

Development of the SSHADE interface

Philippe Bollard

`philippe.bollard@univ-grenoble-alpes.fr`

Second SSHADE partners meeting

4-5 December 2017 - IPAG, Grenoble, France

1. Current state
2. Expected evolutions
3. Optimization of DB search
4. Beta testing

Current state

Deployment of the new search engine is still in progress

New search UI

- a main field for "full-text" query → "simple" search
- a collapsible form with a lot of fields → "advanced" search

Datamodel indexation to be tuned

- select a restricted set of fields for "full-text" search
- adjust the depth of parent/son values linked to a data
- choose the right analyzer/tokenizer for each kind of values

Reorganized data on all pages

- use collapsible blocks
- aggregate sub-data on upper page
- hide some fields to basic users

New navigation layout with sidebar

- tree with experiment/sub-experiments/spectra
- tree with sample/layers/(matters/)materials
- collapsible and always on screen
- provide direct links to related pages

Expected evolutions

Expected evolutions

For delivery

Navigation through sidebar

- keep the tree on sidebar for all sub-pages
- display values related to a data in its parent structure

Global layout

- home page, text and logos...

Export spectrum data and sample metadata

- at least, using import history
- custom ASCII or XML format ?
- VOTable (or through OSUG-VO ?)

Expected evolutions

After delivery

Data validation and access restriction

- don't publish non-validated or "private" data
- use user groups to share a set of data across databases

User dashboard and settings

- data to validate, last imports, recent searches, ...
- preferred units, export format, ...

Cooperation with Unibe (Bern)

- multiangular spectrum visualization tool
- API providing an access to SSHADE data to other softwares

Evolutions of the datamodel

- bandlist, spectral image, ...

Import spectrum data from other file formats

- custom: ascii-sbrdf-ipag (SHINE and SHADOWS instruments)
- generic: netCDF, JCAMP-DX, Brucker, ... ?

Export spectrum data to other file formats

- netCDF, FITS, ... ?

DOI

- using the OSUG-DOI service

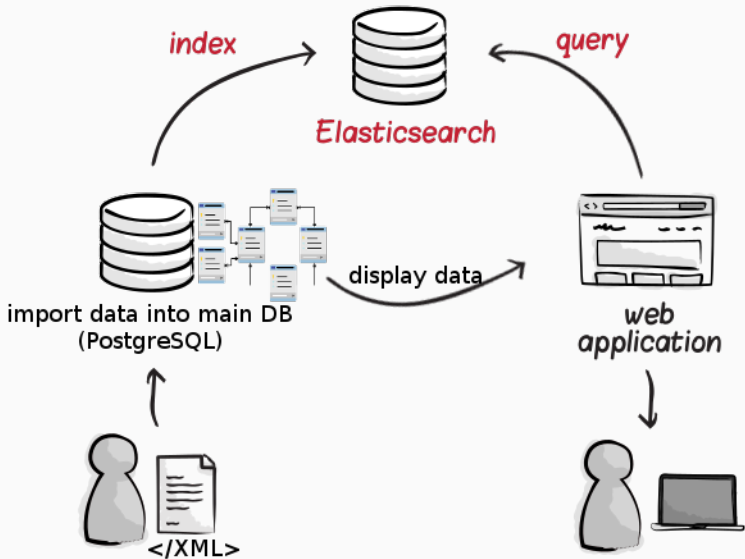
The OSUG-DOI service will not publish a DOI in realtime

- SSHADE: export data into OSUG-DOI custom ASCII format
- OSUG-DOI: fetch the data set (each day?)
- OSUG-DOI: generate static pages describing the data sets
- OSUG-DOI: request a test DOI for each data set
- SSHADE: check generated pages, validate the DOI submission
- OSUG-DOI: request an official DOI for each data set
- OSUG-DOI: publish static pages for each data set
- SSHADE: save generated DOI for each data

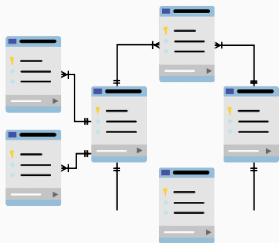
All DOI will be generated within the OSUG namespace

Optimization of DB search

Elasticsearch: the search engine used by SSHADE



Extract a dataset into a JSON document



Relational model



```
1  [
2  {
3    "name": "Top Level",
4    "parent": "null",
5    "children": [
6      {
7        "name": "Level 2: A",
8        "parent": "Top Level",
9        "children": [
10       {
11         "name": "Son of A",
12         "parent": "Level 2: A"
13       },
14       {
15         "name": "Daughter of A",
16         "parent": "Level 2: A"
17       }
18     ]
19   },
20   {
21     "name": "Level 2: B",
22     "parent": "Top Level"
23   }
24 ]
25 }
26 ]
```

JSON document

A good JSON document contains...

All needed properties

- UID
- title, name, formula, ...
- type, family, origin, ...

All needed parent/related data

- spectrum: experiment, sample, ...
- spectrum: parent spectra, ...

All needed children data

- sample: layer
- layer: materials, matters, ...

Indexation of a document

Data type assigned to each property

- text: UID, title, name, ...
- keyword (enum): family, type, ...
- float, integer, boolean

Text

- the string is splitted into a list of words
- each word is processed (lowercase, ...) and indexed

Keyword

- the full string is indexed without any transformation
- useful for enum values or exact matching

Search with Elasticsearch



search
query

search



matching
documents

The two parts of a query

Full-text

- use a hidden field containing all values of the document concatenated into a big string
- return all document with at least an occurrence of the query words regardless of the source fields

Filters

- will return a document only if the target field contains the requested value
- used to refine a search result or to do an "advanced search"

Beta testing

Before delivery

- tested only by the development team
- we need to finalize the UI and remove all "known bugs"
- partners can still import data into `https://pre.sshade.eu`

After delivery

- intensive test period by all partners
- bugs will be reported on a dedicated tool

Questions?