

## Formaterkennung und Formatvalidierung: Theorie und Praxis

Claire Röthlisberger-Jourdan, Koordinationsstelle für die dauerhafte Archivierung elektronischer Unterlagen (KOST), Bern

### Inhaltsverzeichnis

1	Grundlagen.....	2
1.1	Die KOST .....	2
1.2	Der Katalog archivischer Dateiformate (KaD) .....	2
1.2.1	Einleitung .....	2
1.2.2	Formatkategorien .....	2
1.2.3	Analyse und Bewertung .....	2
1.2.4	Umsetzung im einzelnen Archiv.....	3
2	Einführung in Formaterkennung und Formatvalidierung .....	4
2.1	Verifizierung der akzeptierten Formate .....	4
2.2	Unterschied zwischen Formaterkennung und -validierung .....	4
2.2.1	Grundlagen .....	4
2.2.2	Einführendes Beispiel .....	4
2.3	Nutzen und Notwendigkeit der Formaterkennung.....	6
3	Formaterkennung .....	7
3.1	Anforderungen an die Formaterkennung .....	7
3.1.1	Batchtauglichkeit.....	7
3.1.2	Inbetriebnahme .....	7
3.1.3	Qualität und Geschwindigkeit.....	7
3.1.4	Format-Datenbank .....	7
3.2	Die vier bekanntesten Formaterkennungstools.....	7
3.2.1	Grobbewertung der Formaterkennungssoftware .....	8
3.2.2	Analyse der verwendeten Formatdatenbanken .....	9
3.2.3	Detailbewertung von Fido 1.0.0 .....	11
3.2.4	Detailbewertung von DROID 6.0.1 .....	12
3.3	Fazit zu Formaterkennungssoftware .....	12
4	Formatvalidierung.....	13
4.1	Anforderungen an die Formatvalidierung.....	13
4.2	Validierungssoftware .....	13
4.2.1	PDF/A.....	13
4.2.2	TIFF.....	15
4.2.3	JPEG2000.....	15
4.2.4	WAV .....	15
4.2.5	SIARD .....	15
4.3	Fazit zu Formatvalidierungssoftware.....	15

## **1 Grundlagen**

### **1.1 Die KOST**

Die Koordinationsstelle für die dauerhafte Archivierung elektronischer Unterlagen (KOST) ist ein Gemeinschaftsunternehmen des Schweizerischen Bundesarchivs, des Landesarchivs des Fürstentums Liechtensteins, 24 kantonalen und 5 kommunalen Archiven der Schweiz. Sie hat den Auftrag, ihre Träger bei der Archivierung elektronischer Unterlagen zu unterstützen. Dazu erarbeitet sie unter anderem Standards und Richtlinien als Grundlage, stellt Tools und Dienstleistungen zur Lösung konkreter Probleme und Arbeitsschritte zur Verfügung und dokumentiert in Studien und Kolloquien den Wissensstand zu einzelnen Themen. Zudem vermittelt die KOST das zusammenfließende Wissen den beteiligten Archiven in verschiedenen Veranstaltungen<sup>1</sup>.

### **1.2 Der Katalog archivischer Dateiformate (KaD)**

#### **1.2.1 Einleitung**

Der Katalog archivischer Dateiformate (KaD) ist eines der ersten Produkte der KOST. Er wurde in seiner ersten Version 2007 erarbeitet, in Antwort auf ein häufig geäußertes Desiderat vieler KOST-Trägerarchive. Dabei verfolgt er zwei Ziele: Erstens zeigt er auf, welche Formate nach heutigem Kenntnisstand theoretisch archivtauglich sind und als Zielformate für die Migration oder Konversion dienen können. Zweitens dient er im Kontakt mit der Verwaltung als Referenz dafür, welche Formate aus archivischer Sicht im aktiven Lifecycle verwendet (und entsprechend empfohlen) werden können.

Die aktuelle, zweite Version des KaD datiert aus dem Jahr 2009.

Die dritte Version wird 2012 erstellt und soll im Frühjahr 2013 publiziert werden. Neben einer generellen Aktualisierung wird der Katalog insbesondere in den Bereichen Video-, Text- und Bildformate aufdatiert sein.

#### **1.2.2 Formatkategorien**

Für den Katalog archivischer Dateiformate sind die folgenden abstrakten Formatkategorien relevant: Textdaten, Bilddaten, Audiodaten, Videodaten und strukturierte Daten (Tabellenkalkulation, Datenbanken). Hingegen sind Programmdateien für Archive nicht relevant, da diese keine Software archivieren.

#### **1.2.3 Analyse und Bewertung**

Die im Katalog enthaltenen Formate wurden aus verschiedenen Blickwinkeln oder Sichten analysiert (Abb.1):

1. Eine Bewertung anhand archivfachlicher Kriterien legt offen, in welchem Mass ein Format die Anforderungen von Archiven an die Archivtauglichkeit erfüllt und welche Risiken bei seiner Verwendung zu beachten sind. Dazu wurde ein Katalog von sechs unterschiedlich gewichteten Kriterien erarbeitet.

---

<sup>1</sup> Siehe zu den Grundlagen und Produkten der KOST die Beiträge von Georg Büchler, Martin Kaiser und Christian Eugster in früheren Tagungen des Arbeitskreises.

2. Eine Best-Practice-Analyse hält fest, wie jedes Format in der Archivwelt beurteilt und in den Verwaltungen angewendet wird. Diese Sicht lässt sich in zwei weitere Kriterien übersetzen, Best Practice und Perspektive.
3. Eine Klassifizierung der Formate erlaubt es, unterschiedliche Bewertungen gemäss den ersten beiden Sichten zu verstehen, und trägt zur Entscheidung über eine Empfehlung bei. Es wird unterschieden zwischen altbekannten, weit verbreiteten Formaten, die sich auf Grund ihrer Stabilität für die Archivierung eignen; neuen Formaten, die zweifellos grosse Verbreitung erlangen werden; und potentiellen Formaten, bei deren Design die Archivtauglichkeit eine besondere Rolle gespielt hat, deren Zukunft aber noch nicht absehbar ist.

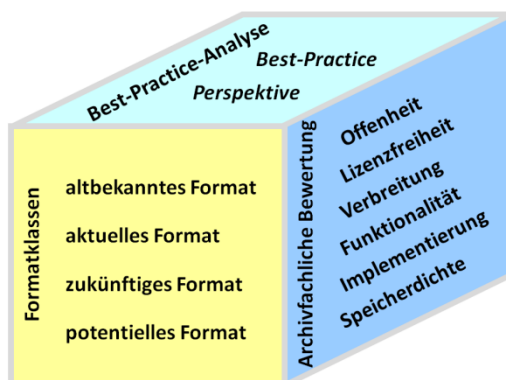


Abb. 1: Kriterien zur Analyse und Bewertung (KaD Version 1 und 2)

Die Bewertung der einzelnen Formate wird zusammengefasst in einer Empfehlung der KOST pro Formatkategorie.

#### 1.2.4 Umsetzung im einzelnen Archiv

Der KaD ist nicht mehr als eine Richtlinie. Der Entscheid, welche Formate für die digitale Archivierung akzeptiert werden, obliegt dem einzelnen Archiv. Dieser Entscheid kann von Archiv zu Archiv unterschiedlich sein, weil die im KaD abgebildeten archivfachlichen Empfehlungen technischen, wirtschaftlichen oder politischen Einflüssen unterliegen (Abb. 2).

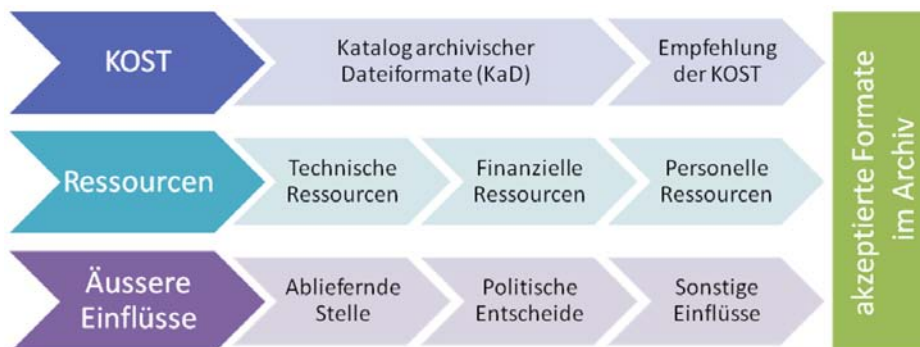


Abb. 2: Umsetzungseinflüsse im Bereich akzeptierter Formate

## 2 Einführung in Formaterkennung und Formatvalidierung

### 2.1 Verifizierung der akzeptierten Formate

In der Praxis reicht es nicht aus, dass ein Archiv definiert hat, welche Formate es für die digitale Archivierung akzeptiert. Die Einhaltung dieses Entscheides muss vor jedem Ingest überprüft werden. Diese Überprüfung vollzieht sich in zwei Schritten: zunächst in einer automatischen Formaterkennung, danach in der Validierung der gesamten Datei.

### 2.2 Unterschied zwischen Formaterkennung und -validierung

#### 2.2.1 Grundlagen

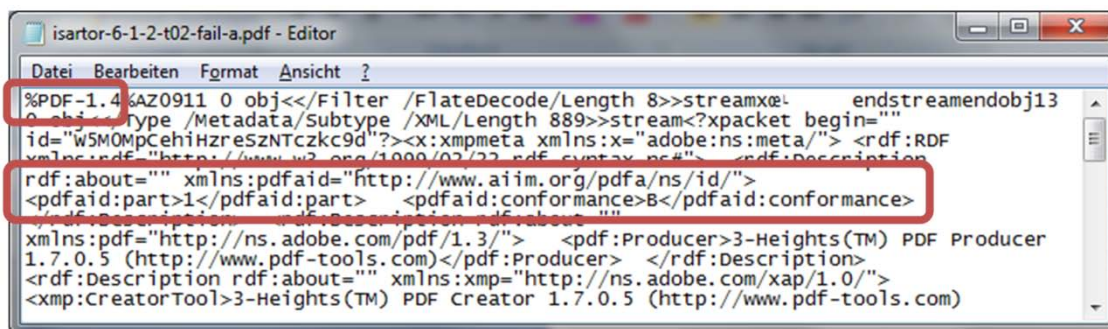
Die Formaterkennung identifiziert das Format einer Datei bis zu einer bestimmten, gewünschten Granularität. Sie stützt sich dabei auf das Vorhandensein besonders charakteristischer Eigenschaften. In der Regel sind dies bestimmte Bytesequenzen innerhalb der Datei.

Die Formatvalidierung überprüft, ob eine Datei der Spezifikation ihres Formats entspricht. Dabei muss jede einzelne der in der Formatspezifikation verlangten Eigenschaften überprüft werden. Nur wenn alle Eigenschaften erfüllt sind, ist die Datei valide.

Wenn für ein Format kein entsprechender Validator existiert, ist einzig eine näherungsweise Validierung möglich durch das Öffnen der Dateien in einem Viewer. Da Viewer in der Regel aber eine gewisse Fehlertoleranz haben, gibt allein die Tatsache, dass eine Datei ohne Fehlermeldung angezeigt werden kann, noch keine Gewissheit über ihre Validität.

#### 2.2.2 Einführendes Beispiel

Eine Datei im Format PDF/A-1b enthält spezielle, charakteristische Bytesequenzen (Abb. 3). Sie beginnt mit den Zeichen %PDF-1 und enthält an einer beliebigen Stelle den RDF-Metadateneintrag, der sie als PDF/A, Version 1, Konformanzlevel B ausweist. Diese Bytesequenzen sind ersichtlich, wenn eine PDF/A-Datei mit einem normalen Texteditor geöffnet wird:



```
isartor-6-1-2-t02-fail-a.pdf - Editor
Datei Bearbeiten Format Ansicht ?
%PDF-1.4 %AZ0911 0 obj<</Filter /FlateDecode/Length 8>>streamxœ...endstreamobj13
0 obj<<type /Metadata/Subtype /XML/Length 889>>stream<?xpacket begin=""
id="w5M0MpcEhiHzreSzNTczkc9d"?><x:xmpmeta xmlns:x="adobe:ns:meta/"> <rdf:RDF
xmlns:pdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"> <rdf:Description
rdf:about="" xmlns:pdfaid="http://www.aiim.org/pdfa/ns/id/">
<pdfaid:part>1</pdfaid:part> <pdfaid:conformance>B</pdfaid:conformance>
</rdf:Description> </rdf:Description> </rdf:about=""
xmlns:pdf="http://ns.adobe.com/pdf/1.3/"> <pdf:Producer>3-Heights(TM) PDF Producer
1.7.0.5 (http://www.pdf-tools.com)</pdf:Producer> </rdf:Description>
<rdf:Description rdf:about="" xmlns:xmp="http://ns.adobe.com/xap/1.0/">
<xmp:CreatorTool>3-Heights(TM) PDF Creator 1.7.0.5 (http://www.pdf-tools.com)
```

Abb. 3: Charakteristische Bytesequenzen von PDF/A-1b

Wenn diese Elemente fehlen, erkennt eine Formaterkennungs-Software eine Datei nicht als PDF/A-1b, auch wenn die Dateiendung .pdf ist. Umgekehrt erkennt sie eine Textdatei fälschlicherweise als PDF/A-1 (Abb. 4), wenn diese Bytesequenzen vorhanden sind. (Erst beim Versuch, eine solche Datei mit einem PDF-Viewer zu öffnen (Abb. 5), oder bei der PDF/A-Validierung wird der Fehler offenbar.)

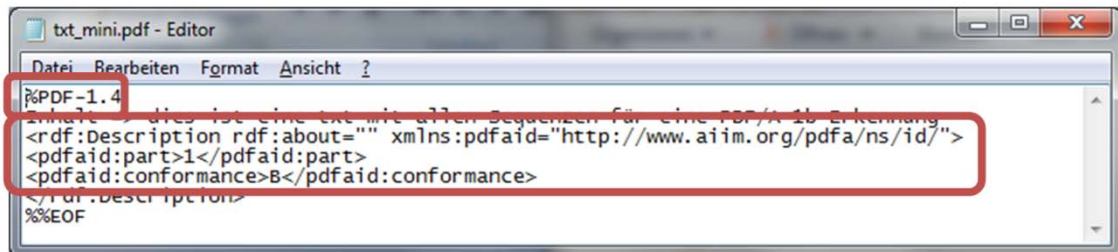


Abb. 4: Textdatei mit enthaltener PDF/A-Sequenz

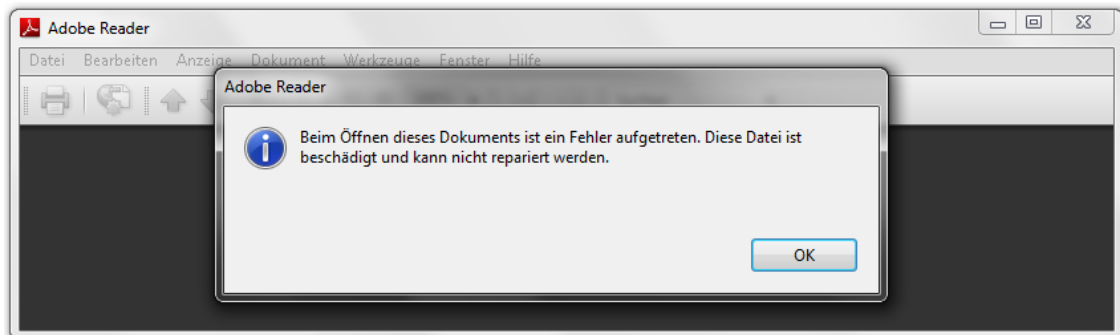


Abb. 5: Viewer-Fehlermeldung bei korrupten PDF-Dateien

Dass das Öffnen einer PDF-Datei mit einem PDF-Viewer nicht ausreicht, zeigt das folgende Beispiel: Es handelt sich hier um eine PDF-Datei, welche die erwähnten Bytesequenzen enthält, die aber mit einem Passwort geschützt ist. Dies verletzt die Spezifikation von PDF/A. Beim Öffnen im Adobe Reader wird die Datei, gestützt einzig auf die integrierte Formaterkennung, fälschlicherweise als PDF/A identifiziert (Abb. 6).

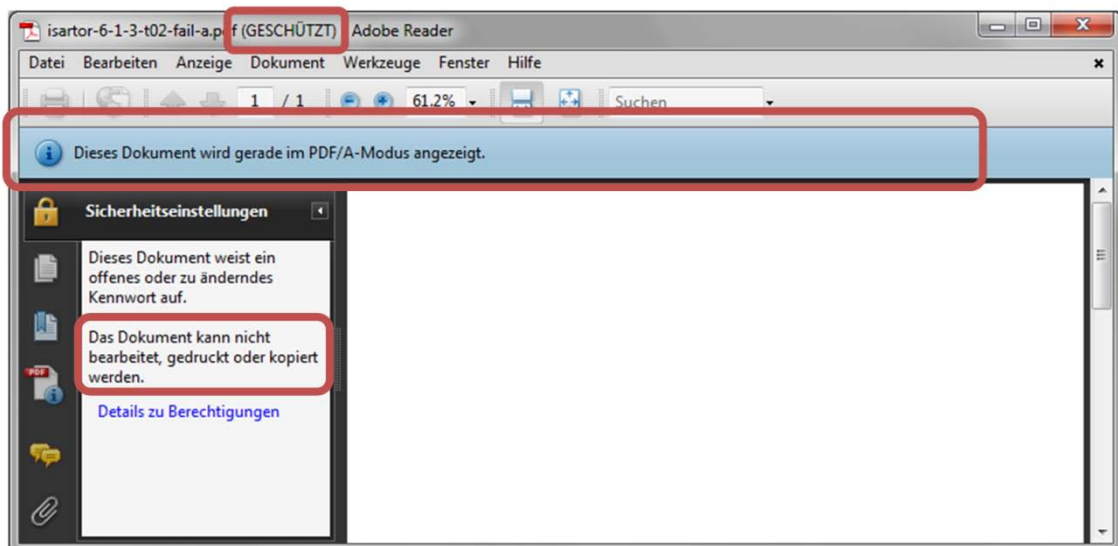


Abb. 6: Grenzen des PDF-Viewers

Die reine Formaterkennung geht hier also fehl. Einzig ein PDF/A-Validator kontrolliert alle Eigenschaften und gewährleistet, dass es sich um ein valides PDF/A-1b handelt, das auch in Zukunft korrekt vermittelt respektive konvertiert werden kann (Abb. 7).

**PDF/A Validation Results (Before Conversion):**

Run DateTime: 2012-Apr-19 15:14:49

Fail:

Error ID	Message	Obj Refs
e_PDFa14	Contains compressed object streams	9, 10, 12, 13, 15, 16, 40
e_PDFa15	Contains cross-reference streams	
e_PDFa341	The font is not embedded	82
e_PDFa361	Widths in embedded font are inconsistent with /Widths entry in the font dictionary	23, 24, 65, 73, 75, 78, 80, 82
e_PDFa723	The XMP Metadata stream is not valid	14
e_PDFa2331	Device-specific color space used, but no GTS_PDFa1 OutputIntent	1, 37

Pass:

Generated using [PDFTron PDF/A Manager V1.02.](#)

Abb. 7: Validierungsergebnisse zweier PDF-Dateien

### 2.3 Nutzen und Notwendigkeit der Formaterkennung

Angesichts der aufgezeigten Grenzen und der Unschärfe der Formaterkennung, und weil immer eine anschliessende Validierung nötig ist, stellt sich die Frage, weshalb es überhaupt eine Formaterkennung braucht.

Ihre Hauptrolle ist die einer Triage. Spätestens wenn die Menge der zu kontrollierenden Daten gross ist, sind die Archive auf maschinelle Unterstützung angewiesen. Anhand der Formaterkennung kann dann automatisch der korrekte Nachfolgeprozess angesprochen werden. In der nachfolgenden Grafik wird zum Beispiel anhand der Formaterkennung die Datei Test.pdf entweder zur PDF/A-Validierung, zur Konvertierung PDF zu PDF/A oder, als weitere Möglichkeit, zur Konvertierung DOC zu PDF/A weitergegeben und verarbeitet (Abb. 8).

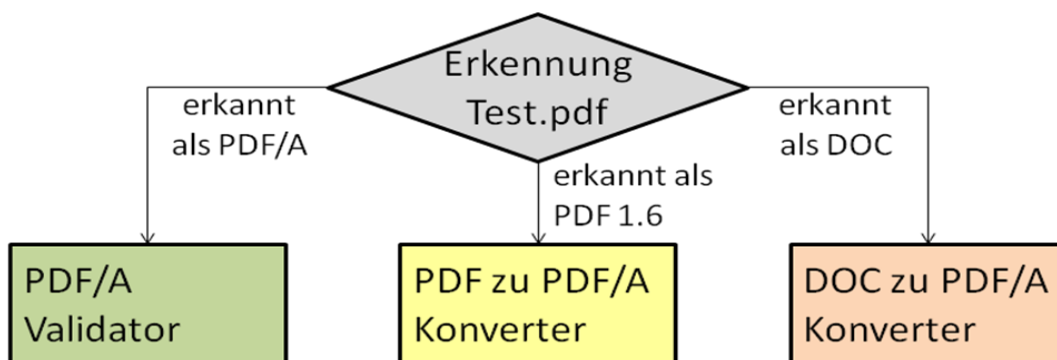


Abb. 8: Formaterkennung im Bereich der automatischen Triage

## **3 Formaterkennung**

### **3.1 Anforderungen an die Formaterkennung**

Damit ein Tool zur Formaterkennung in den Standardprozessen der Archive verwendet werden kann, sollte es die folgenden Anforderungen erfüllen:

- Batchtauglichkeit
- Installation ohne Administratorenrechte
- Schnelle und qualitative Erkennung
- Öffentlich zugängliche sowie editierbare Liste mit den Merkmalen der einzelnen Formate

#### **3.1.1 Batchtauglichkeit**

Die Batchtauglichkeit ist die Grundvoraussetzung, damit die Formaterkennung auch wirklich in den Standardprozess eingebunden werden kann. Zudem müssen die Resultate in einer definierten Form ausgegeben werden, damit der Nachfolgeprozess korrekt angestossen werden kann.

#### **3.1.2 Inbetriebnahme**

Wenn immer möglich soll die Software ohne Administratorenrechte installiert werden können oder eine Installation überhaupt nicht notwendig sein. Dies ist vor allem in der Testphase eine wesentliche Erleichterung.

#### **3.1.3 Qualität und Geschwindigkeit**

Grundsätzlich ist die Qualität höher einzustufen als die Geschwindigkeit. Die Tests, welche die KOST durchgeführt hat, basierten immer auf der Einstellung für die höchst mögliche Qualität mit entsprechendem negativen Einfluss auf die Geschwindigkeit.

Bei der Qualität ist auch die gewünschte Granularität wesentlich. Aus der Sicht der KOST sollte die Granularität genügend fein sein, um zum Beispiel mindestens eine normale PDF-Datei (z.B. PDF-1.4 oder PDF-1.7) von einer PDF/A-Datei (z.B. PDF/A-1b oder PDF/A-2u) unterscheiden zu können. Die Granularität ist auch abhängig von der verwendeten Format-Datenbank.

#### **3.1.4 Format-Datenbank**

Die Merkmale der einzelnen Formate sollten auf einer öffentlich zugänglichen sowie editierbaren Liste beziehungsweise Datenbank basieren. Weil die Anzahl der potenziell zu identifizierenden Formate und Formatversionen sehr hoch und die Identifikation deshalb relativ komplex ist, ist es unabdingbar, dass diese Datenbank von möglichst vielen Akteuren benützt und gepflegt wird. Die in der Archivwelt am weitesten verbreitete Datenbank, welche diese Kriterien erfüllt, ist die PRONOM-Datenbank des Englischen Nationalarchives (TNA).

### **3.2 Die vier bekanntesten Formaterkennungstools**

Im Sinn einer Marktübersicht werden in der Folge die vier bekanntesten Formaterkennungssoftwares für die digitale Archivierung kurz beschrieben:

- **DROID** DROID (Digital Record Object Identification) ist ein Software-Tool entwickelt durch TNA, um die automatisierte Identifikation von Dateiformaten anhand ihrer PRONOM-Datenbank durchzuführen zu können. DROID ist eine plattformunabhängige Java-Anwendung und kann sowohl von einer grafischen Benutzerschnittstelle (Graphical User Interface, GUI) als auch von einer Konsole respektive einer auf Zeichen basierenden Benutzerschnittstelle (Command Line Interface, CLI) aufgerufen werden. DROID ist kostenlos und steht unter der BSD-Lizenz für DROID v4.0. Es kann von <http://sourceforge.net/projects/droid/> heruntergeladen werden.
- **File** File ist ein UNIX-Programm zur Formaterkennung einer Datei. Die *magic numbers*, welche für die Erkennung verwendet werden, sind in einer ASCII-Textdatei namens *Magic* angegeben. File ist kostenlos und kann unter anderem von <ftp://ftp.astron.com/pub/file/> heruntergeladen werden.
- **Fido** Fido (Format Identification for Digital Objects) ist ein CLI-Tool für Windows und Linux zur Identifikation von Dateiformaten. Es wurde von der Open Planets Foundation (OPF) zur einfachen Integration in automatisierte Prozesse entwickelt. Fido verwendet die PRONOM Signaturen, übersetzt sie jedoch in reguläre Ausdrücke und wendet diese direkt an. Fido ist frei erhältlich unter der Apache 2.0 Lizenz und kann von <http://github.com/openplanets/fido/downloads> heruntergeladen werden.
- **Tika** Apache Tika™ Toolkit erkennt und extrahiert Metadaten und strukturierte Text-Inhalte aus verschiedenen Dokumenten in einer der vorhandenen Parser-Bibliotheken. Für die Formaterkennung wird der "Mime Magic Detection" Teil verwendet. Tika ist ein Projekt der Apache Software Foundation unter der Apache License, Version 2.0. Die neueste Tika Version kann von <http://tika.apache.org/download.html> kostenlos heruntergeladen werden.

### 3.2.1 Grobbewertung der Formaterkennungssoftware

Asger Blekinge hat drei der erwähnten Tools für das Scape-Projekt getestet. Die Testresultate sind auf dem Blog der Open Planets Foundation publiziert<sup>2</sup>. Sie werden hier kurz zusammengefasst und ergänzt (Tab. 1). Dabei muss jedoch beachtet werden, dass die gewünschte Granularität und die verwendeten Einstellungen die Resultate stark beeinflussen.

---

<sup>2</sup> Asger Blekinge, Identification tools, an evaluation  
<http://www.openplanetsfoundation.org/blogs/2012-02-23-identification-tools-evaluation>



Name	Stärke	Neutral	Schwäche
DROID	Inbetriebnahme PRONOM-DB Qualität		Batchtauglichkeit Geschwindigkeit
File	Batchtauglichkeit Inbetriebnahme Geschwindigkeit		Qualität Andere DB
Fido	Batchtauglichkeit PRONOM-DB Qualität	Inbetriebnahme	Geschwindigkeit
Tika	Batchtauglichkeit Inbetriebnahme Geschwindigkeit		Qualität Andere DB

Tab. 1: Grobbewertung der Formaterkennungssoftware

### 3.2.2 Analyse der verwendeten Formatdatenbanken

Die vier Formaterkennungstools verwenden drei verschiedene Quellen für die Erkennung einzelner Dateien. Damit sich jeder ein besseres Bild machen kann, werden die einzelnen Quellen anhand des Eintrags für PDF 1.4 erläutert. In den Fällen, in welchen diese Granularität nicht gegeben ist, wurde der PDF-Eintrag verwendet.

#### PRONOM-Datenbank

Die PRONOM-Datenbank ist von TNA erstellt worden und wird laufend durch die weltweiten Benutzer und Softwarehersteller aktualisiert und ergänzt. Die PRONOM-Datenbank wird sowohl von DROID als auch von Fido verwendet. Die Datenbank enthält rund 860 Formateinträge.

Nachfolgend (Abb. 9) ein Auszug aus dem Eintrag von PDF 1.4:

<b>Name</b>	Acrobat PDF 1.4 - Portable Document Format			
<b>Version</b>	1.4			
<b>Other names</b>	PDF (1.4)			
<b>Identifiers</b>	MIME: application/pdf Apple Uniform Type Identifier: com.adobe.pdf PUID: fmt/18			
<b>External signatures</b>	File extension: pdf			
<b>Internal signatures</b>	<b>Name</b>	PDF 1.4		
	<b>Description</b>	Header and footer		
	<b>Byte sequences</b>	<b>Position type</b>	Absolute from BOF	
		<b>Offset</b>	0	
	<b>Byte order</b>			
	<b>Value</b>	255044462D312E34		
	<b>Position type</b>	Absolute from EOF		
	<b>Offset</b>	0		
<b>Byte order</b>				
<b>Value</b>	2525454F(46 460A 460D 460D0A 460D00)			

Abb. 9: PDF-1.4-Eintrag aus der PRONOM-Datenbank

PRONOM gibt nebst dem verbreiteten MIME-Type (application/pdf) auch einen eigenen Identifikator heraus, den PRONOM Unique Identifier (PUID, im Beispiel fmt/18) heraus, welcher die entsprechende Granularität aufweist.

Unter Byte sequences steht, dass das PDF 1.4 mit "%PDF-1.4" (Hexadezimale Schreibweise 255044462D312E34) beginnen muss und mit "%EOF" oder "%EOF." oder "%EOF.." (Hexadezimale Schreibweise 2525454F46, 2525454F460A, 2525454F460D, 2525454F460D0A oder 2525454F460D00) endet.

### Magic-Datensammlung

In der File-Applikation befindet sich auch die Magic-Datensammlung. Diese enthält rund 230 Dateien, welche teilweise mehrere Formatidentifikationen enthalten, jedoch nur einen einzelnen allgemeinen Eintrag zu PDF.

Nachfolgend (Abb. 10) der Eintrag zu PDF:

```

#-----
# $File: pdf,v 1.6 2009/09/19 16:28:11 christos Exp $
# pdf: file(1) magic for Portable Document Format
#
0  string      %PDF-      PDF document
!:mime application/pdf
>5 byte       x          \b, version %c
>7 byte       x          \b.%c

```

Abb. 10: PDF-Eintrag aus der File Magic-Datensammlung

In der Identifikationszeile (0 string %PDF- PDF document) steht, dass ein PDF mit dem Text "%PDF-" beginnen muss und dass bei einer erfolgreichen Erkennung "PDF document" ausgegeben werden soll. File gibt auch den MIME-Type (application/pdf) aus.

## Tika-mimetypes.xml

In Tika `org.apache.tika.detect.MagicDetector` wird Tika-mimetypes.xml zur Verfügung gestellt. Tika-mimetypes.xml enthält rund 170 Dateneinträge, wovon nur ein einzelner allgemeiner Eintrag zu PDF definiert ist.

Nachfolgend (Abb. 11) der Eintrag zu PDF:

```
<mime-type type="application/pdf">
  <acronym>PDF</acronym>
  <comment>Portable Document Format</comment>
  <magic priority="50">
    | <match value="%PDF-" type="string" offset="0" />
  </magic>
  <glob pattern="*.pdf" />
  <alias type="application/x-pdf" />
</mime-type>
```

Abb. 11: PDF-Eintrag aus Tika-mimetypes.xml

Die Tika-Einträge bauen auf den MIME-Type Einträgen (z.B. `application/pdf`) auf.

In der Identifikationszeile (`<match value="%PDF-" type="string" offset="0" />`) steht, dass ein PDF mit dem Text `"%PDF-"` beginnen muss.

## Fazit

Die Qualität der Formaterkennung durch File und Tika ist ungenügend, weil die Granularität nicht fein genug ist. Entsprechend wurden diese beiden Programme keiner Detailbewertung unterzogen. DROID und Fido hingegen wurden detailliert getestet im Hinblick auf ihre Qualität und Geschwindigkeit (als untergeordnetes Kriterium). Für den Test wurde ein Sample von 664 Dateien verwendet sowie (zur besseren Vergleichbarkeit der beiden Tools) einheitlich das `DROID_SignatureFile_V55`.

### 3.2.3 Detailbewertung von Fido 1.0.0

#### Qualität und Geschwindigkeit

Fido benötigte für die Erkennung mit einer theoretischen Buffergrösse von 10GB weniger als 6 Minuten. Fido wendet die PRONOM-Datenbank korrekt an.

Wird die Standard-Buffergrösse von 128KB verwendet, benötigt Fido zwar weniger als 20 Sekunden, erkennt aber 44 Dateien nicht korrekt, weil nicht alle Bytes der grossen Dateien eingelesen wurden und sich jeweils mindestens eine wesentliche Bytesequenz erst am Schluss der Datei befunden hätte. Im Test waren insbesondere PDF/A- und SIARD-Dateien davon betroffen. Entsprechend ist eine solche Reduzierung der Buffergrösse für einen hohen Zeitgewinn in Anbetracht des Qualitätsverlustes nicht zu empfehlen.

Fido bietet im Gegensatz zu DROID jedoch die Möglichkeit, die Anzahl der für die Formaterkennung verwendeten PUIDs einzugrenzen, sei dies durch das Ausschliessen von gewissen PUIDs oder durch das Definieren zu verwendender PUIDs. In einem zweiten Test wurden einzig 10 PUIDs für die Erkennung ausgewählt, wiederum bei einer theoretischen Buffergrösse von 10GB. Dabei benötigte Fido weniger als 40 Sekunden für die Erkennung. Bei den erkannten Formaten wurden durch Fido 43 PDF-Dateien anhand der Dateiendung fälschlicherweise als PDF/A erkannt. Abgesehen von dieser Kinderkrankheit von Fido wurden alle gewünschten PUIDs richtig erkannt. Sobald dieser Fehler behoben ist, ist diese Form von Eingrenzung sehr zu empfehlen, wenn man grundsätzlich mit einer kleinen Anzahl archivtauglicher Formate operiert.

Der kleine Rest von dabei allenfalls nicht erkannten Dateien kann dann in einem zweiten Durchgang ohne Einschränkung der PUIDs durch Fido identifiziert werden.

### **Batchtauglichkeit**

Fido ist ein Konsolenprogramm und erfüllt die Batchtauglichkeit zu 100%. Alle Einstellungen bis hin zum Format der Ausgabe werden über den Erkennungsbefehl Fido übergeben.

### **Inbetriebnahme**

Für die Installation von Fido benötigt man grundsätzlich keine Administratorenrechte. Da Fido jedoch ein Python-Programm ist, muss Python auf dem Rechner vorhanden sein. Es besteht die Möglichkeit, Python 2.7 als Portable-Applikation zu verwenden.

## **3.2.4 Detailbewertung von DROID 6.0.1**

### **Qualität und Geschwindigkeit**

DROID benötigt für die Erkennung mit einer theoretischen uneingeschränkten Buffergröße zirka 1 Minute. Das Resultat der Erkennung entspricht den erwarteten Resultaten anhand der PRONOM-Datenbank und ist identisch mit jenem von Fido.

Wird die Buffergröße von 128KB verwendet, benötigt DROID nur noch 5 Sekunden, erkennt jedoch die gleichen 44 Dateien wie Fido nicht korrekt. Entsprechend ist eine solche Reduzierung der Buffergröße für einen hohen Zeitgewinn in Anbetracht des Qualitätsverlustes nicht zu empfehlen.

### **Batchtauglichkeit**

DROID überzeugt insbesondere bei der Anwendung des GUI. Im Gegensatz dazu ist die Konsolenversion sehr unbefriedigend; es erwies sich im Rahmen der Tests insbesondere als unmöglich, einen brauchbaren Output zu generieren. Da DROID ein quell-offenes Java-Programm ist, wäre es theoretisch möglich, DROID direkt in eine Java-Applikation einzubinden. Leider konnte die KOST DROID Version 6 als Java-API (Application Programming Interface) nicht zum Laufen bringen. Eine Einbindung der Version 5 war zwar möglich, dabei wurden jedoch sehr grosse Dateien trotz uneingeschränkter Buffergröße teilweise nicht korrekt erkannt.

### **Inbetriebnahme**

Für die Installation von DROID werden keine Administratorenrechte benötigt.

## **3.3 Fazit zu Formaterkennungssoftware**

Ihr Umfang und ihre Granularität sprechen für die Verwendung der PRONOM-Datenbank für die Formaterkennung. Es ist für die Archive empfehlenswerter, sich am Ausbau und an der Verbesserung von PRONOM zu beteiligen, als eigene Formaterkennungs-Datenbanken und -Werkzeuge aufzubauen<sup>3</sup>. Als Softwaretools kommen

---

<sup>3</sup> In diesem Sinn ist Steffen Bachmann und Katharina Ernst zu widersprechen, welche eine gewichtete Formaterkennung in einem Modulrahmen postulieren („Formaterkennung – Ziele, Herausforderungen, Lösungsansätze“, in: Matthias Manke (Hg.), Auf dem Weg zum digitalen Archiv. Stand und Perspektiven von Projekten zur Archivierung digitaler Unterlagen. 15. Tagung des Arbeitskreises „Archivierung von Unterlagen aus digitalen Systemen“ am 2. und 3. März 2011 in Schwerin. Veröffentlichungen des Landeshauptarchivs Schwerin, 2012). Die von ihnen geschilderten Probleme bei der Formaterkennung beruhen auf Eigenheiten oder Fehlern von PRONOM, beispielsweise im Verzicht auf eine Erkennung der einzelnen TIFF-Versionen und

folglich DROID oder Fido in Frage. Beide sind im Gegensatz zu Tika und File deutlich langsamer, was aber grösstenteils auf die Granularität der Erkennung respektive die Grösse von PRONOM zurückzuführen ist.

Die Stärken und Schwächen von DROID und Fido werden aus der Detailbewertung deutlich. Wer die Formaterkennung nur ab und zu einsetzen will, ist sicherlich mit dem GUI von DROID bestens bedient. Für den Einsatz in einem maschinellen Prozess empfiehlt sich die Verwendung von Fido, weil hier keine Abstriche bei der Batchtauglichkeit gemacht werden müssen. Die Geschwindigkeit sollte bei der Bewertung eine untergeordnete Rolle spielen. Für Fido spricht weiterhin, dass das Programm aktuell mindestens durch 3 Personen aus verschiedenen Archiven entwickelt wird und sein Quellcode viel übersichtlicher ist, währenddem die kontinuierliche Weiterentwicklung von DROID durch diverse Personalwechsel bei TNA erschwert wurde.

## 4 Formatvalidierung

### 4.1 Anforderungen an die Formatvalidierung

Damit ein Tool zur Formatvalidierung in den Standardprozessen der Archive verwendet werden kann, sollte es folgende Anforderungen erfüllen:

- Batchtauglichkeit
- Installation ohne Administratorenrechte
- Schnelle und qualitative Validierung
- Nachvollziehbarkeit / Überprüfung der Validierung

Die Anforderungen an die Formatvalidierung sind beinahe identisch mit jenen an die Formaterkennung. Bei der Qualität ist nicht nur massgebend, ob die Datei valid oder invalid ist, sondern auch, dass die Fehlermeldung dem effektiven Fehler entspricht.

Die Nachvollziehbarkeit der Validierung muss gegeben sein. Ist dies nicht möglich, weil zum Beispiel die Software nicht quelloffen ist, muss die Validierung mit Hilfe von invaliden Test-Dateien für alle Anforderungen einzeln überprüft werden.

### 4.2 Validierungssoftware

Aufgrund der Anforderungen an die Validierung wird in der Regel ein spezialisiertes Tool pro Format benötigt.

#### 4.2.1 PDF/A

Die Validierung von PDF/A ist sehr komplex und nicht einfach realisierbar. Weil zudem die Nachfrage sehr gross ist, existiert ein Markt für PDF/A-Validatoren und sind quelloffene Tools nicht erhältlich. Die Überprüfung der Qualität der Validatoren wurde durch die KOST durchgeführt und basiert auf den Grundlagen von PDFlib. Die Resultate sind

---

ihrer Ersetzung durch einen generischen TIFF-Eintrag (PUID fmt/353). Für Fehler in PRONOM sollen wenn möglich Korrekturvorschläge erarbeitet und eingereicht werden, um die Qualität der Datenbank zu erhöhen. Selbstverständlich ist zu beachten, dass immer die aktuelle PRONOM-Version respektive das neueste *Signature File* von DROID verwendet wird.

in einer Studie zu den PDF/A-Konvertern zusammengefasst und auf der KOST-Website<sup>4</sup> veröffentlicht.

Nachfolgend sind die 4 bekanntesten Validatoren<sup>5</sup> in einer Produktübersicht aus den Jahren 2009-2010 zusammengefasst (Tab. 2):

PDF/A Validatoren	Adobe: Adobe Acrobat 9.1	Intarsys: PDF/A Live	PDF Tools: 3Heights PDF Validator Shell	PDFTron: PDF/A Manager
<b>Geschwindigkeit &amp; Robustheit:</b> Sehr gut = <1 Minute & ohne Absturz Gut = 1 - 5 Minuten & ohne Absturz Ausreichend = >5 Minuten & ohne Absturz Mangelhaft = Absturz	Ausreichend	Gut	Sehr gut	Sehr gut
<b>Genauigkeit:</b> <sup>6</sup> Sehr gut = Mittelwert >=95% Gut = Mittelwert 90% - 94% Ausreichend = Mittelwert 75 - 89% Mangelhaft = Mittelwert <75%	Gut	Sehr gut	Gut	Sehr gut
<b>Isartor testsuite</b> (non-conforming)				
6.1 File structure 31x	97%	90%	100%	90%
6.2 Graphics 47x	100%	100%	100%	100%
6.3 Fonts 28x	100%	96%	100%	100%
6.4 Transparency 6x	100%	100%	100%	100%
6.5 Annotations 25x	96%	100%	100%	100%
6.6 Actions 37x	100%	100%	100%	100%
6.7 Metadata 27x	100%	100%	100%	96%
6.9 Interactive Forms 3x	100%	100%	100%	100%
<b>Other non-conforming</b>				
ISO 19005 violations 9x	89%	89%	89%	100%
XMP 2004 violations 5x	20%	60%	20%	80%
PDF 1.4 violations 8x	38%	100%	63%	75%
<b>Conforming</b>				
Real world 34x	88%	97%	85%	100%
PDFIib samples 8x	100%	100%	88%	88%
Advanced XMP 16x	100%	100%	100%	100%
<b>Mittelwert</b> (conforming / non-conforming)	<b>91%</b>	<b>97%</b>	<b>90%</b>	<b>95%</b>
<b>Getestete Version:</b>	9.1.0	5.0.4	1.8.32.1	1.00 (CLI)
<b>Tester / Testjahr:</b>	PDFIib / 2009	KOST / 2010	PDFIib / 2009	KOST / 2010
<b>Bemerkungen:</b>	Die Version 9.0 ist bei der Geschwindigkeit und Robustheit mangelhaft.	Test mit einer neueren Version nochmals komplett durchgeführt.		Report ist sehr übersichtlich (Kompakt mit guter Fehlermeldung).

Tab. 2: Auszug der Produktübersicht aus der KOST-Studie

Bei der Validierung von PDF/A muss beachtet werden, dass PDF/A zwar als ISO-Standard genormt ist, jedoch im Standard nur aufgelistet wird, welche einzelnen Funktionen der zugrunde liegenden PDF-Version obligatorisch, empfohlen, eingeschränkt oder verboten sind. Diese werden vereinzelt in den Details unterschiedlich interpretiert. Zudem sind alle Dokumente, die den PDF/A Standard zusammen definieren, sehr umfangreich und sehr technisch. Dies führt dazu dass in einigen Fällen unterschiedliche Validierungsergebnisse existieren.

<sup>4</sup> Studie der KOST zu der Zusammenstellung und Bewertung verschiedener Programme zur Validierung von PDF/A-Dateien [http://kost-ceco.ch/cms/index.php?pdfa\\_validatoren\\_de](http://kost-ceco.ch/cms/index.php?pdfa_validatoren_de)

<sup>5</sup> In der Studie der KOST wurde ebenfalls der pdfaPilot von Callas untersucht. Da dieses Produkt in Adobe Acrobat unter dem Namen Preflight Validator enthalten ist, und da sich die Ergebnisse dieser beiden Validatoren nicht wesentlich unterscheiden, wurde für eine bessere Lesbarkeit pdfaPilot von Callas in der Übersicht weggelassen. Die komplette Produktübersicht ist in der Studie ersichtlich.

<sup>6</sup> Bei der Genauigkeit ist nicht nur das Ergebnis (bestanden / durchgefallen) wesentlich, zusätzlich muss die Fehlermeldung mindestens einen reellen Fehler beschreiben.

#### **4.2.2 TIFF**

Für die Validierung von TIFF-Dateien ist der KOST einzig der JHOVE-Validator bekannt. JHOVE (JSTOR/Harvard Object Validation Environment) ist quelloffen und gratis verfügbar. Eine unabhängige Überprüfung der Validierungsqualität anhand des Quellcodes ist der KOST nicht bekannt.

#### **4.2.3 JPEG2000**

Für die Validierung von TIFF-Dateien ist der KOST nebst dem JHOVE-Validator auch der jpylyzer von OPF bekannt. Beide Validatoren sind quelloffen und gratis verfügbar. Unabhängige Überprüfungen der Validierungsqualität anhand der Quellcodes sind der KOST nicht bekannt.

#### **4.2.4 WAV**

Für die Validierung von WAV-Dateien ist der KOST einzig der JHOVE-Validator bekannt. JHOVE ist quelloffen und gratis verfügbar. Eine unabhängige Überprüfung der Validierungsqualität anhand des Quellcodes ist der KOST nicht bekannt.

#### **4.2.5 SIARD**

Für die Validierung von SIARD-Dateien existierte bis 2012 kein Validator. Deshalb entwickelt die KOST gegenwärtig die Anwendung SIARD-Val (Validator für SIARD-Dateien), die unter der GPL3-Lizenz und gratis zur Verfügung gestellt wird. Geplant ist der Beta-Release für den Herbst 2012. Die Qualität der Validierung muss entsprechend anschliessend durch eine andere Institution anhand des Quellcodes überprüft werden.

### **4.3 Fazit zu Formatvalidierungssoftware**

Nicht nur die Formatvalidierung selber, sondern auch die Qualitätssicherung der Validierungstools ist sehr aufwändig. Für viele Validatoren existieren deshalb noch keine unabhängigen Überprüfungen der Qualität ihrer Resultate. Diese Lücken sollten durch die Archivgemeinschaft geschlossen werden. Die KOST plant deshalb, ihre Arbeit an und ihre Untersuchungen von Validierungstools fortzusetzen. Welche Formate als nächste in den Fokus genommen werden, ist noch offen.