Open geospatial standards and reproducible research

Massimiliano Cannata¹, Gregory Giuliani², Jens Ingensand³, Olivier Ertz³, and Maxime Collombin³

¹SUPSI, Istituto scienze della Terra, DACD, Canobbio, Switzerland (massimiliano.cannata@supsi.ch)
²University of Geneva, Institute for Environmental Sciences/enviroSPACE, geneva, Switzerland
³HEIG-VD, Yverdon-les-Bains, Switzerland

https://doi.org/10.5194/egusphere-egu23-14845
In view of best practice of Open Science principle we’re sharing a project idea we’re currently exploring at IST- SUPSI to stimulate feedbacks, further ideas, collaborations.

OSGeo @istSOS team
we’re starting **creation a proof of concept**
to implement this idea within the SensorThingsAPI

Please refer to this presentation with this DOI: https://doi.org/10.5194/egusphere-egu23-14845
Geospatial research flow is just like any other research

- **Geodata**
  - **01** file data (shp, gpkg, gml, GeoJSON...)
  - **02** databases (postGIS, MongoDB...)
  - **03** geoservices (ESRI Feature Service, OGC services, REST APIs...)

- **Geoprocessing**

- **Geospatial results**

and traditional archiving is an option to pursue reproducible research

1. archive data in an open repository (e.g.: shp, geoJSON, GeoPackage)
2. define and reference archived processing tools (e.g.: github release link, doi)
3. archive your implemented processing routines (e.g.: scripts, GIS workflow, notebook)
4. document and archive your results (e.g.: paper, reports, blog)

img: 10.5281/zenodo.3332807
In Geospatial interoperability is implemented with OGC Services

geospatial data discoverability, accessibility, sharing and re-use is commonly achieved through Spatial Data Infrastructures

Clients, applications, users

Catalog services
- catalog service
- style catalog
- data catalog

Data services
- vector data
- raster data
- sensor data

Processing services
- processing
- navigation

Portrayal services
- map images
- symbology

Encodings
- gml
- O&M
- sld
- filter

not fully FAIR: SDI do not use DOI and DatCite metadata
SDI & long-term preservation (coupling OGC & FAIR repo)

Giuliani et. al (2021) proposed a solution to enhance traditional SDI with a long-term preservation digital repository ensuring full compliance with FAIR principles while at the same time benefiting from geospatial services capabilities.

How to do with dynamic “live” data source with Data Management?

Reference a web service with used parameters to get data is with no guarantee:

- **tomorrow data may vary from today --> not reproducible results**

**EXAMPLE: Monitoring Network from metoffice**

*study*: climate change  
*data web service*: OGC SensorThingsAPI  
*data*: 50 years of 10 min data from 250 stations and 5 parameters (~10GB)

When data size is large and the dataset is highly dynamic (spatio-temporal) saving a snapshot of the dataset at each scientific publication may be:

- **expensive** (10 GB * N publications)
- **inefficient** (download and upload, metadata recreation, etc..)
- **difficult to reproduce** (notebook not working without setup of the same web services)
A Journey through time

Would it be possible to reference data status of a service only using date time?
RDB - SQL “Tempora Data Support”

From 2011 SQL “Tempora Data Support” is an optional feature enabling to access what was the state of data on a specific <time instant>.

- **System time** (ACID time):
  - system maintained (versioned)
  - only past time
  - default is now

- **Business time** (validity time, application time):
  - user maintained
  - future dates may have sense
  - time resolution user defined

Vanroose, P. (2015). Temporal Data & Time Travel in PostgreSQL.
Cadastral traveltime

PATCH: parcel 1900
changed geometry
POST: new parcel 211

POST: new building in parcel 233

POST: new building in parcel 207

2019-03-24T14:55:03.223Z

2020-07-12T09:07:55.841Z

2020-11-19T11:28:37.047Z

Cadastral traveltime
Observation traveltime

- gap filling using spline interpolation techniques
- manual correction of erroneous data
1. **Preliminary proof of concept**

2. **NOW**

3. **AS OF**
Integrating `as_of_system_time` in OGC standards (?)

1. GET data should include the optional `AS_OF_SYSTEM_TIME` parameter, which can optionally be a time instant or a time period.

2. Historic values should be immutable, to guarantee persistence.

Additionally

3. Data returned from a GET request should always refer the `SYSTEM_TIME` to which data refer to.

4. Transactional operations should include commit metadata (git-like):
   a. `commit message` to clarify the operation and reason for data changes
   b. `commiter name` to keep track of author of the change
Thanks

massimiliano.cannata@supsi.ch