Towards rich and standardized metadata in iRODS

Mariana Montes & Paul Borgermans
Metadata schemas in the Mango portal

- Apply rich and standardized metadata
- Hierarchical structure for metadata
- Schema life cycle
First challenge

Applying rich, standardized metadata in a systematic way
Metadata without schemas

Typos, inconsistencies…

color: gray
colour: grey
COLOR: Gray
color: gris
kleur: grijs
colour: slate gray
colour: SlateGray
colour: slat gray
colour: #708090
The alternative: metadata with schemas (forms)

- All necessary and suggested fields.
The alternative: metadata with schemas (forms)

- All necessary and suggested fields.
- Predefined options for the values.
The alternative: metadata with schemas (forms)

- All necessary and suggested fields.
- Predefined options for the values.
- Other types of validation (formats, ranges…)

Metadata schema: Toy block 1.0.0

- Material*: wood
- Color:
  - grey
  - red
  - blue
  - yellow
  - green
  - other
- Purchase date: mm/dd/yyyy
- Input type: date
- Width (in cm)*
  - Input type: float between 0.8 and 8.6

Save metadata
The alternative: metadata with schemas (forms)

- All necessary and suggested fields.
- Predefined options for the values.
- Other types of validation (formats, ranges…)
- Multiple AVUs with the same name AND consistency in names.
Second challenge

Metadata with a hierarchical structure
Hierarchical structure with schemas: namespacing

Schema: “book”
Composite field: “author”

mgs.book.author.given_name: Mariana
mgs.book.author.last_name: Montes
mgs.book.author.email: mariana.montes@kuleuven.be
mgs.book.author.email: montesmarianna@gmail.com
Hierarchical structure with schemas: units

<table>
<thead>
<tr>
<th>Author</th>
<th>Given name*</th>
<th>Montes</th>
<th>Email address</th>
<th><a href="mailto:mariana.montes@kuleuven.be">mariana.montes@kuleuven.be</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Input type: text</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last name*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input type: text</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mgs.book.author.given_name: Mariana 1</td>
<td></td>
<td></td>
<td><a href="mailto:mariana.montes@kuleuven.be">mariana.montes@kuleuven.be</a> 1</td>
<td></td>
</tr>
<tr>
<td>mgs.book.author.last_name: Montes 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mgs.book.author.email: <a href="mailto:mariana.montes@kuleuven.be">mariana.montes@kuleuven.be</a> 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mgs.book.author.email: <a href="mailto:montesmariana@gmail.com">montesmariana@gmail.com</a> 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Schema: “book”
Composite field: “author”

- mgs.book.author.given_name: Mariana 1
- mgs.book.author.last_name: Montes 1
- mgs.book.author.email: mariana.montes@kuleuven.be 1
- mgs.book.author.email: montesmariana@gmail.com 1
- mgs.book.author.given_name: Paul 2
- mgs.book.author.last_name: Borgermans 2
- mgs.book.author.email: paul.borgermans@kuleuven.be 2
Third challenge

Combining stability and evolution potential of metadata schemas
Without life cycle

Every schema editable AND implementable

Each schema editable OR implementable, new schemas every time
Without life cycle

Every schema editable
AND implementable

UNRELIABLE

Each schema editable
OR implementable, new schemas every time
Without life cycle

- Every schema editable AND implementable
- Implementable, new schemas every time

UNRELIABLE

NO HISTORY
With life cycle

draft → published → archived

+ semver
Technical specifications

Metadata schema manager under the hood
metadata schemas under the hood

Python in the backend, JS in the frontend

Bootstrap 5.2
Forms and validation

(Custom) JSON format

```
{
    "schema_name": "book",
    "version": "1.0.0",
    "status": "published",
    "title": "Book information",
    "properties": {
        "name": {
            "type": "text",
            "title": "Book title"
        }
    }
}
```
Thank you!

mariana.montes@kuleuven.be
paul.borgermans@kuleuven.be

DEMO:
https://github.com/kuleuven/mango-metadata-schemas
JSON format: schema

```
{
  "schema_name": "book",
  "version": "1.0.0",
  "status": "published",
  "title": "Book information",
  "properties": {...},
  "edited_by": "username",
  "realm": "project_collection",
  "parent": ""
}
```

- Unique name
- Life cycle (versioning)
- Display name
- Components
- Other metadata
JSON format: simple field

```
"title": {
  "type": "text",
  "title": "Book title",
  "required": true,
  "help": "Title in the book (also translations)",
  "repeatable": true
}
```

Common fields

- Unique name
- Display name
- Type of field
- Boolean values: required, repeatable
- Help text, default value
JSON format: multiple-choice

```
"ebook": {
  "type": "select",
  "title": "Is there an e-book?",
  "multiple": false,
  "ui": "radio",
  "required": true,
  "values": [
    "Available",
    "Unavailable"
  ]
}
```

### Specific fields

- Whether only one or multiple values can be chosen
- How to render (dropdown, checkbox, radio)
- Array of possible values
"author": {
  "type": "object",
  "title": "Author",
  "properties": {
    "name": {
      "type": "text",
      "title": "Name and surname",
      "required": true
    }
  },
  "repeatable": true
}