

# Archiving Electronic Literature

A Workflow

@IIPC 2016

Steffen Fritz  
Stephanie Kuch

14.04.2016

# Project

- *Preserving German Net Literature And Making It Long-Term Available*
- Deutsches Literaturarchiv Marbach<sup>1</sup>
- German Research Foundation
- Project duration: jan 2013 - apr 2016
- Wiki: <https://wwik.dla-marbach.de/line>

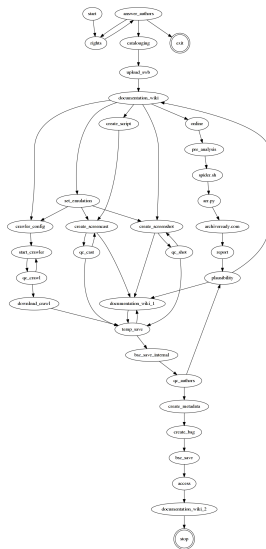
---

<sup>1</sup><http://www.dla-marbach.de>

# Corpus

- Net literature / electronic literature
- 50 works
- Hypertext, Flash, Prezi, Hypercard, ...
- 1989, 1995 - 2012
- Examples:  
[https://wwik.dla-marbach.de/line/index.php/Die\\_Quellen](https://wwik.dla-marbach.de/line/index.php/Die_Quellen)

## Workflow sketch



# Workflow (simplified)

- 1 Obtaining rights for archiving
- 2 Cataloguing
- 3 Begin documentation (project wiki)
- 4 Archivability analysis of the web resource
- 5 Harvesting and documentation
  - Online and crawlable: Crawl
  - Online or emulated: Screenshot
  - Online or emulated: Screenshot
  - Source code/databases available: Source code
- 6 Create metadata.xml and bag
- 7 Transfer bag to long-term preservation service provider (BSZ)
- 8 Make available (BSZ)
- 9 Finish documentation (project wiki)

# Workflow (even more simplified)

- 1 Obtaining rights for archiving
- 2 Cataloguing
- 3 Begin documentation (project wiki)
- 4 **Archivability analysis of the web resource**
- 5 Harvesting and documentation
  - Online and crawlable: Crawl
  - Online or emulated: Screenshot
  - Online or emulated: Screenshot
  - Source code/databases available: Source code
- 6 **Create metadata.xml and bag**
- 7 Transfer bag to long-term preservation service provider (BSZ)
- 8 Make available (BSZ)
- 9 Finish documentation /project wiki)

# Archivability analysis

- Using <http://archiveready.com>'s API
- ArchiveReady analyses one website per request
- Tests/Facettes:
  - Accessibility
  - Standards and Compliance
  - Cohesion
  - Performance
  - Metadata

# Archivability analysis

- Using `http://archiveready.com`'s API
- ArchiveReady analyses one website per request
- Tests/Facettes:
  - Accessibility
  - Standards and Compliance
  - Cohesion
  - Performance
  - Metadata

- 1 Find all URL
- 2 Pass and receive the json report
- 3 Interpret report
- 4 optimize crawl



# Spider

```
#!/usr/bin/env zsh

wget -nv -r --spider -i $1 2 > &1 |
egrep " URL: " |
awk '{ print $3 }' |
sed "s@URL:@@g" >> corpus_link_list.txt

cat corpus_link_list.txt | sort | uniq >
corpus_link_list_sorted_uniq.txt
```

## corpus\_link\_list\_sorted\_uniq.txt

(...)

<http://oliver-gassner.de/textratouren/n99/index.html>

<http://oliver-gassner.de/textratouren/n99/ly.html>

<http://oliver-gassner.de/textratouren/n99/qu.html>

<http://oliver-gassner.de/textratouren/n99/re.html>

(...)

## arrr.py

Python script that ...

- 1 reads a text file line by line (URL by URL)
- 2 passes URL to archiveready.com
- 3 receives json report
- 4 beautifies report
- 5 saves to text file
- 6 prints problematic findings

# Analysis report

Example:

[https://wwik-prod.dla-marbach.de/line/images/4/48/Noise99\\_beurteilung\\_spiegelungsfahigkeit.pdf](https://wwik-prod.dla-marbach.de/line/images/4/48/Noise99_beurteilung_spiegelungsfahigkeit.pdf)

# Procedure

- Depending on the report, create / get
  - Screenshot
  - Screencast
  - Crawl
  - Source code / database

# Objects to archive

Screenshot: Image files

Screencast: Video files

Crawls: Heritrix\_Job\_Directories.tar.gz

Sourcecode: sourcefile.tar.gz

## DLA-Bag

```

bag/
|
|-- data
|   |-- screenshot_00.jpg
|   |-- screenshot_00.tif
|   |-- (...)
|
|-- manifest-sha1.txt
|   51afb385ha019f34b67ae1 data/screenshot_00.jpg
|   (...)
|
|-- metadata.xml
|
|-- bagit.txt
|-- bag-info.txt

```

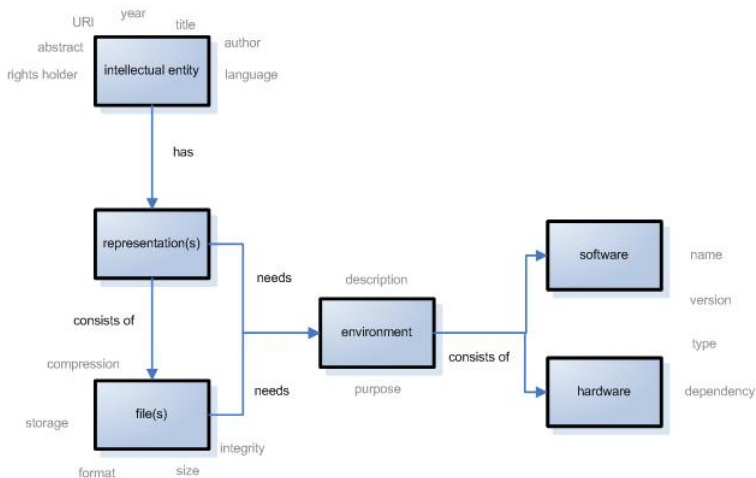
# Metadata

## Documentation of

- bibliographic information
- legal information
- technical information



# Metadata



# Metadata

- METS
- MODS
- PREMIS

# METS

- metsHdr
- dmdSec
- amdSec
- fileSec
- structMap
- structLink
- behaviorSec

# METS

- **metsHdr**
- **dmdSec**
- **amdSec**
- **fileSec**
- **structMap**
- **structLink**
- **behaviorSec**

## MODS

abstract

accessCondition

classification

extension

genre

identifier

language

location

name

note

originInfo

part

physicalDescription

recordInfo

relatedItem

subject

tableOfContents

targetAudience

titleInfo

typeOfResource

# MODS

**abstract**

**accessCondition**

classification

extension

**genre**

identifier

**language**

**location**

**name**

note

**originInfo**

part

**physicalDescription**

recordInfo

relatedItem

subject

tableOfContents

targetAudience

**titleInfo**

**typeOfResource**

# PREMIS - Version 2

- file
- representation

# PREMIS - file

objectIdentifier

preservationLevel

significantProperties

objectCharacteristics

originalName

storage

environment

signatureInformation

relationship

linkingEventIdentifier

linkingIntellectualEntityIdentifier

linkingRightsStatementIdentifier



# PREMIS - file

**objectIdentifier**

preservationLevel

significantProperties

**objectCharacteristics**

originalName

**storage**

**environment**

signatureInformation

**relationship**

linkingEventIdentifier

linkingIntellectualEntityIdentifier

linkingRightsStatementIdentifier

# PREMIS - representation

objectIdentifier  
preservationLevel  
significantProperties  
originalName  
storage  
environment  
relationship  
linkingEventIdentifier  
linkingIntellectualEntityIdentifier  
linkingRightsStatementIdentifier

# PREMIS - representation

**objectIdentifier**

preservationLevel

significantProperties

originalName

storage

**environment**

**relationship**

linkingEventIdentifier

linkingIntellectualEntityIdentifier

linkingRightsStatementIdentifier

## More Information

- 1 Workflow: <https://wwik-prod.dla-marbach.de/line/index.php>
- 2 ArchiveReady: <http://archiveready.com>
- 3 The CLEAR method:  
<https://doi.org/10.1007/s00799-015-0144-4>
- 4 arrr.py: <https://github.com/netzliteratur/arrr>
- 5 SWBcontent:  
<https://www.bsz-bw.de/mare/lza/swbcontent.html>

Thank you!

Stephanie Kuch  
kuch@dla-marbach.de

Steffen Fritz  
fritz@dla-marbach.de