Zeitschrift: Arbido

Herausgeber: Verein Schweizerischer Archivarinnen und Archivare; Bibliothek Information Schweiz

Band: 18 (2003)

Heft: 3

Artikel: XML - ein strategisches Instrument für Archive?

Autor: Heuscher, Stephan / Keller-Marxer, Peter

DOI: https://doi.org/10.5169/seals-769889

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 15.07.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

XML – ein strategisches Instrument für Archive?*

Stephan Heuscher
Schweizerisches Bundesarchiv
Verantwortlich für den Bereich Datenarchitektur im eGovernment-Projekt ARELDA (vgl. S. 13).

Peter Keller-Marxer
Schweizerisches Bundesarchiv
Leiter Fachstelle ARELDA und Gesamtprojektleiter eGovernment-Projekt ARELDA

Es gibt kaum ein Anwendungsgebiet der Informatik, in dem heute die Extensible Markup Language (XML)[1] nicht in irgendeiner Form zum Einsatz kommt, vor allem beim Austausch von strukturierten Informationen zwischen unterschiedlichen Systemen. Im Verlags-, Bibliotheks- und Archivbereich ist XML nichts grundlegend Neues, handelt es sich doch um eine reine Teilmenge der Standardized General Markup Language (SGML, ISO-Standard 8879: 1986), die in diesen Bereichen seit fast zwanzig Jahren bekannt ist. Als ein «SGML für den Hausgebrauch» hat XML dieser komplexen und schwierig zu handhabenden Auszeichnungssprache innert weniger Jahre zum Sprung aus hochspezifischen Fachanwendungen in einen kaum mehr zu überblickenden Kreis von Anwendungsgebieten verholfen.

Es ist deshalb auch nicht das längst von SGML her bekannte Konzept einer semantischen Auszeichnungssprache für strukturierte Daten, das XML für Archive und Bibliotheken heute interessant macht, sondern die Tatsache, dass es sich bei XML um einen in der Industrie weltweit akzeptierten und verbreiteten Standard handelt, für den (ganz im Gegensatz zu SGML) eine breite Palette von günstiger oder kostenfreier Software zur Verfügung steht, mit der sich XML-Dokumente in einfacher Weise erstellen und verarbeiten lassen. Zudem unterstützt heute eine Vielzahl gängiger Anwendungssoftware (z.B. im Datenbankbereich) wenigstens den Import und Export ihrer Daten in XML-Formaten. Damit entsteht die neue Situation, dass zwischen den allgemeinen und vielfältigen Anwendungen der Erzeuger von Daten und den spezifischen Archivapplikationen, welche solche Daten aufnehmen sollen, keine grundsätzlichen technischen Barrieren mehr bestehen, sondern sich mit Hilfe des XML-Einsatzes auf beiden Seiten relativ einfach und effizient «offene Datenkanäle» erstellen lassen.

XML: Strukturierte Daten verständlich bewahren?

In der aktuellen Diskussion um den XML-Einsatz bei der Langzeitarchivierung wird oft vergessen, dass XML selbst keine Archivierungsstrategie resp. keinen Archivierungsansatz darstellt, sondern nur eine Technologie ist, also ein Instrument zur Erarbeitung von möglichen Archivierungslösungen.

Ein verbreitetes Missverständnis ist auch, dass XML das Ende des Datenformat-Wirrwarrs darstelle und XML-Dokumente grundsätzlich «verständlich» oder gar «selbsterklärend» seien. Beides ist falsch. XML ist kein Format, sondern eine Definition zur Definition von semantischen Formaten. Es bildet die Grundlage einer stetig wachsenden Zahl von XML-Formaten[2].

Als Technologie ist XML in gleicher Weise Obsoleszenzen unterworfen wie andere Datenformate. Niemand kann heute voraussagen, wie sich XML weiterentwickeln wird und ob es XML in 20 Jahren überhaupt noch gibt. XML-Dokumente sind zwar reine Textdokumente und damit «human-readable», also im Gegensatz zu binären Daten für den Menschen unmittelbar lesbar; damit ist jedoch die Semantik, die ein XML-Dokument auszeichnet, nicht unbedingt auch unmittelbar verstehbar. Dies sollen die beiden trivialen XML-Dokumente verdeutlichen (siehe Kasten). Sie enthalten beide dieselbe Information, nämlich zwei Einträge aus einer Liste von Personen mit deren Namen, Namenskürzel, Telefonnummer, Computer-Login und Büro.

Die erste Version ist für eine Person ohne genaue Kenntnis der Bedeutung der semantischen Auszeichnungen (XML-Tags) gänzlich unverständlich. Die zweite Version ist für eine solche Person immerhin intuitiv verständlich, da die Parameternamen «Angestellte», «Person», «telefon» etc. immerhin eine allgemein verständliche Bedeutung haben. Allerdings: Die exakte Bedeutung und

XML-Dokumente (Beispiele, siehe Text)

«Unverständliches» XML-Dokument:

```
<?xml version=«1.0»?>
<a a=«BAR»>
   <p l=«U1977» k=«Hs» t=«0313241095» r=«E43»>Heuscher</p>
   <p l=«U1976» k=«Zt» t=«0313250017»>Zürcher</p>
</a>
```

«Verständliches» XML-Dokument:

```
<?xml version=«1.0» encoding=«UTF-16»?>
<a:Angestellte amt=«BAR» xmlns:a=«http://namespaces.arelda.ch/mitarbeiterliste»
          xmlns:xsi=«http://www.w3.org/2001/XMLSchema-instance»
          xsi:schemaLocation=«http://namespaces.arelda.ch/mitarbeiterliste
                    http://schemata.arelda.ch/mitarbeiterliste_2003.xsd»>
   <a:Person login=«U1977» kuerzel=«Hs» telefon=«+41313241095» raum=«E43»>
      Heuscher</a:Person>
   <a:Person login=«U1976» kuerzel=«Zt» telefon=«+41313250017»>Zürcher</a:Person>
</a:Angestellte>
```

* Basiert auf dem Artikel *Softening the Borderlines of Archives through XML – a Case Study, Proceedings of the ERPANET workshop 'XML as a Preservation Strategy', Urbino/Italy, October 2002.*

1 *http://www.w3.org/TR/REC-xml*

2 Über 500 XML-Formate finden sich unter *http://www.oasis-open.org/cover/xml.html#applications*

der genaue Kontext bleiben unbekannt. Und: Wer weiss in 30 Jahren noch, was im Jahr 2003 ein «login» war oder welches Format eine Telefonnummer hatte? Unwahrscheinlich auch, dass man im Jahr 2103 unter dem Begriff «Angestellte» noch dasselbe verstehen wird wie heute. Und ist es ein Fehler des Datensatzes, dass bei der ersten Person eine Nummer des Büroraums angegeben wird, nicht aber bei der zweiten Person? Eine Lösung dieser Fragen deuten die Zeilen mit den Qualifizierern «xmlns» und «xsd» an, welche externe Spezifikationen eines so genannten XML-Namensraumes (namespace)[3] «a:» und eines die gültige Struktur des XML-Dokuments definierenden XML-Schemas[4] referenzieren.

Dieser Namensraum definiert einen historisierbaren Bedeutungskontext der semantischen Tags, in welchem die exakten Bedeutungen der im XML-Dokument verwendeten Semantik festgelegt ist. Dies geschieht wieder in Form von XML. Auf diesen Namensraum können sich dann weltweit alle XML-Dokumente beziehen, welche diese Semantik verwenden. Es stellt sich nun die Frage, wie sich über grosse Mengen solcher Bedeutungsdaten (z.B. Metadaten, Verzeichnungsdaten, Findmittel etc.) Recherchen realisieren lassen, die einerseits unterschiedliche hinterlegte Semantiken resp. Namensräume strikt respektieren, andererseits aber auch tolerante Ähnlichkeitsbeziehungen (z.B. Thesauri) zwischen mehreren Semantiken zulassen.

Diese grundsätzlichen Fragen – historisierte Bedeutungskontexte, Mehrfacherschliessungen durch unterschiedliche Semantiken und nach unterschiedlichen Ontologien, Suchen mittels Thesauri etc. – sind im Archiv- und Bibliotheksbereich natürlich längst bekannt und in unterschiedlichsten Typen von Verzeichnungs-, Erschliessungs- und Katalogisierungssoftware realisiert. Fast immer sind dies aber Systeme, in denen die Daten, Datenmodelle, Datenkonsistenz und Funktionalitäten in höchstem Masse mit herstellerspezifischen, proprietären Produkten (z.B. Datenbanken und Bedienungsoberflächen) verknüpft sind – womit sie selber zum Problemfall für die Langzeitarchivierung werden.

Der Punkt ist nun, dass sich die um XML herum entwickelnden (und ebenfalls auf XML beruhenden) «X-Standards» wie Namespaces, XPath[5], XPointer[6], XML-Schema, XSLT[7], XQuery[8] etc. es relativ einfach und in effizienter Form ermöglichen, all diese längst bekannten archivischen Funktionalitäten und Prinzipien beinahe vollständig aus der Abhängigkeit von spezifischer Herstellersoftware zu lösen und damit in einer für die Langzeitarchivierung tauglichen und zwischen Archiven austauschbaren Form bereitzustellen.

Hier zeigt sich, ob dem Einsatz von XML als technologische Basis einer Archiv- oder Bibliothekslösung tatsächlich strategische Bedeutung zukommt oder ob er bloss eine Modeerscheinung und Verschiebung derselben Probleme in eine neue Technologie darstellt. Es gilt sorgfältig zu identifizieren, welche Archivfunktionen durch den Einsatz von welchen Instrumenten der «schönen neuen XML-Welt» vereinfacht und unterstützt werden können oder sollen.

Im Folgenden werden einige Einsatzgebiete von XML im Archivbereich exemplarisch anhand zweier konkreter Anwendungen im Bundesarchiv dargestellt.

SIARD: System-invariante Archivierung aus relationalen Datenbanken

SIARD ist eine vom Bundesarchiv und der Firma Trivadis AG in der Programmiersprache Java entwickelte Client-Software für relationale Datenbanksysteme. Sie stellt über ein Netzwerk eine Verbindung zu jeder beliebigen Datenbank her, die eine sog. JDBC-Schnittstelle besitzt (was beinahe auf alle Datenbankprodukte zutrifft).

SIARD ist fähig, die Informationen über alle in der Datenbank enthaltenen Elemente (Kataloge, Schemata, Tabellen, Columns, Views, Constraints, Datentypen etc.) zu analysieren und diese komplette Datenstruktur zusammen mit der eigentlichen Datenbasis aus der Datenbank zu extrahieren und in Form von reinen Textdateien, unabhängig von jeglicher herstellerspezifischer Software abzuspeichern. Als Beschreibungssprache für die Datenbankstruktur dient die «Database Definition Language» von SQL3 gemäss ISO-Standard ISO/IEC 9075. Um dies zu erreichen, wandelt SIARD (automatisch oder nach Benutzereingriff) nicht SQL3-konforme, herstellerspezifische Elemente in konforme Elemente um. Wo dies nicht möglich ist, werden diese von der Archivierung ausgeschlossen. Die ausgeschlossenen Elemente und die Gründe für deren Ausschluss werden als Teil des SIARD-Archivs dokumentiert.

Nun erstellt SIARD ein «Zwischenarchiv», welches aus verschiedenen Textdateien besteht: SQL3-Dateien mit der Datenbankstruktur, «Flachtext»-Dateien mit dem Inhalt der Datenbanktabellen (den Daten) und eine XML-Datei. Diese XML-Datei enthält redundant die Struktur der SQL3-Dateien, sie enthält aber auch die Informationen zum Archivierungsprozess (z.B. Angaben über die von der Archivierung ausgeschlossenen Elemente) und allgemeine Angaben über die archivierte Datenbank (z.B. Software-Versionen, Datenmengen etc.). Vor allem enthält diese XML-Datei aber eine vordefinierte Menge von noch leeren Metadatenfeldern zur manuellen Erschliessung und Beschreibung der Datenbank. Insbesondere sind dies Felder, die sich auf archivische und nichttechnische Informationen beziehen, welche zum langfristigen Verständnis der Daten nötig, jedoch nicht in der Datenbank hinterlegt sind. Hier handelt es sich vor allem um eine allgemeine Beschreibung der Datenbank, z.B. der Provenienz und dem Entstehungs- und Verwendungszusammenhang sowie um Klartextbeschreibungen von Tabellennamen, Keywords, Code-Listen etc., also jene Angaben, welche einen dokumentierten Datenkatalog ausmachen. Der Inhalt dieser Felder wird vom Benutzer in SIARD nachgetragen und ist im endgültigen SIARD-Archiv enthalten. SIARD-Archive haben alle dieselbe normierte Struktur, unabhängig vom originalen Datenbanksystem, aus dem archiviert wurde.

Die Langzeitarchivierung von SIARD-Archiven ist aber vor allem auch unabhängig von jeglicher spezifischer Software und basiert ausschliesslich auf vollständig und offen dokumentierten Formatstandards (SQL3, XML, Unicode-Text). SIARD-Archive können zum Zwecke der Benutzung auch in Zukunft mit marginalem Aufwand in jedes proprietäre Datenbanksystem zurückgeladen werden. Dafür ist einzig erforderlich, dass dieses Produkt die Datenbanksprache SQL unterstützt. Dieser Standard stellt seit rund zwanzig Jahren die Basis fast aller relationaler Datenbanken dar, was sich auch in den nächsten zehn Jahren kaum ändern wird. Sollte SQL trotzdem einmal obsolet werden, so garantiert die strikte SQL3-Konformität der SIARD-Archive eine relativ einfache Formatmigration zum zukünftigen, neuen Standard.

XML erfüllt bei SIARD vor allem vier Funktionen:

- Softwareunabhängige Langzeitarchivierung sowie formatunspezifische Integration von technischen und archivisch-beschreibenden Metadaten, basierend auf einem normierten Metadatenmodell mit Überprüfbarkeit der Konsistenz sowie der Möglichkeit zur Historisierung (Versionierung) dieser

[3] *http://www.w3.org/TR/REC-xml-names/*
[4] *http://www.w3.org/TR/xmlschema-0/*
[5] *http://www.w3.org/TR/xpath*
[6] *http://www.w3.org/TR/xptr/*
[7] *http://www.w3.org/TR/xslt*
[8] *http://www.w3.org/TR/xquery/*

Metadatenmodelle (durch XML-Schema);

- Vereinfachung einer späteren Migration von SQL3 zu einem zukünftigen anderen Datenbeschreibungsformat (durch XSLT und die semantische Auszeichnung der SQL-Elemente); ausserdem: Konservierung beliebiger originaler und multilingualer Schriftzeichensätze aller Daten (Unterstützung von Unicode durch XML);
- Direkte Darstellbarkeit und Recherchierbarkeit aller Metadaten (inkl. Erschliessung und allg. Beschreibung) eines SIARD-Datenbankarchivs mit einem Webbrowser (durch XSLT und XQuery);
- Definition und einfacher XML-Import von Subsets der normierten SIARD-Erschliessungsmetadaten in ein proprietäres Verzeichnungssystem (im Bundesarchiv: «scopeArchive»).

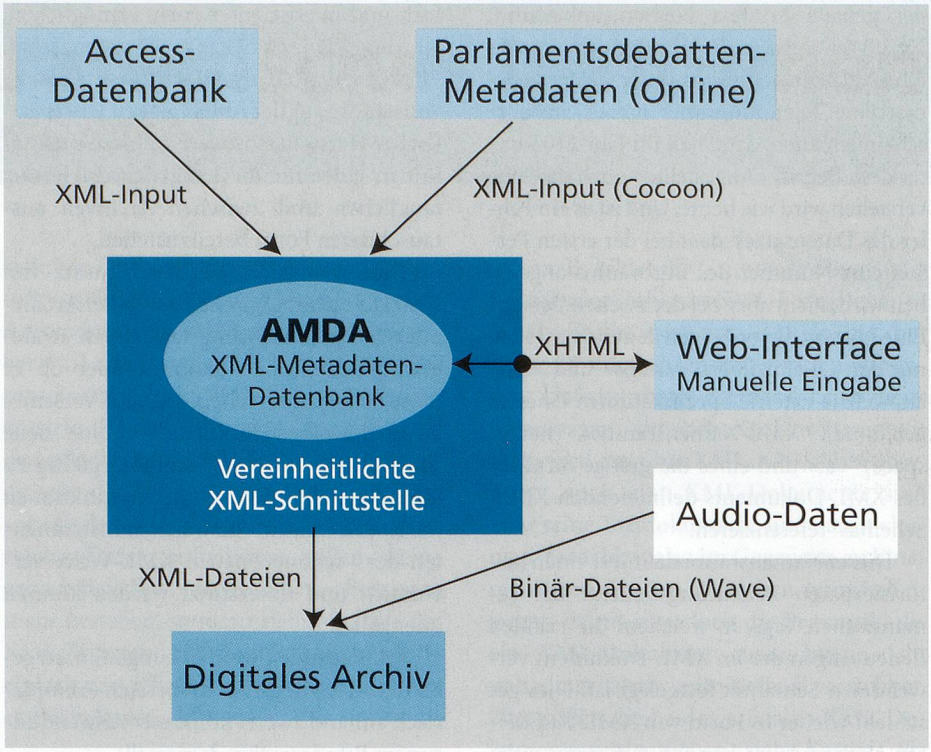

Abbildung 1: Grundablauf von AMDA.

AMDA: Erschliessung von Tonaufnahmen des Parlaments

Dieses Beispiel zeigt einen anderen Aspekt des XML-Einsatzes, nämlich die Metadaten-Akquisition aus mehreren amtsinternen und externen, sehr heterogenen Datenquellen. Den Primärdatenbestand bilden die Akzessionen der Tonaufzeichnungen aus dem Parlament. Diese liegen im Bundesarchiv für den Zeitraum 1980–2001 digital als retrospektive Digitalisierungen von analogen Tonbändern vor. Die Erschliessungsdaten dieses Zeitraums liegen in der Form von Microsoft-Access-Datenbanken vor (Erschliessung mit dem Produkt «Augias»).

Seit 2002 werden die Debatten vom Bundesarchiv im Parlament direkt digital aufgezeichnet. Die dazugehörigen elektronischen Metadaten stammen seit 1999 aus zwei unterschiedlichen Systemen, dem Geschäftsverwaltungssystem «Curia Vista» und dem Stenographiesystem (Datenbank Amtliches Bulletin) des Parlaments und sind auch auf dem Internet verfügbar[9]. Diese Metadaten werden dem Bundesarchiv von den Parlamentsdiensten seit 2002 in einer XML-Rohform pro Session zur Verfügung gestellt. Eine dritte Metadatenquelle bildet die manuelle Verzeichnung und Zusatzerschliessung der digitalen Tonaufnahmen durch das Bundesarchiv.

Die digitalen Metadaten aus diesen vier sehr heterogenen Quellen – MS Access für alte Bestände, Datenbanken «Curia Vista» und «Amtliches Bulletin» für neue Bestände, manuelle Nacherschliessung für alle Bestände – müssen mit den eigentlichen Primärdaten (geschnittene und entrauschte Tondateien im Format WAVE) verbunden werden. Und last but not least: Eine Teilmenge der Erschliessungsdaten muss in das Verzeichnungssystem des Bundesarchivs überführt werden (Produkt: «scopeArchive»).

Diese anspruchsvollen Integrationsaufgaben erfüllt die Anwendung AMDA durch den konsequenten Einsatz von XML als gemeinsames Datenaustauschformat, XSLT als Transformationssprache zwischen unterschiedlichen XML-Formaten für den Import und Export sowie XML-Schema zur Garantie der Datenkonsistenz und -integrität. Abbildung 1 zeigt die Datenflüsse schematisch. Ohne den Einsatz von XML wäre die Zusammenführung der Metadaten aus diesen unterschiedlichen Quellen mit sehr hohen Kosten für die Implementierung von Schnittstellen sowie manuellem Bearbeitungs- und Erfassungsaufwand verbunden.

Fazit

Die Erfahrungen bestätigen die Politik des Bundesarchivs, XML primär für die Metadaten, d.h. für Daten, welche Daten beschreiben, einzusetzen. Für diesen Zweck gibt es unserer Ansicht nach noch keine Standards, die sich durchgesetzt haben und unseren Anforderungen genügen. Zur Auswahl standen in erster Linie die Encoded Archival Description (EAD)[10] und der Dublin Core Metadata Standard (DC)[11]. Dank der relativ einfachen Formatumformung bildet ein eigenes XML-Format hier eine ideale Zwischenlösung, bis sich ein standardisiertes XML-Archivformat durchgesetzt hat.

Aus heutiger Sicht ermöglicht XML eine softwareunabhängige und migrationstolerante Langzeitarchivierung von Metadaten inkl. dazugehöriger Metadatenmodelle (Ontologien) sowie die softwareunabhängige Definition von Datenbeziehungen und Datenrecherchen. In den Erfahrungen hat sich insbesondere gezeigt, dass sich der Datenaustausch zwischen Systemen innerhalb und ausserhalb des Bundesarchivs und die Datenintegration mehrerer heterogener Datenquellen mit XML günstig und technisch einfach realisieren lassen. ■

contact:
E-Mails:
Stephan.Heuscher@bar.admin.ch
Peter.Keller@bar.admin.ch

Arbido
IM ABO
TEL. 031/ 300 63 41, FAX 031/ 300 63 90
E-Mail: abonnemente@staempfli.com

[9] *http://www.parlament.ch/ab/frameset/d/index.htm* (deutsch), *http://www.parlament.ch/ab/frameset/f/index.htm* (französisch)
[10] *http://www.loc.gov/ead/*
[11] *http://dublincore.org/*