MODELLIERUNG VON METADATEN IN DSPACE

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

Philipps Universität Marburg

INHALTSVERZEICHNIS

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

1. METADATEN IN DSPACE

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

Philipps Universität Marburg

DSpace unterstützt/vergibt automatisiert

- `Descriptive Metadata`
- `Administrative Metadata`
- `Structural Metadata`

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

Philipps Universität Marburg

- Jedes Objekt (Item, Community/Collection, Person, etc.) in DSpace besitzt Metadaten
- Metadaten liegen als `String`-Werte in der Datenbank
- (Qualified) Dublin Core gehören zum DSpace-Standard für deskriptive Metadaten

DSpace Intermediate Metadata (internes Schema)

```
<dim:dim xmlns:dim="http://www.dspace.org/xmlns/dspace/dim" dspac
    <dim:field mdschema="dc" element="title" lang="en_US">
        The Endochronic Properties of Resublimated Thiotimonline
    </dim:field>
    <dim:field mdschema="dc" element="contributor" qualifier="aut
        Isaac Asimov
    </dim:field>
    <dim:field mdschema="dc" element="language" qualifier="iso">
        eng
    </dim:field>
    <dim:field mdschema="dc" element="subject" qualifier="other"
        time-travel scifi hoax
    </dim:field>
</dim:dim>
```

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

Philipps Universität Marburg

1.1 ANPASSUNGSMÖGLICHKEITEN

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

Philipps Universität Marburg

Erweiterungen oder neue Schemata sind konfigurierbar:

- `[dspace]/config/registries` im XML-Format
- über ein Kommandozeilenwerkzeug
- über die Rest-Schnittstelle

```
<!-- Used by system: do not remove -->
<dc-type>
    <schema>dc</schema>
    <element>contributor</element>
    <qualifier>author</qualifier>
    <scope_note></scope_note>
</dc-type>
```

- jedem Metadatenfeld kann eine `Authority` angegeben werden, z.B.:
  - über ORCID-Plugin für Autor/innen
  - über selbst implementierte Plugins
- `Authority` ist zum Metadatenfeld gehöriger Wert in der Datenbank
- Wird regulär bei externer Kommunikation nicht mitgegeben

1.2 BESCHRÄNKUNGEN DESKRIPTIVER METADATEN

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

Philipps Universität Marburg

- Maximal ein Attribut pro Metadatenfeld im Form des Qualifiers, z.B. `dc.date.accepted`
- Nur ein Wert pro Metadatenfeld -> keine Verknüpfung von zusammengehörigen Feldern, z.B. `dc.relation` im Blick auf DataCite's `relatedIdentifier`
- Das Bedienen von Schnittstelle nach extern (Crosswalks) nur auf Basis-Anforderungen beschränkt

Beschränkungen werden z.T. in DSpace-Repositorien umgangen:

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

Philipps Universität Marburg

- durch Verwendung "erweiterter" Qualifier, z.B.:
  - **local.source.container-type** = `periodical`
  - **local.source.container-type-name** = `Zeitschrift`
  - **local.item.creatorGND**
- Problem: Keine Zuordnung, wenn mehrere Felder gleichen Typs verwendet werden.

- durch das Speichern mehrere Werte in einem Metadatenfeld, z.B.
  - **dc.contributor.author** = `Müller,Lieschen;Philipps-Universtität Marburg;http://d-nb.info/gnd/0000`
  - **dc.relation** = `hasPart|DOI|https://dx.doi.org/10.5072/0000`
- Problem: Bei Crosswalk könnten Felder falsch ausgelesen werden (z.B. Reihenfolge)

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

2. PROJEKT: METADATENFELDER MIT KOMPLEXEN WERTEN

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

Philipps Universität Marburg

Anforderungen:

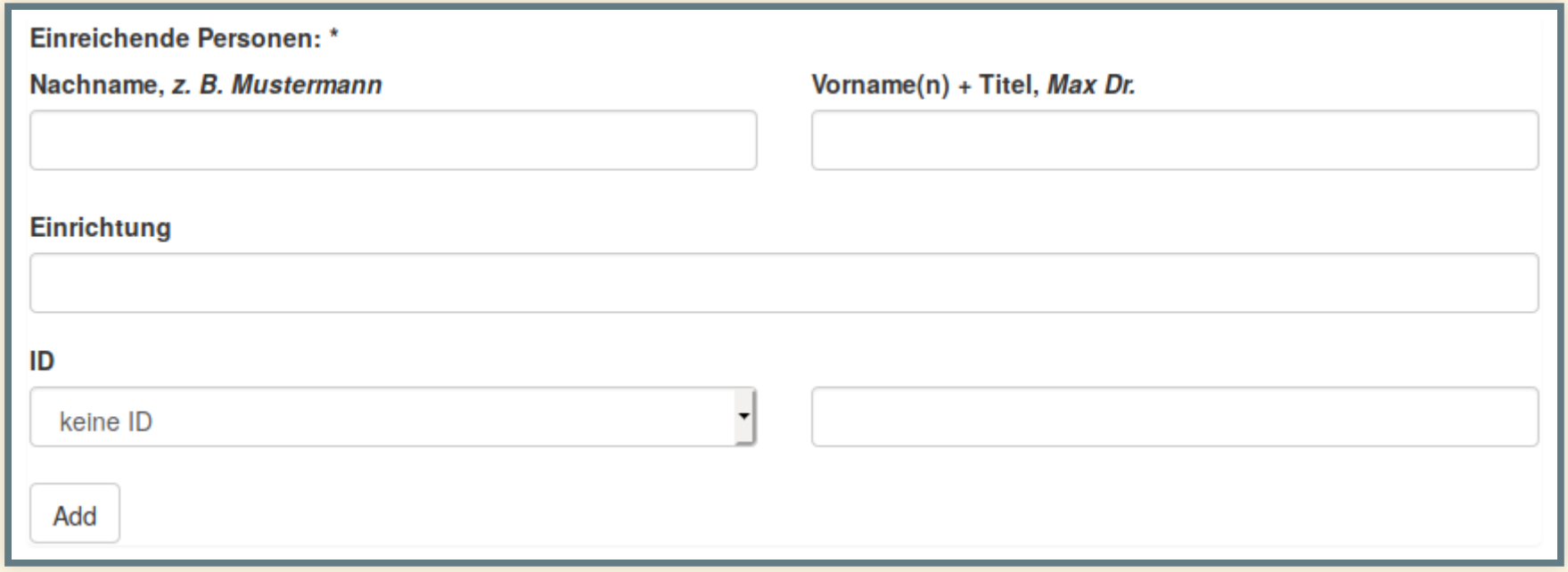
- Zusammengehörige Informationen in einem Metadatenfeld speichern.
- Informationen sollen im standardisierten Format vorliegen.
- Die Eingabe der Informationen in der Submission-Maske soll so komfortabel wie möglich sein.
- Die gespeicherten Informationen sollen bei Crosswalk verlustfrei ausgelesen werden können.

2.1 UMSETZUNG

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

Philipps Universität Marburg

Mehrere Werte zu einem Metadatenfeld können separat eingegeben werden.



Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de

Philipps Universität Marburg

Eingabe wird im **JSON**-Format gespeichert (Teileingaben möglich):

```
{"last":"Müller","first":"Lieschen","affiliation":"Philipps-Unive
"id":"gnd","id_value":"http://d-nb.info/gnd/0000"}
```

- XSL-Template ermöglicht Informationen aus JSON-String separat auszulesen.
- Bei Crosswalk können die Informationen an die entsprechenden Stelle gemappt werden.

```
<xsl:template name="parseAuthor">
    <xsl:variable name="lastName">
        <xsl:call-template name="parseJSON">
            <xsl:with-param name="input" select="."/>
            <xsl:with-param name="key">last</xsl:with-param>
        </xsl:call-template>
    </xsl:variable>
    <xsl:variable name="firstName">
        <xsl:call-template name="parseJSON">
            <xsl:with-param name="input" select="."/>
            <xsl:with-param name="key">first</xsl:with-param>
        </xsl:call-template>
    </xsl:variable>
    <creatorName>
        <xsl:value-of select="$lastName"/>
        <xsl:if test="$firstName!='unknown'">
```

2.2 VORTEILE

- Konfigurierbar
- Sicheres Auslesen über standardisierte JSON-Bibliotheken/-Templates (JavaScript,XSLT)
- Komplexe und nicht komplexe Werte können nebeneinander existieren
- Bietet umfangreiche Anreicherung von Metadaten
- Zukunftsfähig und interoperabel

2.2 NACHTEILE

- Zusätzliche Code-Pflege
- Zusätzliche Anpassung der UI (zur Zeit nur *XMLUI*)

Paul Münch
Kontakt: paul.muench@ub.uni-marburg.de