Shapes, Forms & Footprints

Towards Web generation of RDF data without coding

Patrick Hochstenbach (UGent)
SWIB 2022
It all started at the start of COVID-19 crisis

- Library IT had to quickly invent teleworking tasks for staff working at home
- Crowd sourcing of metadata production
  - Card card catalog
  - Images
  - Digitized materials (with personal information)
- Could we make something quick to generate metadata?
- Couldn't use cloud services because of GDPR reasons
- First app took a weekend to create, the second weeks, months with many variations
- Could this been done easier? Google Forms but in our environment?
- Google Forms but with structured data (not Excel like output)?
Mellon Research Pod project

- Scholarly Communication in a Decentralized Web
- Very decentralized production of (RDF) data
- Every researcher has her own Researcher Pod (mini institutional repository)
- … on which she stores her publications
- … and described these publications with RDF data
- How to produce RDF data in a very decentralized environment?
- … with many local variations in metadata requirements
How to produce RDF data?

(Web) Forms that create/edit data for a knowledge graph.
Decoupled & Decentralised
Decoupled & Decentralised
What components are needed for such apps?

https://ruben.verborgh.org/blog/2019/06/17/shaping-linked-data-apps/
Shapes & Forms

Book shape with all its complexities

Book form you want users to fill in
## Specifications + technologies

<table>
<thead>
<tr>
<th>Shapes</th>
<th>Forms</th>
<th>Footprints</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDF</td>
<td>RDF</td>
<td>RDF</td>
</tr>
<tr>
<td>● SHACL</td>
<td>(with coupling to validation)</td>
<td>● Hydra Core ?</td>
</tr>
<tr>
<td>● ShEx</td>
<td>● SHACL-Form-React</td>
<td>● Shape Trees ?</td>
</tr>
<tr>
<td>XML</td>
<td>● SHACL Forms</td>
<td>● N3Logic ?</td>
</tr>
<tr>
<td>● XForms</td>
<td>● ShExC forms</td>
<td></td>
</tr>
</tbody>
</table>
Towards a pragmatic solution

Focus:

● Decentralised Web Application
  ○ No server components needed (except for the final location where to store the data)

● Lazy User

● Lazy Developer

● Intuitive user interactions > complex data models
  ○ What are the features we need to solve 80% of the use cases
  ○ Data models should bridge the gap from no data -> structured RDF data -> linked data -> your desired data model

● Declarative App (tell what it does, not how and where)
● Should work against authenticated Solid API, possible other LDP endpoints
● Provide inspiration for standards, industry, etc
Architecture

Alice's Pod uses FormViewer App, which reads RDF for Form Definition, Shape Definition, Data, and Footprint Definition. The FormViewer App interacts with Alice's Pod.
Form Viewer App

- [https://github.com/phochste/FormViewer](https://github.com/phochste/FormViewer)
  - Form = RDF resource that defines what to show in an HTML form plus the RDF bindings
    - Currently using Daniël Beeke's [rdf-form](https://github.com/phochste/FormViewer)
  - Data = some RDF resource on the web (possibly on an (authenticated) Pod)
  - Shape = should be defined in the Pod
  - Footprint = RDF resource that defined what to do when the submit button is pressed
    - Currently using Hydra vocabulary
  - Pass all Form, Data, Footprint to the WebApp by reference
Create a ex:Book instance

Create a name/value pair as dc:title

Create a dropdown as ex:rating
Data

@prefix ex: <https://example.org/> .
@prefix dc: <http://purl.org/dc/terms/> .

[] a ex:Book ;
  dc:title "Winne the Pooh" ;
  dc:creator "A.A. Milne" ;
  ex:rating ex:LikedIt ;
  ex:other "Test" ;
  dc:description "Not enough cats".

Data the form can create/update
Footprint (hydra)

@prefix hydra: <http://www.w3.org/ns/hydra/core#> .
@prefix dc: <http://purl.org/dc/terms/> .

[] hydra:endpoint <https://hochstenbach.inrupt.net/inbox> ;
   # Optional define where to go after submitting the form
   # hydra:next <http://some.page.on.the.web>
   # Optional header to show on form
   # dc:title "The title of my form"
   # dc:description "The description of my form"
   hydra:supportedClass [
       a hydra:Class ;
       hydra:method "POST"
   ] .

Send to this resource after submitting the form
Using HTTP POST
Title

Winne the Pooh

Author

A.A. Milne

Rate this book

- Select a value -

Review

Not enough data

Save
RDF-Form supports

- Name value pairs
- Textarea
- Checkboxes
- Selection
- Autocomplete with SPARQL query
- Groups of combination of above
- Repeated fields,
- … but do we want / need all that complexity for Google Forms like use cases?
- What are the minimum requirements?
Form Generator

- Drag & Drop Web App by smessie to create the Shape of a Form
- Based on Google Forms data model
  - Name/Value pairs
  - TextArea
  - Select Dropdown
  - Checkbox
  - Date
- RDF Bindings
- Labels
- Generates shape in format
  - SHACL
  - Solid-UI
  - rdf-form
Create RDF linked data and what next?

- Start a workflow to create better linked data
- Create apps that use known shapes in a nice way

https://github.com/phochste/CVViewer
Next steps

- Adding reasoning to Solid Apps
  - Schema alignment
    - My FormViewer app uses Daniël Beeke's rdf-form vocabulary, but what if I get a form definition in SHACL or Solid-UI?
    - What if the data source I want to edit has a slightly different shape than the form definition?
  - Footprints
    - The FormViewer uses Hydra and can only send data to hardcoded locations.
    - What I want to use multiple locations?
    - What I want to decide only storage location dynamically (based on the data)?
- Currently investigating [N3Logic](https://w3c.github.io/N3/spec/) and [RDF Surfaces](https://josd.github.io/surface/) with [smessie](https://w3c.github.io/N3/spec/).
Questions?

- Patrick.Hochstenbach@UGent.be
- @hochstenbach@scholar.social