ORR / COR
Developer Workshop

Carlos Rueda    –   Monterey Bay Aquarium Research Institute
Beth Huffer    –   Lingua Logica LLC
Lewis McGibbney  –   NASA JPL
ORR - Ontology Registry and Repository

Open source software to facilitate semantic interoperability. Developed by the Marine Metadata Interoperability (MMI) Project, and upgraded via the ‘Cross-Domain Observational Metadata for Environmental Sensing (X-DOMES).

COR - Community Ontology Repository

ESIP’s ORR instance, deployed on ESIP host at https://cor.esipfed.org
Outline of Workshop

- Workshop objectives
- Future Work
- What's the ORR?  Brief History, Who’s using it, ESIP Evaluation
- ORR: Linked Open Data In Practice
  - COR and SWEET
- ORR Development
  - Backend, Frontend, and Integrated ORR system
  - ORR Documentation
- Installing the ORR
Workshop Objectives

- Learn about the ORR in general and the COR instance in particular
  - What they are and what they can do for semantic interoperability
  - Concrete uses including support for SWEET development and hosting

- Familiarize with underlying ORR design and implementation
  - Architecture, standards, and technologies used
  - Git repositories; development workflow
  - ORR and COR Status
    - So you can actively participate in further improvement of the system!

- Install an ORR instance on your laptop (using Docker)
Future Work

● Community
  ○ ORR Software
    ■ Grow a team of developers / maintainers
    ■ Define: Governance; Code of Conduct; Contributing guidelines
  ○ COR Instance and Content
    ■ Operators, Maintainers/Administrator
    ■ Curators, Users
  ○ Promote utilization

● ORR Software
  ○ Address observations from ESIP Evaluation Committee
  ○ Address other pending backend and front-end entries in the trackers
  ○ Improve documentation
  ○ Submit your ideas!
What’s the ORR

● An open-source project to enable repositories of controlled vocabularies and term mappings
● With:
  ○ Web resolvable identifiers for ontologies and terms
  ○ Versioning
  ○ REST API and SPARQL Endpoint
● To:
  ○ Facilitate semantic interoperability
  ○ Enable added-value applications with semantics and inference
ORR - Brief History

● 2004: Marine Metadata Interoperability project (MMI)

“Promoting the exchange, integration and use of marine data through enhanced data publishing, discovery, documentation and accessibility.”

John Graybeal – Luis Bermudez – and many contributors!

● 2008: ORR born as part of MMI’s vision for a Semantic Framework

● 2015: X-DOMES Project; ORR overhauled version 3

● 2016: ESIP Community Ontology Repository (COR)
Marine Metadata Interoperability Project (MMI)

Welcome to the Marine Metadata Interoperability Project

“Promoting the exchange, integration and use of marine data through enhanced data publishing, discovery, documentation and accessibility.”

Our goal is to support collaborative research in the marine science domain, by simplifying the incredibly complex world of metadata into specific, straightforward guidance. MMI encourages scientists and data managers at all levels to apply good metadata practices from the start of a project, by providing the best guidance and resources for data management, and developing advanced metadata tools and services needed by the community.

Posted December 7th, 2008 by graybear

MMI Project Status (Fall 2015)

What’s happening with the MMI project?

The MMI project was first funded by the National Science Foundation in 2004. Today it continues to provide guidance, vocabularies and semantic services, with
X-DOMES
Cross-Domain Observational Metadata for Environmental Sensing

- Tasked to develop prototypes for the application of standards-based description of environmental sensor metadata, including provenance
- The X-DOMES model incorporates semantic content harmonized with the ontologies proposed by the W3C Sensor Network Ontology and with the requirements of Linked Open Data

https://xdomes.org/
X-DOMES Tools

- **ORR instance**
  - To create and maintain fully described, resolvable semantic terms

- **SensorML Viewer/Editor**
  - User friendly technical “spec sheet” form
  - Creates validated OGC/SensorML documents
  - To be owned by sensor manufacturers, field operators and data managers

- **SensorML registry and repository (SRR)**
  - Enables documents to be referenced as URLs with version control and proper authoritative ownership

https://xdomes.org/
ORR deployments

- MMI ORR
  https://mmisw.org/ont

- ESIP COR
  http://cor.esipfed.org/ont/

- XDOMES ORR
  https://xdomes.org/ont/

- SensorML ORR
  http://sensorml.com/ont/
ORR version 3: 2016 major release, 2017 upgrades

- Enhanced user/organization/permission model
- Overhauled authentication mechanism
- Enhanced performance
- REST API
- MongoDB; AllegroGraph
- Backend: Scala; comprehensive tests; Travis CI; good coverage
- Front-end: JavaScript / AngularJS
- Docker images for streamlined installation

https://hub.docker.com/r/mmisw/orr
ESIP Deployment and Evaluation

- **2016**: Deployed to ESIP cloud as COR in 2016 for experimentation and test
- **2017**: Formal evaluation conducted by ESIP
  - ESIP Semantic Portal (based on NCBO BioPortal) was other repository evaluated
  - Feedback by three semantic experts on both systems, based on several use scenarios
- **Feedback Results for COR**
  - All review input was free text; we distilled 21 comments about COR
  - Issues: <5 highlighted key missing features, 2 reflected temporary outages
  - Features: Only 7 comments provided inputs which are arguably still actionable
  - **Biggest technical value**: finding places that UX (either UI or documentation) was weak
- **In the end, ESIP leadership determined COR would be primary repository**
  - Decision took into account formal evaluation results, likely costs, and the value proposition of each system
  - Business discussions continue, including at today’s 1:30 meeting
O R R
Ontology Registry and Repository

Linked Open Data In Practice
Is your data 5 Star?

- Star: Available on the web (whatever format), but with an open license
- Double Star: Available as machine-readable structured data (e.g., excel instead of an image scan)
- Triple Star: As before, but using a non-proprietary format (e.g., CSV instead of excel)
- Quadruple Star: All the above, plus use open standards (RDF & Co.) to identify things, so that people could point at your stuff
- Quintuple Star: All the above, plus link your data to other people’s data to provide context

From https://www.w3.org/People/Ivan/CorePresentations/SWTutorial/Slides.pdf
The Rules of Linked Data

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names
- When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
- Include links to other URIs, so that they can discover more things

See also: “W3C Linked Data Platform” Spec – https://www.w3.org/TR/ldp/
ORR: Linked Open Data In Practice

- Creating Vocabularies
- Creating Term Mappings
- Uploading Ontologies
- Searching Ontologies
- Searching Terms
- Advanced Search – SPARQL
- Admin Features
Capturing Knowledge
... and making it machine-actionable

- Knowledge expressed as *statements*
- Statements modeled as *triples*:

```
Subject - Predicate - Object
```

- Example:

```
Plasma - Is a - State of Matter
```
RDF: Resource Description Framework

- W3C standard to express information about resources
- Anything can be a resource, including physical things, documents, abstract concepts, numbers and strings
- The triple components denote resources:
Resources

- Resources are denoted by **IRIs** and **literals**
- IRI = Internationalized Resource Identifier
- Literals denote values according to known data types (numbers, strings, dates, ...)

So, the example:

- becomes:

```
```
Capturing RDF triple data

<table>
<thead>
<tr>
<th>subject</th>
<th>predicate</th>
<th>object</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>p</td>
<td>j</td>
</tr>
<tr>
<td>a</td>
<td>q</td>
<td>k</td>
</tr>
<tr>
<td>b</td>
<td>r</td>
<td>m</td>
</tr>
<tr>
<td>c</td>
<td>p</td>
<td>j</td>
</tr>
<tr>
<td>c</td>
<td>p</td>
<td>w</td>
</tr>
<tr>
<td>d</td>
<td>t</td>
<td>a</td>
</tr>
</tbody>
</table>

**Triples**

- \((a, p, j)\)
- \((a, q, k)\)
- \((b, r, m)\)
- \((c, p, j)\)
- \((c, p, w)\)
- \((d, t, a)\)
Capturing RDF triple data

<table>
<thead>
<tr>
<th>subjects</th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>y</td>
<td>D</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>w</td>
<td></td>
<td></td>
<td>K</td>
</tr>
</tbody>
</table>

**Triples**

- (x, p1, A)
- (x, p2, B)
- (x, p3, C)
- (y, p1, D)
- (y, p2, E)
- (w, p3, K)
### Querying RDF triple data
(Conceptually)

<table>
<thead>
<tr>
<th>Triple store</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x, p1, A)</td>
</tr>
<tr>
<td>(x, p2, B)</td>
</tr>
<tr>
<td>(x, p3, C)</td>
</tr>
<tr>
<td>(y, p1, D)</td>
</tr>
<tr>
<td>(y, p2, E)</td>
</tr>
<tr>
<td>(w, p3, K)</td>
</tr>
</tbody>
</table>

**Queries:**
- (y, p1, ?)
- (x, ?, ?)
- (?, ?, K)

**Responses:**
- (y, p1, D)
- (x, p1, A)
- (x, p2, B)
- (x, p3, C)
- (w, p3, K)
Vocabularies

- Referring to particular subjects, predicates and objects in triples means we are dealing with vocabularies
- That is, naming things and using names introduced by others
  - This ‘SST’ dataset was produced by organization ‘MBARI’
- Should be controlled vocabularies
  - With names (and associated definitions/attributes) agreed by the community
  - To reduce discrepancies
  - To facilitate data discovery, reuse, and integration
  - To enable crosswalks/mappings
  - In short, to promote and facilitate interoperability
- Standard vocabularies: RDF, RDFS, SKOS, OWL, DCT, Schema{.rdfs}.org, ...
ORR Vocabularies

- ORR enables this:

<table>
<thead>
<tr>
<th>subjects</th>
<th>predicates</th>
<th>objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>p1</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>p2</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>p3</td>
<td>C</td>
</tr>
<tr>
<td>y</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>w</td>
<td>K</td>
<td></td>
</tr>
</tbody>
</table>

Triples:
- (x, p1, A)
- (x, p2, B)
- (x, p3, C)
- (y, p1, D)
- (y, p2, E)
- (w, p3, K)
## Observable Properties

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>id</td>
<td><strong>Definition</strong></td>
</tr>
<tr>
<td>seaWaterTemperature</td>
<td>seaWaterTemperature</td>
<td>heat energy of sea water</td>
</tr>
<tr>
<td>seaWaterPressure</td>
<td>seaWaterPressure</td>
<td>forces on a particular point including sea water and air pressure</td>
</tr>
<tr>
<td>volumetricBackscatter</td>
<td>volumetricBackscatter</td>
<td>portion of an acoustic signal that is returned by hitting a particle</td>
</tr>
<tr>
<td>particleMotion</td>
<td>particleMotion</td>
<td>three dimensional movement of a particle in a fluid</td>
</tr>
<tr>
<td>conductivity</td>
<td>cond</td>
<td>a measure of a water sample volume's ability to conduct an electric current</td>
</tr>
<tr>
<td>temperature</td>
<td>temp</td>
<td>heat energy</td>
</tr>
<tr>
<td>windSpeed</td>
<td>windSpeed</td>
<td>speed of wind</td>
</tr>
<tr>
<td>windDir</td>
<td>windDir</td>
<td>the direction of a wind vector</td>
</tr>
<tr>
<td>windVelocity</td>
<td>windVelocity</td>
<td>speed of wind</td>
</tr>
<tr>
<td>waterVelocity</td>
<td>waterVelocity</td>
<td></td>
</tr>
<tr>
<td>airPressure</td>
<td>Press</td>
<td>airPressure is the pressure exerted by the weight of air in the atmosphere</td>
</tr>
<tr>
<td>barometricPressure</td>
<td>Bpress</td>
<td>Barometric pressure is the pressure exerted by the weight of air in the atmosphere</td>
</tr>
<tr>
<td>humidity</td>
<td>humidity</td>
<td>Humidity is the presence of water vapour in air</td>
</tr>
<tr>
<td>airTemperature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>waterDepth</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[https://docs.google.com/spreadsheets/d/1PEZUVexD9tDIIA-quPmYu_kMZIEVqsbNO_201huYtec/edit#gid=0](https://docs.google.com/spreadsheets/d/1PEZUVexD9tDIIA-quPmYu_kMZIEVqsbNO_201huYtec/edit#gid=0)
... to Linked Open Data!

http://cor.esipfed.org/ont/testorg/obsprop

Observable Properties  Version: 2018/01/07/205013 - Author: Carlos Rueda Owner: testorg

<table>
<thead>
<tr>
<th>Metadata details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: Observable Properties</td>
</tr>
<tr>
<td>Description: ESIP-WM18 demo</td>
</tr>
<tr>
<td>Ontology Creator: Carlos Rueda</td>
</tr>
</tbody>
</table>

### Data

<table>
<thead>
<tr>
<th>ShortName</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>seaWaterTemperature</td>
<td>heat energy of sea water</td>
</tr>
<tr>
<td>seaWaterPressure</td>
<td>forces on a particular point including sea water and air pressure</td>
</tr>
<tr>
<td>volumetricBackscatter</td>
<td>portion of an acoustic signal that is returned by hitting a particle</td>
</tr>
<tr>
<td>particleMotion</td>
<td>three dimensional movement of a particle in a fluid</td>
</tr>
<tr>
<td>conductivity</td>
<td>a measure of a water sample volume's ability to conduct an electric currier</td>
</tr>
<tr>
<td>temperature</td>
<td>heat energy</td>
</tr>
<tr>
<td>windSpeed</td>
<td>speed of wind</td>
</tr>
<tr>
<td>windDir</td>
<td>the direction of a wind or water</td>
</tr>
</tbody>
</table>
ORR Term Mappings

- ORR enables this:

<table>
<thead>
<tr>
<th>subject</th>
<th>predicate</th>
<th>object</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>p</td>
<td>j</td>
</tr>
<tr>
<td>a</td>
<td>q</td>
<td>k</td>
</tr>
<tr>
<td>b</td>
<td>r</td>
<td>m</td>
</tr>
<tr>
<td>c</td>
<td>p</td>
<td>j</td>
</tr>
<tr>
<td>c</td>
<td>p</td>
<td>w</td>
</tr>
<tr>
<td>d</td>
<td>t</td>
<td>a</td>
</tr>
</tbody>
</table>

- Using a given set of standard predicates (SKOS *match; OWL sameAs)
- NOTE: Not automated mapping: User explicitly indicates the mappings
ORR Term Mappings - Predicates

- SKOS - Simple Knowledge Organization System
  [https://www.w3.org/TR/skos-reference/#mapping](https://www.w3.org/TR/skos-reference/#mapping)
  - exactMatch
  - closeMatch
  - broadMatch
  - narrowMatch
  - relatedMatch

- OWL
  - sameAs
Uploading Ontologies

- **File formats recognized by the ORR:** RDF/XML, JSON-LD, Notation3, N-Triples, Turtle, OWL/XML, RDF/JSON
- **You can upload:**
  - A local file
  - A remote URL
Uploading Ontologies: Modes of Registration

- **Fully hosted**
  - ORR will “transfer” the original IRIs (of ontology itself and those of associated terms sharing the same namespace) to IRIs having the URL of the particular ORR instance as a prefix
  - All IRIs will be self-resolvable

- **Re-hosted**
  - No changes at all to the submitted ontology contents
  - IRI Resolvability: Via `iri` parameter
ORR: Three main ontology categories

Only for purposes of dispatch in the GUI

- ORR Vocabularies – Built with ORR Vocabulary tool
- ORR Mappings – Built with ORR Mapping tool
- Externally provided – Via Upload option
Searching Ontologies

- Global filter
## Searching Ontologies

- Global filter
- Facets

### Facets

<table>
<thead>
<tr>
<th>IRI</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://mmisw.org/ont/mmi/cfonmap">http://mmisw.org/ont/mmi/cfonmap</a></td>
<td>ORR-NVS CF standard name mapping</td>
</tr>
<tr>
<td><a href="http://mmisw.org/ont/cf/parameter">http://mmisw.org/ont/cf/parameter</a></td>
<td>Climate and Forecast (CF) Standard Name Mapping</td>
</tr>
<tr>
<td><a href="http://mmisw.org/ont/~ijaztk/patientcare">http://mmisw.org/ont/~ijaztk/patientcare</a></td>
<td>patientcareontology</td>
</tr>
<tr>
<td><a href="http://mmisw.org/ont/~ijaztk/smartoffice">http://mmisw.org/ont/~ijaztk/smartoffice</a></td>
<td>smartofficeont</td>
</tr>
<tr>
<td><a href="https://mmisw.org/ont/testing/SmartOfficeOntology">https://mmisw.org/ont/testing/SmartOfficeOntology</a></td>
<td>Smart Office Ontology</td>
</tr>
<tr>
<td><a href="http://www.w3.org/ns/san/">http://www.w3.org/ns/san/</a></td>
<td>Semantic Sensor Network Ontology</td>
</tr>
<tr>
<td><a href="http://www.dadfha.com/ont/hls">http://www.dadfha.com/ont/hls</a></td>
<td>Human Localization Sensor Ontology</td>
</tr>
<tr>
<td><a href="http://mmisw.org/ont/nasa_pace/taxonomy">http://mmisw.org/ont/nasa_pace/taxonomy</a></td>
<td>Species and Taxonomy</td>
</tr>
<tr>
<td><a href="http://mmisw.org/ont/nasa_pace/observation_mapping">http://mmisw.org/ont/nasa_pace/observation_mapping</a></td>
<td>Observation Mapping</td>
</tr>
<tr>
<td><a href="http://mmisw.org/ont/nasa_pace/environmental_mapping">http://mmisw.org/ont/nasa_pace/environmental_mapping</a></td>
<td>Environmental Mapping</td>
</tr>
<tr>
<td><a href="http://mmisw.org/ont/nasa_pace/environmental_interactions">http://mmisw.org/ont/nasa_pace/environmental_interactions</a></td>
<td>Environmental Interactions</td>
</tr>
<tr>
<td><a href="http://mmisw.org/ont/nasa_pace/pigment_mapping">http://mmisw.org/ont/nasa_pace/pigment_mapping</a></td>
<td>Pigment Mapping</td>
</tr>
</tbody>
</table>
Searching Terms

- String based matches against subjects and objects in triple store
Advanced Search – SPARQL

- ORR’s SPARQL endpoint provides a W3C SPARQL interface as the standards-based mechanism to search ORR's semantic content
- W3C SPARQL Overview: [https://www.w3.org/TR/sparql11-overview/](https://www.w3.org/TR/sparql11-overview/)
- ORR’s SPARQL endpoint powered by AllegroGraph
- ORR’s SPARQL GUI powered by YASGUI
- The URLs for COR:
  - SPARQL Endpoint: [http://cor.esipfed.org/sparql](http://cor.esipfed.org/sparql) (Note: not a GUI)
  - SPARQL GUI: [http://cor.esipfed.org/ont/sparql](http://cor.esipfed.org/ont/sparql)
Advanced Search – SPARQL

Remember this?

<table>
<thead>
<tr>
<th>Triple store</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x, p1, A)</td>
</tr>
<tr>
<td>(x, p2, B)</td>
</tr>
<tr>
<td>(x, p3, C)</td>
</tr>
<tr>
<td>(y, p1, D)</td>
</tr>
<tr>
<td>(y, p2, E)</td>
</tr>
<tr>
<td>(w, p3, K)</td>
</tr>
</tbody>
</table>

Queries:
(y, p1, ?)  
(x, ?, ?)   
(?, ?, K)

Responses:
(y, p1, D)  
(x, p1, A)  
(x, p2, B)  
(x, p3, C)  
(w, p3, K)
From query concept to SPARQL

(y, p1, ?) →

CONSTRUCT { y, p1, ?obj }
WHERE { y, p1, ?obj }

SELECT ?obj
WHERE { y, p1, ?obj }

(? , ? , K) →

CONSTRUCT { ?subj ?pred K }
WHERE { ?subj ?pred K }

SELECT ?subj ?pred
WHERE { ?subj ?pred K }
SPARQL - Examples

List all (OWL) classes defined in SWEET:  (in COR)

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?sweetClass
WHERE { ?sweetClass rdf:type owl:Class .
    FILTER regex(str(?sweetClass), "^http://sweetontology.net/.*") .
}
ORDER BY ?sweetClass
SPARQL - Examples

All states of matter according to SWEET:

Select (in COR)

SELECT ?s
WHERE { ?s a <http://sweetontology.net/statePhysical/StateOfMatter> }

Construct (in COR)

CONSTRUCT { ?s a <http://sweetontology.net/statePhysical/StateOfMatter> }
WHERE { ?s a <http://sweetontology.net/statePhysical/StateOfMatter> }

Convenient shorthand: a ≡ rdf:type
Find devices with temperature tolerance of at least -5 °C

PREFIX dvo: <http://mmisw.org/ont/mmi/device/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?device ?min ?max ?consequence
WHERE {
  ?toleranceLimit dvo:hasConsequence ?consequence .
  ?toleranceLimit dvo:hasTypedValue ?value .
  ?value dvo:hasValueType [ dvo:hasUnitsOfMeasure "°C"] .
  ?value dvo:hasMinimumRangeValue ?min .
  ?value dvo:hasMaximumRangeValue ?max .
  FILTER ( xsd:float(?min) <= -5 )
}
Admin Features

- All ontology entries are visible
- Can “unregister” entries
- Additional operations
Admin Features: Create Organization

Create an organization

- **Short name:**
- **Full name:**
- **URL:**
- **Members:** comma-separated list of usernames

Create organization
# Admin Features: Organizations

## Organizations

<table>
<thead>
<tr>
<th>orgName</th>
<th>Name</th>
<th>Members</th>
<th>Url</th>
</tr>
</thead>
<tbody>
<tr>
<td>xdomes</td>
<td>Earthcube X-DOME...</td>
<td>jfredericks, car...</td>
<td><a href="http://earthcube">http://earthcube</a>...</td>
</tr>
<tr>
<td>esip</td>
<td>ESIP Federation</td>
<td>lmcgibbon, carued...</td>
<td><a href="http://esipfed">http://esipfed</a>...</td>
</tr>
<tr>
<td>testorg</td>
<td>Test Org</td>
<td>carueda</td>
<td><a href="http://example">http://example</a>...</td>
</tr>
<tr>
<td>gcmd</td>
<td>NASA Global Chang...</td>
<td>tstevens</td>
<td><a href="http://gcmd.gsf">http://gcmd.gsf</a>...</td>
</tr>
<tr>
<td>jpl</td>
<td>NASA JPL</td>
<td>lmcgibbon, carued...</td>
<td><a href="http://www.jpl">http://www.jpl</a>...</td>
</tr>
<tr>
<td>obo</td>
<td>OBO Foundry and Li...</td>
<td>plbuttigieg</td>
<td><a href="http://www.obof">http://www.obof</a>...</td>
</tr>
<tr>
<td>sweet</td>
<td>Semantic Web for E...</td>
<td>lmcgibbon, carued...</td>
<td><a href="https://github">https://github</a>...</td>
</tr>
<tr>
<td>w3c</td>
<td>The World Wide We...</td>
<td>carueda, graybeal</td>
<td><a href="https://www.w3">https://www.w3</a>...</td>
</tr>
<tr>
<td>testorg</td>
<td>eos-map test org</td>
<td>carueda, xdomes</td>
<td></td>
</tr>
</tbody>
</table>
Admin Features: Organizations

<table>
<thead>
<tr>
<th>orgName</th>
<th>Name</th>
<th>Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>xdomes</td>
<td>Earthcube X-DOME...</td>
<td>jfrederik, lmcgibbnon, carueda</td>
</tr>
<tr>
<td>esip</td>
<td>ESIP Federation</td>
<td>lmcgibbon, jfrederik, graybeal</td>
</tr>
<tr>
<td>testorg</td>
<td>Test Org</td>
<td>carueda, graybeal</td>
</tr>
<tr>
<td>gcmd</td>
<td>NASA Global Chang...</td>
<td>tstevens, carueda</td>
</tr>
<tr>
<td>jpl</td>
<td>NASA JPL</td>
<td>lmcgibbon</td>
</tr>
<tr>
<td>obo</td>
<td>OBO Foundry and Li...</td>
<td>plbudding, lmcgibbnon</td>
</tr>
<tr>
<td>sweet</td>
<td>Semantic Web for E...</td>
<td>lmcgibbon</td>
</tr>
<tr>
<td>w3c</td>
<td>The World Wide We...</td>
<td>carueda</td>
</tr>
</tbody>
</table>

Organization: esip

- Name: ESIP Federation
- URL: http://esipfed.org/
- Members: carueda, graybeal, lmcgibbon, sweetbot
- Registered: 2016-06-24T17:07:59Z
- Updated: 2017-11-03T06:27:45Z (by carueda)

Ontologies

<table>
<thead>
<tr>
<th>IRI</th>
<th>Name</th>
<th>Author</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://sweetontology.net/sweetAll">http://sweetontology.net/sweetAll</a></td>
<td>SWEET Ontology</td>
<td>ESIP Federation</td>
<td>20171123T183349</td>
</tr>
<tr>
<td><a href="http://sweetontology.net/propSpaceDistance">http://sweetontology.net/propSpaceDistance</a></td>
<td>SWEET Ontology</td>
<td>ESIP Federation</td>
<td>20171123T183132</td>
</tr>
<tr>
<td><a href="http://sweetontology.net/propMass">http://sweetontology.net/propMass</a></td>
<td>SWEET Ontology</td>
<td>ESIP Federation</td>
<td>20171123T183130</td>
</tr>
<tr>
<td><a href="http://sweetontology.net/propSpaceMultidimensional">http://sweetontology.net/propSpaceMultidimensional</a></td>
<td>SWEET Ontology</td>
<td>ESIP Federation</td>
<td>20171123T183132</td>
</tr>
<tr>
<td><a href="http://sweetontology.net/propSpace">http://sweetontology.net/propSpace</a></td>
<td>SWEET Ontology</td>
<td>ESIP Federation</td>
<td>20171123T183131</td>
</tr>
<tr>
<td><a href="http://sweetontology.net/reprSciUnits">http://sweetontology.net/reprSciUnits</a></td>
<td>SWEET Ontology</td>
<td>ESIP Federation</td>
<td>20171123T183332</td>
</tr>
<tr>
<td><a href="http://sweetontology.net/repr">http://sweetontology.net/repr</a></td>
<td>SWEET Ontology</td>
<td>ESIP Federation</td>
<td>20171123T183322</td>
</tr>
<tr>
<td><a href="http://sweetontology.net/reprMath">http://sweetontology.net/reprMath</a></td>
<td>SWEET Ontology</td>
<td>ESIP Federation</td>
<td>20171123T183308</td>
</tr>
<tr>
<td><a href="http://sweetontology.net/reprMathOperation">http://sweetontology.net/reprMathOperation</a></td>
<td>SWEET Ontology</td>
<td>ESIP Federation</td>
<td>20171123T183330</td>
</tr>
<tr>
<td><a href="http://sweetontology.net/groc">http://sweetontology.net/groc</a></td>
<td>SWEET Ontology</td>
<td>ESIP Federation</td>
<td>20171123T183124</td>
</tr>
</tbody>
</table>
Admin Features: Users

<table>
<thead>
<tr>
<th>username</th>
<th>First Name</th>
<th>Last Name</th>
<th>Email</th>
<th>Regist.</th>
<th>Role</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>Adm</td>
<td>In</td>
<td>carueda...</td>
<td>2016-0...</td>
<td>admin</td>
<td></td>
</tr>
<tr>
<td>carueda</td>
<td>Carlos</td>
<td>Rueda...</td>
<td>carueda...</td>
<td>2016-0...</td>
<td>admin</td>
<td></td>
</tr>
<tr>
<td>pepito</td>
<td>Pepito</td>
<td>tester</td>
<td>carueda...</td>
<td>2016-0...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fsgraynilo</td>
<td>Felix</td>
<td>Gaynilo</td>
<td>feilmon....</td>
<td>2016-0...</td>
<td>admin</td>
<td>361825...</td>
</tr>
<tr>
<td>graybeal</td>
<td>John</td>
<td>Graybeal</td>
<td>jgrayb...</td>
<td>2016-0...</td>
<td>admin</td>
<td>650966...</td>
</tr>
<tr>
<td>jfredericks</td>
<td>Janet</td>
<td>Fredericks</td>
<td>jfrederic...</td>
<td>2016-0...</td>
<td></td>
<td>508-289...</td>
</tr>
<tr>
<td>esipdemo</td>
<td>ESIP</td>
<td>Demons...</td>
<td>jgrayb...</td>
<td>2016-0...</td>
<td></td>
<td>650-736...</td>
</tr>
<tr>
<td>akthom</td>
<td>Andrea</td>
<td>thomer</td>
<td>thomer2...</td>
<td>2016-0...</td>
<td></td>
<td>714273...</td>
</tr>
<tr>
<td>janet_esip</td>
<td>JJ</td>
<td>Fredericks</td>
<td>jfrederic...</td>
<td>2016-0...</td>
<td></td>
<td>508-289...</td>
</tr>
<tr>
<td>yongyao</td>
<td>Yongyao</td>
<td>Jiang</td>
<td>J.yongy...</td>
<td>2016-0...</td>
<td></td>
<td>508831...</td>
</tr>
<tr>
<td>druclark</td>
<td>Dru</td>
<td>Clark</td>
<td>clark@...</td>
<td>2016-0...</td>
<td></td>
<td>619222...</td>
</tr>
<tr>
<td>mikebotts</td>
<td>Mike</td>
<td>Botts</td>
<td>mike.bo...</td>
<td>2016-0...</td>
<td></td>
<td>256652...</td>
</tr>
<tr>
<td>tbs1979</td>
<td>Tyler</td>
<td>Stavens</td>
<td>Tyler.B...</td>
<td>2016-0...</td>
<td></td>
<td>301-614...</td>
</tr>
<tr>
<td>mobb</td>
<td>Margaret</td>
<td>O’Brien</td>
<td>margane...</td>
<td>2016-0...</td>
<td></td>
<td>805893...</td>
</tr>
<tr>
<td>lmcgibbon</td>
<td>Lewis</td>
<td>Mcgibbon</td>
<td>lewis.j...</td>
<td>2016-0...</td>
<td>admin</td>
<td>626487...</td>
</tr>
<tr>
<td>jstn</td>
<td>Jstn</td>
<td>One</td>
<td>carueda...</td>
<td>2016-0...</td>
<td></td>
<td>931735...</td>
</tr>
</tbody>
</table>
### Admin Features: Users

<table>
<thead>
<tr>
<th>username</th>
<th>First Name</th>
<th>Last Name</th>
<th>Email</th>
<th>Registred</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>Admin</td>
<td>In</td>
<td>carueda@...</td>
<td>2016-0...</td>
<td>admin</td>
</tr>
<tr>
<td>carueda</td>
<td>Carlos</td>
<td>Rueda...</td>
<td>carueda@...</td>
<td>2016-0...</td>
<td>admin</td>
</tr>
<tr>
<td>pepito</td>
<td>Pepito</td>
<td>tester</td>
<td>carueda@...</td>
<td>2016-0...</td>
<td></td>
</tr>
<tr>
<td>fgayanilo</td>
<td>Feimon</td>
<td>Gayanilo</td>
<td>felmon@...</td>
<td>2016-0...</td>
<td>admin</td>
</tr>
<tr>
<td>graybeal</td>
<td>John</td>
<td>Graybeal</td>
<td>jgray@...</td>
<td>2016-0...</td>
<td>admin</td>
</tr>
<tr>
<td>jfredericks</td>
<td>Janet</td>
<td>Fredericks</td>
<td>jfrederic@...</td>
<td>2016-0...</td>
<td></td>
</tr>
<tr>
<td>esipdemo</td>
<td>ESIP</td>
<td>Demons...</td>
<td>jgray@...</td>
<td>2016-0...</td>
<td></td>
</tr>
<tr>
<td>akthom</td>
<td>Andrea</td>
<td>thomer</td>
<td>thomer2@...</td>
<td>2016-0...</td>
<td></td>
</tr>
<tr>
<td>janet_esip</td>
<td>JJ</td>
<td>Fredericks</td>
<td>jfrederic@...</td>
<td>2016-0...</td>
<td></td>
</tr>
<tr>
<td>yongyao</td>
<td>Yongyao</td>
<td>Jiang</td>
<td>J.yongy@...</td>
<td>2016-0...</td>
<td></td>
</tr>
<tr>
<td>druclark</td>
<td>Dru</td>
<td>Clark</td>
<td>dclark@...</td>
<td>2016-0...</td>
<td></td>
</tr>
<tr>
<td>mikebotts</td>
<td>Mike</td>
<td>Botts</td>
<td>mike.bo@...</td>
<td>2016-0...</td>
<td></td>
</tr>
<tr>
<td>tbs1979</td>
<td>Tyler</td>
<td>Stevens</td>
<td>Tyler.B@...</td>
<td>2016-0...</td>
<td></td>
</tr>
<tr>
<td>mobb</td>
<td>Margaret</td>
<td>O'Brien</td>
<td>margare...</td>
<td>2016-0...</td>
<td></td>
</tr>
<tr>
<td>imgibbn</td>
<td>Lewis</td>
<td>McGibb...</td>
<td>lewis.j@...</td>
<td>2016-0...</td>
<td>admin</td>
</tr>
<tr>
<td>fast</td>
<td>John</td>
<td>One</td>
<td>carueda@...</td>
<td>2016-0...</td>
<td></td>
</tr>
</tbody>
</table>

**Carlos Rueda**

- **Email:** carueda@gmail.com
- **Phone:** 5304008520
- **Updated:** 2018-01-06T00:25:19Z
- **Organizations:** testorg xdomes esip w3c testorg2 acme co. jpl sweet

#### Ontologies

<table>
<thead>
<tr>
<th>IRI</th>
<th>Name</th>
<th>Author</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.w3.org/ns/sosa/">http://www.w3.org/ns/sosa/</a></td>
<td>Sensor, Observation, Sample, and Actuator (SOSA) Ontology</td>
<td>Carlos Rueda</td>
<td>20170522T162410</td>
</tr>
<tr>
<td><a href="http://www.w3.org/ns/ssn/">http://www.w3.org/ns/ssn/</a></td>
<td>Semantic Sensor Network Ontology</td>
<td>Carlos Rueda</td>
<td>20170522T162507</td>
</tr>
</tbody>
</table>
SWEET - Semantic Web for Earth and Environment Terminology

- A mid-level, highly modular ontology suite with ~6000 concepts in ~200 separate ontologies covering Earth system science
- Nine top-level concepts that can be used as a foundation for domain-specific extensions
- Written in W3C Turtle – the Terse RDF Triple Language
- Publicly available under the Apache License v2.0
- Development at https://github.com/ESIPFed/sweet/
SWEET Ontologies

● Registered at the COR: http://cor.esipfed.org/ont/esip
  ○ With automated registration of new versions
    https://github.com/ESIPFed/sweet-tools/tree/master/sc/watchdog

● Transparent http://sweetontology.net IRI resolution via Apache HTTPD mechanisms against the COR
http://sweetontology.net/* Resolution

/etc/httpd/conf.d/sweetontology.conf

<VirtualHost *:80>
  ServerName sweetontology.net

  RedirectMatch "^/?$" "https://github.com/ESIPFed/sweet/blob/master/README.md"

  RewriteRule /(.+)$ http://cor.esipfed.org/ont/api/v0/ont?iri=http://sweetontology.net/$1 [P,QSA,NE,L]
</VirtualHost>

See https://github.com/ESIPFed/sweet/wiki/sweetontology.net
ORR Instance Configurations

- Under version control using Git
  - Mainly as a log (what, when, who)
- Externally hosted in private repos
- Currently at Bitbucket
- This is temporary, esp. for COR
- Should be under particular instance space
Example: COR  https://bitbucket.org/mmiskw/orr-cor/

**Source**

- master -> orr-cor

**configDir**

- `.gitignore` 20 B 2016-08-31  general update reflecting simplified use of docker-compose
- `conf.d_orr.conf` 1.1 KB 2017-09-20  set ServerName cor.esipfed.org (there’s now also a virtual host for sweetontology.net
- `docker-compose.yml` 1.2 KB 2017-11-23  upgrade to mmlsw/orr:3.7.0 - simplified configurability
- `readme-ecite.md` 4.6 KB 2017-09-20  upgrade Docker to CE 17.06.2
- `readme.md` 2.8 KB 2017-11-23  upgrade to mmlsw/orr:3.7.0 - simplified configurability

**Set-up**

Based on docker-compose

```
$ docker-compose up -d
$ docker logs -f -tail=300 orr
```

Basic ORR upgrade procedure upon new version of an image:
- to minimize down time, first pull the new image
- adjust image reference in `docker-compose.yml`
- stop and remove the ORR container
- start the new ORR: `docker-compose up`

Then, commit and push the changes in docker-compose.yml

**Log**

2017-11-23
- upgrade to mmlsw/orr:3.7.0 - simplified configurability

2017-11-03
- upgrade ORR to 3.6.8

2017-10-12
- upgrade ORR to 3.6.7
O R R
Development
Prerequisites

- For installation of the ORR
  - Docker Engine / Docker Compose
    - [https://docs.docker.com/engine/installation/](https://docs.docker.com/engine/installation/)
    - [https://docs.docker.com/compose/install/](https://docs.docker.com/compose/install/)

- For ORR development
  - Git client program; Github account
  - Java, Scala, SBT (for backend development)
    - [http://www.scala-sbt.org/download.html](http://www.scala-sbt.org/download.html)
  - Node/NPM (for front-end development)
    - [https://nodejs.org/](https://nodejs.org/)
  - Docker Engine (to create ORR images)
Can I use my **XYZ IDE**?

- Yes, use whatever preferred editor or IDE
- All builds are done on the command line
- Do not commit your IDE-specific files!
Languages - Technologies

- **Backend**
  - Scala, Java
  - HTTP protocol and REST APIs
  - MongoDB
  - Triple store; SPARQL endpoint (AllegroGraph)

- **Front-end**
  - Javascript
  - AngularJS
  - Other frameworks (many)

- **Devops**
  - Docker images
  - WAR deployable artifact
ORR Components

Front-end

Backend

MongoDB

AllegroGraph
ORR Front-end: Overview

- Language: JavaScript
- UI Framework: AngularJS - https://angularjs.org/

- Exclusively uses backend REST API for all functionality
- Can be deployed independently
ORR Backend: Overview

- Code Repository: [https://github.com/mmisw/orr-ont](https://github.com/mmisw/orr-ont)
- API spec: OpenAPI - [https://github.com/OAI](https://github.com/OAI)
- Database: MongoDB - [https://www.mongodb.com/](https://www.mongodb.com/)
- Triple store / SPARQL endpoint: AllegroGraph [https://franz.com/agraph/allegrograph/](https://franz.com/agraph/allegrograph/)
ORR Integrated System: Overview

- Code Repository: [https://github.com/mmisw/orr](https://github.com/mmisw/orr)
- Languages: Bash, Dockerfile
- Backend and Front-end as git submodules
- Used to build deployable ORR artifacts
  - Docker image - [https://hub.docker.com/r/mmisw/orr/](https://hub.docker.com/r/mmisw/orr/)
  - WAR - [https://github.com/mmisw/orr/releases](https://github.com/mmisw/orr/releases)
ORR Documentation

- Repository: https://github.com/mmisw/mmiorr-docs
- Language: Markdown
- Tool: MkDocs - http://www.mkdocs.org/
- Doc site deployed at: https://mmisw.org/orrdoc/
- Automatically updated upon push to repository
  - Via “webhook” running on mmisw.org
Welcome to the ORR User Manual

Thanks for using the Ontology Registry and Repository (ORR) service.

This is the User's Manual for the ORR and associated MMI tools and services. It applies to all version 3 deployments of the ORR, including the ESIP Community Ontology Repository (COR).

This manual is organized as follows. The first part includes introductory materials, suggestions on getting started, and frequently asked questions. The second part describes specific actions that a user might want to do with the ORR. These actions include working with vocabularies, working with ontologies, and working with mappings. The third part includes more advanced and technical materials:

Please, feel free to use the ORR documentation issue tracker for any topics or sections that should be prioritized, or for any other suggestions. Also, your own help to improve this documentation is welcome!
Client tools and applications

- Use the REST API against the particular ORR instance
- The ORR Portal itself is one such client application
- But you can use:
  - Command-line tools: cURL, HTTPie, ...
  - HTTP client libraries in your preferred language
    - Javascript, Python, Java, Scala, ...
  - Your browser! Try [http://cor.esipfed.org/ont/api/v0/ont](http://cor.esipfed.org/ont/api/v0/ont)
ORR REST API Spec

- Uses OpenAPI specification (fka “swagger spec”)
  - https://github.com/OAI/OpenAPI-Specification (v2)

- Spec for COR instance
  - http://cor.esipfed.org/ontapi/swagger_cor.yaml
Generating client library

Using Swagger Editor - http://editor.swagger.io/
Generating client library

.swagger Editor

swagger: '2.0'
info:
  version: v0
  title: ORR API Documentation
  description: >
  The main ORR documentation is located at: https://mmisw.org/orrdoc/

Bitte note:

- The ORR API is approaching a stable version but still work in progress.
Please [let us know](https://github.com/mmisw/mmiorr-docs/issues) if you have any questions or suggestions.

- Besides the documentation itself, this page is directly exercise and test the API.
  Click on any operation header below to learn details about it, and see a "Try it out" button.
Javascript client example:

Swagger UI

- http://cor.esipfed.org/ontapi/
- Exposes REST API directly
- Powered by https://swagger.io/swagger-ui/
Javascript client example:

COR Portal
Javascript client example:

Simple Demo

Running at https://xdomes.org/orrclientdemo.html

Python Client Tools

- Python client library can automatically be generated from ORR API spec
  - Swagger Codegen - [https://swagger.io/swagger-codegen/](https://swagger.io/swagger-codegen/)
  - Or via Swagger Editor - [http://editor.swagger.io/](http://editor.swagger.io/)

- Some (old) examples / exercises:
  - With generic Python HTTP client libraries [https://github.com/emiliom/mmipy](https://github.com/emiliom/mmipy)
  - SPARQL Queries using SPARQLWrapper:
Scala Client Scripts

- SWEET “watchdog”
  - Checks for updates in the ontologies at the SWEET Github repo
  - Performs automatic registration of new versions

- CF standard names vocabulary to RDF
  - Automated Conversion and Registration
  - [https://github.com/mmisw/cf2rdf](https://github.com/mmisw/cf2rdf)
Installing the ORR [https://mmisw.org/orrdoc/install/]

0. Only requirement: Docker Engine & Compose

1. Get the configuration/settings template

2. Customize the template files for your instance

3. And launch!

**NOTE:** Size of the images to be downloaded:
- MongoDB: 130 MB (compressed)
- AGraph: 626 MB (compressed)
- ORR: 182 MB (compressed)
0. Installing Docker

- **Engine:** https://docs.docker.com/engine/installation/
- **Compose:** https://docs.docker.com/compose/install/

- **MacOS case:**
  - Engine includes Compose
  - Select download from Stable channel https://download.docker.com/mac/stable/Docker.dmg
  - Requires admin privileges
0. Installing Docker (on Mac)

Launch installer

Docker to be used mostly from the command line
1. Get configuration/settings template

- [link](https://github.com/mmisw/orr-instance-template/releases/)
- Expand the tarball or zipball (*) in a location of your choice
- **NOTE:**
  - For a basic launch test (e.g., no email dispatch), all given settings are in general usable
  - NOT a production-ready installation anyway!

(*) Or, clone the repo:
```
git clone https://github.com/mmisw/orr-instance-template.git myORR
cd myORR
rm -rf .git
```
2a. Environment variables

```bash
# Configuration directory on the host
export HOST_CONFIG_DIR=$PWD/config

# Host ORR data directory
export ORR_HOST_DATA=$PWD/orr_data

# Host port for the ORR service
export ORR_HOST_PORT=9090

# Host Mongo data directory
export MONGO_HOST_DATA=$PWD/mongo_data

# Host port for the Mongo service
export MONGO_HOST_PORT=27017

# Host port for the AGraph service
export AGRAPH_HOST_PORT=10035
```

Make sure the host ports are available
## There will be an "admin" user automatically created upon initial start of
## the ORR. This user will have all privileges on your ORR instance.

```yaml
admin {
    ## (required) Password to be given to the ORR 'admin' user.
    password = "actual-admin-pw"

    ## (required) Email address associated with the ORR 'admin' user.
    email = "jane@example.com"

    ## (optional) comma-separated list of additional usernames to be granted
    ## admin privilege right upon creation of the corresponding account.
    extra = "foo, bar"
}
```

## User authentication

```yaml
auth {
    ## (required) A strong password to generate authentication tokens.
    secret = "replace-with-actual-strong-password"
}
```
## Info about the deployment of this ORR instance in terms of associated externally visible URLs

```yaml
deployment {
    url = "http://localhost:9090/ont"
    sparqlEndpoint = "http://localhost:10035/repositories/mmiorr"
}
```
2b. Configuration - Branding

```yaml
branding {
    ## (required) A short name used in email messages and other places.
    instanceName = "MyORR"

    ## (optional) HTML fragment to be inserted right before </body>
    footer = """"<div align="center"> MyORR footer</div> """"

    ## (optional) URL of image to show in the page header.
    ## By default, this will be MMI ORR's logo.
    #logo = "http://example.net/somelog.png"

    ## (optional) URL for "Terms of Use" link.
    ## No default value (so no such link is shown).
    #tou = "http://example.net/sometou"

    ## (optional) "Contact us" link.
    ## No default value (so no such link is shown).
    ## Examples: "https://somewhere/contactus", "mailto:addr@example.net"
    #contactUs = "mailto:${admin.email}"
}
```
2b. Configuration - Email server

```yaml
email {
  ## (required) account used for sending emails.
  account {
    username = "baz@example.net"
    password = "actual-pw"
  }

  ## (required) corresponding email server for the account above.
  server {
    host = "smtp.somewhere.example.net"
    port = 465
    prot = "smtps"
    debug = false
  }

  ## Info used to compose emails:
  from = "${branding.instanceName}" "${admin.email}"
  replyTo = ${admin.email}
  mailer = ${branding.instanceName}
}
```
## Optional section.

### googleAnalytics

```
#
propertyId = "?"
#
```

## Optional section.

### See https://www.google.com/recaptcha/intro/.

### recaptcha

```
#
privateKey = "?"
#
siteKey = "?"
#
```
## Note: No changes are in general needed in this section (unless relevant pieces in docker-compose.yml need to be changed for some reason).

```yaml
mongo {
  ## (required) Name of Mongo database to be used for ORR purposes.
  ## Can be changed if desired/needed.
  db = "orr-ont"

  ## (required) Mongo server host. Should be same value as indicated in
  ## `docker-compose.yml` for the 'MONGO_HOST' environment variable of the orr service.
  host = "mongo"

  ## (required) Mongo server port. Should be same value as indicated in
  ## `docker-compose.yml` for the 'MONGO_PORT' environment variable of the orr service.
  port = 27017
}
```
2b. Configuration - AllegroGraph

```plaintext
agraph {
    ## (required) The desired AG username to access the ORR triple store
    userName = "test"

    ## (required) The desired password for the AG username above:
    password = "xyzzy"

    ## (required) Desired name for the AG repository to be used for ORR purposes.
    repoName = "mmiorr"

    ## (required) AGraph server host. Should be same value as indicated in
    ## `docker-compose.yml` for the 'AGRAPH_HOST' environment variable of the orr service.
    host = "agraph"

    ## (required) AGraph server port. Should be same value as indicated in
    ## `docker-compose.yml` for the 'AGRAPH_PORT' environment variable of the orr service.
    port = 10035

    ...
}
```
# 2c. Configuration - Notification emails

# List of email addresses to be notified upon any registration event (users, organizations, and ontologies).
# This file is checked whenever any of such events occurs.
# Put each email address in a line by itself below.

foo@example.net
bazbar@other.example.net
3. Launch your ORR instance

- On your terminal:

  ```
  $ source setenv.sh
  $ docker-compose up -d
  ...
  Starting agraph
  Starting mongo
  Starting orr
  ```

- Login with “admin” and corresponding password indicated in orront.conf
3. Launch your ORR instance

- On your terminal:
  
  ```bash
  $ source setenv.sh
  $ docker-compose up
  ...
  Starting agraph
  Starting mongo
  Starting Orr
  ```

- Open http://localhost:9090/ont/
- Login with "admin" and corresponding password indicated in orront.conf
ORR / COR
Developer Workshop

Github: https://github.com/mmisw
ORR Doc: https://mmisw.org/orrdoc
Google Group: https://groups.google.com/forum/#!forum/orr-users
Slack: https://mmisw.slack.com

Thank you!

Carlos Rueda - Beth Huffer - Lewis McGibbney

2018 ESIP Winter Meeting – Jan 11, 2018 – Bethesda, MD