GeoPackage
alternative encoding
for INSPIRE

60th meeting MIG-T, Prague

Francisco J. Lopez-Pellicer
fjlopez@unizar.es

IAAA
Grupo de Sistemas de
Información Avanzados
Universidad Zaragoza

GOBIERNO DE ESPAÑA
MINISTERIO DE FOMENTO
INSTITUTO GEOGRÁFICO NACIONAL

Agenda

- Is Geopackage fit for INSPIRE?
- Geopackage ES – Experience overview
- First iteration – Proof of concept
- Second iteration – Model driven
  - Model driven development of schemas
  - Pragmatic creation of GeoPackages
- Three insights
  - It's just plain SQL for implementers!
  - Encoding for bulk download
  - Three simple encoding profiles
- Next steps

Is Geopackage fit for INSPIRE?

- Shapefile is today a bad format
  - Good side → The most widely supported format
  - Bad side → Stuck in the 80-90's informatics

- Alternatives
  - **OGC GML** → **INSPIRE** official encoding, supported in many software packages
  - **GeoJSON** → JSON format, IETF Standard
  - **OGC GeoPackage** → File based SQLite database for vector, raster, etc. and OGC standard

- OGC GeoPackage is a set of conventions for storing in SQLite
  - Vector features (with linear and non-linear geometry types)
  - Tile matrix sets of imagery, raster maps at various scales
  - Tile gridded coverage data
  - Attributes (non spatial data)
  - Metadata (coarse to fine grained) and schema constraints
  - And more

Is Geopackage fit for INSPIRE?

- Characteristics useful for INSPIRE
  - SQLite is the most used database in the world (is embedded anywhere)
  - Pre-defined datasets ready to direct use
  - Large file support
  - Complex model support

- As *alternative* encoding could be the INSPIRE workhorse for data transfer
  - Can encode data from all themes (but 3D data) without loss of information
  - Ideal to transfer large/complex data from one system to another
    - GIS desktops, network servers, corporate databases, web and mobile applications.

- Pre-Defined Download Service
  - One or more Geopackages representing either the full dataset or subsets
  - Requires schema conform to relevant INSPIRE data specification(s)

Geopackage ES - Experience overview

- Use cases
  - Addresses data usability in desktop (QGIS3, ArcGIS), native mobile (NGA iOS and Android SDK), web (NGA JS), servers (NGA Java) and geoprocessing software (GDAL).
  - We have tested desktop (QGIS3), geoprocessing software (GDAL) and server (NGA Java)

- Themes tested
  - We have tested Annex I: Geographical names and Administrative units

- Technical Issues found
  - GDAL do not support yet all extensions
  - NGA Java supports all extensions and is OGC certified

Geopackage ES - Experience overview

- First iteration: 2019-Feb
  - Textual rules and handmade schema creation from INSPIRE data specifications
- Second iteration: 2019-Oct
  - Formal rules expressed as a program and automatic schema creation from INSPIRE UML XMI
- Identified schema conversion rules
  - Feature Types → Feature Table per geometry attribute
  - Data Type → Attribute Table
  - Data Type with Geometry → Feature Table
  - ISO 19103 basic types → Equivalent GeoPackage types or TEXT
  - ISO 19107 geometry types → Equivalent linear and non-linear GeoPackage geometry types
  - Flattening of data types with maximum multiplicity of 1
  - Enumeration and code lists → GeoPackage data column constraints of type enum
  - Voidable → Explain null or empty content as additional Metadata

Geopackage ES - Experience overview

Nomenclátor Geográfico Básico de España

Descripción: relación de topónimos del NGBE.

SGR: ETRS89 en la Península, Islas Baleares, Ceuta y Melilla, y REGCAN95 en las Islas Canarias (ambos sistemas compatibles con WGS84). Coordenadas longitud y latitud y UTM en su huso correspondiente.

Ud. descarga: toda España

Formato: .accdb de Access

Ver +

Base de datos con la relación de topónimos con sus correspondientes coordenadas y resto de atributos que constituye el Nomenclátor Geográfico Básico de España.

Metadatos

Descargar

Geographical Names

Líneas límite municipales

Descripción: recintos municipales y líneas límite (municipales, provinciales y autonómicos).

SGR: ETRS89 en la Península, Islas Baleares, Ceuta y Melilla, y WGS84 en las Islas Canarias. Coordenadas geográficas longitud y latitud.

Ud. descarga: toda España

Formato: shape (.shp)

Ver +

Metadatos

Descargar

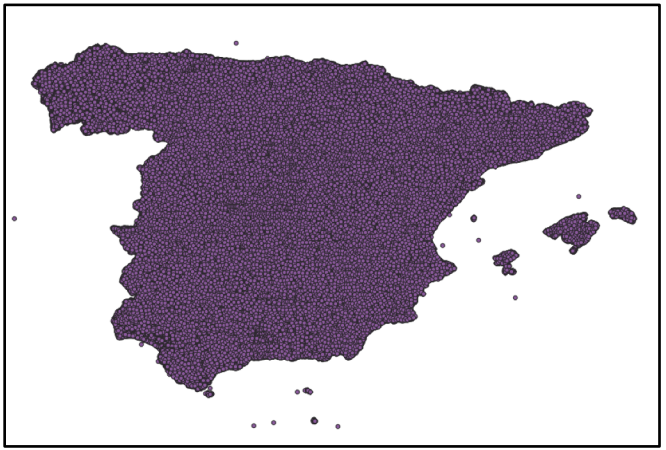
Administrative Units

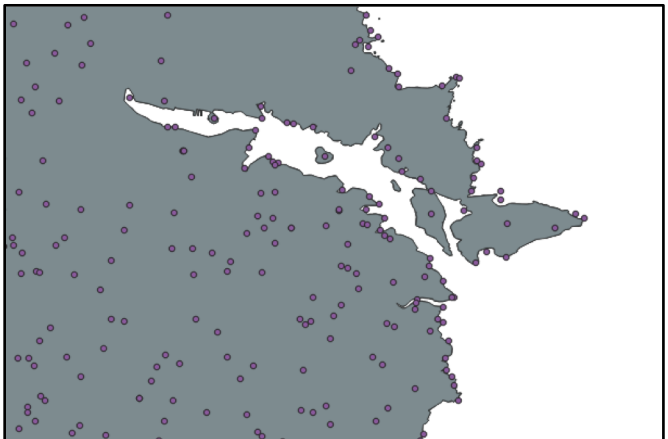
First iteration – Proof of concept

- NGA GeoPackage Java (certified by OGC)
  - https://github.com/ngageoint/GeoPackage
- In-house DSL (written in Kotlin)
  - Handmade creation of any INSPIRE schema within GeoPackage
  - ETL Postgis → GeoPackage
    - Data migration to Postgis → GeoPackage
    - INSPIRE schema creation
    - Population INSPIRE Schema
- Lessons learned
  - Think as SQL developer, not as GIS developer→ faster development
  - Performance improved if model transformation is executed within SQLite → FME-like scripts will be slow
  - INSPIRE schema templates per theme (empty GeoPackages) → Reusable by-product
  - Our in-house DSL is not sufficient flexible for population → We need a different approach

First iteration – Proof of concept

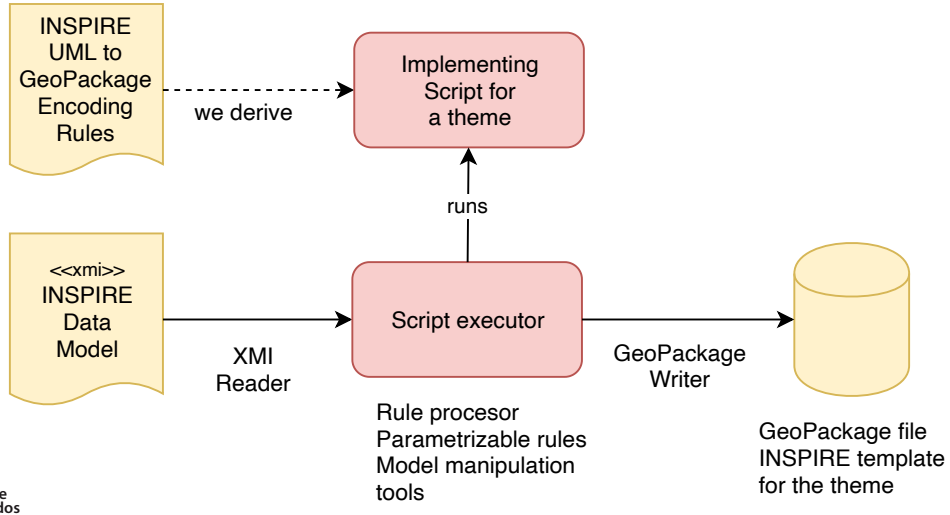
- Geographical Names
  - File size 352 MB
  - Features: NamedPlace (1132583 rows)
  - Attributes: GeographicalName (1317590 rows), SpellingOfName (1317590 rows), + 3 more tables for attributes with multiplicity (localType, relatedSpatialObject, type)
- Administrative Units
  - File size 646 MB
  - Features: AdministrativeUnit (8283 rows), AdministrativeBoundary (45844 rows)
  - Attributes: Identifier (54127 rows), GeographicalName (8283 rows), SpellingOfName (8283 rows), + 1 more table for M:N relation between Boundary and Unit
- Transformation run: 40min
- Tested with QGIS3
  - View, edit… fast and satisfactory
  - Join tables feasible (NamedPlace x GeographicalName x SpellingOfName)





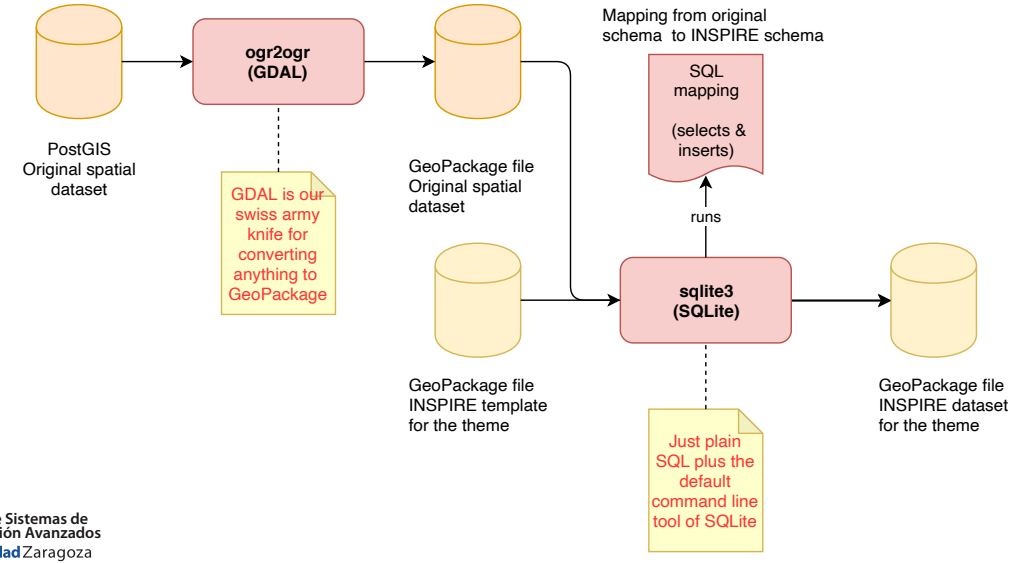
Second iteration – Model driven

- INPIRE UML to GeoPackage schema (written in Kotlin)
  - Inspired by ShapeChange
  - Input: Official data models in XMI
  - Output: GeoPackage with the data model encoded
- Developed alongside with a textual specification (similar to GeoJSON)
- Work in progress available at IDEE repository at GitHub



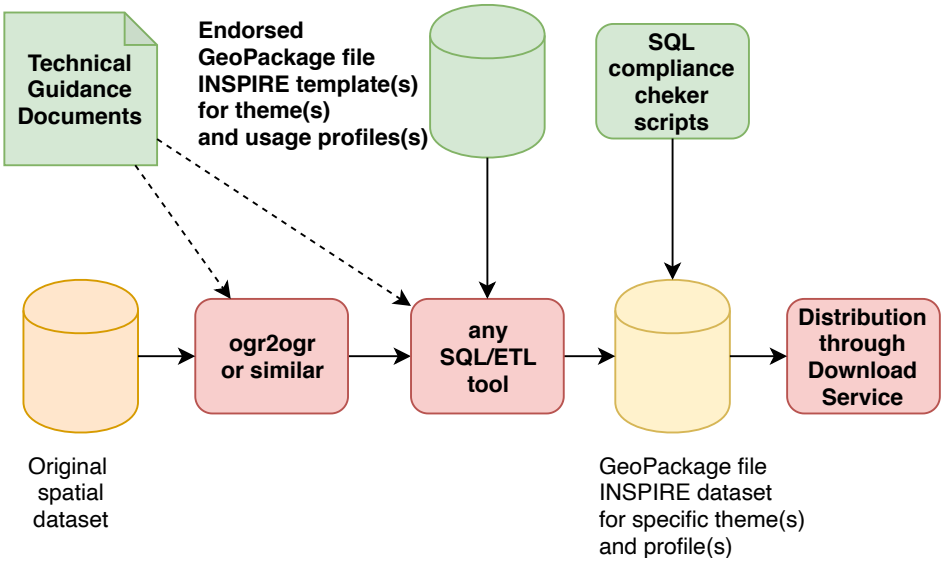
Second iteration – Model driven

- Pragmatic creation of a Geopackage for a Pre-Defined Download Service
  - Input #1: GeoPackage template from the previous step
  - Input #2: A spatial database with the dataset
  - Output: GeoPackage file ready to be served by the pre-defined download service
- Pragmatic implies no special tools required, only basic SQL knowledge required



Insight #1 - It's just plain SQL for implementers!

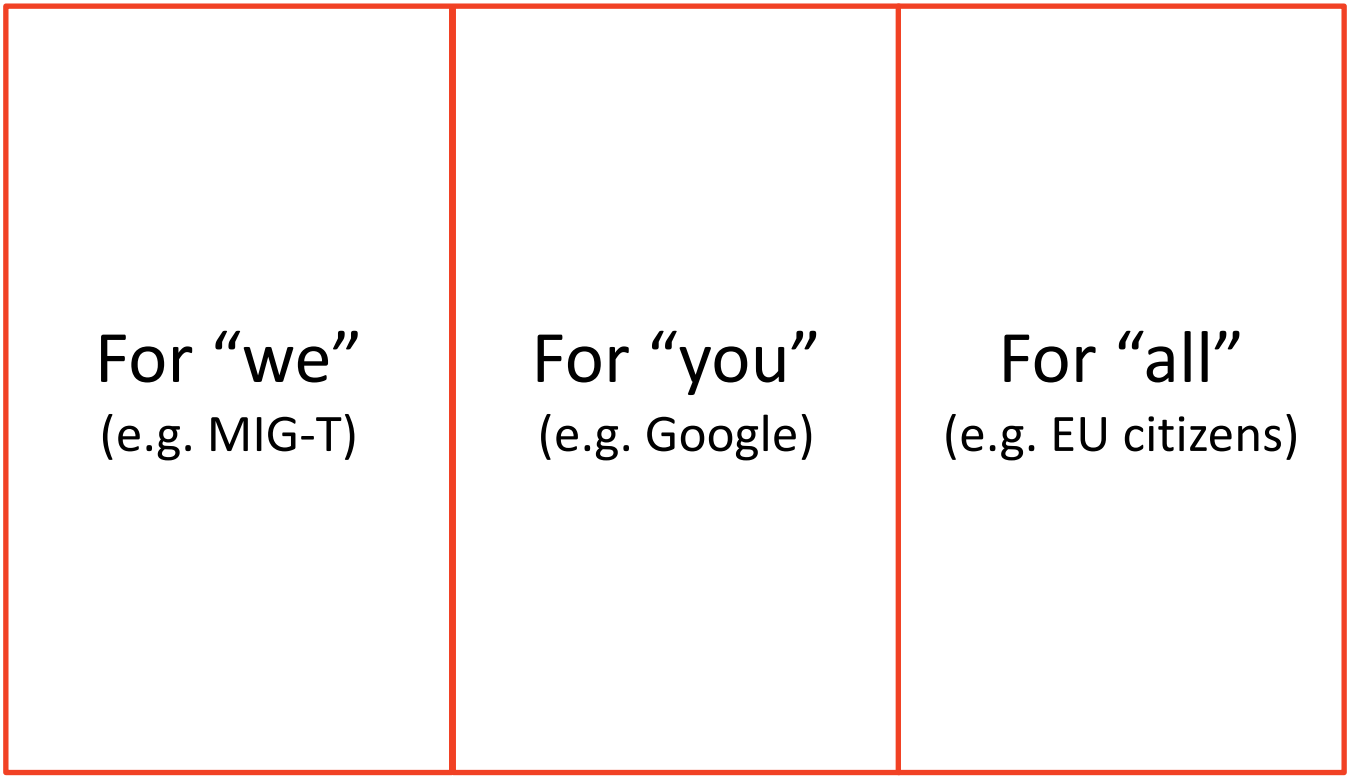
- Implementers only require
  - GeoPackage Template files
  - Encoding rules plus examples written in SQL
  - Compliance checkers written in SQL
- No "new" tools/skills required
  - Implementers are accustomed to performing similar processes with Shapefiles and other geodatabases
  - IT without specific geo skills can implement this workflow



Insight #2 – Encoding for bulk download

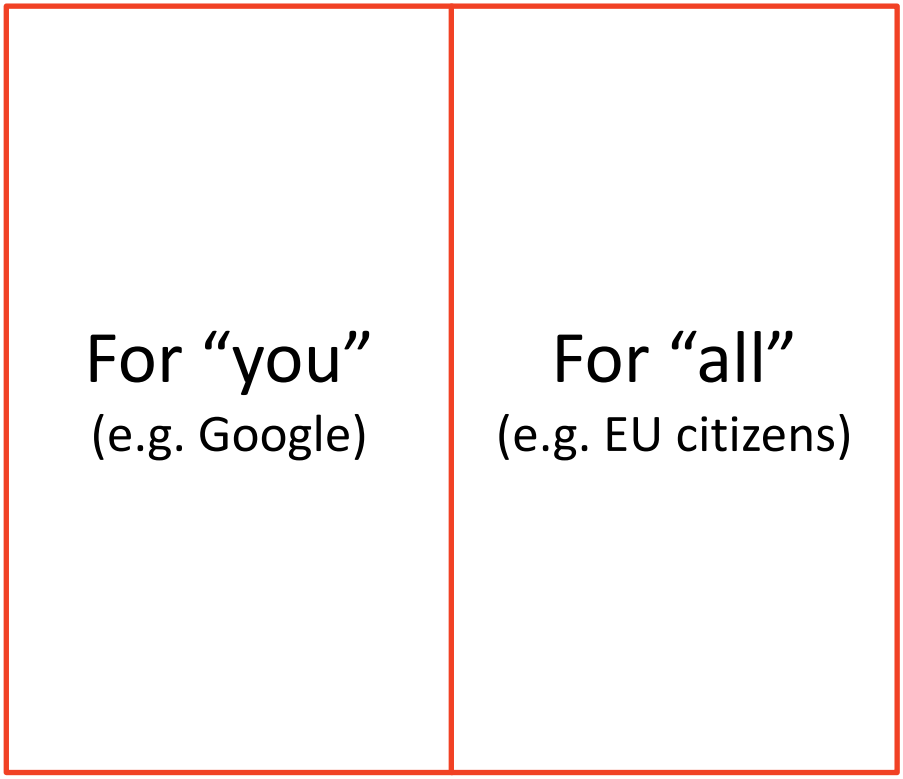
- ES experience with GML
  - WFS/ATOM for GN and AU are a success
  - Users and apps use them → Simple models,
- In which scenarios could GeoPackage replace GML?
  - Direct access → No, GeoJSON is the best candidate
  - Pre-defined → Yes, specially in large/complex models
- Rule of thumb for GML replacement by GeoPackage
  - GML files are too big to be created easily for direct access
  - GML files are too big to be processed by user tools
  - GML files are contains objects highly connected each other
- Evaluation of the replacement → Planned conversion of TN to GeoPackage

Insight #3 – Three simple encoding profiles



Insight #3 – Three simple encoding profiles

- INSPIRE compliance for MIG
  - Full schema
  - Simple flattening
  - Voidable metadata
  - Document other compliance requirements
- Level 1 = full
  - The original encoding rule that we are developing



Insight #3 – Three simple encoding profiles

- INSPIRE compliance for MIG
  - Full schema
  - Simple flattening
  - Voidable metadata
  - Document other compliance requirements
- Data transfer for data consumers/organisations
  - Full schema
  - Simple flattening
  - No loss of information

- Level 1 = full
  - The original encoding rule that we are developing
- Level 2 = only data
  - Apply the original encoding rule
  - Do not add to the geopackage voidable metadata

For “all”
(e.g. EU citizens)

Insight #3 – Three simple encoding profiles

- INSPIRE compliance for MIG
  - Full schema
  - Simple flattening
  - Voidable metadata
  - Document other compliance requirements
- Data transfer for data consumers/organisations
  - Full schema
  - Simple flattening
  - No loss of information
- "Bunch of shapefiles" for final users/mobile apps
  - "Aggressive" flattening
  - Reduce clutter by removing nil properties
  - No voidable metadata

- Level 1 = full
  - The original encoding rule that we are developing
- Level 2 = only data
  - Apply the original encoding rule
  - Do not add to the geopackage voidable metadata
- Level 3 = easy to use
  - Apply the original encoding rule
  - Apply an optional rule to flatten low cardinality associations
  - Remove empty tables and columns

Next steps

- Test with large datasets
- Develop the encoding rules for TN
- Create one/many GeoPackage(s) with the dataset TN ES
- Check usability
  - Can be directly used in GIS Desktop tools?
  - Is it only useful for data transfer?
- … and of course, sharing this work with MIG-T members and other stakeholders

GeoPackage
alternative encoding for INSPIRE

Thanks to:

GOBIERNO DE ESPAÑA
MINISTERIO DE FOMENTO
INSTITUTO GEOGRÁFICO NACIONAL

GeoSLab
https://www.geoslab.com/

Francisco J Lopez-Pellicer
IAAA, Universidad Zaragoza
fjlopez@unizar.es
twitter @fjlopezpellicer
https://www.linkedin.com/in/franciscojlopezpellicer

IAAA
Grupo de Sistemas de Información Avanzados
Universidad Zaragoza