kopal
Daten für die Zukunft

Projekt *kopal*
Kooperativer Aufbau eines Langzeitarchivs digitaler Informationen

Universelles Objektformat

Ein Archiv- und Austauschformat für digitale Objekte

Von

Dipl.-Inform. Tobias Steinke

Frankfurt am Main 2006

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

1. Einleitung

Im vorliegenden Text wird ein Format definiert, mit dem digitale Objekte zusammen mit Metadaten archiviert und zwischen Institutionen und Archivsystemen ausgetauscht werden können. Es basiert auf den Formaten *Metadata Encoding and Transmission Standard* (METS) in der Version 1.4 und den *Langzeitarchivierungsmetadaten für elektronische Ressourcen* (LMER) in der Version 1.2 und entstand im Rahmen des Projekts *kopal*[1]. Das verwendete Archivsystem in *kopal* ist *DIAS*[2]. Die Version dieser Software, welche im Projekt Verwendung findet, bedingt bestimmte technische Vereinbarungen, die sich auf die Beschreibung im Folgenden auswirken. Solche Systemabhängigkeiten werden entsprechend mit „***DIAS***" gekennzeichnet und kursiv gesetzt.

In der Beschreibung wird die Kenntnis der Formate METS und LMER vorausgesetzt. Insbesondere gilt das für folgende Dokumente:

- METS: An Overview & Tutorial[3]
- LMER 1.2 - Referenzbeschreibung[4]

Beispieldateien zum Universellen Objektformat (UOF) können auf der kopal-Projektseite heruntergeladen werden. Dort findet sich auch die freie Software *kopal Library for Retrieval and Ingest* (koLibRI), mit der Archivobjekte gemäß dem UOF erzeugt werden können.

[1] http://kopal.langzeitarchivierung.de/
[2] http://www.ibm.com/nl/dias/
[3] http://www.loc.gov/standards/mets/METSOverview.v2.html
[4] http://nbn-resolving.de/?urn=urn:nbn:de:1111-2005041102

2. Objektformat

Ein Archivobjekt besteht beim UOF aus einer gepackten Datei, die eine beliebige Ordnerstruktur mit beliebig vielen Dateien enthält. Auf der Wurzelebene der Ordnerstruktur muss sich eine Datei namens „mets.xml“ befinden, die eine gültige XML-Datei gemäß des Schemas von METS 1.4[5] darstellt. Die Pfade in der gepackten Datei müssen immer relativ angegeben sein. Für den Dateinamen des Archivobjekts gibt es keine besonderen Vorgaben, er sollte jedoch den Vorgaben der verwendeten Dateisysteme entsprechen und sicherstellen, dass unterschiedliche Archivobjekte auch unterschiedliche Namen tragen.

DIAS: *Erlaubte Packformate sind ZIP (kompatibel zu PKZIP Version 2.50 und höher, aber niedriger als 5.0, nur Standardkompression oder unkomprimiert), GNU-TAR (bis Version 1.15.1) und TAR. Ein Archivobjekt darf nicht mehr als 5000 Dateien enthalten. Bei Wahl des Formats ZIP dürfen die enthaltenen Dateien die maximale Größe von 2 GB pro Datei nicht überschreiten.*

[5] http://www.loc.gov/standards/mets/version14/mets.xsd

3. Aufbau der Datei mets.xml

Grundsätzlich sind gültige METS-Dateien erlaubt, die den nachfolgenden Kriterien entsprechen. Über die im Folgenden genannten Metadaten hinaus sind dabei auch beliebige weitere möglich, allerdings hängt deren praktischer Nutzen stark vom jeweiligen Archivsystem ab.

DIAS: *In DIAS wird die METS-Datei komplett mit den anderen Dateien des Archivobjekts archiviert und auch ausgeliefert. Allerdings wird nur ein Teil der Metadaten bei der Einlieferung ausgewertet und überprüft. Dieser Teil wird im so genannten Data Management gehalten und kann gezielt zu Zwecken der Langzeitarchivierung abgefragt werden.*

Die Spezifikationen von METS 1.4 führen eine Reihe von Abschnitten und Unterabschnitten in einer METS-Datei auf. Im UOF sind von diesen Abschnitten nur METS Header, Descriptive Metadata (dmdSec), Administrative Metadata (amdSec) mit den Unterabschnitten Technical Metadata (techMD) und Digital Provenance Metadata (digiprovMD), sowie File Section und Structural Map definiert. Die anderen Abschnitte dürfen aber trotzdem auftauchen, wenn sie gültiges METS 1.4 sind.

METS 1.4 erlaubt für die Abschnitte in dmdSec und amdSec beliebige Metadaten, wenn diese ein gültiges XML-Schema haben. Dabei können diese Metadaten referenziert in externen Dateien liegen oder direkt eingebunden in derselben Datei sein. Beim UOF ist nur letzteres erlaubt, d. h. alle Metadaten des Archivobjekts müssen sich in der einen Datei mets.xml befinden.

In den amdSec-Abschnitten von METS im UOF wird LMER 1.2 verwendet. Dabei wird der modulare Aufbau von LMER 1.2 genutzt. Von den LMER-XML-Schemas werden nur die folgenden eingesetzt: lmer-object.xsd, lmer-file.xsd und lmer-process.xsd. Es werden also jeweils nur die einzelnen Elemente und deren Attribute in METS integriert, nicht die LMER-Struktur, wie sie in lmer.xsd definiert ist. Somit gelten auch nicht die obligatorischen Felder, wie sie in der LMER-Referenzbeschreibung aufgelistet sind. Tatsächlich werden insbesondere alle LMER-Elemente weggelassen, die bereits durch gleiche METS-Elemente oder METS-Attribute beschrieben sind. Elemente aus LMER-Object und LMER-File finden sich in METS-Abschnitten der Kategorie amdSec/techMD und Elemente aus LMER-Process in METS-Abschnitten der Kategorie amdSec/digiprovMD.

Im Folgenden wird von einer externen ID und einer internen ID zur Kennzeichnung des Archivobjekts gesprochen. Die externen IDs sind Persistent Identifiers[6] (***DIAS:*** *URN*), welche somit weltweit eindeutig sind. Interne IDs sind dagegen nur innerhalb eines Archivsystems eindeutig. Archivobjekte, die Migrationen von anderen Archivobjekten sind, haben dieselbe externe ID, aber eine unterschiedliche interne ID.

[6] http://www.persistent-identifier.de/

3.1. METS Header

Im <mets>-Tag steht im Attribut OBJID die interne ID, wenn es sich um ein Exportobjekt aus einem Archiv handelt (nach OAIS[7]-Terminologie ein DIP).

DIAS: *Das Attribut OBJID muss leer (OBJID=“”) angegeben werden für ein Importobjekt (nach OAIS ein SIP).*

Das Attribut CREATEDATE vom <metsHdr>-Tag muss ein gültiges Datum beinhalten, welches die Erstellung bzw. letzte Aktualisierung der vorliegenden Metadaten in der Datei mets.xml benennt.

Das <agent>-Element muss mit den Attributen ROLE und TYPE, sowie mit dem Unterelement <name> vorhanden sein. Dabei wird Auskunft über die erstellende Institution (bei einem SIP) oder das Archivsystem (bei einem DIP) gegeben. In letzterem Fall kann so die interne ID in OBJID im Kontext interpretiert werden.

3.2. Descriptive Metadata (dmdSec)

In diesem Abschnitt können den Inhalt des Archivobjekts beschreibende Metadaten abgelegt werden (z. B. das bibliothekarische Katalogisat). Da Langzeitarchive zur Erhaltung ihrer Aufgaben vor allem technische Informationen benötigen und die Systeme meist ohnehin mit optimierten Katalogen verbunden sind, ist dieser Abschnitt im UOF optional. Er kann allerdings mehrfach vorkommen (***DIAS:*** *maximal 5*) und so Metadaten mit verschiedenen XML-Schemas aufnehmen (z. B. DC Simple, MODS, MABxml, etc.).

DIAS: *Wenn dmdSec-Abschnitte mit dem Attribut MDTYPE=“DC“ vorhanden sind, werden Elemente des Dublin Core Metadata Element Set, Version 1.1 (DC Simple)[8] erwartet. DIAS speichert diese im Data Management, so dass sie für gezielte Abfragen zur Verfügung stehen.*

Jede dmdSec hat ein ID-Attribut, welches in der Structural Map vom Typ „ASSET“ referenziert werden muss (siehe 3.5).

3.3. Administrative Metadata (amdSec)

Im UOF kann es eine oder mehrere amdSec-Abschnitte geben. Darin enthalten sind mindestens ein techMD-Abschnitt für Metadaten über das ganze Archivobjekt und je einen techMD-Abschnitt pro Datei, die zum Objekt gehört. Des Weiteren kann es digiprovMD-Abschnitte für das ganze Objekt und für jede Datei geben. Alle techMD- und digiprovMD-Abschnitte müssen ID als Attribut haben, damit sie in der File Section referenziert werden können.

DIAS: *Es darf maximal 5000 amdSec-Abschnitte, 5001 techMD-Abschnitte und 5001 digiprovMD-Abschnitte geben.*

---

[7] Reference Model for an Open Archival Information System:
http://www.ccsds.org/documents/650x0b1.pdf
[8] http://www.dublincore.org/documents/dces/

Im techMD-Abschnitt zum ganzen Archivobjekt finden sich Elemente aus LMER-Object. Verpflichtend ist dabei nur <persistentIdentifier>, was die externen ID benennt. <groupIdentifier> kann mehrfach vorkommen. <objectVersion> legt die Zustand als Original oder Migration fest und sollte beim Original „1“ sein. Wenn <startFile> vorkommt, dann beinhaltet dieses Element den Wert, den das ID-Attribut des zugehörigen <file> in der File Section von METS hat. Wenn vorhanden, muss <numberOfFiles> der Anzahl der in der File Section von METS aufgeführten Dateien entsprechen.

Im jeweiligen techMD-Abschnitt zu jeder einzelnen Datei finden sich Elemente aus LMER-File. Verpflichtend ist dabei nur <format> mit einem passenden Attribut REGISTRYNAME zur Kennzeichnung des verwendeten Namensraum (***DIAS:*** *Es werden URNs wie urn:diasid:fty:kopal:020050705000000000001 für PDF 1.4 verwendet und REGISTRYNAME="DIAS"*). <linkedTo> ist wiederholbar und bezeichnet wie bei <startFile> die ID des zugehörigen <file>-Elements von METS. Folgende Elemente aus LMER-File sollten weggelassen werden, da sie in der File Section von METS ohnehin zwingend auftauchen: fileIdentifier, path, name, size, fileDateTime, fileChecksum und mimeType.

Im LMER-Element <xmlData> können für jede Datei weitere technische Metadaten in einem beliebigen XML-Schema aufgeführt werden (***DIAS:*** *Das zugehörige XML-Schema muss lokal beim Archivsystem vorliegen. An dieser Stelle wird in der Praxis vor allem die XML-Ausgabe des Tools JHOVE*[9] *eingefügt.*). Das Element ist wiederholbar.

Es kann digiprovMD-Abschnitte zum ganzen Objekt und zu einzelnen Dateien geben. Die Elemente <oldMetadataRecordCreator>, <oldObjectIdentifier> und <oldObjectVersion> kann es nur beim Bezug zum Objekt geben. In <oldObjectIdentifier> wird die interne ID des Vorgängers der beschriebenen Migration aufgeführt. <oldMetadataRecordCreator> gibt dazu den Kontext an, in dem die interne ID gilt.

3.4. File Section

Die File Section von METS beschreibt alle zum Objekt gehörenden Dateien über relative Links. Sie beinhaltet genau ein <fileGrp>-Element, in dem für jede Datei ein <file>-Element existiert, welches wiederum je ein <FLocat>-Element hat. Die dadurch mit xlink:href benannten Referenzen müssen alle LOCTYPE="URL" sein und als relative Links mit „file://./“ anfangen. Folgende Attribute von <file> sind verpflichtend:

- ID
- MIMETYPE
- CREATED
- SIZE
- CHECKSUM
- CHECKSUMTYPE (***DIAS:*** *Für CHECKSUMTYPE sind nur „SHA-1“ und „MD5“ erlaubt*).

[9] http://hul.harvard.edu/jhove/

Im Attribut ADMID von <fileGrp> muss die ID des techMD-Abschnitts referenziert werden, welche Metadaten zum ganzen Objekt beinhaltet. Weiterhin werden dort die IDs der digiprovMD-Abschnitte aufgeführt, welche Migrationen des ganzen Objekts beschreiben. Entsprechend müssen im Attribut ADMID von <file> die ID des zugehörigen techMD-Abschnitts mit den Metadaten zur jeweiligen Datei und die IDs von dateibezogenen digiprovMD-Abschnitten vorkommen. Für die Reihenfolge bei der jeweiligen Auflistung der IDs in ADMID gibt es folgende Festlegung: Zuerst werden die IDs von digiprovMD-Abschnitten absteigend von der jüngsten Migration ausgehend aufgelistet und als letztes die ID des techMD-Abschnitts (es kann pro <file> bzw. in <fileGrp> nur genau ein techMD-Abschnitt zugeordnet sein).

3.5. Structural Map

Es kann beliebig viele Structural Maps geben. Allerdings muss genau eine mit dem Attribut TYPE="ASSET" vorhanden sein, welche folgenden Aufbau hat: Es gibt nur ein -Element mit TYPE="ASSET" und dieses umschließt eine Liste von <fptr>-Elementen für jede Datei des Objekts. Im jeweiligen Attribut FILEID wird die entsprechende ID aus der File Section referenziert. Im Attribut DMDID des -Elements müssen die IDs aller vorhandenen dmdSec-Abschnitte aufgeführt sein.

Neben den verpflichtenden <fptr>-Elementen können im ASSET- auch <mptr>-Elemente aufgelistet werden, welche auf externe IDs von anderen Archivobjekten verweisen, die technisch in einem Zusammenhang mit dem Objekt stehen (z. B. ein XML-Schema, welches von den enthaltenen XML-Dateien des Objekts referenziert wird). Inhaltliche Zusammenhänge (z. B. Zeitschriften) sollten über diesen Mechanismus nicht abgedeckt werden, dafür bieten sich Metadaten im dmdSec-Abschnitt an.