KOST.07
Archivierung von Unterlagen aus Geschäftsverwaltungssystemen (AUGev)
Beurteilungen verschiedener AIP-Lösungen

Inhalt

1 Einleitung 1
2 DIAS-AIP (METS-DC-LMER, Prototyp des StASG) 4
3 AIP nach OAIS (Prototyp des StAZG) 7
4 KOST-Referenzimplementierung (JAR, EAD, PREMIS) 12
5 AIP-Lösung des Stadtarchivs Baden (METS-EAD-PREMIS) 16
6 METS pur 19
7 METS-DC 22

1 Einleitung

1.1 Allgemeines

Im Projekt AUGev wurden verschiedene Ansätze zu einem AIP diskutiert. Diese stammen einerseits aus den Pilotimplementationen in St. Gallen, Zug und bei der KOST, anderseits aus der Literatur oder aus eigenen Überlegungen innerhalb des Projektteams. 6 relevante Lösungen sind hier beschrieben und beurteilt. Zur Beurteilung werden die folgenden 7 Kriterien herangezogen, die in Kapitel 2 der Empfehlungen, "Bewertungskriterien für ein AIP", detailliert beschrieben und motiviert sind[1]:

- Aus dem AIP können die notwendigen Metadaten für ein ISAD(G)-konformes Findmittel extrahiert werden.
- Notwendige ISAD(G)-Metadaten sind schemakontrolliert und nicht als Freitext oder *Key-Value*-Paare abgelegt.
- Aus dem AIP kann wieder eine universelle GEVER-Sicht (DIP) generiert werden.
- Formatmigrationen bei Übernahme oder Formatobsoleszenz können im AIP schemakonform und mit vernünftigem Aufwand abgebildet werden.
- Ordnungssystem, Metadaten und digitale Objekte sind eng miteinander verbunden.
- Unnötige Redundanz ist vermieden, d.h. die Vererbung von Informationen erfolgt entlang der Ordnungsstruktur oder durch Referenzierung.
- Es sollen möglichst wenige Standards verwendet werden.

[1] Siehe *AUGev-Empfehlungen.pdf*.

Weitere Lösungen wären denkbar, indem in den Vorschlägen einzelne Standards ausgetauscht würden; siehe dazu die separat publizierten Überlegungen zu METS-XFDU-EAD sowie zu PREMIS-LMER[2].

1.2 Verwendete Standards

1.2.1 DC

Das *Dublin Core Metadata Element Set* umfasst 15 Metadatenelemente zur Beschreibung von (analogen und digitalen) Ressourcen. DC ist vor allem im Bibliotheksbereich sowie auf dem WWW verbreitet. http://dublincore.org/documents/dces/

1.2.2 EAD

Encoded Archival Description ist ein dokumentarischer XML-Standard zur Beschreibung von Findbüchern und anderen Findhilfen in Archiven, Museen und Bibliotheken, der von der Library of Congress herausgegeben wird. http://www.loc.gov/ead/

1.2.3 ISAD(G)

General International Standard Archival Description (ISAD[G]) ist eine Beschreibung und Standardisierung von Verzeichnungstechniken und Verzeichnungselementen für Archive, die vom ICA herausgegeben wird. ISAD(G) ist ein narrativer Standard, es existiert keine Umsetzung in XML.
http://www.ica.org/en/node/30175

1.2.4 JAR

Java Archive ist eine ZIP-Datei, die zusätzliche Metadaten in einer Datei „META-INF/MANIFEST.MF“ enthält. In der Datei MANIFEST.MF sind alle Dateien mit ihrem relativen Pfad innerhalb des JAR-Files verzeichnet. Es besteht zudem die Möglichkeit, alle Dateien zur Sicherung der Integrität mit einem Hash-Wert zu versehen und die gesamte JAR-Datei zu signieren. JAR wurde ursprünglich in der Programmierung zur Verteilung von Java-Klassenbibliotheken entwickelt. Durch die grosse Verbreitung ist es auch für die digitale Archivierung interessant.
http://java.sun.com/javase/6/docs/technotes/guides/jar/jar.html
JAR Tools: *jar*, http://java.sun.com/javase/6/docs/technotes/tools/windows/jar.html;
jarsigner, http://java.sun.com/javase/6/docs/technotes/tools/windows/jarsigner.html;
keytool, http://java.sun.com/javase/6/docs/technotes/tools/windows/keytool.html.

1.2.5 LMER

Langzeitarchivierungsmetadaten für elektronische Ressourcen ist der Standardvorschlag der Deutschen Nationalbibliothek für die technischen Metadaten in der digitalen Archivierung. Er basiert auf einem Modell der *National Library of New Zealand*.
http://www.d-nb.de/standards/lmer/lmer.htm

---

[2] *Vergleich_METS-XFDU-EAD.pdf* und *Vergleich_LMER-PREMIS.pdf*.

1.2.6 METS

Der *Metadata Encoding & Transmission Standard*, entstanden aus der DTD für die digitalen Objekte im "Making of America II"-Projekt (MOA2, http://sunsite.berkeley.edu/moa2/), ist ein XML-Standard zur Verzeichnung von deskriptiven, administrativen und strukturellen Metadaten für Objekte in digitalen Bibliotheken und Archiven.
http://www.loc.gov/standards/mets/

1.2.7 MIME-Type

Multipurpose Internet Mail Extensions, auch *Internet Media Type*, ist eine Klassifikationsmethode für Dateien, die über das Internet verschickt werden. Durch die breite Verwendung im Internetverkehr bietet sich der MIME-Type auch als De-facto-Standard zur Fileformat-Bezeichnung an.
http://www.iana.org/assignments/media-types/
http://www.mimetype.org/

1.2.8 OAIS

Das *Open Archival Information System* (ISO 14721:2003) ist ein Referenzmodell für die Archivierung digitaler Unterlagen. Es spezifiziert einerseits eine funktionale Sicht auf das Archiv, anderseits ein Informationsmodell für Archivalien. Das OAIS ist als gemeinsame konzeptionelle Basis für die digitale Archivierung weitgehend anerkannt.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24683
http://public.ccsds.org/publications/archive/650x0b1.pdf

1.2.9 PREMIS

PREservation Metadata: Implementation Strategies ist eine internationale Arbeitsgruppe, die sich mit der Entwicklung von Metadatenstandards für die digitale Archivierung beschäftigt. Sie hat als Resultat ihrer Arbeit einen Standard zur Verzeichnung von administrativ-technischen Metadaten publiziert. http://www.loc.gov/standards/premis/

Anmerkung: Die Arbeiten im Projekt AUGev beruhen weitgehend auf der Version 1.0 von PREMIS. Inzwischen wurde die Version 2.0 publiziert. Für die im Rahmen des Projekts betrachteten Elemente ergibt sich dabei keine Änderung.

1.2.10 UOF

Das *Universal Object Format* ist der Ansatz für ein *Archival Information Package* (AIP), der für das Projekt kopal der Deutschen Nationalbibliothek und der Staats- und Universitätsbibliothek Göttingen entwickelt wurde. Es basiert auf METS und LMER.
http://kopal.langzeitarchivierung.de/downloads/kopal_Universal_Object_Format.pdf

1.3 Autoren

Das vorliegende Dokument wird von der AUGev-Projektgruppe als ganzes verantwortet. Die einzelnen Kapitel wurden verfasst von Martin Lüthi (2), Olivier Debenath (3), Martin Kaiser (4, 7), Georg Büchler (5) und Lambert Kansy (6).

2 DIAS-AIP (METS-DC-LMER, Prototyp des StASG)

2.1 Einleitung

Während der ersten Projektphase von AUGev entschied sich das StASG für die Umsetzung des Konzepts in einem Pilotprojekt mit dem Produkt DIAS von IBM. Es war schon vor Beginn des gemeinsamen Projektes AUGev mit der KOST und den Staatsarchiven Basel-Stadt und Zug in St. Gallen geplant, ein Pilotprojekt an einem praktischen Beispiel, wenn möglich mit einem im Handel befindlichen Produkt, durchzuführen, um vor allem auch praktische Erfahrung zu sammeln. Dabei waren mehrere Faktoren für die Wahl ausschlaggebend:

- Die Verwendung von Standardsoftware in DIAS.
- Die bereits vorhandenen Kontakte zu IBM sowie die guten Kenntnisse der Software durch den Projektleiter.
- Das Interesse von IBM, selber Manpower und Wissen in das Projekt einzubringen und auch ein gewisses Risiko einzugehen.
- Der Wunsch, ein Produkt, welches für den Bibliothekbereich konzipiert wurde, auch für den Archivbereich zu prüfen.

2.2 Aufbau

Grundlage für das AIP in DIAS ist der METS-Standard. METS beinhaltet zunächst eine zentrale *Structural Map (structMap)*, welche die (hierarchische) Struktur der Daten durch Schachtelung von *div*-Verzeichnungseinheiten beschreibt. In der *structMap* werden zudem Verzeichnungsebene *(TYPE)* und Verzeichnungstitel *(LABEL)* festgehalten. Aus der *structMap* werden eine *File Section (fileSec)*, beschreibende Metadaten (*dmdSec*) und administrative Metadaten (*amdSec*) referenziert. In der DIAS-Lösung werden für die beschreibenden Metadaten in der *dmdSec Dublin Core*, für die technischen Metadaten in der *amdSec* LMER als Metadatenstandard verwendet.

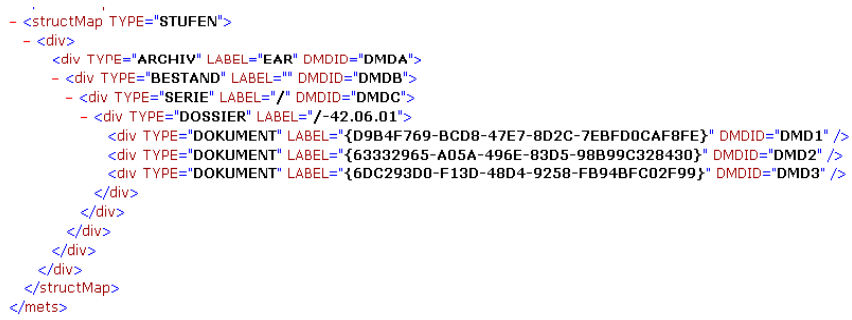
2.3 Beurteilungskriterien

2.3.1 Extraktion der Metadaten für ein ISAD(G)-konformes Findmittel

Die aus archivischer Sicht notwendigen Metadaten (minimales Set von sechs Verzeichnungselementen nach ISAD[G]) können nicht direkt im METS-Schema erfasst werden. Es muss mindestens ein weiteres Metadatenschema (z.B. Dublin Core) eingesetzt werden.

	Verzeichnungselement ISAD(G)	METS
1.1	Signatur	ID
1.2	Titel	LABEL
1.3	Entstehungszeitraum / Laufzeit	-
1.4	Verzeichnungsstufe	Position von DIV bzw. ORDER
1.5	Umfang (Menge und Abmessung)	-
2.1	Name der Provenienzstelle	-

ISAD(G)-Verzeichnungselemente und ihre Abbildung in METS



Ein Mangel von METS ist die fehlende Abbildung eines minimalen Metadatensets nach ISAD(G) innerhalb der eigentlichen Schemastruktur. Die Wrapper-Lösung ist zwar äusserst flexibel, erhöht aber die Komplexität.

Die Referenzierung von Struktur zu Metadaten entspricht eher einem relationalen Datenansatz als einem hierarchischen XML-Schema. Das Schema kann aber in eine hierarchische Form (Nutzung einer vorhandenen Archiv-Informationssystem-Software) überführt werden.

2.3.2 Schemakontrolle der notwendigen ISAD(G)-Metadaten

Die notwendigen ISAD(G)-Metadaten sind schemakontrolliert vorhanden, teils im METS-Schema, teils gemäss Dublin Core, da wie erwähnt das METS-Schema alleine nicht ausreicht.

2.3.3 Generierung einer universellen GEVER-Sicht (DIP)

Aus dem AIP können verschiedene Sichten für ein DIP erstellt werden. In DIAS wurde eine Umwandlung in HTML durchgeführt. Es besteht aber auch die Möglichkeit, dazu einen XSL-Prozess zu verwenden. Allerdings können in METS und DC zusätzliche GEVER-Metadaten nicht als *Key-Value*-Paare abgelegt und deshalb später auch nicht mehr rekonstruiert werden.

2.3.4 Abbildung von Formatmigrationen

Die technischen Metadaten in *fileSec* erlauben zusammen mit den Metadaten in der *amdSec* eine Formatmigration. Diese Konstellation wird bereits für Migrationszwecke genutzt.

2.3.5 Enge Verbindung von Ordnungssystem, Metadaten und digitalen Objekten

Sämtliche Metadaten sind in der METS-Datei gespeichert. Auf Dateien wird in der *fileSec* extern verwiesen. Die Aufteilung in Struktur, Dateiverweis und Metadaten ist nicht so konsequent, wie die Verwendung dreier unterschiedlicher Standards nahelegt, denn bereits in *structMap* (TYPE, LABEL) und in *fileSec* (MIMETYPE, SIZE, CHECKSUM etc.) sind gewisse beschreibende und technische Metadaten definiert.

2.3.6 Vermeidung unnötiger Redundanz

Die Metadatencontainer *amdSec, dmdSec, fileSec* sind über Referenzen mit der logischen Datenstruktur in *structMap* verbunden. Dies dient der Vermeidung von Redundanzen bei der Metadatenspeicherung.

2.3.7 Verwendung von möglichst wenigen Standards

Wenn der METS-Standard selber einige wenige Möglichkeiten mehr anbieten würde, könnte man auf die Nutzung von *Dublin Core* für die ISAD(G)-Werte verzichten. Ansonsten wird nur noch der Standard LMER für die technischen Metadaten genutzt. Die Nutzung von verschiedenen Standards innerhalb von METS ist eigentlich genau ein Vorteil von METS, läuft aber der Verwendung von möglichst wenigen Standards entgegen.

2.4 Schlussfolgerung

Die Vorteile von METS können sich gleichzeitig auch als Nachteile erweisen. Die grosse Vielfältigkeit lässt jedem eine grosse Freiheit in der Wahl der nutzbaren Standards. METS alleine fehlt leider die Vollständigkeit, um alle Anforderungen abzudecken. Es stellt sich jedoch die Frage, ob es nun wirklich so entscheidend ist, ob die Metadaten in METS alleine stehen oder z.B. noch *Dublin Core* genutzt wird. Vor allem im technischen Bereich bei der Nutzung von LMER bzw. PREMIS erübrigt sich wohl die Frage, da beide Standards die Anforderungen erfüllen. In METS können alle benötigen Metadaten abgelegt werden. Der Vorteil der Referenzierung ist, dass dadurch Redundanzen vermieden werden können.

3 AIP nach OAIS (Prototyp des StAZG)

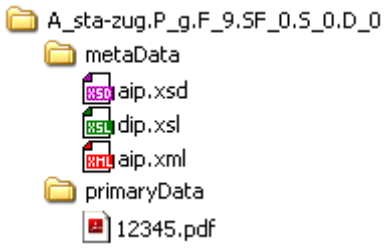
3.1 Strategische Überlegungen

Als Anfang 2007 sich das Staatsarchiv Zug entschied, am *KOST-Modellprojekt 2007* teilzunehmen, standen bereits einige strategische Entscheidungen für die elektronische Langzeitarchivierung fest. Einige dieser Vorgaben entstanden während des Projektes, andere wurden nachhaltig diskutiert und revidiert. Für die vorliegende Implementation *ch.stazg.joai.aip* wurden schliesslich die folgenden Vorgaben berücksichtigt:

- Plattform- und applikationsunabhängige Speicherung
- Dateisystembasierende Speicherung
- Archivgut soll in einem OAIS-konformen Paket (AIP) verzeichnet und gespeichert werden.
- Das AIP soll einen hohen Autonomiegrad erhalten. Die daraus resultierende Redundanz wurde in Kauf genommen.
- Das AIP soll physisch ein Verzeichnis darstellen, welches Primärdaten und Metadaten in Dateiform enthält.
- Das AIP ist eine Teilmenge eines SIP. Üblicherweise wird auf Stufe Dossier erschlossen.
- Die Beschreibung und Validierung der Paketbeschreibung erfolgt mit Hilfe von XML/XSD-Technologie. Die hierfür notwendigen Daten sind im AIP mitgespeichert. Dadurch ist das AIP selbstbeschreibend. Für die syntaktische und semantische Überprüfung der Paketbeschreibung ist ein XML-Schema mitgespeichert.
- Das AIP enthält sowohl semantische als auch technische Metadaten.
- Das zugehörige DIP wird mit Hilfe einer XSL-Transaktion ad hoc erstellt. Das XSL-Stylesheet ist im AIP mitgespeichert.
- Das AIP wird als unkomprimiertes JAR (Java Archive) gespeichert.
- Eine allfällige Komprimierung wird an das DMS/Dateisystem delegiert.

3.2 Aufbau

Die physikalische Speicherung der Primär- und Metadaten erfolgt in einem normierten Verzeichnis:



Der Einhängepunkt (*mount point*) ist ein Verzeichnis, welches den Signaturnamen trägt, z.B. *A_sta-zug.P_g.F_9.SF_0.S_0.D_0.* Dieses Verzeichnis ist in zwei Bereiche unterteilt, in *metaData* und *primaryData*. MetaData enthält die Paketbeschreibung *(aip.xml)*, das zugehörige XML-Schema *(aip.xsd)* sowie das XSL-Stylesheet *(dip.xsl)*, welches die Transformation zwischen AIP und DIP ausführt. Im Weiteren wird nur auf die Paketbeschreibung *aip.xml* eingegangen.

Die Paketbeschreibung *aip.xml* besteht aus drei "Namensräumen", die jeweils einen Teil der Metadaten abdecken. (Der Begriff Namensraum wird hier nicht im Sinne der Sprachbeschreibung von XML verwendet.) Aus pragmatischen Gründen besteht die Paketbeschreibung aus dem monolithischen Namensraum *joai*. Der äussere Rahmen dieses Namensraumes ist durch das Objektmodell des *OAIS* gegeben:

```
<archiveInformationPackage>
    <preservationDescriptionInformatio>
        <provenanceInformation>
        </provenanceInformation>
        <referenceInformation>
        </referenceInformation>
        <contextInformation>
        </contextInformation>
    </preservationDescriptionInformatio>
    <contentInformation>
    </contentInformation>
</archiveInformationPackage>
```

Im *OAIS* werden keine näheren Angaben zum Bereich **<contextInformation>** gemacht. Daher wird für Verzeichnungsangaben die ISAD(G)-Nomenklatur hinzugezogen:

```
<contextInformation>
    <archive>
        <partition>…
            <fonds>…
                <subFonds>…
                    <series>…
                        <subSeries>…
                            <dossier>…
                                <content>…
                                    <object/>
                                <content>
                            <dossier>
                        </subSeries>
                    </series>
                </subFonds>
            </fonds>
        </partition>
    </archive>
</contextInformation>
```

Für die Abbildung der technischen Metadaten wird schliesslich der LMER-Standard verwendet, indem dieser in den Bereich der **<contentInformation>** eingebaut wird:

```
<contentInformation>
    <object>
        <process>
        </process>
        <file>
        </file>
    </object>
</contentInformation>
```

Damit ergibt sich für die Beschreibung eines AIP folgendes Gerüst:

```
<archiveInformationPackage>
    <preservationDescriptionInformatio>
        <provenanceInformation>
        </provenanceInformation>
        <referenceInformation>
        </referenceInformation>
        <contextInformation>
            <archive>
                <partition>…
                    <fonds>…
                        <subFonds>…
                            <series>…
                                <subSeries>…
                                    <dossier>…
                                        <content>…
                                            <object/>
                                        <content>
                                    <dossier>
                                </subSeries>
                            </series>
                        </subFonds>
                    </fonds>
                </partition>
            </archive>
        </contextInformation>
    </preservationDescriptionInformatio>
    <contentInformation>
        <object>
            <process>
            </process>
            <file>
            </file>
        </object>
    </contentInformation>
</archiveInformationPackage>
```

3.3 Bewertungsschema

3.3.1 Extraktion der Metadaten für ein ISAD(G)-konformes Findmittel

Die semantischen Metadaten werden nach ISAD(G) gespeichert:

```
<contextInformation>
    <archive>
        <signature/>
        <title/>
        <systemOfArrangement/>
        <note/>
        <partition>
            <signature/>
            <title/>
            <systemOfArrangement/>
            <fonds>
                <signature/>
                <title/>
                <creationDateBegin/>
                <creationDateEnd/>
```

```
                    <administrativeHistory/>
                    <appraisalInformation/>
                    <schedulingInformation/>
                    <systemOfArrangement/>
                    <subFonds>
                        <signature/>
                        <title/>
                        <creationDateBegin/>
                        <creationDateEnd/>
                        <appraisalInformation/>
                        <destructionInformation/>
                        <systemOfArrangement/>
                        <series>
                            <signature/>
                            <title/>
                            <creationDateBegin/>
                            <creationDateEnd/>
                            <appraisalInformation/>
                            <systemOfArrangement/>
                            <subSeries>
                                <dossier>
                                    <signature/>
                                    <signatureOld/>
                                    <creationDateBegin/>
                                    <creationDateEnd/>
                                    <title/>
                                    <producer/>
                                    <actor/>
                                    <registry/>
                                    <legalStatus/>
                                    <systemOfArrangement/>
                                    <language/>
                                    <content>…
                                        <object/>
                                    <content>
                                <dossier>
                            </subSeries>
                        </series>
                    </subFonds>
                </fonds>
            </partition>
        </archive>
</contextInformation>
```

3.3.2 Schemakontrolle der notwendigen ISAD(G)-Metadaten

Sämtliche Metadaten, welche in der Datei *\metaData\aip.xml* gespeichert sind, werden durch das monolithische Schema *\metaData\aip.xsd* kontrolliert.

3.3.3 Generierung einer universellen GEVER-Sicht (DIP)

Die AIP/DIP-Transformation wird mittels eines XSL-Prozesses realisiert. Die XML-Repräsentation *\metaData\aip.xml* enthält einen statischen Verweis auf das XML-Stylesheet *\metaData\dip.xsl.*

Die statische Referenz könnte von der Logik-Schicht dynamisch verändert werden, so dass verschiedene "Sichten" auf das AIP hergestellt werden könnten. Das vorliegende

XML-Stylesheet liest eine Teilmenge des AIP aus und stellt diese tabellarisch dar. Dabei wird striktes HTML 4.0 verwendet, ohne CSS-Technologie.

3.3.4 Abbildung von Formatmigrationen

Die Genealogie migrierter Versionen wird auf der untersten Ebene der Tektonik im Bereich `<contextInformation>` abgebildet. Dort werden Contentobjekte angelegt, die auf dazugehörige LMER-Dateien verweisen. Mit einer XPath-Anfragen kann die Genealogie ausgelesen werden.

3.3.5 Enge Verbindung von Ordnungssystem, Metadaten und digitalen Objekten

Die Ordnungsstruktur-Metadaten werden zusammen mit den technischen Metadaten in der XML-Darstellung *metaData**aip.xml* abgebildet. Die Ordnungsstruktur-Metadaten sind im Bereich der `<contextInformation>` gespeichert. Unter "digitalem Objekt" wird in der vorliegenden AIP-Implementation die Einheit von elektronischer Datei und zugehörigen Prozessinformationen (nach LMER-Standard) verstanden, die eigentlichen technischen Metadaten also. Diese werden ebenfalls in der Datei *metaData**aip.xml* im Bereich `<contentInformation>` gespeichert. Die tatsächlichen digitalen Primärdaten – die PDF/A-Dateien – liegen im Verzeichnis *primaryData* und werden aus *metaData**aip.xml* mittels einer URL referenziert.

3.3.6 Vermeidung unnötiger Redundanz

Das AIP ist in sich redundanzfrei. Dafür sorgt zunächst das OAIS-Datenmodell, welches bereits normalisierte "Informationseinheiten" formuliert. Die eingebaute ISAD(G)-Verzeichnung sorgt durch die mehrstufige Verzeichnung für Redundanzfreiheit, da die Kontextinformationen jeweils auf der richtigen Ebene untergebracht werden (sollen). Die technische Metadaten-Speicherung nach LMER ist ebenfalls redundanzfrei, da die Entitäten "Objekt", "Prozess", "Datei" normalisiert abgebildet werden.

Zwischen den verschiedenen AIPs gibt es allerdings Redundanzen: Da in jedem Paket das zugehörige XML-Schema und XML-Stylesheet mitgespeichert werden, ergeben sich Mehrfachspeicherungen. Ebenfalls sind die Kontextinformationen in diesem Sinne redundant mitgespeichert, nämlich dadurch, dass jedes Paket "genotypisch" die gesamte Tektonikverzeichnung in sich mitträgt. Diese "externen" Redundanzen wurden aber bewusst in Kauf genommen als Preis dafür, dass den einzelnen AIP eine hohe Autonomie zukommt.

4 KOST-Referenzimplementierung (JAR, EAD, PREMIS)

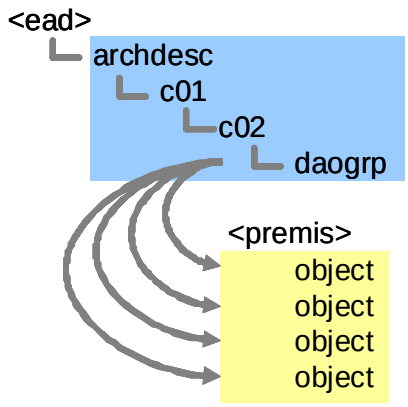
4.1 Kurzbeschreibung

Die KOST-Referenzimplementierung präsentiert sich wie folgt:

- Die archivischen Metadaten werden in einer EAD-Datei festgehalten (Ordnungssystem und ISAD[G]-Metadaten).
- In der EAD-Struktur wird auf Dokumentebene *(level item)* mit *X-Pointer* auf einen entsprechenden Objekteintrag in einer PREMIS-Datei verwiesen.
- Die PREMIS-Datei enthält pro Dokument einen Objekteintrag mit den technischen Metadaten und der URL der zugehörigen Primärdateien. Ein Eventeintrag beschreibt die administrativen Abläufe (z.B. Formatmigration).
- Die Primärdaten sind in einem Ordner *primaryData* (optional mit Unterverzeichnis), die XML- und Schemadateien (*ead.xml* und *premis.xml*) sind in *metaData* abgelegt.
- Beide Ordner bilden in einer JAR (Java Archive Format)-Datei das AIP.
- In der JAR-Manifest-Datei wird jede Datei im AIP zur Sicherung der Integrität mit einem Hash-Wert versehen.

4.2 Aufbau des AIP

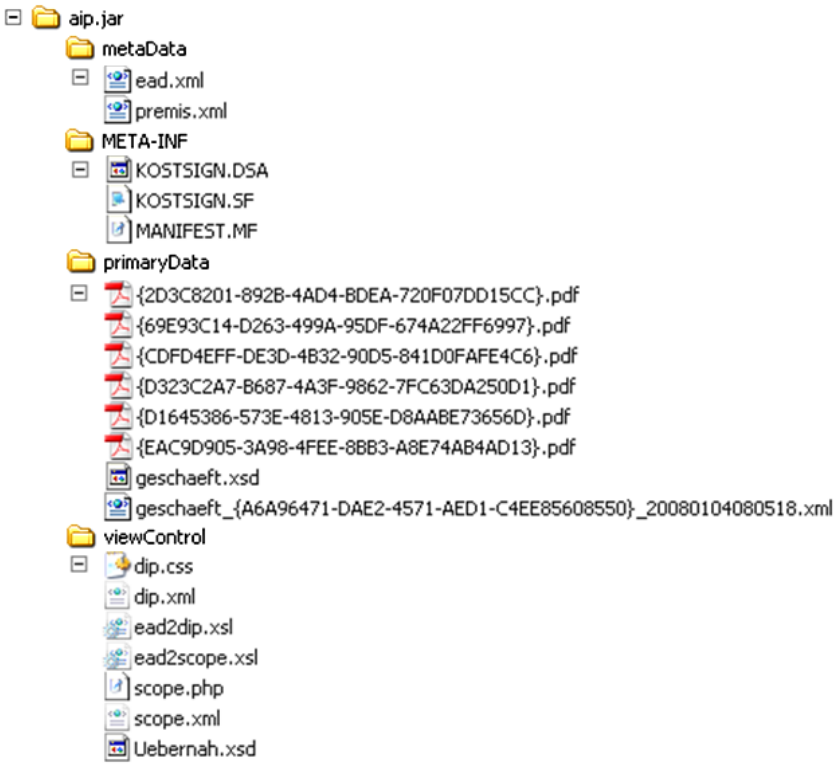
Die inhaltliche Struktur des AIP ist in der EAD-XML-Datei festgehalten. Das AIP wird dadurch Teil der Archivtektonik. Metainformationen auf Verzeichnungsebene (c01 bis c0x) werden hierarchisch vererbt. In der PREMIS-XML-Datei hingegen sind die technischen und administrativen Eigenschaften der archivierten digitalen Objekte beschrieben.



EAD und eine PREMIS-Datei durch XPointer verbunden

Der physische Aufbau des AIP ist schliesslich durch die Verzeichnisstruktur in der JAR-Datei bestimmt:

- *META-INF* beschreibt Struktur und Dateien im JAR-File
- *metaData* enthält die XML-Dateien.
- *primaryData* enthält die Primärdateien und die Ablieferungsmetadaten
- *viewControl* erzeugt ein DIP und eine scopeArchiv-Übernahmedatei



Die Dateien im JAR-File

4.3 Analyse gemäss Kriterien

4.3.1 Extraktion der Metadaten für ein ISAD(G)-konformes Findmittel

Sämtliche für ein ISAD(G)-konformes Findmittel notwendigen Metadaten sind im EAD-File *ead.xml* abgelegt. Mittels einer XSLT-Transformation *ead2dip.xsl* und eines Stylesheets *dip.css* kann dieses File direkt als DIP für das Dossier benutzt werden. Zudem kann es mit einer weiteren Transformation *ead2scope.xsl* in ein Archivinformationssystem importiert werden. Diese Lösung implementiert ein *self-contained AIP*: Alle archivisch notwendigen Informationen sind im AIP selber enthalten.

4.3.2 Schemakontrolle der notwendigen ISAD(G)-Metadaten

Da für die deskriptiven und strukturellen Metadaten EAD verwendet wird, für welches ein *Crosswalk* zu ISAD(G) besteht[3], sind die ISAD(G)-Metadaten schemakontrolliert speicherbar. Es gilt dies insbesondere für die aus archivischer Sicht zentralen Elemente Signatur, Titel, Entstehungszeitraum/Laufzeit, Verzeichnungsstufe und Provenienz. Es muss allerdings darauf hingewiesen werden, dass die Zuordnung ISAD(G)→EAD oftmals mehrdeutig ist und erst durch zusätzliche Spezifikationen, wie SIP-Schnittstelle und Findmittelschnittstelle, definiert wird.

[3] Siehe dazu *ISAD(G)-Metadaten.pdf*.

4.3.3 Generierung einer universellen GEVER-Sicht (DIP)

GEVER-spezifische Metadaten, welche keine Entsprechung im ISAD(G)-Modell finden, werden in eine *Key-Value*-Struktur in der EAD-Datei abgelegt. Sie können so in einem DIP in ihrer semantischen Bedeutung wieder sichtbar gemacht werden.

4.3.4 Abbildung von Formatmigrationen

In PREMIS verweist ein Event (z.B. Ingest) auf die betroffenen Objekte. Im Detail ist der Fall Migration in der Referenzinstallation noch nicht realisiert.

4.3.5 Enge Verbindung von Ordnungssystem, Metadaten und digitalen Objekten

Die Ordnungsstruktur-Metadaten sind im EAD-File *ead.xml* abgelegt. Daraus wird über die PREMIS-Datei *premis.xml* auf die im gleichen AIP liegenden primären digitalen Objekte verwiesen. Das JAR-Format hält alle zum AIP gehörigen logischen Dateien in einer physischen Datei zusammen.

4.3.6 Vermeidung unnötiger Redundanz

Minimale Redundanz entsteht durch die Tatsache, dass in jedem AIP das gesamte Ordnungssystem dokumentiert ist. Dies ist jedoch kein Unterschied zu den anderen untersuchten Lösungen. Durch die Vererbung innerhalb der EAD-Struktur *(archdesc)* wird diese Redundanz aber auf ein Minimum beschränkt. Es gibt keine Überschneidung/Doppelspurigkeit, was inhaltliche und technische Metadaten betrifft; diese sind klar in EAD und PREMIS getrennt.

4.3.7 Verwendung von möglichst wenigen Standards

Es werden EAD und PREMIS als in der Archiv- und Bibliothekswelt auch international akzeptierte Metadatenstandards verwendet. Das Containerformat JAR ist in der Welt der Informatik gut verankert.

4.4 Zusätzliche Überlegungen

4.4.1 Logische Struktur

Das AIP (eine Ablieferung, ein oder mehrere Dossiers) wird physisch in einer JAR-Datei zusammengehalten. Sämtliche Dateien des AIP sind in der JAR-Manifest-Datei referenziert und signiert, das heisst, eine Veränderung im AIP kann leicht erkannt werden. Archivische Metadaten sind alleine in der EAD-Datei festgehalten, administrative und technische Metadaten in der PREMIS-Datei.

Folgende Verknüpfung wird damit realisiert:
AIP (JAR)→EAD (logische Struktur)→PREMIS (physische Struktur)→Primärdaten

Jede Ebene beschreibt die jeweils zugehörigen Metadaten ohne Redundanz.

4.4.2 Datenmigration

Bei eine späteren allfälligen Datenmigration bleibt die EAD-Datei unberührt. Die Migration wird auf der Ebene *event* alleine in der PREMIS-Datei festgehalten. Über die

PREMIS-Dateien werden Zustand und Veränderungen im Archiv dokumentiert und damit ein eigentliches Protokoll erstellt.

4.5 Zusammenfassung

Die Lösung JAR–EAD–PREMIS zeigt eine relativ einfache Metadatenstruktur: eine klare Trennung von GEVER-Metadaten, administrativen-technischen Daten und Primärdaten; auch sind die Dokumente als logische Objekte klar von den physischen Primärdateien getrennt, was die Verwaltung von Versionen und Migrationen erleichtert. Ungewöhnlich ist JAR als Containerformat. JAR ist jedoch im Prinzip ein ZIP mit einer Verzeichnis-/Signatur-Datei und kann auch als ganz gewöhnliches ZIP behandelt werden.

5 AIP-Lösung des Stadtarchivs Baden (METS-EAD-PREMIS)

Dieser Ansatz für ein AIP wurde von Docuteam GmbH[4] entwickelt und im Stadtarchiv Baden implementiert. Docuteam bewirbt den Ansatz auch als generische Lösung für digitale Archive, siehe http://www.docuteam.ch/darc.htm. Zur Badener Implementation siehe ferner auch die Präsentation von Tobias Wildi am AeA-Workshop vom 21. November 2007, http://www.vsa-aas.org/uploads/media/Wildi_docuteam.pdf. Die KOST konnte die Lösung nicht detailliert anschauen oder testen; die hier vorgelegte Analyse beruht deshalb ausschliesslich auf den genannten Quellen.

5.1 Aufbau des AIP

ZIP-File	***Dossier***
L mets.xml	*Packaging Information*
L ead.xml	*Descriptive Information, Reference Information*
L premis.xml	*Preservation Description Information, Fixity Information, Provenance Information*
L datei1.pdf	*Digital Object*
L datei2.pdf	*Digital Object*
L …	…

Die Badener Lösung versteht sich als direkte Umsetzung des OAIS. Das AIP wird als Dossier implementiert, verpackt in einem ZIP-File. (Dafür verwendet Docuteam den Begriff "digitales Objekt", was angesichts der unterschiedlichen Verwendung im OAIS leicht missverständlich ist.) Im ZIP-File liegen in flacher Ordnerstruktur die Primärdaten (im Beispiel ausschliesslich, aber nicht notwendigerweise im PDF-Format) sowie drei Metadatenfiles: *mets.xml*, *ead.xml*, *premis.xml*.

- *Descriptive Information* nach OAIS wird im EAD-File abgelegt.
- *Preservation Description Information* nach OAIS wird im PREMIS-File abgelegt. Darunter fällt insbesondere auch die Dokumentation zu Migrationen.
- *Packaging Information* nach OAIS wird im METS-File abgelegt. Dieses dient dabei ausschliesslich als Referenz auf die anderen Files, also gewissermassen als Manifest.

Die drei Metadateien beschreiben somit unabhängig jeweils einen andern Aspekt des Dossiers und sind untereinander nur indirekt, nämlich durch ihre Verlinkung der Primärdateien, verbunden.

Das EAD-File bietet einen strukturierten Zugriff auf das digitale Objekt. Es kann zusätzlich in eine Archivdatenbank importiert werden.

Die digitalen Objekte, d.h. die ZIP-Container, werden mit einem *Persistent Identifier* versehen, welcher eine rein interne Laufnummer ist. Sie können als einfachste Lösung auf einem Fileserver abgelegt werden; für grössere Bestände empfiehlt Docuteam ein Fedora-Repository[5].

[4] http://www.docuteam.ch/.
[5] http://www.fedora-commons.org/.

5.2 Analyse gemäss Kriterien

5.2.1 Extraktion der Metadaten für ein ISAD(G)-konformes Findmittel

Sämtliche für ein ISAD(G)-konformes Findmittel notwendigen Metadaten sind im EAD-File abgelegt. Mittels eines Stylesheets kann dieses File direkt als Detailfindmittel für das Dossier benutzt werden. Zudem kann es in ein Archivinformationssystem importiert oder mit anderen Detailfindmitteln konkateniert werden. Diese Lösung implementiert ein *self-contained AIP*: Alle archivisch notwendigen Informationen sind im AIP selber enthalten.

5.2.2 Schemakontrolle der notwendigen ISAD(G)-Metadaten

Da EAD für die deskriptiven und strukturellen Metadaten verwendet wird, zu welchem ein Crosswalk zu ISAD(G) besteht, sind die ISAD(G)-Metadaten schemakontrolliert speicherbar. In diesem Zusammenhang muss allerdings auf gewisse Kompatibilitätsprobleme hingewiesen werden, wie sie auch im KOST-Prototypen dokumentiert sind.

5.2.3 Generierung einer universellen GEVER-Sicht (DIP)

Es ist aus unseren Unterlagen nicht ersichtlich, wo und in welcher Form GEVER-spezifische Metadaten abgelegt werden. Strukturmetadaten finden zwar im EAD-File ihren Platz, aber zusätzliche, aus archivischer Sicht sekundäre GEVER-spezifische Metadaten können nirgends speziell abgelegt werden.

5.2.4 Abbildung von Formatmigrationen

Wie die Datenmigration unterstützt werden soll, ist unklar. Grundsätzlich ist davon auszugehen, dass eine migrierte Datei zusätzlich in den ZIP-Ordner eingefügt würde. Dazu müssten die Referenzen in den drei Metadaten-Files angepasst werden. Die Geschichte der einzelnen Files wäre in den PREMIS-Metadaten nachzuvollziehen. In der KOST-Referenzimplementierung wurde vermieden, das EAD-File, d.h. den Archivkatalog, anpassen zu müssen, da technische Informationen nicht in das Findmittel gehören.

Da der Badener Ansatz eine Konvertierung der Dateien in langfristig aussichtsreiche Dateiformate zum Zeitpunkt der Archivierung vorsieht, wobei die Originaldateien offensichtlich nicht aufbewahrt werden, kann davon ausgegangen werden, dass in absehbarer Zeit keine Datenmigration nötig sein wird. Dementsprechend sind diesbezügliche Überlegungen nicht prioritär.

5.2.5 Enge Verbindung von Ordnungssystem, Metadaten und digitalen Objekten

Die Ordnungsstruktur-Metadaten sind im EAD-File abgelegt. Daraus wird jeweils auf die im gleichen Dossier liegenden primären digitalen Objekte verwiesen.

5.2.6 Vermeidung unnötiger Redundanz

Minimale Redundanz entsteht durch die Tatsache, dass in jedem Dossier der gesamte Pfad zum logischen Ort des Dossiers im EAD-File dokumentiert ist. Dies ist jedoch kein Unterschied zu den anderen untersuchten Lösungen.

5.2.7 Verwendung von möglichst wenigen Standards

Die Badener Lösung verwendet die Metadatenstandards METS, EAD und PREMIS sowie das Paketformat ZIP, somit relativ viele Standards, welche aber alle als gut eingeführt und verbreitet bezeichnet werden können.

5.3 Zusätzliche Überlegungen

5.3.1 Logische Struktur

Das Dossier = AIP = digitales Objekt wird physisch in ZIP-Form zusammengehalten. Zudem sind sämtliche Dateien des AIP auch in der Datei *mets.xml* referenziert. Das AIP ist *self-contained*.

Die Tatsache, dass zur Beschreibung des Dossiers drei selbständige Metadatenfiles verwendet werden, hat den Vorteil, dass die einzelnen Metadaten nach Sinneinheiten getrennt sind und unabhängig voneinander manipuliert werden können: deskriptive Metadaten in EAD, administrativ-technische in PREMIS, METS als reines Manifest. Damit verknüpft ist jedoch der Nachteil, dass diese Metadatenfiles untereinander nicht verbunden sind. Jedes beschreibt nur einen Aspekt; die einzige Möglichkeit, sie miteinander in Verbindung zu bringen, ist der Umweg über die von ihnen beschriebenen Dateien.

5.4 Zusammenfassung

Die METS-EAD-PREMIS-Lösung besticht durch ihre relative Einfachheit und durch die saubere Trennung der verschiedenen Metadaten. Negativ bewertet werden die mangelnde Verlinkung der Metadaten sowie die unklare Situation in Sachen Migration.

6 METS pur

6.1 Beschreibung

Bei der Lösung METS-pur wird ein AIP mithilfe der Standards METS, *Dublin Core* und PREMIS gebildet. Hierbei wird die logische, meist aber nicht zwingend hierarchische, Struktur eines AIP in der *Structural Map* in METS abgebildet. Alle Einheiten des Ordnungssystems und Verzeichnungseinheiten werden dabei durch je ein *div*-Element repräsentiert. Auf diese Weise können beliebige hierarchische Ordnungsstrukturen umgesetzt werden.

Für die Erschliessungsmetadaten wird *Dublin Core* eingesetzt. Die DC-Metadaten werden in Form eines *mdWrappers* erfasst. Für jedes *div*-Element besteht demzufolge ein eigener DC-Metadatensatz resp. *dmdSection*. Die technischen Metadaten werden in Form einer PREMIS-Datei im Abschnitt *techMD* referenziert. Als weiteres Element werden die dem AIP zugehörigen Dateien im der *FileSection* aufgelistet.

Die Referenzierung zwischen den Elementen

dmdSec
amdSec
techMD
fileSec
structMap

erfolgt über die Vergabe von IDs in jedem Element und entsprechenden ID-Referenzen respektive im Fall der Dateien URN o. ä. Die technischen Metadaten eines *div*-Elements werden über Referenzierung des PREMIS-Files mit Angabe des spezifischen Einsprungs- und Ausstiegspunktes verknüpft.

6.2 Analyse gemäss Kriterien

6.2.1 Extraktion der Metadaten für ein ISAD(G)-konformes Findmittel

Die für ein ISAD(G)-konformes Findmittel notwendigen minimalen Metadaten können aus der *dmdSection* extrahiert werden:

ISAD(G)	*DC*
Signatur	Identifier
Titel	Title
Entstehungszeitraum	Date, Coverage
Provenienz	Creator, Publisher, Contributor
Umfang	Format
Verzeichnungsstufe	Type(?)

Mit Ausnahme des letzten Elements können alle ISAD(G)-Metadaten DC-Elementen zugeordnet werden – allerdings nicht immer eindeutig. Damit ist der Einsatz von *Dublin Core* nicht selbsterklärend möglich, sondern muss zusätzlich definiert werden. Die Verzeichnungsstufe lässt sich im Element *Type* abbilden, allerdings muss dies ebenfalls

als Konvention festgehalten werden und ist überdies nicht ohne weiteres nachvollziehbar.

6.2.2 Schemakontrolle der notwendigen ISAD(G)-Metadaten

Durch den Einsatz von *Dublin Core* für die Beschreibungsmetadaten ist weitgehend sichergestellt, dass die für ein ISAD(G)-konformes Findmittel erforderlichen Metadaten schemakontrolliert abgebildet werden. Durch die Mehrdeutigkeit einiger Zuordnungen sowie die nicht befriedigende Abbildung der Verzeichnungsstufe ist dieses Kriterium jedoch nicht vollkommen erfüllt.

6.2.3 Generierung einer universellen GEVER-Sicht (DIP)

Es ist durch die Verwendung von *Dublin Core* für die beschreibenden Metadaten nicht möglich, beliebige Metadaten aus GEVER-Systemen schemakontrolliert abzubilden im AIP. Auch innerhalb von METS selbst fehlt diese Möglichkeit. Wäre freilich ein akzeptiertes XML-Schema eines GEVER-Metadatenstandards vorhanden, liesse sich dieses Kriterium in METS erfüllen.

6.2.4 Abbildung von Formatmigrationen

Durch die Verwendung von PREMIS und den Einsatz des Elements *filegroup* in der *techMD*-Section können Formatmigrationen eindeutig abgebildet werden. Es muss dazu die PREMIS-Datei geändert werden und zusätzlich innerhalb jeder *filegroup* ein Dateieintrag hinzugefügt werden. Dies ist eine Schwachstelle, da bei einer Migration zwei voneinander weitgehend unabhängige Bereiche des AIP manipuliert und aktualisiert werden müssen.

6.2.5 Enge Verbindung von Ordnungssystem, Metadaten und digitalen Objekten

Dieses Kriterium wird weitgehend erfüllt, da Ordnungssystem- und beschreibende Metadaten in einer Datei vorliegen und die technischen Metadaten sowie die Primärdaten über eindeutige Referenzierungsregeln mit dieser Datei verknüpft sind.

6.2.6 Vermeidung unnötiger Redundanz

Dieses Kriterium ist weniger von der Struktur des AIP abhängig als vielmehr von der Granularität des AIP. In dieser Hinsicht kann das Kriterium voll erfüllt werden, wenn ein DIP in ein AIP umgewandelt wird und dieses sämtliche Inhalte einer Ablieferung enthält. Repräsentiert ein AIP jedoch ein Dossier, dann wird das gesamte Ordnungssystem in jedem AIP wiederholt und das Kriterium nicht erfüllt.

6.2.7 Verwendung von möglichst wenigen Standards

Mit drei Standards bewegt sich die Lösung im Bereich anderer Lösungen.

6.3 Zusätzliche Überlegungen

In Bezug auf die Verwendung von *Dublin Core* wäre zu überlegen, ob nicht eher das vom Projekt *eDavid* in Belgien entwickelte ISAD(G)-Schema verwendet werden sollte (http://www.edavid.be/xmlschemas/isad.xsd). Dieses kann zwar nicht als Standard be-

zeichnet werden, doch hat sich im Verlauf der Arbeiten gezeigt, dass Bedarf nach einer XML-Umsetzung von ISAD(G) besteht; ein Bedarf, den keiner der bestehenden Standards (EAD, DC) wirklich befriedigend erfüllen kann.

Der Einsatz von METS einerseits als Containerformat und andererseits zur Abbildung des Ordnungssystems kann kritisch betrachtet werden, insofern ein Standard unterschiedliche Funktionen übernimmt bei der Bildung des AIP. Dies senkt die Flexibilität der Lösung in Hinblick auf den Ersatz der eingesetzten Standards.

Bei dem Kriterium, aus dem AIP wieder ein SIP resp. eine universelle GEVER-Sicht zu generieren, das nicht erfüllt werden kann, stellt sich ohnehin die Frage, wie dies in Anbetracht der vorhandenen Standards überhaupt machbar ist und welche Konsequenzen aus dieser Schwierigkeit gezogen werden sollen. So könnte eine Folgerung sein, dass das AIP doch der GEVER-Struktur folgt. Eine andere Folgerung wäre denkbar, nämlich auf die Forderung nach einer solchen Umwandlung zu verzichten.

6.4 Fazit

Die vorgestellte Lösung für die Bildung von AIPs lässt sich realisieren und arbeitet mit akzeptierten Standards. Dass METS zugleich als Containerformat eingesetzt wird, verringert die Zahl der eingesetzten Standards, jedoch auch die Flexibilität der Lösung.

Metadaten und Primärdaten sind ebenso klar getrennt wie eindeutig miteinander referenziert, wie auch innerhalb der Metadaten beschreibende, technische und strukturierende deutlich voneinander geschieden sind.

Die Kriterien, die sich mit der Frage der Metadaten-Interoperabilität befassen (SIP/GEVER – AIP – Findmittel), werden jedoch nicht immer zufriedenstellend erfüllt.

Insgesamt resultiert eine übersichtliche AIP-Struktur, die zudem grundlegenden Anforderungen genügt. Der Bereich der beschreibenden Metadaten ist mit *Dublin Core* aus archivischer Sicht jedoch nicht vollkommen befriedigend abgedeckt.

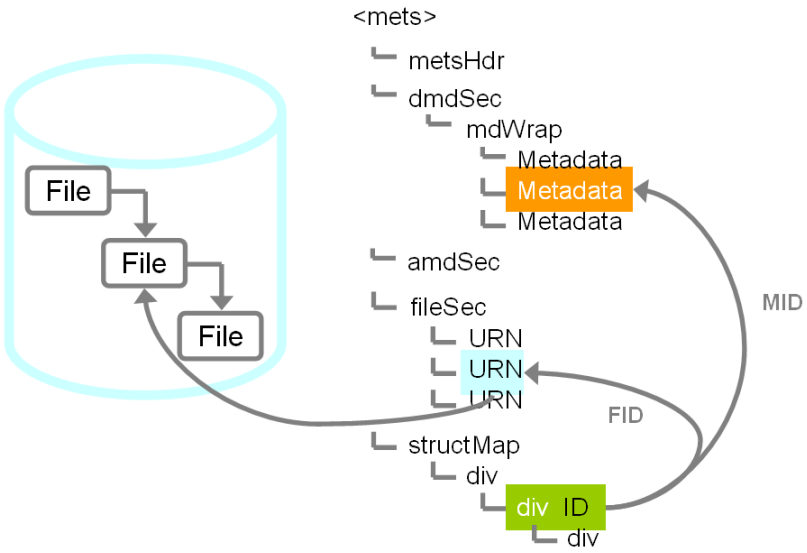
7 METS-DC

7.1 Kurzbeschreibung des AIP-Aufbaus

Es handelt sich hier um die Analyse einer AIP-Lösung mit einem METS-Schema für Struktur und technische Metadaten sowie deskriptive Metadaten in *Dublin Core*, ohne PREMIS oder LMER. Diese Lösung entspricht in etwa der DIAS-kopal-Lösung, bzw. dem *Universal Object Format* (UOF), wie es von Tobias Steinke vorgeschlagen wurde. Es wird jedoch bewusst auf eine *amdSec* mit PREMIS- oder LMER-Metadaten verzichtet.

Die Struktur der Ablage, das Ordnungssystem, wird in der METS *structMap* abgebildet, dort kann durch rekursive *div*-Elemente eine beliebige hierarchische Struktur dargestellt werden.

Metadaten zum Ordnungssystem bzw. zu Verzeichnungseinheiten werden in der *dmdSec* jeweils in einem *Dublin Core*-Wrapper abgelegt. DC stellt mit den 15 Standard-DC-Elementen ein entsprechendes Metadatenschema zur Verfügung. Für jede Verzeichnungseinheit bzw. jedes digitale Objekt wird ein *dmdSec*-Eintrag aus der METS *structMap* referenziert. Die *structMap* selber enthält konsequenterweise keine Metadaten.



<METS> cross reference mit ID, IDREF, IDREFS und URN

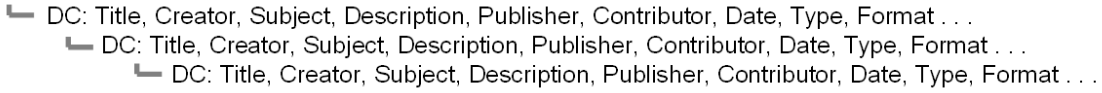
Technische Metadaten werde alleine in der METS *fileSec* im Element *file*[6] festgehalten METS stellt dafür die folgenden Attribute zur Verfügung: MIMETYPE, SEQ, SIZE, CREATED, CHECKSUM, CHECKSUMTYPE.

[6] Siehe METS Strukturdiagramm: http://sunsite3.berkeley.edu/mets/diagram/.

7.2 Analyse gemäss Kriterien

7.2.1 Extraktion der Metadaten für ein ISAD(G)-konformes Findmittel

Sämtliche für ein ISAD(G)-konformes Findmittel notwendigen Daten sind pro Verzeichniseinheit in der METS *dmdSec* abgelegt. Die *structMap* bildet die Hierarchie der Verzeichnungseinheiten ab. Schemasprache für die Metadaten ist *Dublin Core*.



Hierarchisch strukturierte Dublin Core-Metadaten

7.2.2 Schemakontrolle der notwendigen ISAD(G)-Metadaten

Dublin Core-Metadatenschema und ISAD(G) lassen sich mit einigem Aufwand ineinander überführen[7], damit sind die ISAD(G)-Metadaten schemakontrolliert speicherbar. Es muss dies insbesondere für die aus archivischer Sicht zentralen Elemente Signatur, Titel, Entstehungszeitraum/Laufzeit, Verzeichnungsstufe und Provenienz gelten. Schwierigkeiten bereitet dabei die Verzeichnungsstufe, eine Entität die in DC nicht vorgesehen ist. Wir können hier aber auf das Attribut */ead/structMap/div@LABEL*[8] in der METS *structMap* zurückgreifen:

ISAD(G)	**Dublin Core**	**METS**
Titel	Title	
Provenienz	*Creator*	
	Subject	
	Description	
Provenienz	*Publisher*	
	Contributor	
Entstehungszeitraum/Laufzeit,	Date	
	Type	
	Format	
Signatur	Identifier	
	Source	
	Language	
	Relation	
	Coverage	
	Rights	
Verzeichnungsstufe		*//div@LABEL*

7.2.3 Generierung einer universellen GEVER-Sicht (DIP)

GEVER-spezifische Metadaten, welche keine Entsprechung im Dublin Core Modell finden, können allenfalls unter *DC:description* abgelegt werden. Weder METS noch DC stellen einen Mechanismus zu Speicherung von nicht vorgängig spezifizierten *Key-*

[7] Mapping between metadata formats http://www.ukoln.ac.uk/metadata/interoperability/
[8] Attribut LABEL http://www.loc.gov/standards/mets/docs/mets.v1-5.html#div

Value-Paaren zur Verfügung. Damit lässt sich eine GEVER-Sicht zwangsläufig nur in Form einer eingeschränkten *Dublin Core*-Sicht darstellen.

7.2.4 Abbildung von Formatmigrationen

Der beschränkte Umfang an technischen Metadaten in METS erlaubt es dennoch, Formatmigration und Dateiversionen abzubilden. In *structMap* können mehrere *file* Elemente in einer *fileGrp* zusammengefasst werden. In *fileGrp* kann mit den Attributen VERSDATE und USE die Abfolge der Migration minimal beschrieben werden.

7.2.5 Enge Verbindung von Ordnungssystem, Metadaten und digitalen Objekten

Sämtliche Metadaten sind in der METS-Datei gespeichert, einzig auf Dateien wird extern (im Container) verwiesen. Im Prinzip wäre es möglich, auch die Dateien in der XML-Datei als CDATA[9] *base64-encoded* zu speichern.

7.2.6 Vermeidung unnötiger Redundanz

Die Darstellung ist äusserst kompakt, Redundanz wird vollständig vermieden, technische Metadaten sind auf das absolut Notwendige beschränkt.

7.2.7 Verwendung von möglichst wenigen Standards

Es werden mit METS und DC vor allem in der Bibliothekswelt auch international akzeptierte Metadatenstandards verwendet.

7.3 Zusätzliche Überlegungen

Angenommen, wir wählen eine externe Verzeichnisstruktur für die AIP-Primärdateien, dann ist es denkbar, in einer zweiten METS-*structMap* die Dateien des AIP und ihren Speicherort zu beschreiben *(Asset Map)*. Damit könnte auf einen zusätzlichen Container verzichtet werden.

7.4 Zusammenfassung

Die Lösung METS-DC zeigt eine einfache Metadatenstruktur. Es besteht eine klare Trennung von archivischen Metadaten, administrativen-technischen Daten und Strukturdaten in den METS-Sections *dmdSec*, *fileSec* und *structMap*. Die METS-Datei kann gleichzeitig auch die Funktion eines AIP-Containers übernehmen. Die Lösung ist damit in jeder Beziehung einfach und kompakt.

Durch den Verzicht auf die METS *amdSec*[10] ist der administrativ/technische Teil auf ein Minimum beschränkt. So kann eine Formatmigration nur rudimentär im AIP abgebildet werden. Es besteht zudem keine Möglichkeit, den Formattyp genauer denn als MIME-Type anzugeben.

Nicht *Dublin Core*-kompatible Metadaten können ohne Verletzen der klaren Struktur nicht festgehalten werden.

---

[9] CDATA – XML (Unparsed) Character Data

[10] Die METS *amdSec* hat ohne den Einsatz weiterer Metadatenschemata (PREMIS, LMER etc.) keine Einsatzmöglichkeiten.