Nutzen und Grenzen der Formaterkennung

(Zu-)Fälle bei PRONOM und DROID

Stephanie Kortyla, Christian Treu

Ziel jeder Archivierung ist Nutzbarmachung. Im digitalen Bereich sind Objekte nur mit Hilfsmitteln, genauer mit einer technischen Darstellungsumgebung wahrnehmbar. Digitale Objekte werden in Repräsentationen abgelegt, von denen jede aus mindestens einer Datei besteht.[1] Folglich liegt pro Objekt mindestens ein Dateiformat vor. Doch um welches handelt es sich explizit? Die Dateinamenserweiterung, auch Extension genannt, kann trügen, unbekannt sein oder auch fehlen. Eine Identifizierung des Formats, als Erstbestimmung oder Verifizierung einer Vermutung, ist unabdingbar. Mit diesem Anhaltspunkt können Daten lesbar und nutzbar gemacht werden. Nicht nur für den aktuellen, sondern auch für einen zukünftigen Gebrauch von Objekten, z.B. im Zuge von Bestandserhaltung wie Formatmigration, sind passende Werkzeuge auszuwählen. Formaterkennung bildet nur einen ersten Schritt im Lebenszyklus von Unterlagen, die für eine Langzeitspeicherung und/ oder dauernde Archivierung vorgesehen sind.[2]

Das Sächsische Staatsarchiv betreibt seit 2013 sein Elektronisches Staatsarchiv (el_sta) und setzt sich fortlaufend mit Formaterkennung auseinander. Im Folgenden wird auf die in der Praxis eingesetzten Werkzeuge mit ihrem Nutzen und ihren Grenzen näher eingegangen.[3] Beim el_sta werden das technische Register

1 Vgl. PREMIS Editorial Committee (Hg.): PREMIS Data Dictonary for Preservation Metadata.Version 3.0., 2015, S. 8. http://www.loc.gov/standards/premis/v3/premis-3-0-final.pdf. (Sämtliche Weblinks wurden am 19.02.2018 zuletzt aufgerufen.)

2 Im hiesigen Beitrag soll es dagegen nicht um Bewertung von «archivtauglichen» Formaten gehen.

3 Mit den aufgekommenen Fragen wandte sich das Sächsische Staatsarchiv im Sommer 2015 in der Community zunächst an die nestor-AG Formaterkennung und daraufhin an Jay Gattuso von der Neuseeländischen Nationalbibliothek, siehe unten. Die nestor-AG hat auf ihrer Wiki-Seite einige der hier geschilderten Gegebenheiten ebenfalls aufgenommen. Vgl. Tunnat, Yvonne: PRONOM. Persistenz von PUIDs. Wiki-Unterseite der nestor AG-Formaterkennung, 2016. https://wiki.dnb.de/display/NESTOR/PRONOM%3A+Persistenz+von+PUIDs. Nutzererfahrungen mit PRONOM und DROID sind bisher marginal veröffentlicht worden: http://coptr.digipres.org/DROID_(Digital_Record_Object_Identification). Vgl. auch Gattuso, Jay: Throughput efficiencies and misidentification risks in DROID, 2012. http://ndha-wiki.natlib.govt.nz/assets/NDHA/Reading/MSB+DROID+v1-05.pdf. Ders.: Evaluating the historical persistence of DROID asserted PUIDs, 2012. Auf dieser Website sind weitere Testberichte zu finden. Vgl. auch Jackson, Andy: Formats over Time: Exploring UK Web History. In: iPres2012. Proceedings of the 9th International Conference on Preservation of Digital Objects, 2012, S. 155-158. https://ipres-conference.org/ipres12/sites/ipres.ischool.utoronto.ca/files/iPres%202012%20Conference%20Proceedings%20Final.pdf. Vgl. auch Tarrant, David; Carr, Les: LDS³: Applying Digital Preservation Principals to Linked Data Systems. In: iPres2012. Proceedings of the 9th International Conference on Preservation of Digital Objects, 2012, S. 77-84, hier S. 83. https://ipres-conference.org/ipres12/sites/ipres.ischool.utoronto.ca/files/

Informationswissenschaft: Theorie, Methode und Praxis, Bd. 5 (2018), Nr. 1 — http://dx.doi.org/10.18755/iw.2018.9

bzw. die Formatdatenbank PRONOM, die Schnittstelle zum eigentlichen Werkzeug bzw. die sogenannte Signature-File sowie das Erkennungswerkzeug DROID eingesetzt. Hierbei handelt es sich um eines der gängigsten Verfahren zur Formatidentifizierung.[4]

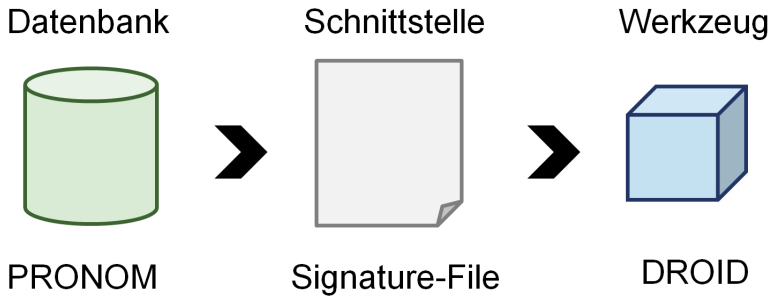

Abbildung 1: Zur Formaterkennung eingesetzte Komponenten, hier PRONOM, PRONOM-Signature-File, DROID. Eigene Darstellung

Formaterkennung – wofür?

Durch Erkennung und Analyse der vorliegenden Formate lässt sich eine Übersicht erstellen, die dem Preservation Planning als Grundlage dienen kann. Zudem lassen sich durch die Formaterkennung bereits vor dem Ingest etwaige Unstimmigkeiten und Fehler, wie falsche Endungen und unter Umständen beschädigte Dateien, erkennen und beheben. Unerwünschte System- und Vorschaudateien (z. B. Thumbs.db) lassen sich ebenso automatisiert erkennen und filtern. Die bei der Ana-

iPres%202012%20Conference%20Proceedings%20Final.pdf. Vgl. auch Töwe, Matthias; Geisser, Franziska; Suri, Roland E.: To Act or Not to Act – Handling File Format Identification Issues in Practice. Poster in: iPRES2016. 13th International Conference on Digital Preservation. Proceedings, 2016, S. 288f. https://ipr16.organizers-congress.org/frontend/organizers/media/iPRES2016/_PDF/IPR16.Proceedings_4_Web_Broschuere_Link.pdf.

4 Einen Überblick über weitere Formaterkennungswerkzeuge bietet z.B. folgende Website: http://coptr.digipres.org/Category:File_Format_Identification. «The 'big 3' file format identification tools, DROID, Tika and File (…)», in: Wheatley, Paul; Pennock, Maureen: Supporting practical preservation work and making it sustainable with SPRUCE. In: iPres2013. Proceedings of the 10th International Conference on Preservation of Digital Objects, 2013, S. 73-77, hier S. 74. Bewertungen von Erkennungswerkzeugen sind zu finden bei: Knijff, Johan van der; Wilson, Carl: Evaluation of Characterisation Tools. Part 1: Identification. SCAPE Project, 2011. http://scape-project.eu/wp-content/uploads/2014/08/SCAPE_PC_WP1_identification21092011.pdf sowie bei Röthlisberger-Jourdan, Claire; Koordinationsstelle für die dauerhafte Archivierung elektronischer Unterlagen (KOST): Formaterkennung und Formatvalidierung: Theorie und Praxis, 2012. https://kost-ceco.ch/cms/index.php?format_validation_de.

lyse generierten technischen Metadaten können in die Metadaten des AIP aufgenommen werden.

Nicht zuletzt bietet die Formaterkennung auch die Möglichkeit des Risikomanagements. Um spätere Kosten für spezielle Viewer und lizensierte Fachanwendungen einzudämmen oder wenigstens deren Bedarf kritisch beurteilen zu können, bietet sich die Begrenzung und Kontrolle der eingehenden Formate an. Probleme, die sich zum Beispiel aus obsoleten oder proprietären Formaten ergeben, können besser eingeschätzt werden. In aggregierter Form können die technischen Metadaten der bereits erfolgten Ingests in einer Statistik zusammengefasst werden.

Erkennung vs. Validierung

Hauptziel der Formaterkennung ist die Identifizierung eines bestimmten Dateiformates. Die Validierung überprüft hingegen die Regelhaftigkeit bzw. die Normkonformität einer Datei[5] und grenzt dabei gegebenenfalls Formatvarianten voneinander ab. Dies geht weit über die bloße Erkennung hinaus, erfordert aber spezielle Tools. Ein Datei kann nutzbar und dennoch nicht valide sein, wenn sich z. B. eine Datei als PDF/A zu erkennen gibt, aber nur dem PDF-1.4-Standard entspricht.[6] In der Praxis existieren bisher Validatoren für nur wenige Formate.

Formaterkennung – wie ?

Um Dateiformate zu identifizieren, gibt es verschiedene Ansätze. Drei Methoden sollen hier beispielhaft dargestellt werden, wobei die zuletzt genannte den größten Nutzwert bietet. Für viele selbstverständlich, aber bei weitem nicht ausreichend, ist die Formaterkennung anhand der Dateiendung oder Extension. Die Vergabe von Dateiendungen auf verschiedenen Betriebssystemen ist weder obligatorisch noch einheitlich. Ein Bild im JPEG Interchange Format (JIF) kann die Endungen .jpeg, .jpe, .jpg oder schlicht keine Endung tragen. Endet eine Datei auf .pdf, kann dies repräsentativ für eine von 37 PDF-Varianten stehen (nach aktuellen Stand der Signature-File V90). Vergleichbares ergibt sich auch bei der Bestimmung des MIME Types. Nicht immer lässt sich der MIME Type ermitteln und selbst wenn, ist dieser nicht sehr trennscharf. Die Standards HTML 4.01, XHTML 1.0 und HTML5 können sich alle den gleichen MIME Type, hier «text/html», teilen. Für eine genauere Unterscheidung kann eine Validierung vorgenommen werden.

5 Vgl. Röthlisberger-Jourdan, Formaterkennung und Formatvalidierung (siehe Anm. 4), S. 4.
6 PDF/A-1 basiert auf PDF 1.4.

Die zuverlässigste und eindeutigste Formaterkennung erfolgt durch die Analyse und Identifizierung charakteristischer Bytesequenzen,[7] so genannter *Signatures*.[8] Dabei wird versucht, bestimmte Bitfolgen, auch *Magic Numbers* genannt, zu erkennen und einem Format zuzuordnen. Diese Muster können am Anfang, am Ende oder einer anderen Stelle im Bitstrom auftauchen. Bisweilen enthalten die Signatures versteckte Botschaften in ASCII oder in hexadezimaler Kodierung, welche zum Beispiel in einem Hex-Editor lesbar gemacht werden können. So sind alle zip-basierten Dateiformate mit der Magic Number «pk», für den Entwickler Phil Katz, versehen. Die Signature kann aber auch vergleichsweise offensichtlich, wie «%PDF-1.4» für PDF 1.4 oder «DOCTYPE HTML[…]» für HTML sein.

Nutzen der Formaterkennung

PRONOM

Um die Signaturen und weiterführende Informationen zu verschiedenen Formaten zusammenzutragen, bedarf es einer Datenbank. Die bisher verbreitetste Datenbank dieser Art ist die 2002 durch das britische Nationalarchiv (TNA) begonnene PRONOM-Datenbank.[9] Kernaspekt von PRONOM ist die Bereitstellung und Pflege von eindeutigen Identifiern für Formate (Persistent Unified IDentifier) in einer kostenfreien, webbasierten Datenbank. Ergänzungs- und Verbesserungsvorschläge durch die Community werden regelmäßig durch TNA eingepflegt.[10] Aus der Datenbank wird wiederum mehrmals im Jahr eine DROID Signature-File generiert, deren Hauptzweck es ist, die Grundlage zur automatisierten Formaterkennung zu stellen. Als XML-Datei bildet sie die Schnittstelle zwischen Datenbank und Erkennungstool.[11]

DROID

Das Tool DROID (Digital Record Object IDentification) wurde ebenfalls von TNA entwickelt und der Community im Jahre 2005 zum ersten Mal zur Verfügung gestellt. 2017 ist die Version v6.3 veröffentlicht worden.[12] Die Java-Anwendung kann

7 Vgl. Röthlisberger-Jourdan, Formaterkennung und Formatvalidierung (siehe Anm. 4).

8 Unter unixoiden Betriebssystemen wird ein solcher Ansatz mit dem Tool «file» schon länger verfolgt.

9 http://www.nationalarchives.gov.uk/aboutapps/pronom/default.htm. Andere «Projekte» wie GDFR bzw. UDFR konnten sich nicht etablieren. Vgl. http://www.udfr.org/.

10 Inwieweit PRONOM zukunftssicher, etwa finanziell gesehen, bzw. zweckmäßig ausreichend ist, soll im Rahmen dieses Beitrags nicht erörtert werden.

11 Eine Übersicht über bisherige Signature-Files ist zu finden unter http://www.nationalarchives.gov.uk/aboutapps/pronom/droid-signature-files.htm.

12 Eine Bedienungsanleitung findet sich hier: The National Archives (TNA): Droid User Guide, 2017. http://www.nationalarchives.gov.uk/documents/information-management/droid-user-guide.pdf.

plattformunabhängig über Kommandozeilenanweisung oder per GUI (Desktop-Version) eingesetzt werden. Sie kann in Prozess-/ Systemlandschaften integriert werden. DROID ist weltweit verbreitet und frei verfügbar.

DROID differenziert Typen, das heißt, es unterscheidet zwischen Dateien, Verzeichnissen und «Archiv»-Dateien (z.B. zip, tar). Das Werkzeug ermöglicht Stapelverarbeitung und führt die Analyse verzeichnisübergreifend durch. Die Struktur der zu analysierenden Einheit ist für das Werkzeug somit irrelevant. Die dabei angewandten Methoden zur Erkennung sind Signature-, Extension- und Containeridentifizierung, und nur wenn keine dateiinterne Signature gefunden wird, folgt eine Erkennung über die Extension. Voraussetzung ist hier, dass überhaupt eine existiert, andernfalls bleibt das Format unerkannt. Für eine Erkennung über Signatures ist die Existenz einer Extension unerheblich. Über ein Scanprofil können Einstellungen für eine Analyse vorgenommen werden. Ergebnis ist eine Informationssammlung über die gescannten Objekte, die z.B. im GUI tabellarisch ausgegeben wird. Optional können die Ergebnisse in verschiedenen Varianten exportiert werden, so dass beispielsweise die zu Objekten erstellten Metadaten in nachfolgenden Prozessschritten weiterverarbeitet werden können. DROID ermöglicht so Angaben zu Extension (auch Anzeige, wenn nicht vorhanden), PUID (PRONOM), Formatname und -version, Mime-Type, Erkennungsmethode und -status (erkannt | nicht erkannt | Ambiguität), optional Hashwert. Gelegentlich treten Performanceprobleme auf (Scanprofileinstellung, Absturz). Aufgrund dessen sind in der Community vereinzelt bereits Überlegungen angestellt, die komplexen Signature-Files zu reduzieren.[13]

Grenzen der Formaterkennung

DROID

Nachfolgend wird auf Gegebenheiten und sich daraus ergebende Konsequenzen im Umgang mit DROID und PRONOM eingegangen. Gemäß dem Prinzip, dass zuerst eine Erkennung über Signatures angestoßen wird und im negativen Fall anschließend eine Erkennung über Extensions, muss berücksichtigt werden, dass Formate zufällig über Muster verfügen können, welche als Signature eines anderen Formates

13 Reduktion des Umfangs der von der einsetzenden Institution akzeptierten Formate bei Hoppenheit, Martin: Minimizing the DROID signature file, 2017. http://hoppenheit.info/blog/2017/minimizing-the-droid-signature-file/. Auf diese Weise konkret umgesetzt für die Praxis ist die Signature-File bei der KOST für den KOST-Val, s.u.; Syntaxreduktion durch Auslassen von sog. Shift-Bytes bei Spencer, Ross: Hacking the DROID Signature File: Keep It Simple Stupid! 2012. http://exponentialdecay.co.uk/blog/hacking-the-droid-signature-file-keep-it-simple-stupid/.

erkannt werden,[14] so dass die Formaterkennung zwar technisch korrekt verläuft, das Ergebnis praktisch aber eine falsche Formatangabe liefert. So erging es dem Staatsarchiv beispielsweise mit einer Übernahme von Daten in Plain Text mit spezieller Kodierung, hier EBCDIC (fmt/159). Zu erwarten war eine Erkennung über Extension (hier .ebcdic), was in den meisten Fällen zutraf. Im Scanergebnis traten aber auch unerwartete Formate auf, so z.B. das Microsoft Owner File Format (fmt/473). Bei Überprüfung des Scanergebnisses für jene Datei wurde deutlich, dass DROID im ersten Prüfvorgang eine Signature fand, so dass die Formaterkennung für diese Datei nach diesem Prozessschritt beendet war. Einzig der Warnhinweis eines Mismatches, genauer dass die tatsächliche Extension (hier .ebcdic) nicht zum vermeintlich erkannten Format (MS Owner File) gehört, deutete daraufhin, dass hier evtl. ein Erkennungsproblem vorliegen könnte. Wie aber bereits erwähnt, ist die Extension, sofern sie überhaupt vorhanden ist, keine Garantie für Korrektheit. Im hiesigen Fall war die Extension (.ebcdic) allerdings zutreffend, so dass die durch die Formaterkennung automatisch erhobenen Metadaten für den Archivierungsprozess (DROID ist in die Prozesslandschaft im Background des el_sta integriert) nach AIP-Generierung und noch vor Ingest in das Repository manuell anzupassen waren.[15] Bei einem Ingest von knapp 90 AIP mit insgesamt über 1000 Dateien, wobei EBCDIC nur eines von mehreren Formaten war, und aufgrund geringer Erfahrung in Bezug auf das EBCDIC-Format wurde eine Qualitätskontrolle stichprobenhaft durchgeführt.

Um diesen Fehler zu reproduzieren, kann ein simpler Test mit einer Datei eines vermeintlichen Formats durchgeführt werden. Dieser ist auch bei Röthlisberger-Jourdan[16] beschrieben. Eine einfache Plain Text-Datei optional mit .pdf-Extension beinhaltet ausnahmslos die zur Signature-Erkennung notwendigen Angaben (hier: «%PDF-1.4» sowie «%%EOF»). DROID findet im ersten Schritt zur Formatidentifizierung bereits eine Signature, hier für das Format PDF 1.4 (fmt/18) und gibt demzufolge als Ergebnis fmt/18 aus. Dabei ist offensichtlich, dass die PDF-Datei nicht funktionsfähig ist respektive sich nicht über entsprechende Software wie einen PDF-Reader darstellen lässt. Hier wird deutlich, dass die Erkennungssoftware erwartungsgemäß und zuverlässig arbeitet, einem Nutzer jedoch

14 Hierauf weisen z.B. Dunckley und Rankin hin: Dunckley, Matthew; Rankin, Stephen: The Use of File Description Languages for File Format Identification and Validation, 2007, S. 1. https://epubs.stfc.ac.uk/work/50089.

15 Mitcham wirft eine Frage zum «over-ride» auf: «to over-ride file identifications - eg – 'I know this isn't really xxxx format so I'm going to record this fact' (and record this manual intervention in the metadata)». Mitcham, Jenny: File identification… let's talk about the workflow, 2015. http://digital-archiving.blogspot.de/2015/11/file-identification-lets-talk-about.html.

16 Vgl. Röthlisberger-Jourdan, Formaterkennung und Formatvalidierung (siehe Anm. 4), S. 4f.

auch bewusst sein muss, dass Ergebnisse stets stichprobenhaft kritisch geprüft werden sollten.

Ein weiterer Aspekt, der seit Inbetriebnahme des el_sta aufgetreten ist, betrifft unvorhergesehene Scandifferenzen. Diese treten bei Einsatz verschiedener Erkennungswerkzeuge[17] bzw. Datengrundlagen (hier verschiedene Signature-File-Versionen) auf. Bedingt durch die Fortschreibung bzw. Aktualisierung der Signature-Files werden derlei Scanergebnisdifferenzen kontinuierlich zu erwarten sein.[18] Dahingehend stellt sich die Frage, in welchem Ausmaß Differenzen zu akzeptieren wären und ob bei wissentlicher Datengrundlagenänderung (z.B. Veröffentlichung einer neuen Signatur-File mit relevanten Änderungen in Bezug auf eigene Echt-Ingests)[19] eine Re-Identifizierung[20] durchzuführen wäre. Diskutabel wäre hier der Grad der Granularität, beispielsweise im Hinblick auf Dateiformatversionen. So werden PDF-Dokumente zunächst in Hauptgruppen wie PDF, PDF/A, PDF/E, PDF/UA und PDF/X unterschieden. In den Gruppen wiederum gibt es weitere Unterversionen wie PDF 1.0-1.7, PDF/A-1a sowie PDF/A-1b etc. In PRONOM sind aktuell (Stand Signatur-File v90 vom 30.03.2017) 37 verschiedene PDF-Versionen mit PUID aufgenommen. Eine falsche Zuordnung eines Formats (wie oben beschrieben anhand von EBCDIC) wäre gegenüber einer fehlerhaften Version (PDF/A-1a oder PDF 1.4) stärker zu gewichten. Abschließend sei auf die KOST verwiesen, die eine eigene Signature-File (KaD-Signature-File[21]) aus den PRONOM-Signature-Files generiert, in der einige Formatversionen und demzufolge auch mehrere PUID zusammengefasst sind. Diese Signature-File wird im eigens erstellten Validierungswerkzeug KOST-Val eingesetzt.[22]

Neben der sich ändernden Datengrundlage (Signature-Files) können die Scandifferenzen auch mit der Scanprofileinstellung begründet werden. Seit der Version v6.3 bietet DROID die Option, anstelle der gesamten Datei (full scan) lediglich einen abgesteckten, selbstgewählten Abschnitt zu scannen. Bei diesem Scanmodus, dem sog. «Max-Byte-Scan» (MBS), wird sowohl Dateianfang als auch

17 Neben DROID kamen bspw. KOST-Val und Fits zum Einsatz.

18 Gattuso, Jay als Kommentar zu Blogeintrag von Wheatley, Paul: Don't panic!: What we might need format registries for, 2012. http://openpreservation.org/blog/2012/07/05/dont-panic-what-we-might-need-format-registries/. Auch Mitcham greift einige Fragen zu Ergebnisdifferenzen und (Nicht-)Identifikation auf: Mitcham: File identification… (siehe Anm. 15).

19 Zumindest sollte die jeweils eingesetzte Signature-File-Version dokumentiert sein. Im weiteren Sinne auch bei Tarrant und Carr: «[…] facts might include the format identification information (at the time)». Tarrant, Carr, LDS³ (siehe Anm. 3), S. 82.

20 Töwe, Geisser, Suri, To Act or Not to Act (siehe Anm. 3), S. 288f. Spencer ist pro Rerun bei neuer Signature: «Pronom is a continuum», Spencer, Ross: Tweet von @beet-keeper, 2015. https://twitter.com/beet_keeper/status/626890544631812097.

21 Vgl. https://github.com/KOST-CECO/KaD_SignatureFile.

22 https://kost-ceco.ch/cms/index.php?kost_val_de. Hierbei ist zu erwähnen, dass das Werkzeug KOST-Val in erster Linie zur Validierung einiger Formate eingesetzt wird.

-ende auf Signatures hin abgetastet. Signatures finden sich häufig in diesen Abschnitten. Es wird deutlich, dass je kleiner der Wert respektive Abschnitt ist, desto schneller die Erkennung verläuft. Allerdings ist auch eine höhere Fehlerquote zu erwarten. Hierauf wird später anhand eines Beispiels näher eingegangen. Je größer der Scanabschnitt (an Dateianfang und -ende) ist, desto länger dauert der Scanprozess, umso wahrscheinlicher ist jedoch die Erkennungsquote und umso kleiner die zu erwartende Fehlerquote. Zudem muss auch bedacht werden, dass bei kleineren Dateien der MBS nicht vorteilhaft ist, wenn der doppelte MBS-Wert die jeweilige Dateigröße überschreitet, da sich die Scanabschnitte von Dateianfang und -ende mittig überlappen und somit ein Teil doppelt gescannt würde.[23] Der empfohlene MBS-Wert liegt bei 65536 Bytes. Dieser basiert auf umfangreichen Tests mit verschieden großen Dateien und Formaten durch die Neuseeländische Nationalbibliothek und ist auch der Default-Wert für die MBS-Einstellung in Droid.[24]

Wie angedeutet, kann immer eine gewisse Fehlerquote innerhalb der Erkennungsquote bestehen.[25] Als Beispiel in Bezug auf MBS mögen Formatversionen dienen, die auf einer anderen Version basieren, wie bei einigen Fällen in der PDF-Gruppe.[26] So basieren auf PDF 1.4 und PDF 1.7 jeweils weitere PDF-Versionen. Konkret basiert beispielsweise PDF/A-1a auf PDF 1.4. Dies spiegelt sich in der Signature wider So beginnen beide Dateien mit dem Signature-Bestandteil %PDF-1.4 und enden mit %%EOF.[27] Diese Signature-Teile treten dabei an fest definierten Stellen auf. DROID gibt so zuverlässig das Format fmt/18 für PDF 1.4 aus. Bei PDF/A-1a kommt zusätzlich als Distinktionsmerkmal ein Signature-Teil vor, der variabel im Bitstrom auftritt. Das Problem wird hierbei bereits sichtbar: Es kann nicht vorhergesehen werden, an welcher Stelle dieser Teil auftritt, ergo kann kein zuverlässiger MBS-Wert vorab definiert werden. Dementsprechend ist das Ergebnis nicht vorhersehbar. Liegt der variable Signature-Teil der PDF/A-1a-Datei außerhalb des Scanbereichs, wird der Signature-Teil am Dateiende (%%EOF) von DROID aufgegriffen, PDF 1.4 zugeordnet und als Ergebnis ausgegeben.

23 Gattuso, Throughput efficiencies and misidentification risks in DROID (siehe Anm. 3), S. 12. Schaubild zum MBS auf S. 4

24 Vgl. ebd., S. 7. Für umfangreiche Tests s. weitere Quellen von Gattuso.

25 Auch Bachmann u.a. weisen darauf hin. Vgl. Bachmann, Steffen; Ernst, Katharina: Formaterkennung – Ziele, Herausforderungen, Lösungsansätze. In: Manke, Matthias (Hg.): Auf dem Weg zum digitalen Archiv. Stand und Perspektiven von Projekten zur Archivierung digitaler Unterlagen. 12. Tagung des Arbeitskreises «Archivierung von Unterlagen aus digitalen Systemen» am 2. und 3. März 2011 in Schwerin, 2012, S. 69-73, hier S. 72.

26 Auf die Problematik bei der Erkennung von bestimmten PDF-Versionen weisen auch Knijff und McGath hin. Vgl. Knijff, Johan van der: PDF version numbers based on deprecated mechanism #114. Mit Kommentar von McGath, Gary, 2016. https://github.com/digital-preservation/droid/issues/114.

27 %%EOF ist für die Signature-Erkennung von PDF/A-1a irrelevant. Bei einem Full Scan würden sowohl die festen als auch das variable Signature-Teil erfasst. Um hier einem Konflikt vorzubeugen, ist in PRONOM vermerkt, dass PDF/A-1a eine höhere Priorität gegenüber PDF 1.4 genießt.

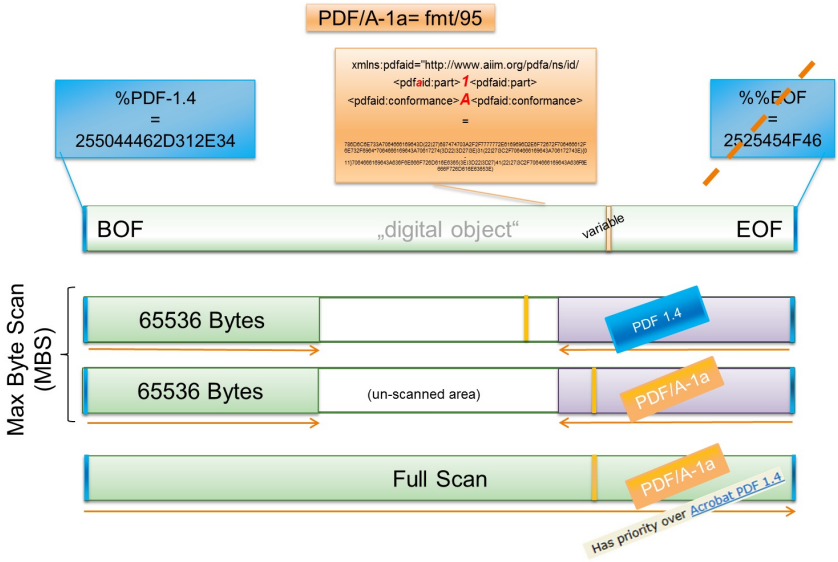

Abbildung 2: Je nach Auftreten des variablen Signature-Bestandteils in der PDF/A-1a-Datei wird jener vom Max-Byte-Scan erfasst oder nicht. Bei Nichterfassung findet DROID den Signature-Bestandteil von PDF 1.4 und gibt dies als Ergebnis aus. Darstellung in Anlehnung an Gattuso, Throughput efficiencies and misidentification risks in DROID, (siehe Anm. XX), S. 4. http://ndha-wiki.natlib.govt.nz/assets/NDHA/Reading/MSB+DROID+v1-05.pdf.

Wie schwerwiegend diese Scandifferenz ist, die sich auf eine Version einer Formatgruppe bezieht, muss jede Institution selbst werten.

PRONOM – Weiterentwicklung der Datengrundlage

Wie bereits angedeutet, wird PRONOM kontinuierlich durch TNA und die Community weiterentwickelt. Angestrebt wird dabei, ca. 100 neue PUIDs pro Jahr einzupflegen. Es kommen also regelmäßig neue Einträge hinzu, jedoch werden ältere gegebenenfalls modifiziert, zusammengelegt oder als überholt («Deprecated») gekennzeichnet. Die Zusammenhänge zwischen PUID, Signature und Extension sind bisweilen relativ komplex. Beispielsweise wird das PNG 1.1 Format unter der PUID «fmt/12» geführt, es existieren dafür in der Signature File die internen IDs 183, 184 und 185.[28] Andere Formate wie Plain Text («x-fmt/111») besitzen hinge-

28 Die Signature File vergibt für die Signatures interne IDs, welche bestimmten Formaten zugeordnet werden. Zum Aufbau der Signature-File, vgl. z.B. Gattuso, Jay (2012): How to write a new signature

gen gar keine Signature. Zudem kann die «generic» Signature mit der ID 78 nicht spezifisch einem Format zugeordnet werden, sondern verweist auf zwei verschiedene Versionen von Excel-Dateien.

Seit 2010 gibt es einen sprunghaften Anstieg bei den Signatures zu verzeichnen, was deren gewachsene Bedeutung für die Formaterkennung unterstreicht. Insgesamt gibt es jedoch mehr Dateiendungen (Extensions) als Formate und mehr Formate als Signatures.[29] Die dynamische Weiterentwicklung von PRONOM ist für die Nutzer nicht nur mit verbesserter Erkennung von Formaten verbunden. Unweigerlich stellt sich auch die Frage nach der Persistenz, denn die Daten, die vor Jahren bei einem Ingest generiert wurden, können nun veraltet sein. Neue Erkenntnisse zu Formaten sind für die Persistenz der Datenbank relativ unproblematisch, da der alte Informationsstand nicht falsch, sondern aktualisiert ist. Zunächst konnten beispielsweise die verschiedenen Microsoft Office Formate nur unter dem Sammelformat OLE2 Compound Document (fmt/111) erkannt werden. Heutzutage ist eine Unterscheidung möglich.[30]

Eine größere Hürde für die Persistenz[31] der erkannten Formate ergibt sich bei den ursprünglich vorläufigen, zusammengelegten oder überholten PRONOM-Einträgen. Da die Community schneller weitere PUID benötigte,[32] als TNA diese in PRONOM einpflegen konnte, bestand die Möglichkeit, vorläufige Einträge zu generieren. Diese sollten durch ein vorangestelltes «x» markiert werden (x-fmt). Da auf etliche x-fmt bereits verwiesen war, stellte sich eine spätere Löschung als problematisch heraus. Aus Stabilitätsgründen werden die vorhandenen x-fmt-Einträge daher weiterhin gepflegt. Zukünftig sollen jedoch keine weiteren vorläufigen PUIDs vergeben und bestehende zum Teil migriert werden.[33] Da sowohl die endgültigen als auch die vorläufigen PUIDs aus demselben Zahlenpool geschöpft haben, ohne in einem inhaltlichen Zusammenhang zu stehen, ergibt sich daraus eine gewisse Ambiguität: Das PDF 1.4 Format hat die PUID fmt/18, das Format Comma Separated Values besitzt die PUID x-fmt/18. Insgesamt haben aktuell ca. 450 PUIDs uneindeutige Nummern, die nur durch das vorangestellte x zu unterscheiden sind.

---

file for DROID. A guide by NLNZ [National Library of New Zealand]. http://openpreservation.org/system/files/how%20to%20write%20a%20sig%20file%20v1.1.pdf.

29 Vgl. Diagramm in Young, Paul: Identifiying digital file formats – a collaborative effort (2016). http://blog.nationalarchives.gov.uk/blog/identifying-digital-file-formats-collaborative-effort/.

30 Vgl. Tunnat, PRONOM (siehe Anm. 3).

31 «File format registries are expected to be persistent, trustworthy, and publicly discoverable.» Barve, Sunita (2007): File Formats in Digital Preservation. In: Proceedings of the International Conference on Digital Libraries, S. 239-248. http://dlissu.pbworks.com/f/File+format1.pdf.

32 Es wurde dadurch möglich, die Metadateien für ein AIP mit einem PUID zu bestücken, bevor diese als endgültige Formate in die PRONOM-Datenbank eingeflossen waren.

33 Vgl. http://www.nationalarchives.gov.uk/aboutapps/pronom/puid.htm.

Aufgrund von Problemen in der Praxis der Formaterkennung wurden PUIDs zum Teil umstrukturiert. Die PUIDs fmt/7 bis fmt/10 waren für das Format TIFF bestimmt. Da aus technischen Erwägungen diese TIFF Varianten nun unter fmt/353 zusammengefasst werden, sind die vorgenannten PUIDs als veraltet gekennzeichnet und verweisen nun aus Stabilitätsgründen auf fmt/353.[34] Mit Stand Anfang 2017 sind 64 PUIDs aus vergleichbaren Gründen zurückgezogen wurden.[35]

Nicht abschließend geklärt ist indes die Frage, wie mit einem anstehenden Ingest verfahren werden soll, wenn noch kein PUID vorhanden ist.[36]

— Abwarten/ Ingest aufschieben.
— Signature bei PRONOM einreichen, damit PUID zugewiesen werden kann.[37]
— Ingest ohne PUID durchführen, nicht zuletzt auch deswegen, da, wie erwähnt, die Datenbasis der Formaterkennung und somit das Ergebnis (z.B. Scandifferenzen) aus verschiedenen Gründen dynamisch ist.

Das el_sta hat bereits eine Handvoll Ingests ohne PUID vorgenommen, so zum Beispiel mit SQLite-Dateien. Zeitgleich wurde der Vorschlag für eine neue Signature von verschiedenen Seiten bei TNA/PRONOM eingereicht. Inzwischen existiert für derartige Dateien ein PUID (fmt/729). Bisher hat das Staatsarchiv aber generell noch keinen Rerun der Formaterkennung durchgeführt.

Fazit

Sowohl die Datengrundlage (= PRONOM, seit 2002) als auch die Schnittstelle (= Signature-File,[38] seit 2005) als auch das Werkzeug (= DROID, seit 2005) sind im Laufe der letzten Jahre weiterentwickelt worden, umfangreicher und mächtiger geworden. Trotz aller Verbesserungen kann zu einem bestimmten Zeitpunkt jedoch immer nur eine gewisse Erkennungsquote erreicht werden. Innerhalb dieser Quote existiert auch immer eine Fehlerquote, wie oben beschrieben wurde.

34 Vgl. Tunnat: PRONOM (siehe Anm. 3).
35 Suche nach «deprecated» in der PRONOM Datenbank. http://www.nationalarchives.gov.uk/PRONOM/BasicSearch/proBasicSearch.aspx?status=new.
36 Frage auch bei Töwe, Geisser, Suri, To Act or Not To Act (siehe Anm. 3) sowie bei Mitcham, File identification… (siehe Anm. 15).
37 Signature-Vorschläge können unter https://www.nationalarchives.gov.uk/contact-us/submit-information-for-pronom/ eingereicht werden.
38 Ebenso werden parallel Container-Signatures gepflegt. Für weitere Informationen siehe http://www.nationalarchives.gov.uk/aboutapps/pronom/droid-signature-files.htm.

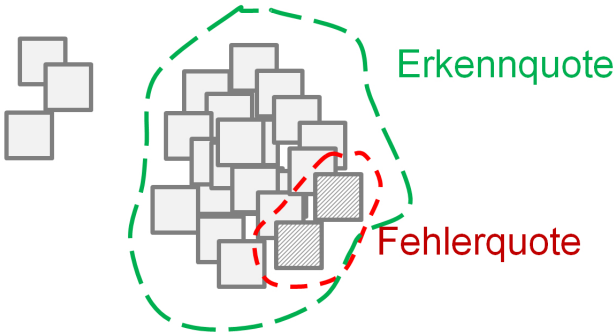

Abbildung 3: Bei der Formaterkennung besteht immer nur eine gewisse Erkennungsquote, innerhalb derer wiederum eine Fehlerquote zu erwarten ist. Eigene Darstellung.

Formaterkennung an sich ist «a work in progress, and therefore could not be considered complete at this time»,[39] sagt Jay Gattuso von der NZLZ im Jahre 2012. Er fügt hinzu: «Perhaps it is only possible to have a relative 'truth'".[40] Oben geschilderte An- und Auffälligkeiten im Umgang mit PRONOM und DROID sollen andeuten, dass bei Formaterkennungswerkzeugen Nutzen, aber auch Grenzen existieren. Formaterkennung bildet im Umfeld von Langzeitspeicherung und digitaler Archivierung nur einen ersten Schritt, jedoch gleichzeitig auch eine wichtige Voraussetzung.[41] Es sollte ein verlässlicher Überblick darüber zu schaffen sein, was im eigenen Repository vorliegt.[42] Dennoch soll auch darauf hingewiesen sein, dass Werkzeuge und Datengrundlagen daher immer nur als temporär und pragmatisch anzusehen sind.

Wie kann eine Institution auf dieser Grundlage handeln? Kurzfristig können andere Werkezuge hinzugezogen bzw. DROID komplett ersetzen werden. Als Alternative böte sich das Programm Siegfried von Richard LeHane an.[43] Dieses Werkzeug arbeitet u.a. auf der Basis von PRONOM-Signature-Files. Langfristig

39 Gattuso, Jay als Kommentar zu Blogeintrag von Wheatley: Don't panic! (siehe Anm. 18).

40 Gattuso, Throughput efficiencies and misidentification risks in DROID (siehe Anm. 3), S. 14.

41 Bachmann; Ernst, Formaterkennung (siehe Anm. 25), S. 69; Spencer, Ross: Generation of a Skeleton Corpus of Digital Objects for the Validation and Evaluation of Format Identification Tools and Signatures. In: The International Journal of Digital Curation 8 (1), 2013, S. 120-130, hier S. 120. http://www.ijdc.net/index.php/ijdc/article/view/8.1.120.

42 Weitergehende Informationen z.B. über Content Profiling bei Petrov, Petar; Becker, Christoph: Large-scale content profiling for preservation analysis, 2012. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.303.6330&rep=rep1&type=pdf.

43 Für weitere Informationen vgl. LeHane, Richard: Siegfried – a PRONOM-based, file format identification tool, 2014. http://openpreservation.org/blog/2014/09/27/siegfried-pronom-based-file-format-identification-tool/.

wäre eine noch stärkere Vernetzung auf Arbeitsebene anzustreben bzw. die schon vorhandenen Plattformen auszubauen. Das nestor Netzwerk bietet mit seinen AGs, Praktikertagen, Workshops und Publikationskanälen vielfältige Optionen.[44] Weitere Einrichtungen wie die KOST oder der LWL (hier Archivberatung zur elektronischen Archivierung) wären wünschenswert. Nicht zuletzt sollten Anwendertreffen und Fachtagungen näher in den Fokus gerückt werden.

44 Siehe Website von nestor. http://www.langzeitarchivierung.de/.