

2017.2 meeting #6 2018-12-17/18

Logistics

Date: Monday/Tuesday, 17-18th of December 2018

Location: Rooms Leonardo/Michelangelo, building 26a, Joint Research Centre, Ispra, Italy

Agenda

Time	Agenda item	Document(s)
Monday, 17 December 2018		
12:00 – 13:00	<i>Arrival & buffet lunch</i>	
13:00 – 13:15	Welcome and approval of the agenda (JRC)	20181217-18 2017.2 sub-group meeting 6 - Final agenda.pdf
13:15 – 15:00	Scoping of the GeoJSON Alternative Encoding – INSPIRE themes, INSPIRE requirements (WeTransform) <ul style="list-style-type: none">• Presentation of 3 examples (Sub-group members, 10 mins each)• Specific Technical problems to be solved and their priority• Discussion of potential encoding rules based on GeoJSON examples	
15:00 – 15:30	<i>Coffee break</i>	
15:30 – 17:30	Discussion of open issues (GeoJSON) (all)	
20:00	<i>Social dinner (offered by JRC)</i>	
Tuesday, 18 September 2018		
09:00 – 10:30	Good Practice document on Simplification Rules – Presentation Current Draft (WeTransform) Discussion of open issues (simplification rules) (all)	
10:30 – 11:00	<i>Coffee break</i>	
11:00 – 11:30	Good Practice document on GeoJSON – Presentation Current Draft (WeTransform)	
11:30 – 12:30	Discussion of open issues (GeoJSON) (all)	
12:30 – 14:00	<i>Lunch break</i>	
14:00 – 15:30	Presentation of current draft of work on 2017.3 Discussion of data usability testing activities (all)	
15:30 – 16:00	<i>Coffee break</i>	
16:00 – 16:30	Wrap-up and next steps (JRC, WeTransform)	

Attendees

- Sub-group members:
- JRC contractors: Stefania Morrone (Epsilon Italia), Thorsten Reitz (WeTransform)

Draft Notes & actions

Item	Notes / Actions
Welcome and approval of the agenda	The agenda has been approved.
Scoping of the GeoJSON Alternative Encoding	<p>To scope the JSON encoding, the group discussed whether to start with specific examples or whether to start with broad, generally applicable rules.</p> <p>General requirements and rules:</p> <ul style="list-style-type: none"> • Work should be directly usable, so if it requires other parts that are not specified there is less value • If we only change the encoding, the main issues will still remain, so we need to look at these things (simplification + alternative encoding). • How is an alternative encoding developed and documented? • Maintain lifecycle information of INSPIRE features • Themes can be from Annex I, II and III • Make implementation of INSPIRE easier • Do not require special software, often complexity is the issue, not so much the encoding • This group will NOT discuss the conceptual models. <p>Alternative/Additional Encoding?</p> <p>One major question is whether this encoding can be an alternative to providing INSPIRE GML for the themes in scope, or whether it is always to be provided in addition. If it is an additional encoding, do providers need to go via the default encoding or can they deliver the alternative encoding (with a simpler model) directly? Key points from the discussion:</p> <ul style="list-style-type: none"> • alternative only if there is no information loss • "This encoding complies to the IR if your source data only has..." • Clarification must go to the terminology or into final report ("DS have to potentially uploaded to clarify relationship"). Will there still be a <i>default</i> encoding? • Relation between the alternative encoding and the data specification • Relation from Data specification to Alternative Encoding provided in 9.3.x <p>Conclusion:</p> <p><i>Within this work, we will define an alternative encoding that can be used instead of the default encoding for simple data, where there is no information loss.</i></p> <p>Examples & Themes</p> <p>The decision was made to start from concrete examples and to apply these to specific themes. Members of the group presented various examples:</p> <ul style="list-style-type: none"> • GN: +++ -- <ul style="list-style-type: none"> ◦ Is used in a lot of places in other themes ◦ If we define rules for GN, would they be applicable to AD as well? • AD: ++++++ <ul style="list-style-type: none"> ◦ Address Size issue, e.g. for gazetteer usage as in the French case? ◦ Very valuable, but currently hard to use ◦ Many use cases, extensively used also by non-GIS communities ◦ Good template case to develop generic rules as well as specific rules ◦ A working encoding is achievable in this group ◦ In the JRC example: very long names (how to cut), simplifications on the GN, no lifecycle, removed id ◦ Issue with presence of multiple geometries: GeoJSON doesn't allow semantic for geometries, you have to provide an extra property which tells you the semantics. • O&M: ++++++ <ul style="list-style-type: none"> ◦ used by several themes, important environmental use case ◦ Can be extended to more challenging aspects such as profiles and 3D geometries ◦ Use EMF as context theme for O&M; makes this data more accessible to environmental user communities +++ ◦ EMF would include references to legal resources ◦ GE - Boreholes as potential second context ◦ Add Atmospheric Conditions as third context ◦ NOTE: there are lots of efforts on how meteorological info looks like on the web (e.g. as JSON) already ◦ NOTE: For sensorthings, there is a JSON encoding proposal; Feedback to OGC should be provided • TN: +++++ <ul style="list-style-type: none"> ◦ Properties attached to Networks ◦ Support for visualisation, some themes such as TN are currently not suited for that <p>The group then decided whether to include these in the scope, or not (+ / - on the items above).</p> <p>Conclusion:</p> <p><i>The draft Alternative Encoding encompasses the INSPIRE themes Addresses (including GeographicalName properties) and Environmental Monitoring Facilities (including O&M properties).</i></p> <p>Use Cases</p>

	<p>The group also discussed whether to focus on specific use cases such as web applications for this encoding.</p> <ul style="list-style-type: none"> Data sharing / exchange / transfer of vector information <ul style="list-style-type: none"> towards non-domain experts towards developers who do not know much about geo standards towards GIS users (desktop, mobile, web) explorative work with transient features (from client object lifecycle perspective) Consume the data in the most commonly used clients +++ Load and visualise the data in common GIS easily +++ Think in terms of delivery via APIs At OS, requests for JSON are mostly from non-GIS people Select relevant parts from JSON for their own analytics and tools Usage of environmental data in environmental and scientific communities <ul style="list-style-type: none"> Make it easy for DG ENV officer to analyse / visualise data Change detection Quality Assurance/Validation: Would have been done before, GeoJSON is a publishing format Each theme has a list of specific (business) use cases, need to investigate these as a validation step <p>Conclusion:</p> <p><i>The draft Alternative Encoding shall be optimized for the following use cases: Viewing and analysing in mainstream GIS clients. This means that no generally unsupported features such as JSON References are to be used.</i></p> <p>Specific technical issues</p> <p>The encoding should also resolve specific technical issues that have been problematic when using the default encoding.</p> <ul style="list-style-type: none"> See findings in https://github.com/INSPIRE-MIF/2017.2/issues/48 For ArcGIS: https://github.com/INSPIRE-MIF/2017.2/issues/18 Abstract geometry types for an object (mixed geometry types in a FeatureCollection) Be able to display types on map, e.g. date/time, xlinks, coded values and codelists ...
Model Simplification Rules	<p>The initial draft of the best practice paper simply collects those rules found in the examples. After discussion, the group decided to change the it from a catalogue of examples to a list of patterns / best practices. The current template used for each pattern was discussed and should be amended as follows:</p> <ul style="list-style-type: none"> reference where the pattern/rule has been used have at least those rules needed for the themes Add UML class diagram to the description of each rule <ul style="list-style-type: none"> Have both UML models (Conceptual Model -> Implementation Model) When should the rule be used, when should it not be used? Under which conditions there is no information loss? As far as possible, rules should be independent from the encoding Flag rules specific to an encoding, e.g. Geometry in GeoJSON specific rules can go to GeoJSON doc? Examples as primary driver of the how-to; should examples use a homogeneous encoding? Yes. <p>Specific Rules under discussion</p> <ul style="list-style-type: none"> How to deal with the same local names being used by members from different namespaces Voidable properties: <ul style="list-style-type: none"> Make a simplification rule that takes any property that has the stereotype <<voidable>> and makes it optional (0..) result: nilReason dropped, not bijective, change in semantics (empty/nil values) corresponding instance transformation rule: <ul style="list-style-type: none"> Omit the member if its value is empty make it an optional rule to drop unpopulated members
Usability Testing	<p>Conclusions</p> <ul style="list-style-type: none"> Do not focus on producer/server side Concentrate on client-side issues

Follow-Up Actions	<ul style="list-style-type: none"> • WE/TR: Timeline for next deliveries <ul style="list-style-type: none"> ◦ 2nd draft for 29.01.2019 (originally scheduled 15.01.2019), includes EMF (which includes O&M) and AD (which uses GN) ◦ Meeting to consolidate open issues around end of February/earliest March ◦ Final draft for 15.03.2019 -> will be sent to MiG <input checked="" type="checkbox"/> WE/TR: Clarify base types usage / simplification <ul style="list-style-type: none"> ◦ GN, Identifiers, Citations <input type="checkbox"/> WE/TR: Plan communication towards Community <input checked="" type="checkbox"/> BRGM/Marie: FR BRGM has Borehole INSPIRE GML example using O&M - Marie will ask/provide links <input type="checkbox"/> JRC/ML: Determine how to reconcile NilReason Model Transformation Rule with the requirements in the IR --> "You shall provide a value of ... or document ..."
-------------------	---

Annex 1: Annotated Examples

Addresses IGN-F 1

```
{
  "type": "FeatureCollection",
  "crs": {
    "type": "name",
    "properties": {
      "name": "urn:ogc:def:crs:EPSG::4258"
    }
  },
  "features": [
    {
      "type": "Feature",
      "id": "AD_ADDRESS_FR_IGNF_BDUniGE_Adresses_MET_ADRNIVX_0000000283326117", // added; might add
      recommendation on patterns to use; should use same pattern of generation as for GML encoding if that is also
      delivered

      // Have the ID for filtering by ID in a download service --> TODO Test this

      // Where there is a 1:1 relationship before and after model transformation, we should use the value that
      would be used to populate gml:id

      // do not make a recommendation for cases outside the 1:1 scope

      "properties": {
        "gml_id": "AD_ADDRESS_FR_IGNF_BDUniGE_Adresses_MET_ADRNIVX_0000000283326117", // should be removed, there
        is no GML in JSON; but need to add id to feature

        "inspireId_localId": "ADRNIVX_0000000283326117", // flattening like this is fine

        "inspireId_namespace": "FR_IGNF_BDUniGE_Adresses_MET", // clarify whether to use . or _ or - to indicate
        different semantics, e.g. as in OCL; clarify whether to keep the typename in the flattened property name; where
        might this lead to issues?

        // Use the common flattening rule for the InspireIdentifier

        "position_specification": "http://inspire.ec.europa.eu/codelist/GeometrySpecificationValue/parcel", //
        how would we add readable text (titles)?

        "position_method": "http://inspire.ec.europa.eu/codelist/GeometryMethodValue/fromFeature",

        "position_default": true, // this attr is only useful if we have more than 1 point

        "locator_designator": [
          "698", // Currently two arrays need to be parsed together, should rather group by locator_type (e.g.
          locator_1 + locator_1_type, locator_addressNumber)

          "A"
        ]
      }
    }
  ]
}
```



```

"inspireId_namespace": "FR_IGNF_BDUniGE_Adresses_MET",

"position_specification": "http://inspire.ec.europa.eu/codelist/GeometrySpecificationValue/parcel",

"position_method": "http://inspire.ec.europa.eu/codelist/GeometryMethodValue/fromFeature",

"position_default": true,

"locator_designator": [

    "698",

    "A"

],

"locator_type": [

    "http://inspire.ec.europa.eu/codelist/LocatorDesignatorTypeValue/addressNumber",

    "http://inspire.ec.europa.eu/codelist/LocatorDesignatorTypeValue/addressNumberExtension"

],

"level": "http://inspire.ec.europa.eu/codelist/LocatorLevelValue/siteLevel",

"component": [ // In-Place encoding will be a case by case decision; In this case it makes sense for
usability; might want to get rid of components and instead directly use fields like adminUnitName_2nd,
adminUnitName_3rd, ...; If there is a short list of soft types, it could be transformed/promoted to a set of
properties which are hard types each and which indicate specific types by their name (derived from the codelist
value?)

    // If we use a reference instead of in-place encoding, display value/name and de-referencable URI;
Does a certain style of reference indicate the (media) type of the linked resource? Do we use $ref JSON
references? We could also look at JSON-LD when including a object linking mechanism into the encoding.

    // --> Use name/label and simple, resolvable link

    // 3 types of "links" - between objects, to codelists, to external resources (e.g. in citations),
Identifiers/URIs

    // "nativeness:ref"/nativeness_link, ...Id, ...

    {

        "gml_id": "AD_ADMINUNITNAME_FR_IGNF_BDUniGE_Adresses_MET_codeINSEE_24309",

        "inspireId_localId": "codeINSEE_24309",

        "inspireId_namespace": "FR_IGNF_BDUniGE_Adresses_MET",

        "status": "http://inspire.ec.europa.eu/codelist/StatusValue/current",

        "name_language": "fra",

        "name_nativeness": "http://inspire.ec.europa.eu/codelist/NativenessValue/endonym",

        "name_nameStatus": "http://inspire.ec.europa.eu/codelist/NameStatusValue/official",

        "name_sourceOfName": "BD TOPO@",

        "name_spelling_text": "Neuovic", // strip down GN to just the name text when the data sets in
monolingual? For multiple languages, how to indicate which name is in which language if it it would be a simple
array or multiple members? In XML, additional info can be put into attributes on the tag (e.g.
LocalisedCharacterString?)

        // always provide "default/official" name in "first" national language? Additional info and
names can go into more fields?

        // diff names by language code in property name? name_en? No, provide a flattened list (name_1)
with the additional info in extra members (name_1_status, name_1_lang, name_1...)

        // make it a content negotiation issue between client/service - if you request one language the
names will be in that language

        "name_spelling_script": "Latn"

    },

    {

        "gml_id": "AD_ADDRESSAREANAME_FR_IGNF_BDUniGE_Adresses_MET_24309_LABATUT",

        "inspireId_localId": "24309_LABATUT",

        "inspireId_namespace": "FR_IGNF_BDUniGE_Adresses_MET",

```

```

        "status": "http://inspire.ec.europa.eu/codelist/StatusValue/current",
        "name_language": "fra",
        "name_nativeness": "http://inspire.ec.europa.eu/codelist/NativenessValue/endonym",
        "name_nameStatus": "http://inspire.ec.europa.eu/codelist/NameStatusValue/official",
        "name_sourceOfName": "BD ADRESSE@",
        "name_spelling_text": "Labatut",
        "name_spelling_script": "Latn"
    }
]
},
"geometry": {
    "type": "Point",
    "coordinates": [
        0.460372,
        45.078589
    ]
}
}
]
}

```

Open Actions

Task report

Looking good, no incomplete tasks.