

A Linked Data approach to surfacing 101's software chrestomathy

Ralf Lämmel (Software Languages Team, University of Koblenz-Landau) on behalf of the 101companies contributors. Acknowledgements: Kevin Klein, Martin Leinberger, Thomas Schmorleiz, and Andrei Varanovich.

<http://softlang.uni-koblenz.de/101explorer/>

Basic Linked Data principles

- Use URIs as names for 'things'.
- Use HTTP URIs to enable lookup of names.
- Provide useful information in the HTTP response.
- Use standards for responses (RDF, ...).
- Provide standardized query endpoints (SPARQL, ...).
- Include URIs into response to cater resource recovery.

Linked Data accessors for 101

101explorer
101wiki's endpoint
101wiki's triplestore
Dumps
Derived resources

101explorer

Summary

Headline
The namespace for namespaces used on this wiki

Name Namespace

Namespace Namespace

Classifier Namespace

Links

[Github Link](#)

[Wiki Link](#)

[Sesame Link](#)

[Endpoint Link](#)

Parts

Members

- [101](#)
- [Concept](#)
- [Contribution](#)
- [Contributor](#)
- [Course](#)
- [Dotnet](#)
- [Feature](#)
- [Information](#)
- [Java](#)
- [Language](#)
- [Module](#)
- [Programming language](#)
- [Property](#)
- [Resource](#)
- [Script](#)
- [Service](#)
- [Technology](#)
- [Theme](#)
- [Vocabulary](#)

[What's the 101explorer?](#)

Switch from HTML format to [JSON format](#) or [RDF format](#).

Powered by [Twitter Bootstrap](#), [JQuery](#) and [Masonry](#)

101explorer

<http://101companies.org/resources?format=json>

```
{
  wiki: "http://101companies.org/wiki/namespace:Namespace",
  github: "https://github.com/101companies/101repo/tree/master/",
  name: "Namespace",
  headline: "The namespace for namespaces used on this wiki",
  namespace: "Namespace",
  sesame: "http://triples.101companies.org/openrdf-workbench/repositories/wiki101/explore?resource=%3Chttp%3A%2F%2F101companies.org%2Fresource%2FNamespace-3ANamespace%3E",
  endpoint: "http://101companies.org/endpoint/namespace:Namespace/json",
  - members: [
    - {
      resource: "http://101companies.org/resources/101",
      classifier: "Namespace",
      name: "101"
    },
    - {
      resource: "http://101companies.org/resources/concepts",
      classifier: "Namespace",
      name: "Concept"
    },
    - {
      resource: "http://101companies.org/resources/contributions",
      classifier: "Namespace",
      name: "Contribution"
    }
  ]
}
```

wiki

101explorer

<http://101companies.org/resources?format=rdf>

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:co="http://101companies.org/property/">
<rdf:Description rdf:about="http://101companies.org/resources?format=rdf">
<rdf:type rdf:resource="http://101companies.org/property/namespace"/>
<co:name>Namespace</co:name>
<co:namespace>Namespace</co:namespace>
<co:classifier>Namespace</co:classifier>
<co:member rdf:resource="http://101companies.org/resources/101?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/concepts?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/contributions?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/contributors?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/courses?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/features?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/information?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/languages?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/modules?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/properties?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/resources?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/scripts?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/services?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/technologies?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/themes?format=rdf"/>
<co:member rdf:resource="http://101companies.org/resources/vocabularies?format=rdf"/>
<co:wiki>http://101companies.org/wiki/namespace:namespace</co:wiki>
<co:github>https://github.com/101companies/101repo/tree/master/</co:github>
<co:sesame>http://triples.101companies.org/openrdf-workbench/repositories/wiki101/explore?
resource=%3Chttp%3A%2F%2F101companies.org%2Fresource%2Fnamespace-3Anamespace%3E</co:sesame>
<co:endpoint>http://101companies.org/endpoint/namespace:namespace/json</co:endpoint>
</rdf:Description>
</rdf:RDF>
```

101explorer

<http://101companies.org/endpoint/Contribution:prologStarter/json>

101companies.org/endpoint/Contribution:prologStarter/json

```
[
- [
  "http://101companies.org/resources/contributions/prologStarter",
  "http://101companies.org/property/uses",
  "http://101companies.org/resources/languages/Prolog"
],
- [
  "http://101companies.org/resources/contributions/prologStarter",
  "http://101companies.org/property/developedBy",
  "http://101companies.org/resources/contributors/Ralf_Lämmel"
],
- [
  "http://101companies.org/resources/contributions/prologStarter",
  "http://101companies.org/property/instanceOf",
  "http://101companies.org/resources/namespaces/Contribution"
],
- [
  "http://101companies.org/resources/contributions/prologStarter",
  "http://101companies.org/property/mentions",
  "http://101companies.org/resources/languages/Prolog"
]
]
```

[0]

resources (2) resources (1)

101wiki endpoint

<http://triples.101companies.org/openrdf-workbench/repositories/wiki2/query>

Let's enter this query:

```
PREFIX concept: <http://101companies.org/resources/concepts/>
SELECT *
WHERE {
?s ?p concept:Model.
}
```

101wiki triplestore

Sesame server

Repositories

New repository

Delete repository

Explore

Summary

Namespaces

Contexts

Types

Explore

Query

Export

Modify

SPARQL Update

Add

Remove

Clear

System

Information

Current Selections:

Sesame server: <http://triples.101companies.org/openrdf-sesame> [change]

Repository: [wiki2 \(wiki2 \)](#) [change]

Query Result (23)

Limit results:

S	P
http://101companies.org/resources/101s/%40contribution	http://101companies.org/property/mentions
http://101companies.org/resources/concepts/Software_design	http://101companies.org/property/mentions
http://101companies.org/resources/concepts/Metamodel	http://101companies.org/property/mentions
http://101companies.org/resources/themes/Java_mapping	http://101companies.org/property/mentions
http://101companies.org/resources/contributions/ArgoUML	http://101companies.org/property/instanceOf
http://101companies.org/resources/contributions/Atl	http://101companies.org/property/mentions
http://101companies.org/resources/languages/XMI	http://101companies.org/property/mentions
http://101companies.org/resources/contributions/AtlPluginUsage	http://101companies.org/property/mentions
http://101companies.org/resources/contributions/EmfGenerative	http://101companies.org/property/mentions
http://101companies.org/resources/contributions/EmfReflexive	http://101companies.org/property/mentions
http://101companies.org/resources/contributions/Gra2mol	http://101companies.org/property/mentions
http://101companies.org/resources/contributions/Xtext	http://101companies.org/property/mentions

101wiki triplestore

<http://101companies.org/resources/contributions/antlrObjects/src/main/java/org/softlang/company/model/Company.java?format=html>

Derived files

- [Company.java.commitInfo.json](#)
- [Extracted Facts](#)
 - Produced by [extractFacts](#)
- [Company.java.fragments.json](#)
- [Company.java.geshi.html](#)
- [Matches](#)
 - Produced by [matches101meta](#)
- [Metrics](#)
 - Produced by [extractTokens](#)
- [Predicate Matches](#)
 - Produced by [matchImports](#)
- [Refined Tokens](#)
 - Produced by [analyzeTokens](#)
- [Company.java.summary.json](#)
- [Tokens](#)
 - Produced by [extractTokens](#)
- [Company.java.validator.json](#)

Derivatives

Linked Data scenarios in software (language) engineering

Scenario I: Code-sharing management

Find all equal (or similar) files
across the different contributions
in 101repo.

Some groups of perfect clones found

```
[
  dom/Company.xsd",
  jdom/Company.xsd",
  sax/Company.xsd",
  scalaXML/Company.xsd",
  xom/Company.xsd",
  xquery/Company.xsd",
  xslt/Company.xsd"
],
[
  haskellSyb/src/Company/Data.hs",
  monoidal/src/Company/Data.hs",
  nonmonadic/src/Company/Data.hs",
  writerMonad/src/Company/Data.hs"
],
[
  jaxbChoice/src/test/java/org/softlang/tests/Operations.java",
  jaxbExtension/src/test/java/org/softlang/tests/Operations.java",
  jaxbSubstitution/src/test/java/org/softlang/tests/Operations.java"
]
```

Code-sharing management

- Expressiveness
 - Access to full 101repo tree and content
- Basic Linked Data convenience
 - Abstraction from confederation of 101repo
 - Abstraction from file-system access

Clone detection

```
# Iterate over all contributions
filesList = []
root = 'http://101companies.org/resources/contributions'
contributions = loadPage(root)
for member in contributions['members']:
    filesList += extractFilesFromFolder(member)

# Hash map content to file URIs
contents = {}
for file in filesList:
    content = file['data']['content']
    if not content in contents:
        contents[content] = []
    contents[content].append(file['uri'])
```

I/II

.py

Clone detection

```
# Collect files and content recursively
def extractFilesFromFolder(folder):
    files = []
    data = loadPage(folder['resource'])
    for file in data.get('files', []):
        fileData = loadPage(file['resource'])
        fileData = fileData.get('content')
        files.append({
            'uri' : file['resource'],
            'data': fileData
        })
    for folder in data.get('folders', []):
        files += extractFilesFromFolder(folder)
    return files
```

II/II

.py

Misc details

- Working code available:

<https://github.com/101companies/101worker/tree/master/modules/cloneDetection>

- Code deployed on 101worker

- Results deployed data.101companies.org:

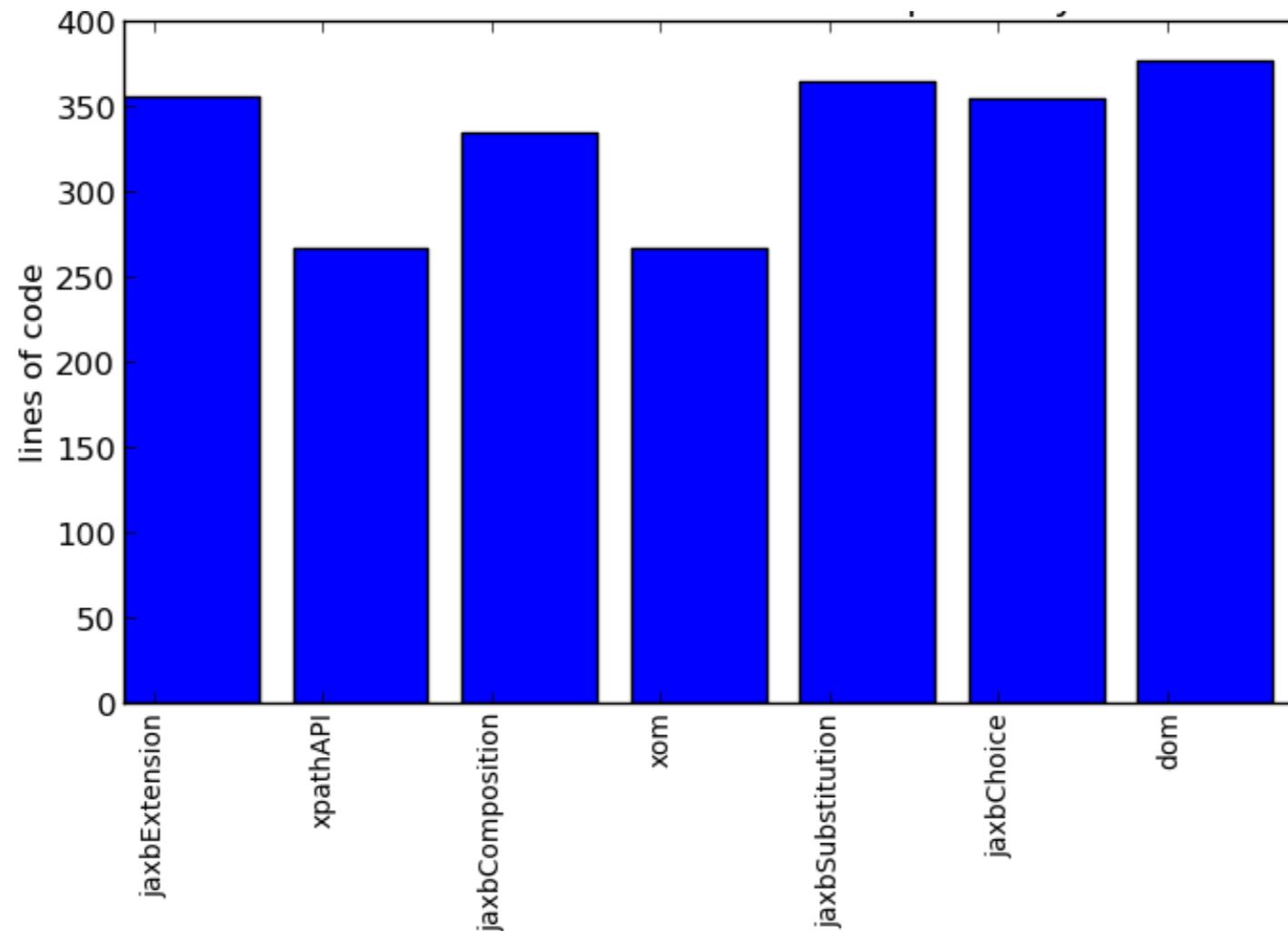
<http://data.101companies.org/views/clones.json>

Scenario II: Metrics-based comparison

Compare contributions per
feature set by (NC)LOC metric.

A feature set with contributions and (NC)LOC

```
[  
  "http://101companies.org/resources/features/Hierarchical_company",  
  "http://101companies.org/resources/features/Mapping",  
  "http://101companies.org/resources/features/Open serialization",  
  "http://101companies.org/resources/features/Total",  
  "http://101companies.org/resources/features/Cut"  
]
```



Metrics-based comparison

- Expressiveness
 - Access to full 101repo tree and (NC)LOC
 - Access to feature sets per contribution
- Additional Linked Data convenience
 - Access to feature sets via linked query endpoint
 - Access to (NC)LOC via associated derived resource

Metrics-based comparison

```
configs = {} # Associate feature configurations with contributions
metrics = {} # Associate contributions with LOC metric

# Iterate over all contributions
contribs = loadPage('http://101companies.org/resources/contributions')
for contrib in contribs['members']:
    data = loadPage(contrib['resource'])

    # Collect features for contribution
    features = retrieveFeatures(data['triplestore'])
    key = tuple(features)
    if not key in configs:
        configs[key] = {'features': features, 'contribs': []}

    # Map feature configuration to contribution
    configs[key]['contribs'].append(contrib['name'])

    # Aggregate LOC for all files of the contribution
    files = collectFiles(contrib['resource'])
    loc = 0
    for file in files:
        mdata = retrieveMetrics(file['resource'])
        if not mdata == {}:
            if not 'relevance' in mdata or mdata['relevance'] == 'system':
                loc += int(mdata['loc'])
    metrics[contribution['name']] = loc
```

I/II

.py

Metrics-based comparison

```
# Retrieve features for a contribution
def retrieveFeatures(url):
    triples = loadPage(url)
    features = []
    for triple in triples: # Filter triples
        predicate = triple[1]
        object = triple[2]
        if predicate == 'http://101companies.org/property/implements':
            features.append(object.replace(
                'http://101companies.org/resources/features/', ''))
    return features

# Retrieve metrics for a file
def retrieveMetrics(uri):
    file = loadPage(uri)
    derivatives = file['derivatives']
    for derivative in derivatives: # Find associated metrics file
        if derivative['name'].endswith('metrics.json'):
            return loadPage(derivative['resource'])
    return {}

# Collect files and metrics in a folder recursively
def collectFiles(uri):
    folder = loadPage(uri)
    files = folder['files']:
    for subfolder in folder['folders']:
        files += collectFiles(subfolder['resource'])
    return files
```

II/II

.py

Misc details

- Working code available:

<https://github.com/101companies/101worker/tree/master/modules/featureMetrics>

- Code deployed on 101worker

- Results deployed data.101companies.org:

<http://data.101companies.org/views/features>

Scenario III: Concept analysis

Infer OOP vs. FP concepts by means of their association with corresponding contributions.

Concepts associated with FP and OOP

Functional programming

Concept	#Occs	Unique
GUI	12	false
Class	10	false
101implementation	8	false
Server	8	false
Zipper	8	false
Library	8	false
Functional programming	8	false
Float	8	false
String	8	false
Pure function	8	false
Algebraic data type	7	true

OO programming

Concept	#Occs	Unique
GUI	16	false
Class	16	false
MVC	14	false
POJO	14	true
Metamodel	13	true
101implementation	12	false
Client	12	false
Model	11	true
Server	10	false
Annotation	8	false
OO programming	8	true

Concept analysis

- Expressiveness
 - Full graph-based queries over 101wiki
- Additional Linked Data convenience
 - Abstraction from 101wiki implementation

Concept analysis

```
static final String resources = 'http://101companies.org/resources/'
static final String properties = 'http://101companies.org/property/'

public findConcepts(paradigm) {

    def concept = getResource(resources + 'namespaces/Concept')
    def concepts = graph.v(paradigm).
        inE(properties + 'instanceOf').outV. // Languages
        inE(properties + 'uses').outV. // Contributions
        outE(properties + 'mentions').inV. // Mentions
        toList().findAll{ // Concept mentions only
            it.outE(properties + 'instanceOf').inV.
            filter{it == concept}.toList().size() > 0
        }
    return concepts
}
```

Misc details

- Working code available:

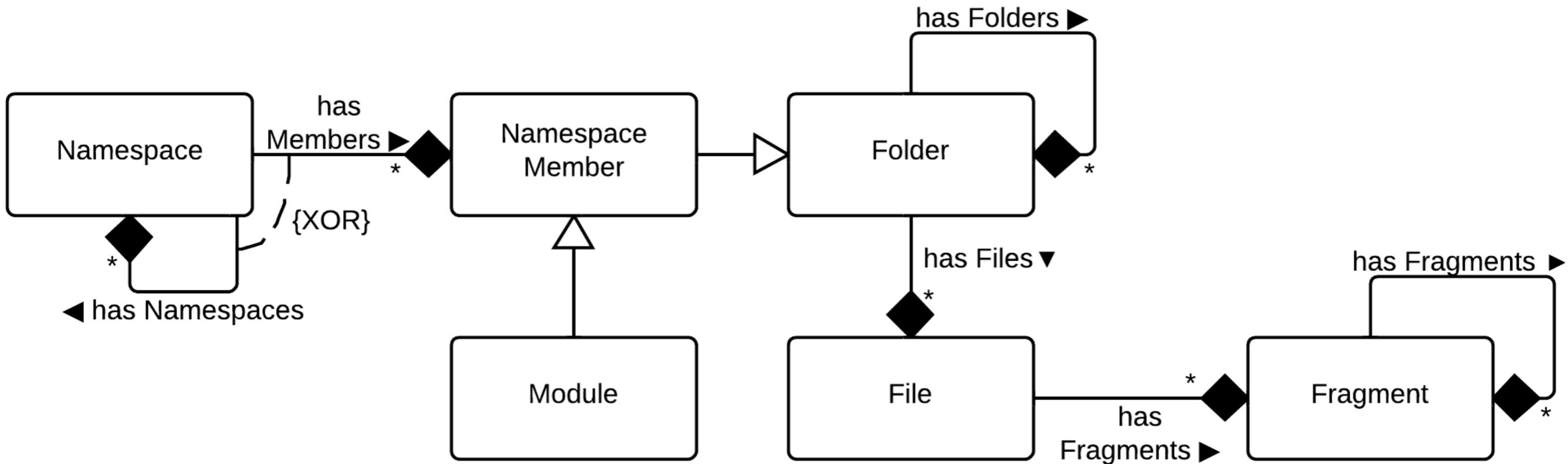
https://github.com/avaranovich/linked_data_experiments/blob/master/src/main/groovy/Main.groovy

- Code not deployed on 101
- Results only computed locally

A heterogeneous schema for 101's Linked Data

https://github.com/101companies/101worker/tree/master/services_temp/discovery/schemas

UML-based metamodel of the 10l repo (summary)



JSON schema for fragments

```
{
  "title": "Fragment schema",
  "type": "object",
  "properties": {
    "name": { "type": "string" },
    "namespace": { "type": "string" },
    "headline": { "type": "string" },
    "wiki": { "type": "url" },
    "github": { "type": "url" },
    "triplestore": { "type": "url" },
    "classifier": { "type": "string" },
    "language": { "type": "string" },
    "content": { "type": "string" },
    "fragments": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "resource": { "type": "resource" },
          "classifier": { "type": "string" },
          "name": { "type": "string" },
        }
      }
    }
  }
}
```

Likewise for
folders and files.

JSON schema for module descriptions

```
{
  "title": "Module description schema",
  "type" : "object",
  "properties": {
    "derivatives" : {
      "type" : "array",
      "items": {
        "type": "object",
        "properties": {
          "headline" : { "type": "string" },
          "scope" : { "enum": ["file", "folder", "dump"] },
          "suffix" : { "type": "string" },
          "filename": { "type": "string" },
          "language" : { "enum": ["JSON", "RDF"] }
        }
      }
    }
  }
}
```

<https://github.com/101companies/101worker/tree/master/modules>

Example of a module description

```
{
  "derivatives" : [
    {
      "headline" : "Metrics",
      "scope" : "file",
      "suffix" : ".metrics.json",
      "language" : "JSON"
    },
    {
      "headline" : "Tokens",
      "scope" : "file",
      "suffix" : ".tokens.json",
      "language" : "JSON"
    }
  ]
}
```

<https://raw.githubusercontent.com/101companies/101worker/master/modules/metrics101meta/module.json>

RDF schema for 101wiki's triplestore

```
<rdf:RDF xmlns:rdf="..." xmlns:rdfs="...">

  <rdfs:Class rdf:ID="http://101companies.org/schemas/wiki#Namespace"/>
  <rdfs:Class rdf:ID="http://101companies.org/schemas/wiki#Language"/>
  <rdfs:Class rdf:ID="http://101companies.org/schemas/wiki#Technology"/>
  <!-- Further classes (namespaces) omitted. -->

  <rdf:Property rdf:about="http://101companies.org/schemas/wiki#uses">
    <rdfs:range rdf:resource="http://101companies.org/schemas/wiki#Technology"/>
    <rdfs:range rdf:resource="http://101companies.org/schemas/wiki#Language"/>
    <rdfs:domain rdf:resource="http://101companies.org/schemas/wiki#Contribution"/>
  </rdf:Property>

  <rdf:Property rdf:about="http://101companies.org/schemas/wiki#implements">
    <rdfs:range rdf:resource="http://101companies.org/schemas/wiki#Feature"/>
    <rdfs:domain rdf:resource="http://101companies.org/schemas/wiki#Contribution"/>
  </rdf:Property>

  <!-- Further properties omitted. -->

</rdf:RDF>
```

Meaning of predicates

Predicate	Meaning	# Triples
uses	A resource uses a language or technology.	565
implements	A contribution implements a feature.	628
instanceOf	'instance of' relationship	1578
isA	'is-a' relationship on concepts	84
developedBy	A contribution is developed by a contributor.	191
reviewedBy	A contribution is reviewed by a contributor.	16
relatesTo	A resource relates to (ontological) to another resource.	80
mentions	A resource mentions another resource (weak internal link).	4717

Please start exploring now!

- <http://101companies.org/> Landing page
- <http://101companies.org/wiki> Wiki
- <http://101companies.org/resources?format=html> Resource explorer
- <http://data.101companies.org/dumps/> Dumps
- <http://data.101companies.org/resources/> Derived resources

Exercises -- Proposals

Determine average and median of file counts for contributions.

- Rationale: the idea is to get a feeling to the size of contributions: the scale, the average, the median. Contributions are tiny systems!
- Possible enhancements
 - Plot distribution of file counts.
 - Count only files with "system relevance".

List features with missing implementations per language

- Rationale: The idea is to favor (by sorting or otherwise) “popular languages” (i.e., languages with a noteworthy number of contributions) and to show unimplemented features for those.
- Possible enhancements
 - Plot distribution of file counts.
 - Count only files with “system relevance”.

Hands on

Search the 101wiki for your favorite language.

- Go here: <http://101companies.org/wiki/>
- Enter "Haskell" in Search field.

Study the metadata of an interesting Java file.

- Go here: <http://101companies.org/resources/contributions/antlrObjects/src/main/java/org/softlang/company/features/Parsing.java?format=html>
- Select "Predicate Matches": <http://data.101companies.org/resources/contributions/antlrObjects/src/main/java/org/softlang/company/features/Parsing.java.predicates.json>

Review the triples for a software technology.

- Go here: <http://101companies.org/resources/technologies/Ruby%20on%20Rails?format=html>
- Select "Endpoint": <http://101companies.org/endpoint/Technology:Ruby%20on%20Rails/json>

What's next?

- Let's exercise SHARE and 101 some more.
- Let's discuss (SE) Research 2.0:
 - Integration of SHARE and 101.
 - ... and megaplanet.
 - Contributions to S(L)EBOK.
 - Other forms of R2.0.
- What are you thoughts?