin sollte die Quellen sowie die Art der Verwendung von normalisierten Ansetzungen beschrieben werden.

Erfasse die Begriffe für <persname> (Personen), <corpname> (Korporationen), <famname> Familien, <geogname> (geographische Bezeichnungen), <subject> (Sache), <genreform> (Materialform), <title> (Titel), <occupation> (Beruf), and <function> (Funktion) in der Form, wie sie in einer passenden Ansetzungsdatei wie etwa bei der Library of Congress in dem „Name Authority file“ oder dem „Art and Architecture Thesaurus“ gefunden werden. Eine Liste möglicher Quellen wird im Abschnitt über Attribute in der *EAD Tag Library* angegeben.

Auch das SOURCE-Attribut kann verwendet werden, um den Namen der Ansetzungsdatei, aus der der Begriff übernommen wurde, anzugeben.

Unterelemente können rekursiv nach Typen getrennt in Gruppen organisiert werden, indem nachgeordnete <controlaccess>-Elemente mit jeweils erläuternden <head>-Elementen eingesetzt werden. Rekursive <controlaccess>-Elemente brauchen nicht denselben ID-Attribut-Wert wie das übergeordnete <controlaccess>-Element zu besitzen, da SGML einmal vorkommende Werte für jedes ID-Attribut erfordert.

Erfasse ENCODINGANALOG-Attributwerte für die entsprechenden MARC 21-Felder, um die Nutzung derselben Daten in Katalogen zu ermöglichen und um die Herstellung von Dublin Core Metadaten in HTML-Ausgaben zu unterstützen. Zum Beispiel verursacht der Wert „600“ in diesem Attribut bei der Benutzung der



Stylesheets, dass ein Personennamen in einem Subject-Element in einem <meta>-Tag in der HTML-Ausgabe erscheint, während der Wert „700“ den Namen in dem <meta>-Tag als Beitragender erscheinen lässt. Die unten angegebenen Elemente werden als Standard-Angaben in den beigefügten Formatvorlagen verwendet. Für die Fälle, in denen mehr als ein ENCODINGANALOG-Wert vorhanden ist, wird der standardmäßig verwendete Wert unten angegeben. Falls es nicht vorgesehen ist, Daten aus EAD in MARC-Datensätze umzuwandeln, oder wenn die standardmäßig verwendeten Werte für eine Anwendung ausreichend präzise sind, können sie beibehalten und kann die folgende Tabelle ignoriert werden.

Die passenden MARC-Werte sind:

<persname>	(600 als Sachbegriff oder 700 als Ersteller; 700 ist der Standard)
<corpname>	(610 als Sachbegriff oder 710 als Ersteller; 710 ist der Standard)
<famname>	(600 als Sachbegriff)
<geogname>	(651)
<subject>	(650)
<genreform>	(655)
<title>	(630 als Sache oder 730 als Teil eines Bestandes; Standard: 630)
<occupation>	(657)
<function>	(656)

Die Standardwerte werden in den beigefügten Formularen und Macros verwendet.

### 3.3.9 Administrative Information <admininfo>

Die Unterelemente von <admininfo> werden unten beschrieben. Um die Flexibilität bei der Bearbeitung zu erhöhen, wird empfohlen, dass jedes dieser Elemente in ein eigenens übergeordnetes Element <admininfo> eingefügt wird, statt mehrere Unterelemente in einem <admininfo>-Element zusammenzufassen.

#### 3.3.9.1 Zugangsbeschränkungen <accessrestrict>

Erfasse das ENCODINGANALOG-Attribut als "506".

Erfasse das ID-Attribut als "a14".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt des Elements als Text mit Absätzen mit dem Element <p>.

#### 3.3.9.2 Benutzungsbedingungen <usesrestrict>

Erfasse das ENCODINGANALOG-Attribut als "540".

Erfasse das ID-Attribut als "a15".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt des Elementes als Text mit Absätzen.

#### 3.3.9.3 Bestandsgeschichte <custodhist>

Erfasse das ENCODINGANALOG-Attribut als "561".

Erfasse das ID-Attribut als "a16".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt des Elementes als Text mit Absätzen.

#### 3.3.9.4 Andere verfügbare Formen <altformavail>

Erfasse das ENCODINGANALOG-Attribut als "530".

Erfasse das ID-Attribut als "a17".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt des Elementes als Text mit Absätzen.

#### 3.3.9.5 Zitierweise <prefercite>

Erfasse das ENCODINGANALOG-Attribut als "524".

Erfasse das ID-Attribut als "a18".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt des Elementes als Text mit Absätzen.

#### 3.3.9.6 Akzessionsbeschreibung <acqinfo>

Erfasse das ENCODINGANALOG-Attribut als "541".

Erfasse das ID-Attribut als "a19".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt des Elementes als Text mit Absätzen.

#### 3.3.9.7 Bearbeitungshinweise <processinfo>

Erfasse das ENCODINGANALOG-Attribut als "583".

Erfasse das ID-Attribut als "a20".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt des Elementes als Text mit Absätzen.

#### 3.3.9.8 Bewertung <appraisal>

Erfasse das ENCODINGANALOG-Attribut als "583".

Erfasse das ID-Attribut als "a21".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt des Elementes als Text mit Absätzen.

#### 3.3.9.9 Übernahmen <accruals>

Erfasse das ENCODINGANALOG-Attribut als "584".

Erfasse das ID-Attribut als "a22".

Erfasse ein <head>-Element, das den Inhalt dieses Elements beschreibt.

Kodierte den Inhalt des Elementes als Text mit Absätzen.

### **3.3.10 Beschreibung nachgeordneter Strukturbestandteile <dsc>**

Für die Kodierung des <dsc>-Elementes wird das "kombinierte Modell" verwendet. Im Kapitel 3 der *EAD Application Guidelines* werden mögliche Strukturmodelle für die Beschreibung der nachgeordneten Strukturbestandteile erläutert. Die beigefügten Stylesheets 1 bis 3 präsentieren <dsc> die Unterelemente von <dsc> der Reihe nach. Stylesheet 4 präsentiert sie als separate, zusammenfassende Beschreibung der Akten "Summary Description of the Records" und als Abschnitt "Detailed Description of the Records" im Druck. Die Stylesheets benutzen HTML-Tabellen, um die beabsichtigte tabellarische Ansicht darzustellen. Die Tabellenstruktur wird aus dem kodierten Inhalt abgeleitet. Es ist keine ausdrückliche Tabellencodierung in EAD erforderlich. Weitere Details sind in Abschnitt 6.1 beschrieben.

Erfasse das ID-Attribut als "a23".

Erfasse das TYPE-Attribut als "combined".

Erfasse ein <head>Element, das den Inhalt dieses Elements beschreibt. Die beigefügten Formatvorlagen benutzen die Standard-Formulierungen: "Detailed Description of the Records" oder "Detailed Description of the Collection."

Füge eine Einleitung hinzu, die diesen Abschnitt des Findbuchs erläutert.

### Strukturbestandteil <c>

Dieses Modell unterstellt, dass die Strukturbestandteile der Ersten Ebene eine Gliederungsgruppe („series“) darstellt.

Erfasse die folgenden Informationen für jede Gliederungsgruppe.

Erfasse das ID-Attribut als „series $nn$ “, wobei  $nn$  die laufende Nummer der Gliederungsgruppe ist, z.B. series1, series2, series3, etc.

Erfasse das LEVEL-Attribut als „series“.

(Anm. zur Übersetzung: Für die Abbildung von Gliederungsgruppen nach dem in deutschen Archiven üblichen Verfahren wurde das LEVEL-Attribut=class in die EAD-DTD eingefügt. Wenn die Stylesheets des Kochbuchs verwendet werden sollen, muss das LEVEL-Attribut=series verwendet werden.)

Für alle Ebenen der Strukturbestandteile werden die passenden <did>-Elemente erfasst. Dabei kann die Laufzeit innerhalb oder neben <unittitle> angebracht werden. Im Gegensatz zu den Empfehlungen im Kapitel 4 der *EAD Application Guidelines*, können durchaus Trennungsinterpunktionen zwischen den einzelnen Unterelementen der <did>-Elemente in den Strukturbestandteilen, etwa zwischen Titel und Datum verwendet werden. Während der Testphase des Kochbuches wurde festgestellt, dass es unmöglich ist in einem Stylesheet, alle möglichen Trennzeichen zu berücksichtigen. Deshalb wird in den beigefügten Stylesheets ein Leerraum zwischen den Elementen eingefügt, soweit sinnvoll.

Erfasse Informationen über Schachteln oder Bündel (Box or Folder), wie es den jeweiligen Hausregeln entspricht. Die Möglichkeiten bei der Präsentation umfassen die Anzeige beliebiger Kombinationen dieser Angaben. Die beigefügten Stylesheets bieten fünf Varianten an. Wenn Behälternummern erfasst werden, müssen sie für jede Einheit angegeben werden, selbst wenn ein Eintrag die Informationen aus einer vorangehenden Einheit wiederholt. Die beigefügten Stylesheets unterdrücken die Darstellung wiederholter Behälternummern. (Weitere Informationen dazu in Abschnitt 6) Das Kochbuch plädiert eindeutig für die Aufnahme von Behälterangaben für alle Einheiten als eine wünschenswerte Praxis, die eine spätere Weiterverwendung der Daten erleichtern wird.

Wenn Informationen zu den Behältern für die meisten Einheiten angebracht werden sollen, nicht aber für alle, z.B. wenn eine Einheit als „Reihe, nach Datum sortiert“ erfasst werden soll, sollte das Behälter-Element, allerdings ohne Eintragung eingefügt werden. Aus Gründen der Programmierung legen die Stylesheets eine Präsentation fest, die sich an dem Erscheinen des Behälterelements sowie dem Vorhandensein oder Fehlen von Eintragungen orientiert. Einfacher ausgedrückt, heißt das folgendes: Wenn Behälterelemente für einige Einheiten angebracht werden, für andere aber nicht, werden die Stylesheets

damit Probleme haben. Glücklicherweise ist es nicht schwierig, die Behälter-Tags einzufügen, weil die Tastatur-Makros in den beigefügten Formatvorlagen sie automatisch bereitstellen.

Wie Veränderungen der Formatvorlagen und Makros vorgenommen werden, um sie an den Typ von Behältern, der verwendet werden soll, anzupassen, wird im Einzelnen für XMetaL, Author/Editor und WordPerfect in den entsprechenden Teilen von Abschnitt 4 beschrieben.

Erfasse das entsprechende TYPE-Attribut für alle Behälter-Elemente.

Erfasse andere Elemente, soweit erforderlich. Die beigefügten Stylesheets unterstützen im Besonderen die Darstellung von <scopecontent>, <bioghist>, <note>, <odd>, <userrestrict> und <accessrestrict> innerhalb der Strukturbestandteile.

### **3.4 Tabellen**

Die Stylesheets des EAD-Kochbuchs unterstützen die Einfügung einfacher Tabellen zur Präsentation von Daten in zwei oder mehr Spalten innerhalb von Textabsätzen in narrativen Elementen wie Umfang und Aussage <scopecontent>, Biographie oder Geschichte <bioghist> und Bemerkung <note>. Die Tabellen können jede Zahl von Spalten enthalten. Zur weitere Information über die Kodierung von Tabellen vgl. den Abschnitt: Überblick über EAD und die Beschreibungen und Beispiele von einzelnen Tabellen-Elementen und –Attributen in der *EAD Tag Library*.

#### **3.4.1 Tabelle <table>**

Erfasse die Zahl der Spalten in der Tabelle in dem COLNUM-Attribut.

##### 3.4.1.2 Tabellengruppe <tgroup>

Füge eine Spezifikation für Tabellenspalten mit dem Element <colspec> für jede Spalte ein. Definiere die Breite der Spalte mit dem COLWIDTH-Attribut. Die Breite kann als absolute Zahl für die Anzahl der Pixel oder als relative Prozentzahl angegeben werden.

##### 3.4.1.3 Table Head <thead>

Erfasse einen Kopf für jede Spalte als ein <entry>-Element innerhalb eines <row>-Elements.

#### 3.4.1.4 Table Body <tbody>

Erfasse den Hauptteil der Tabelle als Serie von Datenzellen, die horizontal mit dem <row>-Element definiert werden und deren Inhalt mit je einem <entry>-Element erfasst werden.

### **3.5 Referenzen**

Referenzelement <ref> stellt einen Mechanismus bereit, mit dem interne Hyperlinks zwischen den Abschnitten eines Findbuchs hergestellt werden können. Alle solche Links enthalten zwei Teile, die Quelle und das Ziel. Das Ziel wird durch ein hinzugefügtes, einmaliges ID-Attribut bei dem Element, das als Ziel dient, hergestellt.

Die Quelle wird durch Einfügen des Textes, der als Link genutzt werden soll, in eine <ref>-Element hergestellt.

Erfasse den Wert des ID-Attributs beim Zielelement als Wert des TARGET-Attribut dieses <ref>-Elementes.

Die Kochbuch-Stylesheets unterstützen die Verwendung von <ref>-Elementen in einer begrenzten Zahl von Situationen. Links können zu <c01>-Elementen („Serien“-titeln) in den Erschließungskomponenten und zu <p>-Elementen an jeder beliebigen Stelle hergestellt werden.

## **Abschnitt 4: Die beigefügten Werkzeuge**

Das Kochbuch stellt eine Reihe von Werkzeugen bereit, die eine Kodierung neuer Findbücher erleichtern. Sie umfassen Tabellen, Makros und andere angepasste Dateien. Vergleichbare Werkzeuge sind verfügbar für die drei Anwendungen: XMetaL, Author/Editor, und WordPerfect 9. Unterstützung für andere Software kann eventuell in Zukunft bereitgestellt werden. Im Folgenden werden Anweisungen für die Installation, Anpassung und Nutzung jeder Anwendungen gegeben. Die zugehörigen Dateien können von den EAD-Hilfe-Seiten heruntergeladen werden.

(Anm. zur Übersetzung: Die folgenden Anweisungen wurden nicht übersetzt. Sie beziehen sich auf die englischen Versionen von XMetaL, Author/Editor und WordPerfect 9 . Für eine Nutzung wird auf die englische Version des Cook Book verwiesen.)

## Abschnitt 5: Die Konvertierung von SGML Dateien nach XML

Falls EAD-Dokumente in der klassischen SGML-Syntax erstellt werden, etwa mit einem Werkzeug wie Author/Editor, und die Veröffentlichungswerkzeuge in XSL, die hier beschrieben werden, verwendet werden sollen, müssen die SGML-Dateien zunächst in XML umgewandelt werden.

### 5.1 EAD als XML

Das XML eines EAD-Findbuchs unterscheidet sich in fünf Aspekten von der klassischen SGML-Syntax:

1. Der Einschluß einer XML-Deklaration am Anfang der Datei.
2. Das Erfordernis einer Systemidentifikation, die auf eine Kopie der EAD-DTD verweist (optional in SGML) am Schluss der DOCTYPE-Deklaration, die sich am Anfang des Dokumentes befindet.
3. Alle Element- und Attributbezeichnungen werden in Kleinbuchstaben erwartet. SGML schreibt das nicht vor.
4. Elemente, die als EMPTY definiert sind, etwa lb, extptr, ptr, ptrloc, extptrloc, müssen mit dem Tag für leere Elemente dargestellt werden also zum Beispiel als <lb/> statt <lb></lb>. Das Kochbuch benutzt diese Elemente nicht.
5. Eine kleine Modifikation der Datei "ead.dtd" ist nötig. Eine Kopie der erforderlichen XML version der "ead.dtd" ist dem Kochbuch beigelegt.

Die Unterschiede zwischen der SGML- und der XML-Syntax von EAD und die notwendigen Modifikationen der DTD sind im Abschnitt 4.3.2 der *EAD Application Guidelines* beschrieben.

Dieser Abschnitt beschreibt zwei Wege für die Konversion von SGML in XML.

(Anm. zur Übersetzung: Dieser Abschnitt ist nicht übersetzt worden, da SGML kaum im deutschsprachigen Raum für Findbuchkodierung eingesetzt wurde. Für eine Nutzung wird auf die englische Version des Cook Book verwiesen.)



## Abschnitt 6: Publikationswerkzeuge – die Stylesheets

Dem Kochbuch beigelegt sind mehrere Stylesheets, die EAD-Findbücher, die entsprechend dem in Abschnitt 3 geschilderten Vorgehen in XML kodiert wurden, in HTML-Syntax umwandeln, die mit Standard-Webbrowsern gelesen werden kann. Diese Stylesheets sind keine umfassenden Werkzeuge für die Präsentation jedes gültigen EAD-Findbuchs. Sie sind so programmiert worden, dass sie speziell mit den Findbüchern funktionieren, die nach dem hier dargestellten Verfahren erstellt wurden. Natürlich können sie so modifiziert werden, dass sie besser an hausinterne Präferenzen bei der Kodierung und Präsentation angepasst sind. Der Konversionsprozess nutzt die Sprache des XSL-Transformationsprotokolls, das vom World Wide Web Konsortium als Empfohlene Praxis entwickelt wurde. Eine Softwareanwendung, die XSLT für Transformationen benutzt, liest den Text eines XML-Dokumentes, wendet die Syntax, die in einem speziellen Stylesheet beschrieben ist, darauf an und erstellt eine HTML-Ausgabedatei. Mehrere solcher Anwendungen sind als „Freeware“ von Microsoft, IBM, Oracle und anderen Herstellern verfügbar. Die dem Kochbuch beigelegten Stylesheets wurden mit dem XT-Package von James Clark getestet. Dieses Produkt ist ziemlich vollständig, was seine Funktionen angeht, außerdem ist es schnell und einfach zu installieren. Es kann als Batch-Anwendung laufen oder im Real-Time-Modus auf dem Server als http-Servlet eingesetzt werden. (Vg. Abschnitte 7 und 8)

Die beigelegten Stylesheets gehen relativ wortreich mit der Kodierung und den erklärenden Kommentaren um. Die Struktur wird modular aufgebaut, um die Lesbarkeit, Übertragbarkeit und spätere Modifikation zu erleichtern, auch wenn das die Performance für eine spezielle Datei beeinträchtigen kann. Jedoch wurden XSL-Funktionen, von denen bekannt ist, dass sie die Performance spürbar reduzieren, vermieden. Die Kodierung enthält viele Angaben und bedingte Verfahren, um eine größere Variantenbreite zwischen individuellen Findbüchern und Institutionenmodellen zu erlauben.

Die Stylesheets unterstützen fünf Typen für die Kodierung der Informationen über Behälter, also der Schachtel- oder Bündelnummern. So kann nur die Schachtelnummer, die Schachtel- und die Bündelnummer, nur die Bündelnummer, eine kombinierte Schachtel- und Bündelnummer oder gar keine Behälterangabe angezeigt werden. Falls eine Behälternummer angegeben wurde, unterstellt das Stylesheet, dass sie für jede Struktureinheit, außer <c01> vergeben wurde, da diese Erschließungsebene einer ganzen Gliederungsgruppe (entire series) normalerweise keine Behälternummer trägt. Wie in den Abschnitten 3 und 4 beschrieben, muss das Behälter-Element für jede Einheit vorhanden sein, braucht aber nicht immer Daten zu enthalten. Die Stylesheets sind so abgefasst, dass sie wiederholte Nummern unterdrücken. Wenn also etwa sowohl Schachtelnummern als auch Bündelnummern oder nur Schachtelnummern angegeben werden, wird eine Schachtelnummer, die bereits innerhalb eines <c01>-Elementes angezeigt wurde, nicht noch einmal für andere Einheiten innerhalb derselben Gliederungsgruppe wiederholt. Ebenso werden, wenn nur Bündelnummern oder kombinierte Schachtel- und Bündelnummern eingegeben wurden, die jeweilige Schachtel- oder Bündelnummer innerhalb derselben Gruppe nicht wiederholt.

Die beigefügten Formatvorlagen und Tastaturmakros fügen eventuell Elemente in die Dokumente ein, die nicht immer benötigt werden. Es gibt einige gute Gründe, um leere Elemente aus dem Dokument zu entfernen. Doch ist es für das Funktionieren der Stylesheets nicht erforderlich. In allen Fällen, in denen ein Problem durch leere Elemente entstehen könnte, stellt das Stylesheet fest, ob Eintragungen oder nur Tags vorhanden ist.

## 6.1 Die Ausgabeformate

Vier XSL-Stylesheets werden mit dem Kochbuch bereitgestellt. Die Grundzüge der Präsentation sind in allen vier Fällen gleich. Die Formate 1,2 und 3 unterscheiden sich nur in der Platzierung des Inhaltsverzeichnisses innerhalb des Findbuchs.

**6.1.1 *Format 1*** erstellt die Standardpräsentation des Findbuchs mit dem Inhaltsverzeichnis am Anfang des Dokumentes, wie bei einem Buch. Interne Hyperlinks verknüpfen dieses Inhaltsverzeichnis mit den verschiedenen Abschnitten des Findbuchs. Das Stylesheet hat den Namen: **EADCBS1.xsl**. Ein Beispiel für eine Anwendung des Stylesheets kann angesehen werden unter:

<http://www.mnhs.org/library/findaids/2468a.html>

**6.1.2 *Format 2*** erstellt die Standardpräsentation des Findbuchs mit einem Inhaltsverzeichnis in einem Fenster auf der linken Seite des Bildschirms. Interne Hyperlinks verknüpfen dieses Inhaltsverzeichnis mit den verschiedenen Abschnitten des Findbuchs. Dieses Format benutzt eine HTML-Tabelle, um die linke Seite mit dem Inhaltsverzeichnis zu erstellen. Das kann für Institutionen nützlich sein, deren Internetpolitik die Nutzung von Rahmen untersagt. Das Stylesheet hat den Namen **EADCBS2.xsl**. Ein Beispiel, das mit diesem Format erstellt wurde, findet sich unter:

<http://www.mnhs.org/library/findaids/2468.html>

**6.1.3 *Format 3*** erstellt ebenfalls die Standardpräsentation des Findbuchs mit dem Inhaltsverzeichnis in einem Fenster auf der linken Seite des Bildschirms. Interne Hyperlinks verknüpfen das Inhaltsverzeichnis mit den verschiedenen Abschnitten des Findbuchs. Dieses Format nutzt HTML-Rahmen für das Inhaltsverzeichnis. Der XSL Transformationsprozess, der in Abschnitt 7 beschrieben wird, kann die vier dazu erforderlichen Datei in einem Schritt herstellen. Der Name des Stylesheets ist: **EADCBS3.xsl**. Ein Beispiel für ein hiermit erstelltes Findbuch findet sich unter:

<http://www.mnhs.org/library/findaids/2468f.html>

**6.1.4 *Format 4*** erstellt eine Variante der Standardpräsentation für die Druckausgabe. Es unterscheidet sich von den vorausgegangenen Formaten in drei Aspekten. Da es für den Ausdruck bestimmt ist, besitzt es keine internen

Hyperlinks. Das Dokument beginnt mit einer ausführlicheren Beschreibung des Bestandes, die sich enger an eine übliche Titelseite eines Buches anlehnt. Außerdem ist die Beschreibung nachgeordneter Struktureinheiten in zwei Abschnitte unterteilt, die jeweils mit der Zusammenfassung einer Gliederungsgruppe beginnen, worauf die detaillierte Information zu den Verzeichnungseinheiten folgt. Dieser Ansatz gibt Lesern einen schnellen Überblick über den Bestand, ohne dass sie das ganze Findbuch nach Gruppentiteln, Laufzeiten etc. durchsuchen müssen. Er folgt mehr dem linearen Lesen eines gedruckten Dokumentes im Gegensatz zur Hypernavigation im Online-Findbuch. Die Präsentation mit der vorausgehenden Zusammenfassung ist in der Online-Version nicht vorgesehen, damit dort keine störenden Redundanzen entstehen.

Eine Druckausgabe kann hergestellt werden, indem ein EAD-Findbuch mit dem Stylesheet gelesen wird. Die daraus resultierende HTML-Datei wird dann in eine Textverarbeitung importiert, wo sie endgültig mit Einfügung von Kopf- und Fußzeilen und Seitenumbrüchen überarbeitet werden kann. Das Stylesheet wurde mit Microsoft Word 97 getestet, das HTML in das Wordformat konvertiert, ohne die Formatierung zu verlieren. Test mit Word2000 haben gezeigt, dass noch verfeinerte Funktionen wie eine hängende Einrückung bei der Konversion unterstützt werden können.

Das Stylesheet heißt **EADCBS4.xsl**. Ein Beispiel für seine Anwendung kann angesehen werden unter:

<http://www.mnhs.org/library/findaids/2468prt.html>

## 6.2 Mögliche Anpassungen

Jedes dieser Stylesheets enthält einen HTML-Tag <img> am Anfang, das dazu dient, ein Archivlogo an der Spitze des Dokuments einzufügen. Die Datei erscheint in dem SCR-Attribut mit dem zu ersetzenden Namen „youtlogo.gif“. Entweder sollte der Dateiname geändert oder das Element ganz gelöscht werden, je nachdem, ob ein Logo eingefügt werden sollt oder nicht.

## Abschnitt 7: Die Umwandlung von XML-Dateien in HTML

Die oben beschriebenen Stylesheets können XML-Dateien in HTML umwandeln. Diese Umwandlung kann entweder im Voraus, vor einer Benutzeranfrage, also im Batch-Modus, geschehen, oder in „Real-Time“, dann, wenn die Datei auf dem Server aufgerufen wird. Mit der neuesten Generation von Webbrowsern, kann diese Umwandlung statt auf dem Server auch beim Benutzer erfolgen. Allerdings beschreibt das Kochbuch noch die servergestützte Umwandlung als vorläufig vorgezogene Methode, solange XML-fähige Browser noch nicht weiter verbreitet sind. In diesem Abschnitt wird die Umwandlung im Batch-Modus beschrieben. Die Real-Time-Umwandlung wird in Abschnitt 8 beschrieben.

Es werden zwei Umwandlungsverfahren vorgestellt. Das erste nutzt das Programm XT, eine freie Software, mit dem jedes Findbuch für sich oder ein ganzer Dateiordner auf einmal konvertiert werden können. Das zweite Verfahren nutzt die XMetaL-Software, die, ab Version 1.2, eine integrierte Version von XT verwendet und damit die Umwandlung als integralen Bestandteil des Bearbeitungsprozesses stattfinden lässt.

### 7.1 XT

Dieses Werkzeug ist ein Freeware-Produkt für nicht kommerzielle Anwendung. Es ist verfügbar bei der Webadresse:

<http://www.jclark.com/xml/xt.html>

Es läuft unter einer Umgebung, die Java zulässt. Im Folgenden wird die Installation unter Windows beschrieben.

#### 7.1.1 Die Installation des Werkzeugs

- Ein Dateiordner wird angelegt, in dem Konversion der Dateien stattfinden soll.
- Die Windows-Version der XT-Software wird von der oben angegebenen Website heruntergeladen. Sie wird entpackt und installiert sich dabei.
- Die erforderlichen EAD Dateien - eadbase.ent und eine Version der ead.dtd-Datei für XML (vgl. Abschnitt 4.3.2.1 der *EAD Application Guidelines*) werden installiert.
- Eine Kopie der Microsoft-Version der Java Virtual Machine, bei Microsoft Virtual Machine, genannt wird installiert. Sie ist verfügbar als Teil der benutzergesteuerten Installation des Internet Explorers oder kann von der Microsoft-Website herunter geladen werden. Die URL ist: ([http://www.microsoft.com/java/vm/dl\\_vm32.htm](http://www.microsoft.com/java/vm/dl_vm32.htm))

### 7.1.2 Die Nutzung des Werkzeugs für die Dateiumwandlung

- Gehe zu der Kommandozeile im Verzeichnis, in dem XT gespeichert ist. Das DOS-Fenster erscheint.
- Schreibe die folgende Syntax, um die Konversion zu starten:

```
xt name_der_xml_quelle datei_name_des_xsl_stylesheets  
datei_name_der_ausgabedatei
```

for example    `xt fonds.xml eadcb3.xsl fonds.html`

Das Werkzeug wird daraufhin die DTD parsen, das Dokument parsen, das Stylesheet parsen und dann die konvertierte Datei ausgeben oder eine Fehlermeldung produzieren, wenn es ein Syntaxproblem gegeben hat..

**7.1.3** XT kann auch alle Dateien eines Verzeichnisses mit einem einzigen Befehl umwandeln.

- Schreibe die folgende Syntax, um die Konversion durchzuführen:

```
xt name_des_quellverzeichnisses name_des_xsl_stylesheets  
name_des_zielverzeichnisses
```

for example    `xt c:\xmldocs c:\styles\eadcb3.xsl  
c:\htmldocs`

- Das Stylesheet sollte nicht im Quellverzeichnis liegen.

## 7.2 XMetaL

Die XMetaL-Software enthält eine eingebaute C++ Version der XT XSL Transformationsmaschine ab der Version 1.2. Es benötigt die Version 5.0 des Microsoft Internet Explorers. Die Umwandlungsfunktion erscheint als Teil der Option für die Druckvorschau im Ansicht-Menü. Das Stylesheet für das Kochbuchformat 3 benutzt eine Ergänzung von XSL, genannt "xt:document", um die vier HTML-Dateien für die Rahmendarstellung der HTML-Version herzustellen und kann nicht mit der Version 2.1 von XMetaL wie unten beschrieben genutzt werden. Mit diesem Stylesheet sollte also die XT-Anwendung wie in Abschnitt 7.1 beschrieben, eingesetzt werden.

### 7.2.1 Installation des Werkzeuges

- Die für den Einsatz von XT erforderliche Konfiguration von XMetaL findet sich in der Datei `ead.mcr`, die dem *Kochbuch* beigelegt ist. Die Standard-Einstellung sucht das Stylesheet in dem Ordner "Display" im XMetaL-Verzeichnis. Installiere eine Kopie des Stylesheets in diesen Ordner.
- Als Standard-Wert für den Dateinamen des Stylesheet in der Datei `ead.mcr` ist die Datei `eadcbs2.xsl` angegeben. Wenn ein anderes Stylesheet verwendet werden soll, muss die Datei `ead.mcr` entsprechend verändert werden. Dazu ist folgendes zu tun:
  - ◆ Öffne die Datei mit einem Texteditor, etwa Windows Notepad. Nach etwa zwei Dritteln der Datei findet man den folgenden Text:
 

```
// Path to XSLT stylesheet is hard codedvar xslPath =
Application.Path + "\\Display\\eadcbs2.xsl";
```
  - ◆ Ändere den Dateinamen am Ende der zweiten Zeile entsprechend dem Dateinamen des zu verwendenden Stylesheets. Speichere die Datei in das entsprechende Unterverzeichnis für Makros.

### 7.2.2 Verwendung des Werkzeugs

Im Ansichtsmenü wird Seitenvorschau ausgewählt. Die Konversion geschieht und die Ergebnisdatei wird im Browser, der in XMetaL geöffnet wird, angezeigt. Die Datei kann gespeichert werden. Sie befindet sich in demselben Verzeichnis wie das Stylesheet. XMetaL gibt der Datei einen zufälligen, aber unterscheidbaren Namen, den man beliebig ändern kann.

## Abschnitt 8: Bereitstellung von EAD im Internet

Dieser Abschnitt beschreibt, was als letzter Schritt im Bereitstellungsprozess auf dem Webserver oder mit zugehörigen Dateien unternommen werden muss. Dabei ist die Unterstützung des Webmasters oder des System- oder Netzwerkadministrators erforderlich.

### 8.1 Links von anderen Webseiten

Wenn Links von anderen Webseiten zu dem Findbuch erstellt werden sollen, ist es erforderlich,

- in dem Quelldokument einen HTML-Pointer zu dem Findbuch anzulegen. Dazu sind Grundkenntnisse von HTML erforderlich. Normalerweise benötigt man das HTML-Anker-Element `<a>` mit dem HREF-Attribut.
- die Findbuchdatei in einem passenden, öffentlich zugänglichen Ort auf Ihrem Webserver abzulegen. Dazu ist der Zugang zum Dateiserver und Kenntnis über seine Dateistruktur erforderlich. Deshalb sollte der Webadministrator herangezogen werden.

### 8.2 Links von Online-Katalogen

Wenn ein Link aus MARC-Datensatz im Katalo zu dem Findbuch hergestellt werden soll, sind mehrere Schritte außerhalb der EAD-Kodierung nötig. Diese Methode setzt natürlich voraus, dass ein webfähiger Online-Katalog vorhanden ist, in dem Links zu externen Zielen verwaltet werden können.

- In dem MARC-Datensatz wird mit Hilfe des Editier-Werkzeugs, das oft Teil eines integrierten Bibliothekssystems (ILS) ist, ein Link-Feld erstellt. MARC 21 wurde vor kurzem so verändert, dass es ein zusätzliches Unterfeld bereithält. Im Moment hat man noch zwei Optionen. Ein 856-Feld, Elektronische Lokalisierung und Zugang, kann erstellt werden, das einen Pointer zum Findbuch enthält. Er kann etwa so aussehen:

856    42 \$ 3 An electronic version of the inventory for this collection is available at \$u <http://www.mnhs.org/library/findaids/00054.html>

Wenn der Hersteller des Bibliothekssystems das neue Unterfeld u im Feld 555 Cumulative Index/Finding Aids Note, unterstützt, kann der Link wie folgt aussehen:

555    0 \$ a An inventory that provides additional detailed information about this collection is available at \$u <http://www.mnhs.org/library/findaids/00054.html>

- Die Findbuch-Datei wird in einem passenden, öffentlich zugänglichen Ort auf dem Webserver abgelegt. Dazu sind der Zugang zum Server und die Kenntnis der Datei-Struktur erforderlich.

### 8.3 Zugang von Datenservern aus

Eine Reihe von Produkten sind verfügbar, wenn eine Stichwortsuche oder ein strukturierter Zugang zu den Findbüchern gewünscht wird. Dabei geht es um komplexe technische Zusammenhänge, die nicht im Kochbuch abgehandelt werden können. Verschiedene Softwaresysteme sind momentan verfügbar und es erscheinen regelmäßig neue Systeme auf dem Markt. Das Kapitel 5 der *EAD Application Guidelines* und andere Quellen geben weitere Informationen dazu.

Die XSL-Liste und ihr Archiv:

<http://www.mulberrytech.com/xsl/xsl-list>

Die XML Cover Pages (die beste Quelle für Link zu SGML/XML-Information)

<http://www.oasis-open.org/cover/>

Steve Pepper's Whirlwind Guide to SGML sowie XML Tools und Hersteller

<http://www.infotek.no/sgmltool/guide.html>

## 8.4 Transformation nach HTML auf dem Server

Abschnitt 7 beschreibt, wie Dateien im Vorraus konvertiert werden können. Mehrere Werkzeuge sind inzwischen verfügbar, die eine „Real-Time“-Konversion von XML nach HTML bei jeder Benutzeranfrage bewirken. Das Kochbuch beschreibt ihre Anwendung nicht im Detail. Der technische Prozess erfordert zusätzliche IT Kenntnisse. Deshalb ist es erforderlich, mit der IT-Abteilung zusammen zu arbeiten. Im April 2000 waren mehr als 6 größere Produkte vorhanden.

### 8.4.1 XT

Dieselbe Anwendung, die in Abschnitt 7 beschrieben wurde, kann auch als Servlet auf einem http-Server laufen. Im Archiv der XSL-Liste können, vor allem in dem Thread „XT as servlet“ weitere Informationen gefunden werden. Grundsätzliche Informationen sind verfügbar unter

<http://www.jclark.com/xsl>



### 8.4.2 SAXON

SAXON ist ein XSL-Prozessor, der als Applet oder Servlet laufen kann. Eine reduzierte Version unter dem Namen Instant Saxon kann, ähnlich wie es für XT in Abschnitt 7 beschrieben wurde, im Batch-Modus von der Kommandozeile aus gestartet werden. Informationen über SAXON sind unter der folgenden URL zu finden:

<http://users.iclway.co.uk/mhkay/saxon/>

### 8.4.3 Microsoft

Microsoft bietet einen XSLT-Prozessor als Erweiterung des ISAPI zur Nutzung mit seinen Internetservern an. Als dieser Abschnitt des Kochbuchs geschrieben wurde, waren einige der Funktionen, die zum XSLT-Standard gehören und auch in den beigefügten Stylesheets verwendet werden, noch nicht implementiert worden. Microsoft hat jedoch angekündigt, die volle Kompatibilität in regelmäßig bereitgestellten neuen Versionen herzustellen. Informationen findet man unter:

<http://www.microsoft.com/downloads/webtechnology/xml/xslisapi.asp>

### 8.4.4 Xalan

Xalan wird als XSL-Software für Apache Webserver vom Apache XML-Projekt bereitgestellt. Weitere Informationen:

<http://www.apache.org/xalan/index.html>

### 8.4.5 Cocoon

Das ist eine weitere XSL-Publikationsumgebung für Apache Webserver, entwickelt vom Apache XML Project. Weitere Informationen:

<http://www.apache.org>

### 8.4.6 LotusXSL

Das ist ein XSL-Werkzeug von IBM/Alphaworks und gehört zu dem umfangreichen Freewareangebot des Alphaworks XML/XSL Toolkit. Weitere Information:

<http://www.alphaworks.ibm.com/LotusXSL>