

Table of Contents

1Doel.....	2
2Bestaande software.....	2
1.1Verpakken van metadata en leggen van mappingen.....	2
Delving SIP Creator.....	2
Europeana SIP Creator.....	3
U of Florida - SobekCM Editor.....	3
1.2Verpakken van data.....	3
BagIt.....	3
Library of Congress – TransferTools – BagIt.....	4
UGent – BagKit.....	4
1.3Verpakken van data en metadata.....	5
METS.....	5
Fedora SIP Creator.....	6
Library of Congress – TransferTools – Bagger.....	6
1.4Wijzigingen in de data verifiëerbaar maken.....	6
1.5Batch renamers.....	6
Batchfilerenametool.....	6
Bulk rename utility.....	7
File renamer Basic.....	7
Advanced Renamer.....	7
PFrank.....	7
RenameWand.....	7
1.6Conclusies.....	8
3Functionaliteit in Bagger library of congress.....	9
1.1Beschrijving.....	9
1.2Aandachtspunten.....	12
1.3TODO.....	13

4Verklarende woordenlijst.....	14
--------------------------------	----

1 Doel

- Creatie van een grafische user interface (GUI), om data én metadata samen in een containerformaat (i.e. Exportformaat) te verpakken.
- Het moet mogelijk zijn om een lijst van metadata-records toe te voegen aan deze container.
- wijzigingen in de data moet verifiëerbaar zijn
- Men moet de structuur van de data kunnen duidelijk maken.
- Gebruik van open standaarden, zowel bij het verpakken als bij het formaat van de metadata.
- algemene bruikbaarheid van het containerformaat. Vele containers zijn op maat gemaakt van één soort repository.
- De GUI moet het formaat van de toegevoegd data kunnen herkennen. Op die manier wordt verhinderd dat zogezegde JPEG-bestanden worden opgeslagen, terwijl ze in werkelijkheid TIFF-bestanden zijn.
- Container moet opgeladen kunnen worden naar een online repository.

Het is dus de bedoeling om van een bestaande GUI te vertrekken, en daar de nodige functionaliteit aan toe te voegen. In wat hierna volgt, geven we een overzicht van de bestaande tools en standaarden die ons geholpen hebben in het vormen van een beeld van wat deze GUI moet kunnen, en wat niet.

2 Bestaande software

1.1 Verpakken van metadata en leggen van mappen

Delving SIP Creatorⁱ

Delving SIP Creator is een programma dat als doel heeft digitale pakketten aan te maken die naar het online platform CultureHubⁱⁱ kunnen opgeladen worden. CultureHub aanvaardt records in meerdere XML-formaten, maakt de records beschikbaar via een OAI-PMH-interfaceⁱⁱⁱ (in de geleverde formaten), en voorziet een zoekinterface om de pakketten te ontsluiten. Om de zoekervaring te uniformiseren verplicht CultureHub dat de data ook in het XML-formaat AFF (Another Fine Format) wordt aangeleverd. AFF is eigen XML-formaat dat juist het nodige bevat om te kunnen zoeken in CultureHub.

Daarom moeten de velden van een van de meegeleverde formaten gemapt worden naar AFF. XML kan worden getransformeerd met behulp van XSLT, maar dat vereist de tussenkomst van technici. Voor vele XML-formaten beschikt CultureHub over een mapping. Maar voor zij die liever zelf de

mapping leggen, biedt deze tool een grafische manier om dat te verwezenlijken. De mapping kan opgeslagen worden als profiel, en toegepast worden op een reeks XML-bestanden.

De gemaakte pakketten zijn enkel nuttig voor ingest naar CultureHub.

Het programma is geschreven in Java, en is dus beschikbaar voor elk besturingssysteem die beschikt over Java-ondersteuning. Uitgebracht in Open-Source.

Europeana SIP Creator^{iv}

De Europeana SIP Creator biedt dezelfde functionaliteit als de Delving SIP Creator, maar is bedoeld om pakketten aan te maken om op te laden naar Europeana. Beide programma's gelijken zéér sterk op elkaar.

De gemaakte pakketten zijn enkel nuttig voor ingest naar Europeana.

Eveneens geschreven in Java, en uitgebracht in Open-Source.

U of Florida - SobekCM Editor^v

Programma om op een visuele manier METS-documenten aan te maken, te bekijken, en aan te passen.

Enkel beschikbaar voor Windows.

1.2 Verpakken van data

BagIt^{vi}

BagIt is een bestandsformaat die speciaal ontwikkeld is om volledige bestandsstructuren – i.e. mappen, en onderliggende bestanden – op te slaan onder één noemer, en die bestanden 'veilig' te stellen door het bijhouden van controlesommen of 'checksums' (zie 1.4).

Hiermee lijkt het zeer sterk op bestandsformaten als ZIP en TAR, maar het voegt de mogelijkheid aan toe om wijzigingen in de data te verifiëren. De checksums kunnen onder andere uitgedrukt worden in MD5 of SHA1.

In zijn meest eenvoudige vorm, is een BagIt een map met de volgende structuur:

```
<bagit-directory>
  bagit.txt
  manifest-md5.txt
  bag-info.txt
  data/
    payload1.jpg
    payload2.jpg
```

De bestanden en mappen die men wil bijhouden worden *payloads* genoemd, en worden bewaard in de submap *data*. Van die *payloads* worden *checksums* bijgehouden in het manifestbestand *manifest-md5.txt*, in dit geval uitgedrukt in MD5:

b4a88128dfba780402150a8b223f26a2 data/payload1.jpg

b4a88128dfba780402150a8b223f26a2 data/payload2.jpg

Tussen zender en ontvanger van een bagit bestaat een overeenkomst: de zender maakt het manifestbestand aan; de ontvanger herberekent de checksums van de payloads, en vergelijkt die met de checksums zoals vermeld in het bestand *manifest-md5.txt*. Wanneer die niet met elkaar overeenkomen, zijn er wijzigingen binnengeslopen, hetzij bewust (meestal met kwaadwillig opzet), hetzij onbewust (wegens netwerktransport).

Als map lijkt een bagit eerder een bestandsstructuur dan een formaat in strikte zin. Maar de standaard laat ook toe dat de map verpakt wordt als een archiefbestand, i.e. als ZIP of TAR.

BagIt kan ook metadata bevatten. Het bestand *bag-info.txt* bevat een vlakke beschrijving van de payloads in hun geheel:

Bag-Size: 758.7 KB

Bagging-Date: 2011-01-09

DC-AccessRights: ugent

DC-Creator: Verleysen, Sabine

DC-DateAccepted: 2011-01-09

DC-Identifier: rug01:001371848

Er zijn bestaan verscheidene softwarebibliotheken, die deze specificatie omzetten naar bruikbare code. De bekendste is de gelijknamige java-bibliotheek *BagIt Library* van de Library Of Congress. Maar ook voor andere programmeertalen zijn bibliotheken geschreven: *BagIt gem* (Ruby), *bagit* (Python), *pybagit* (Python) en *BagItPHP* (PHP).

Library of Congress – TransferTools – BagIt^{vii}

Console-applicatie waarmee bagits gemaakt kunnen worden, gevalideerd, en aangepast. Bagit is een open standaard om data ('payloads') te verpakken, metadata aan toe te voegen, en de wijzigingen in de data verifiëerbaar te houden. Speciaal hiervoor werd een software-bibliotheek BagIt geschreven in Java. De console-applicatie is een wrapper rondom die bibliotheek.

UGent – BagKit^{viii}

GUI rondom de hierboven vermelde software-bibliotheek BagIt. Bevat dezelfde functionaliteit als hierboven vermelde applicatie, en is eenvoudig in gebruik.

1.3 Verpakken van data en metadata

METS^{ix}

METS (Metadata Encoding and Transmission Standard) is een metadatastandaard om beschrijvende, administratieve en structurele metadata van digitale objecten uit een collectie te beschrijven. De voertaal is daarbij XML. METS combineert met andere woorden de digitale wereld met de wereld van de metadata.

De nood aan zo'n standaard komt voort uit de complexiteit van dergelijke digitale data. Wanneer men bijvoorbeeld de pagina's van een dichtensbundel inscant, en die wilt beschrijven in een document, moet men de volgende dingen bekijken:

- technische metadata van elk bestand (locatie, formaat, checksum, type).
- de bibliografische metadata . Vaak bestaat die in meerdere formaten. Uit het document moet ondubbelzinnig blijken dat de metadata handelt over ofwel het geheel van de beschreven bestanden, ofwel een deel ervan.
- Administratieve metadata moet gekoppeld worden aan ofwel het geheel van de bestanden, ofwel aan een deel ervan. Voorbeeld: auteursrechten.
- Er moeten verbanden gelegd kunnen worden tussen de verschillende onderdelen. Hoe maak je een computer bijvoorbeeld wijs dat een bibliografisch beschrijving in het document van toepassing op pagina 1 en 2 van de dichtensbundel, en niet op de rest?

METS lost deze complexiteit door het probleem te ontbinden in zijn onderdelen, en de linken later te leggen.

Een typisch METS-document bestaat uit de volgende “losse” onderdelen:

- één of meerdere *dmdSec* (Descriptive Metadata Section). Elke *dmdSec* bevat XML-data met beschrijvende metadata, of verwijst naar een externe locatie. Elke *dmdSec* heeft een identifier.
- één of meerdere *amdSec* (Administrative Metadata Section). Elke *amdSec* bevat XML-data met administratieve metadata, of verwijst naar een externe locatie. Elke *amdSec* heeft een identifier.
- één *fileSec* (File Section), met één of meerdere *fileGrp* en *file*. Elke *file* heeft een identifier.

Omdat die onderdelen op zich staan, kunnen we meerdere verbanden tussen hen leggen, louter door naar hun identifier te verwijzen. METS gebruikt hiervoor een *structMap*. Zo kunnen we in *structMap* 1 van elke ingescande pagina zeggen, welke gedichten er afgebeeld worden, door een *file* met een *dmdSec* te koppelen.

De structuur van een *structMap* is vrij.

Er bestaan verschillende softwarebibliotheken^x, voornamelijk geschreven in Java: METS Java Toolkit^{xi}, METS API 1.2^{xii}, jpstroom / mets-api^{xiii}. 7train METS Generation

Tool^{xiv} is een commandline tool om METS-documenten te genereren. Voor meer informatie omtrent de [GUI's voor METS](#), zie website van Library Of Congress.

Fedora SIP Creator^{xv}

Webapplicatie om digitale pakketten aan te maken, die in een Fedora-repository kunnen opgeladen worden. Zo'n digitaal pakket bestaat uit een lijst aan bestanden, samen met metadata in meerdere formaten. De bestanden kunnen geselecteerd worden via een bestandskiezer. Het biedt de mogelijkheid om mappen te maken naar een ander metadata-formaat. Het voortgebrachte pakket is enkel nuttig voor ingest in een Fedora repository.

Library of Congress – TransferTools – Bagger^{xvi}

Eveneens gemaakt door de Library of Congress. Ditmaal werd een GUI gemaakt rondom de software-bibliotheek BagIt geschreven. Interessant is de toevoeging van de BagView, een grafisch paneel waarmee de metadata, opgeslagen in het bestand bag-info.txt, kan worden aangepast. De velden van de bag-info.txt kunnen eveneens onderworpen worden aan regels zoals vastgelegd in een profiel. Handig dus voor organisaties die op een vaste manier hun metadata beschrijven.

1.4 Wijzigingen in de data verifiëerbaar maken

Er zijn bestaan verschillende – algemeen beschikbare – algoritmes om wijzigingen in data op te merken. Deze algoritmes aanvaarden als input binaire data, en geven een 'code' terug. Ze zijn zó gemaakt dat de wijziging van één bit in de data een andere code oplevert. Zo'n code wordt aangemaakt, en meegeleverd met de data bij een digitale overdracht (CD-ROM, USB, Internet ..). De ontvanger van de data produceert met hetzelfde algoritme de code opnieuw, en vergelijkt die met de meegeleverde code. Vandaar dat deze codes 'checksums' genoemd worden.

Voorbeelden van deze algoritmes zijn MD5 en SHA1.

1.5 Batch renamers

Batchfilerenametool^{xvii}

Grafische tool, gemaakt door ontwikkelaars van Google, om een reeks bestanden in een map te hernoemen. De procedure is als volgt:

- Selecteer een map. Alle operaties zullen toegepast worden op de onderliggende bestanden in deze map. De bestandsnaam wordt opgesplitst in de volgende onderdelen: naam + punt + extensie.
- selecteer 'prefix' om aan 'naam' toe te voegen (optioneel)
- geef woord op als vervanger voor 'naam' (optioneel)
- geef aan hoe uitkomsten genummerd moeten worden, hoe, en hoeveel 'padding' er gegeven moet worden (optioneel)

- geef suffix op (optioneel)
- vink aan of de extensie moet bewaard worden (optioneel)

Voorbeeld:

- Map: c++, met onderliggende bestanden Hello.cpp en Test.cpp
- prefix: hello-
- suffix: test
- padding: 4
- uitkomst: hello-Hello0001-test.cpp en hello-Test-0002.cpp

De tool is geschreven in Java, en dus beschikbaar voor elk systeem dat Java ondersteunt.

Bulk rename utility^{xviii}

Grafische tool, enkel voor Windows beschikbaar. De tool is zeer populair op het internet door de grote waaier aan opties bij het hernoemen van bestanden. Bestanden kunnen bijvoorbeeld hernoemd worden op basis van reguliere expressies. Maar juist deze grote schat aan mogelijkheden kan de modale gebruiker te veel overdonderen. Alle mogelijke opties zijn zichtbaar in het hoofdscherm, waardoor de interface onoverzichtelijk wordt.

File renamer Basic^{xix}

Grafische tool, enkel voor Windows beschikbaar, en niet gratis beschikbaar. De selectie van bronbestanden gebeurt op basis van reguliere expressies, net zoals de hernoeming. Technische metadata van het bestand kan ook worden opgenomen in de nieuwe bestandsnaam.

Advanced Renamer^{xx}

Grafische tool, enkel voor Windows beschikbaar. De tool is een stuk eenvoudiger dan de Bulk rename utility. Zo worden geen reguliere expressies gebruikt, maar wel een eenvoudigere syntax, vergelijkbaar met die van RenameWand (zie verder). Technische metadata van het bestand kan ook worden opgenomen in de nieuwe bestandsnaam.

PFrank^{xxi}

Grafische tool, enkel voor Windows beschikbaar. De tool is even complex als Bulk rename utility, en bevat grosso modo dezelfde functionaliteit, maar is veel overzichtelijker. Zo worden de meest gebruikte functies in het hoofdscherm gezet, terwijl de geavanceerde opties verbanden worden naar het menu.

RenameWand^{xxii}

Console applicatie, geschreven in Java, en dus beschikbaar voor alle platformen. De applicatie hanteert een eigen syntax om bestanden te

hernoemen, die veel eenvoudiger is dan de syntax van een reguliere expressie. De applicatie selecteert bestanden in een opgegeven map op basis van een *source pattern*, en hernoemt ze op basis van een *target pattern*.

Voorbeeld:

map: bhs1-pap, met bestanden BHS1-PAP-0064_2012_0001_MA.tif, BHS1-PAP-0064_2012_0002_MA.tif en BHS1-PAP-0064_2012_0001_MA.jpg

source pattern: <prefix>_<year>_0001_MA.<extension>

target pattern: <year>_<prefix>_0001_MA.<extension>

result:

BHS1-PAP-0064_2012_0001_MA.tif => 2012_BHS1-PAP-0064_0001_MA.tif

BHS1-PAP-0064_2012_0002_MA.tif => 2012_BHS1-PAP-0064_0002_MA.tif

1.6 Conclusies

- Een reeks tools en standaarden bevatten interessante ideeën, elk met hun voordelen en nadelen. Als basis van het containerformaat hebben we ervoor gekozen om de Bagit-standaard te combineren met het gebruik van METS. Met andere woorden: een bagit met een METS-document erin. Bagit kan beter om met het opslaan van de data, en het bijhouden van de checksums, terwijl METS beter is in het beschrijven van de metadata, en het leggen van de link tussen de metadata en de data.
- METS:
 - open standaard
 - verpakt zowel metadata als data. Een METS-document kan beide 'opnemen' op twee verschillende manieren: 1) door het kopiëren van de data in het document (mits speciale encoding); 2) door een verwijzing op te nemen naar de bestanden, vaak door middel van hyperlinks. De eerste optie is interessant omdat we streven naar het creëren van een exportformaat. Hyperlinks kunnen verwijzen naar data, die in de toekomst niet meer aanwezig is op de aangewezen plaats. In principe zouden we alleen METS kunnen gebruiken als containerformaat. Maar als we de data opnemen in het bestand, kan dit bestand te groot en onhandelbaar worden. Beeld je een METS-document in van een paar gigabytes, die niemand meer kan openen. Daarom is het beter om – enkel in het geval van de data – gebruik te maken van hyperlinks. Metadata moet in principe letterlijk opgenomen worden. We moeten ernaar streven dat de data niet naar externe metadata-records verwijst, omwille van dezelfde reden.
 - Checksums van data kunnen bijgehouden worden, maar zijn niet verplicht.
- Bagit:
 - open standaard

- verpakt zowel metadata als data. Verpakken van data is handiger. Metadata wordt stiefmoederlijk behandeld. De bag-info.txt kan enkel een vlakke beschrijving bevatten, geen gestructureerde.
- Checksums zijn verplicht
- Voor het hernoemen van bestanden moeten we op de volgende punten letten:
 - de te incorporeren software moet een bepaalde filosofie, i.e. een syntax bevatten om bestanden te hernoemen, en moet liefst mogelijk zijn in de vorm van een softwarebibliotheek. Een dergelijke software was helaas niet (direct) te vinden in open source. RenameWand, de console applicatie, kwam wel al aardig in de buurt. Het bevat een handige syntax, en is zo modulair gemaakt dat het niet zo moeilijk was om er een bibliotheek uit te extraheren. De eigen versie heet RenameWandLib.
 - er bestaan verschillende manieren om bestanden te hernoemen:
 - de gebruiker maakt de selectie van de bestanden
 - de gebruiker selecteert de map. Dat wil niet zeggen dat alle bestanden en mappen hieronder objecten van hernoeming zijn. Meestal moet de lijst nog wat uitgefilterd worden op basis van een patroon. RenameWand helpt hierbij.
 - Onderscheid maken tussen modale en geavanceerde gebruiker. RenameWand heeft een syntax voor beide gebruikers.
 - niet alle functionaliteit proberen op te dringen aan de gebruiker, of in de interface te steken.
- Keuze van globale grafische interface als vertrekpunt: Bagger van Library Of Congress:
 - gebruik van BagIt van Library Of Congress, die alle opties van de standaard ondersteunt.
 - Geschreven in Java, en dus te bruikbaar voor elk systeem dat Java ondersteunt.
 - BagKit bevat ook volledige ondersteuning, maar is in Ruby geschreven, waarmee ik minder vertrouwd ben.

3 Basis van de GUI: Bagger van Library Of Congress

1.1 Beschrijving

- Bagit aanmaken:
 - selecteer versie van bagit.

- selecteer profiel voor bag-info.txt
- Voeg payloads en tagbestanden toe:
 - klik op '+' boven de lijst om bestanden te selecteren
 - klik op '-' om een geselecteerd bestand of map terug uit de lijst te verwijderen.
 - Verplaats bestanden naar de lijst via drag and drop.
 - Verplaats url vanuit browser naar lijst via drag-and-drop
 - men kan met andere woorden data verzamelen van verschillende bronnen, een mappenstructuur maken,
- Opslaan van bagit:
 - kies opslagplaats
 - kies seriëel (zip, tar en tar.gz) of niet-seriëel formaat (bestandsmap)
 - kies checksum algoritme voor manifestbestanden.
 - Vink aan of een bag 'holey' is. Dat wil zeggen dat de payloads niet zullen opgenomen worden in de bag. Een bestand fetch.txt wordt gecreëerd met daarin de namen, en de locaties online waar de inhoud zich bevindt. Waar die bestanden zich bevinden, kan je aangeven door 'baseurl' op te geven. Die bestanden uit de lijst fetch.txt moeten gedownload worden, en aan de payloads toegevoegd vooraleer de validatie kan starten. Zo wordt de payload data/test.jpg toegevoegd aan fetch.txt als "<baseurl>/data/test.jpg".
 - Mappen met enkel het bestand .keep worden mee opgenomen in de payloads; lege mappen niet, conform de bagit specificatie.
- Pas de bag-info aan:
 - selecteer een veldnaam uit de standaardlijst, of vink 'standard' uit om zelf een veldnaam op te geven. Klik op add om veld en waarde toe te voegen aan de bag-info. Houd ermee rekening dat bagit-specifieke velden zoals Bag-Size, Bagging-Date en Payload-Oxum bij het opslaan steeds overschreven zullen worden.
 - Indien bij het aanmaken van de bag een profiel werd geselecteerd, worden de bag-info.txt voorzien van velden met al dan niet vooraf ingevulde waarden. Velden die nog aangevuld moeten worden, worden aangeduid. Het profiel controleert of een verplicht veld bestaat, en of de waarde voorkomt in een lijst van toegelaten waarden.
- Profiel voor bag-info.txt:
 - geschreven in JSON^{xxiii}, en ingeladen bij het aanmaken van een nieuwe bag. In zo'n bestand kan je het volgende opgeven:
 - veldnaam

- veld is vereist: ja/nee
- verplichte waarde
- default waarde
- mogelijke waarden die het veld mag bevatten.
- Bij het laden van het profiel wordt de bag-info voorzien van de in het profiel opgegeven velden. Default waarden of verplichte waarden worden vooraf ingevuld. In het geval van een verplichte waarde, wordt het veld read-only gemaakt. Het profiel controleert of alle verplichte velden opgegeven zijn, en de waarden voorkomen in de lijst van toegelaten waarden, indien zo'n lijst is opgegeven in het profiel.
- Validatie profiel is niet noodzakelijk om de bag op te slaan. Het levert wel een waarschuwing op.
- Een extra veld "Profile Name" wordt aan de bag-info toegevoegd, en is read-only.
- Controleer of bag 'complete' is
 - klik op het icoon 'is bag complete'.
 - Controleert of alle verplichte bestanden aanwezig zijn, en alle payloads zoals opgegeven in het manifestbestand.
- Valideer bag:
 - klik op het icoon 'validate bag'
 - controleert of de bag 'complete' is, en controleert de checksums van alle payloads
 - geeft een waarschuwing indien het profiel fouten opmerkt in de bag-info.
- Create Bag in Place:
 - klik op het icoon 'create bag in place'.
 - een bestandskiezer verschijnt die je toelaat een map te selecteren. De bestanden in die map worden beschouwd als de payloads van de te creëren bag.
 - een seriële formaat is niet mogelijk, enkel een eenvoudig bagit directory.
 - vink aan of rekening gehouden moet worden met '.keep'-bestanden. Dat zijn lege bestanden die in lege mappen geplaatst worden, om ervoor te zorgen dat de mappen worden meegeleverd. Lege mappen worden normaal gezien niet opgenomen.
- Console output handig om na te trekken waar het fout ging
- Mogelijkheid tot vertalen door middel van "messages" in textbestand.

- Bekijk tag files in viewer, in geformateerde modus

1.2 Aandachtspunten

- Vertalen niet overal mogelijk. Hier en daar staan nog hardgecodeerde boodschappen in het Engels.
- Bestanden toevoegen is onhandig op enkele vlakken:
 - drag-and-drop van bestanden naar de lijst lukt niet, zowel in Linux als in Windows.
 - Bestanden worden steeds rechtstreeks onder de map 'data' gevoegd. Er is geen mogelijkheid om een reeks bestanden aan een submap onder 'data' toe te wijzen. Verplaatsen van mappen en files in de bestandshiërarchie is ook niet mogelijk.
 - Geen mogelijkheid tot samenvoegen van mappenstructuren ('mergen').
 - Bestandskiezer geeft de indruk dat het mogelijk is om bestanden uit de lijst te verplaatsen via drag-and-drop, maar wanneer je een element uit de lijst verplaatst, verdwijnt hij bij het loslaten.
 - De standaardmap van de bestandskiezer is de root van het bestandsysteem (i.e. C:/). Het zou handiger zijn mocht de laatst geselecteerde map gekozen worden, of – indien de bestandskiezer voor het eerst wordt opgestart – de home directory van de gebruiker. Zowel in Linux als in Windows.
 - Optie nodig om te controleren of een map gewijzigd is in het bestandsysteem, en dan node in kwestie updaten? Is waarschijnlijk niet wenselijk, zeker als je files in de hiërarchie wilt verplaatsen. De 'reload' optie zou de bestandslijst in de map alleen maar opnieuw vullen met de bestanden die je zonet verplaatst hebt.
- Baginfo:
 - Profielen zijn bestanden, geschreven in JSON. Geen GUI^{xxiv} om zo'n profiel aan te maken of aan te passen. Te technisch dus, hoewel de semantiek heel eenvoudig is.
 - De veldnamen van het profiel verschijnen niet in de lijst van toe te voegen velden.
 - Profielen niet aanpasbaar eenmaal bag geladen
 - Betere feedback bij validatie profiel nodig.
 - Optie bij profiel toe te voegen: opslaan bij foute profiel-validatie ja/nee.
 - Bij het openen van een nieuwe bag kan het profiel gekozen worden. Niet meer daarna.

- Create bag:
 - Bij 'create bag' wordt er niet gecheckt of er een bag open staat. De open bag wordt gewoon gesloten, en wijzingen niet opgeslagen. In dat geval moet er de vraag komen of de bag niet opgeslaan moet worden.
- Bag in place:
 - Zou net als bij 'create bag' een waarschuwing moeten geven wanneer een openstaande bag gesloten wordt.
- Holey bag:
 - Begrip van 'holey' komt niet helemaal overeen met de specificatie. 'Holey' betekent gewoon dat bijkomende files gedownload moeten worden, en aan de payloads toegevoegd. Bagger geeft de indruk dat een holey bag geen enkele payload opneemt. Waarschijnlijk bedoelt de auteur dat de bag 'holey' gemaakt wordt door middel van de functie 'makeHoley' uit de bagit library.
 - verwarrend: je selecteert eerst bestanden uit het bestandstelsel, of uit je browser, om dan vervolgens aan te geven dat ze niet op je lokale schijf staan, maar elders.
 - Misschien beter:
 - make holey (maak je volledige bag holey)
 - voeg extern bestand toe (url)

1.3 TODO

- Vertalen naar Nederlands via configuratiebestand messages.properties.
- Metadata:
 - Toevoegen metadatabestanden via bestandslijst zoals er reeds bestaat voor de payloads en de tagbestanden.
 - Validatie metadatabestanden aan bestaande standaarden (EAD, MODS, Dublin Core, MADS en Premis).
 - Automatische generatie van mets.xml op basis van de bag-info.txt, manifest-m5.txt en meegeleverde metadata. Meegeleverde mets.xml wordt aangevuld.
- Bestanden:
 - Bestandsherkenning, en controles op bestandstype.
 - Opkuisen van bestandsnamen, i.e. verwijderen van tekens uit de namen die de overdracht van bestanden kan bemoeilijken. vb. spaties vervangen door underscores.

- Bestanden hernoemen volgens patronen (en het gebruiksvriendelijk houden). Die renamer moet automatiseren wat men ook handmatig kan doen.
- Opladen van een bagit naar een externe repository.
- Profielen bag-info.txt aanmaken via GUI.
- (Creëren van bags in batch-modus.)

4 Verklarende woordenlijst

AFF	Another Fine Format
GUI	Grafic User Interface
JSON	Javascript Object Notation
METS	Metadata Encoding and Transmission Standard
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations

- i <https://github.com/delving/sip-creator>
- ii <http://www.delving.eu/the-delving-platform>
- iii <http://www.openarchives.org/pmh/>
- iv <http://europeanalabs.eu/sip-creator/>
- v http://ufdc.ufl.edu/?m=hesoftware_mets
- vi <http://www.digitalpreservation.gov/documents/bagitspec.pdf>
- vii <http://loc-xferutils.sourceforge.net/>
- viii <https://github.com/nics/BagKit>
- ix <http://www.loc.gov/standards/mets/>,
<http://www.loc.gov/standards/mets/METSPrimerRevised.pdf>
- x <http://www.loc.gov/standards/mets/mets-tools.html>
- xi <http://hul.harvard.edu/mets/>
- xii <http://sourceforge.net/projects/mets-api/>
- xiii <https://github.com/jpstroop/mets-api>
- xiv <http://sourceforge.net/projects/seventrain>
- xv <http://www.fedora-commons.org/download/2.2/services/sipcreator/doc/index.html>
- xvi <http://sourceforge.net/projects/loc-xferutils/files/loc-bagger/>
- xvii <http://code.google.com/p/batchfilerenametool/>
- xviii http://www.bulkrenameutility.co.uk/Main_Intro.php
- xix http://download.cnet.com/File-Renamer-Basic/3000-2248_4-11753579.html
- xx <http://www.advancedrenamer.com>

xxi <http://www3.telus.net/pfrank/index.html>

xxii <http://renamewand.sourceforge.net/>

xxiii Javascript Object Notation (JSON): *JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.* (<http://www.json.org/>)

xxiv Graphic User Interface (GUI)