Dynamic and Observational Spatial Data

Bridging the Gap
Observational (Meta)Data

Color: YELLOW

Expression: HAPPY
Observational (Meta)Data

Result:

ObservedProp.: HAPPY

Sensor: Face Recognition

DQ: Good

Result:

Swath Comparison

DQ: High

Result:

Color

SWATH

Expression

Face Recognition
Using Observational (Meta)Data
Using Observational (Meta)Data

<table>
<thead>
<tr>
<th>Object</th>
<th>Descriptor</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face1</td>
<td>Color</td>
<td>Yellow</td>
</tr>
<tr>
<td>Face1</td>
<td>Expression</td>
<td>Happy</td>
</tr>
<tr>
<td>Face1</td>
<td>Name</td>
<td>Abbie</td>
</tr>
<tr>
<td>Face2</td>
<td>Color</td>
<td>Green</td>
</tr>
<tr>
<td>Face2</td>
<td>Expression</td>
<td>Unhappy</td>
</tr>
<tr>
<td>Face2</td>
<td>Name</td>
<td>Bob</td>
</tr>
<tr>
<td>Face3</td>
<td>Color</td>
<td>Blue</td>
</tr>
<tr>
<td>Face3</td>
<td>Expression</td>
<td>Neutral</td>
</tr>
<tr>
<td>Face3</td>
<td>Name</td>
<td>Mary</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Color</th>
<th>Expression</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face1</td>
<td>Yellow</td>
<td>Happy</td>
<td>Abbie</td>
</tr>
<tr>
<td>Face2</td>
<td>Green</td>
<td>Unhappy</td>
<td>Bob</td>
</tr>
<tr>
<td>Face3</td>
<td>Blue</td>
<td>Neutral</td>
<td>Mary</td>
</tr>
</tbody>
</table>
Using Observational (Meta)Data

Data Quality?
Methodology?
UoM?

SensorThings API V1.1
OGC & Sensor Measurements

• Sensors, Actuators & Simulations usually have Location

• OGC Sensor Web Enablement (SWE)
  • Enable developers to make all types of sensors, transducers and sensor data repositories discoverable, accessible and useable via the Web
  • Since 1990 by NASA
  • Since 2001 in OGC
    • SensorML
    • Sensor Observation Service (SOS)
    • Web Processing Service (WPS)
    • Sensor Planning Service (SPS)
    • Observations & Measurements (O&M)
    • SensorThings API (STA)

• Sensor Data & Measurement Metadata
• Core of INSPIRE
Relevant Domains (on beyond Sensors)

• Environmental:
  • Air quality, Meteorology
  • Water quality and quantity
  • Biodiversity occurrence data
  • Soil and Geological data

• Demography
• BMS
• Industry 4.0
• Smart Cities
• ...

©OGC: http://www.opengeospatial.org/ogc/markets-technologies/swe
OGC SensorThings API

• A standard for exchanging sensor data and measurement metadata
  • Historic data & current data
  • JSON Encoded
  • RESTful
  • Adapting OASIS Odata URL patterns and query options
  • Supporting ISO MQTT messaging

• Easy to use & understandable
  • Discoverable with only a web browser
OGC SensorThings API

- A standard for exchanging sensor data and measurement metadata
- Historic data & current data
- JSON Encoded
- RESTful
- Adapting OASIS Odata URL patterns and query options
- Supporting ISO MQTT messaging
- Easy to use & understandable
- Discoverable with only a web browser
Getting to your data

- Based on OASIS OData
- Base URL: http://server.de/FROST-Server/v1.1
- Read: GET
  - v1.1 → Get collection index
  - v1.1/Collection → Get all entities in a collection
  - v1.1/Collection(id) → Get one entity from a collection
- Create: POST
  - v1.1/Collection → Create a new entity
- Update: PATCH
  - v1.1/Collection(id) → Update an entity
- Update: PUT
  - v1.1/Collection(id) → Replace an entity
- Delete: DELETE
  - v1.1/Collection(id) → Remove an entity

Query Parameters
- $skip: pagination
- $top: pagination
- $count: entity count
- $select: result customization
- $expand: result customization
- $filter: data search
STA Query Logic

Query Logic differs from basic spatial feature APIs
• Data is not static in time – may change during viewing
• Data is not tile-based
• Multiple dimensions through underlying complex data model
  • Time series data
  • Observed Properties
• Massive not-tile-based data, requires intelligent queries depending on
  • Zoom level
  • Presentation requirements – displaying location vs. data time-series
• Complex queries across multiple objects core in STA

→ Additional support required, existing tools for static data don’t fulfil requirements
Realtime Air Quality

Bad Vöslau, Gainfarn
- NO
- NO2
- PM10
- O3
- PM2.5

Graph showing fluctuations in air quality from September 1 to September 28, 2020.
Multi Resolution Data

• European NUTS regions with demography data
  • Regions in 5 scales. Which to choose?
• A STA Thing can have multiple Locations (Geometries)
  → Add All scales!
  → Store the scale for each in the Location properties
  → Build cool queries!

https://api4inspire.k8s.ilt-dmz.iosb.fraunhofer.de/servlet/is/163/
Multi Resolution Data

• European NUTS regions with demography data

• Regions in 5 scales. Which to choose?

• A STA Thing can have multiple Locations (Geometries)

→ Add All scales!

→ Store the scale for each in the Location properties

→ Build cool queries!

https://api4inspire.k8s.dmz.iosb.fraunhofer.de/servlet/is/163/
STA Mapper - STAM

• JavaScript library
• Displays Things/Features-of-Interest
• Handles groupings by zoom level
• Integrated call-back for displaying time series
• Integrates with Leaflet/OpenLayers map.

https://github.com/DataCoveEU/STAM
STAM Functionality

- **Display Things/FoIs on Map**
  - Takes zoom level into account
  - For tiles with many entities, only count requested
  - Groupings based on OSM Tiles
  - Custom Icons, influenced by response data

- **Identify Things/FoIs**
  - All associated Datastreams listed

- **Show Observations**
  - Callback can be configured for custom display
  - Plotly integrated for default display of time series
STAM Configuration Options

- **baseUrl**: string  //The base url of the Sensorthings API
- **markerStyle**: Function | string  //Specifies the color of the marker. Functions get geoJSON as parameter
- **clusterStyle**: Function  //Used to specify the style of the circle or polygon
  - circle
  - polygon
- **markerMouseOver**: Function  //Callback receiving feature on marker hover
- **markerClick**: Function  //Callback receiving feature on marker click
- **clusterMouseOver**: Function  //Callback receiving feature on cluster hover
- **clusterClick**: Function  //Callback receiving feature on cluster click
STAM Configuration Options II

- **plot:** {} // Temporal range for plot. Offset OR endDate may be specified
- **cachingDuration:** number //Time in seconds to cache the data. null = forever
- **cluster:** Boolean //Defaults to true, if false no clustering applied
- **clusterMin:** number //Minimal count within tile, so that a cluster is displayed
- **queryObject:** {} //Can be a array of a ranges or directly a queryObject. Queries can be specified for given zoomlevels or ranges.
Conclusions

• SensorThings API is being increasingly deployed
• New domains still discovering power of STA
• Map based visualization still in development
• STAM a first approach to providing simple mapping support

More examples and demos at:
https://datacoveeu.github.io/API4INSPIRE/
Thanks for your Attention!

Kathi Schleidt
Kathi@DataCove.eu

Hylke van den Schaaf
hylke.vanderschaaf@iosb.fraunhofer.de

Thomas Usländer
thomas.uslaender@iosb.fraunhofer.de