universitätbonn

Web-based ontology analysis and partitioning tool

Ahmad Alzeitoun
Imuwahen Osazuwa
Iulia Buga

Supervised by:
Dr. Gökhan Coskun and Irlan Grangel

universitätbonn

Agenda

- Objective
- Organization
- Solution
- Challenges
- Demo
- Q&A

universitätbonn

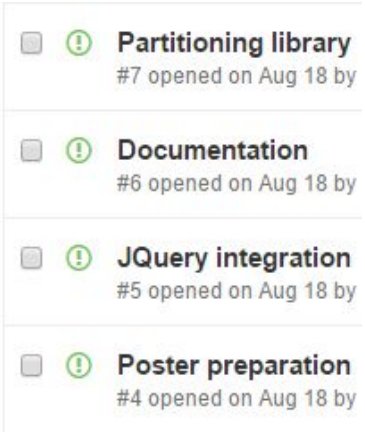
Objective

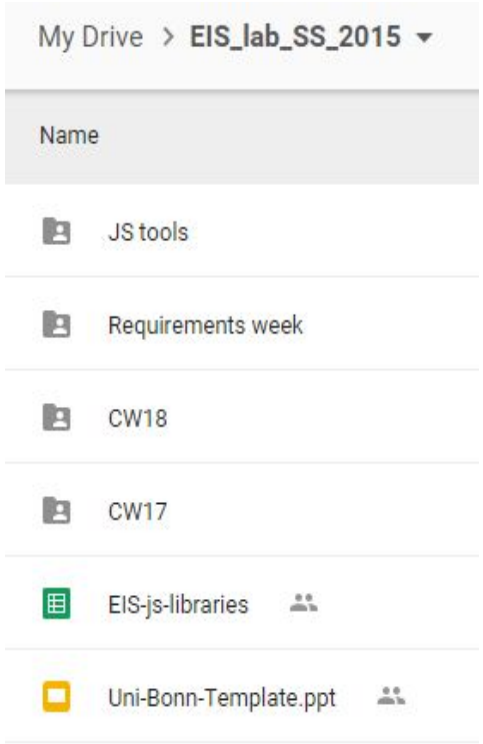
- Decompose large ontologies into smaller modules.
- Support the comprehension of an existing ontology and improve the process of ontology (modular) reuse and interlink.

Organization

- **Weekly*** team meetings
  - with mentors on Wednesdays
  - with team on Sundays

- **Github** for managing code/tasks
- **Google Drive** for managing files





universitätbonn

Solution

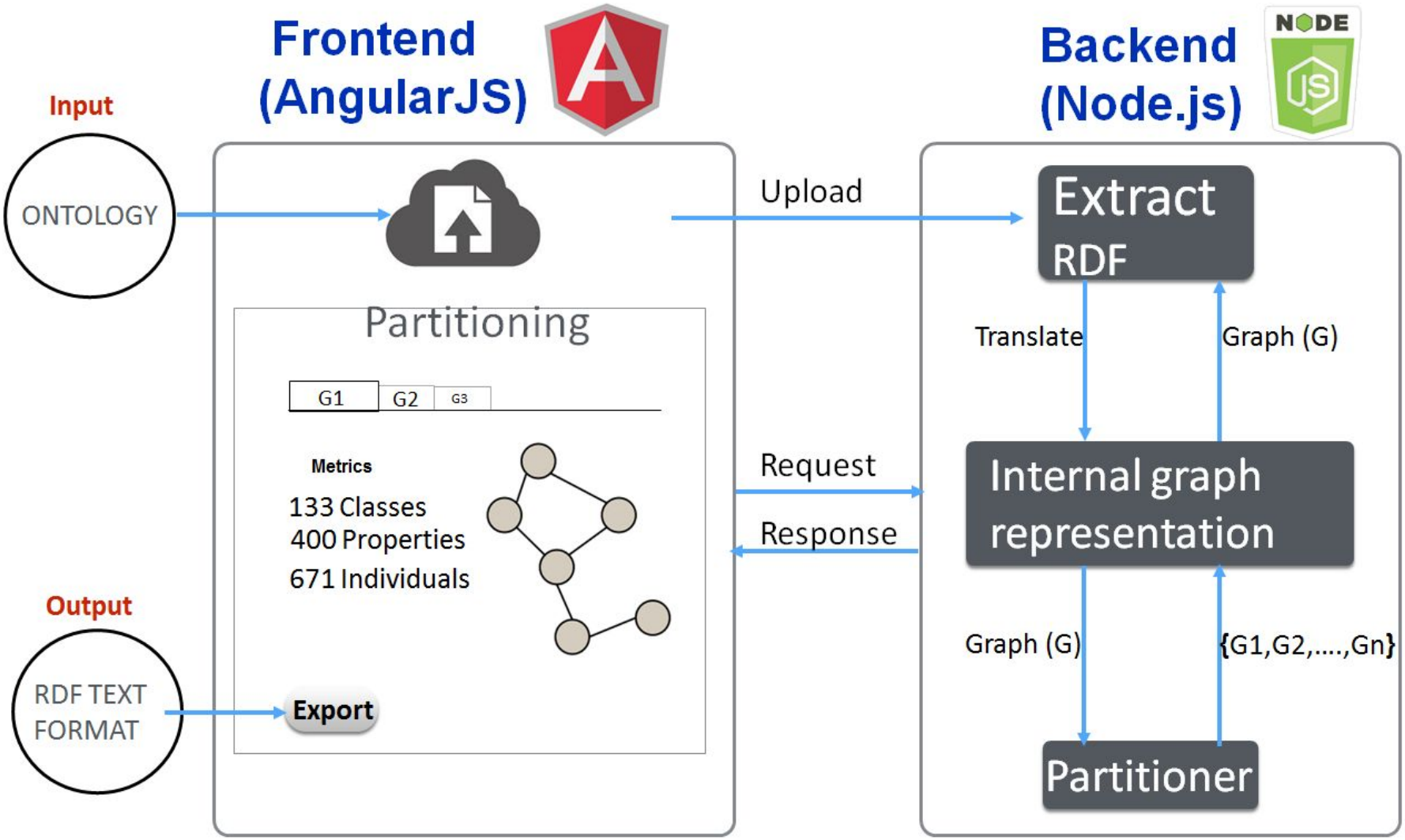
- Programming language: Javascript
- Environment: Windows, Mac Os
- Tools: SublimeText, Atom, GitSCM

universitätbonn

Tasks

1. Technology stack
2. Upload
3. Parse
4. Metrics
5. Filtering
6. Visualize and modify
7. Partition and export
8. Document
9. Test
10. Research

Application flow



Technology stack

- MEAN
  - AngularJS
  - ExpressJS
  - Node.js
- Package managers:
  - Bower (frontend)
  - npm (backend)
- Libraries
  - RdfStore (backend)
  - VisJS (frontend)

Image credit: http://adrianmejia.com/blog/2014/10/03/mean-stack-tutorial-mongodb-expressjs-angularjs-nodejs/

universitätbonn

Upload

Local upload

- OWL file as input
- processed by NodeJS and RDFStore
- dedicated folder: *uploads*

URL upload

- custom code to add bonus metrics
- processed by NodeJS and RDFStore
- *no storage*

Output: triples (subject, predicate, object)

universitätbonn

Parse

OWL2VOWL

- local/URI upload
- generate JSON in VOWL format

RDFStore

- local/URI upload
- generate custom JSON of edges, nodes, graph

Output: REST endpoints

- nodes


```
[{"id":"family-ontology","label":"family-ontology"},{"id":"Alice","label":"Alice"},{"id":"Bob","label":"Bob"},
{"id":"Mary","label":"Mary"},{"id":"Ontology","label":"Ontology"},{"id":"ObjectProperty","label":"ObjectProperty"},
```


- edges


```
[{"from":"family-ontology","to":"Ontology","label":"rdf:type","arrows":"from","filter":["resource"]},
{"from":"hasChild","to":"ObjectProperty","label":"rdf:type","arrows":"from","filter":["object-property","resource"]},
{"from":"_:7","to":"Restriction","label":"rdf:type","arrows":"from","filter":["resource"]},
{"from":"Person","to":"Class","label":"rdf:type","arrows":"from","filter":["class","resource"]},
```

universitätbonn

Metrics

OWL2VOWL

- already available
- generate VOWL JSON

VOWL	
Name	**Count**
Class	3
Datatype	0
Object	1
Datatype property	0
Property	1
Axioms	15

RDFStore

- not available
- custom code to add bonus metrics

Rdfstore	
Blank node	2
Literals	2

universitätbonn

Filtering

- generate input for visualization
- process the nodes and edges
- add a filter to each edge
- Implementation:
  - Node.js using RDFStore

universitätbonn

Visualization

VisJS

- highlight
  - color
  - shape
- metrics
- add edit node
- add edge
- save
- export as image

universitätbonn

Partitioning

- Constructing weighted matrix
- Weight Normalization
- Neighborhood random walk distance method
- Silhouette: criterion function
- Agglomerative algorithm

universitätbonn

Export

Triplles generation:

<http://example.org/#spiderman> <http://www.perceive.net/schemas/relationship/enemyOf> <http://example.org/#green-goblin> .

Export as text file:

universitätbonn

Results

universitätbonn

Challenges

- technology stack: setting up the stack
- AngularJS and Node.js connection
- javascript libraries shortcomings
- converting JSON to OWL

universitätbonn

Q&A

Image credit: http://blog.mypermissions.com/new-study-shows-comprehension-rates-for-tos-of-googlefacebook-is-less-than-40/