ARCHIVING DIGITAL MAPS WITH GEOPACKAGE AND VECTOR-TILE DISSEMINATION

Yunnan Chen

Supervisors: Dr. Markus Jobst (BEV)
Prof. Dr. Georg Gartner (TUW)
Dr.-Ing. Christian Murphy (TUM)
Outline

1. Introduction
2. Literature Review and Theoretical Background
3. Methodology and Case Study
4. Results
5. Discussion and Future Work
6. Conclusions
7. References
1. Introduction

• Background and Motivation
• Research Goal and Objectives
• Research Questions
1. Introduction

1.1 Background and Motivation

“Currently, there are no easy answers as how best to guarantee that digital data will last into the future without active management to preserve the data.” - Bleakly, D. R., 2002

- Limitations of current geospatial data format in archiving digital maps
- Time consuming operation for archiving digital maps
- Digital map distribution Issues
1. Introduction

1.2 Research Goal and Objectives

• Goal:

  An alternative method for long-term archiving and distribution of digital maps

• Objective:

  1. To determine and analyze a suitable data format that has the ability to store geospatial data and its relevant information such as metadata and style.

  2. To establish a suitable method for archiving, publishing, and distributing digital maps with customized styling.

  3. To determine a suitable programming and to develop a script that minimizes the human effort during the procedure.
1. Introduction

1.3 Research Questions

• Is it possible to embed relevant metadata for future usage in digital maps using GeoPackage?

• Is it possible to embed styling from GeoPackage when distributing digital maps that use vector-tile format?

• Is it possible to create a Python script that partially automate the procedures of embedding relevant information into GeoPackage?
2. Literature Review and Theoretical Background

- Principle of Digital Maps Preservation and Archiving
- Status Quo and Challenges
- Evaluation of Geospatial Data Format for Archiving Purpose
2. Literature Review and Theoretical Background

2.1 Principle of Digital Maps Preservation and Archiving

- Archiving begins at the point of data creation.
- Data should be preserved in a way that non-specialists can understand.
- Information objects should be independently understandable.
- A graphical representation of the data should be preserved with the data itself.
- Avoid restrictive assumptions about future use of the data.

Five principles for the preservation of geospatial data (Clark, 2016)
2. Literature Review and Theoretical Background

2.2 Status Quo and Challenges

- Modern cartography through digital approach requires a more complicated framework to ensure the processes of reproduction, sharing, and publishing. (Jobst, M & Gartner, G, 2011)
- Backup and storage of digital maps; The importance of storing data representation. (Lauriault et al., 2011)
- Proprietary nature of geospatial data format raises a technical risk for long-term archiving. (Clark, 2016)
- Preserve the application schemas and its relationships with the relating geospatial dataset. (Shaon et al., 2011)
- Decentralized Production; Complexity of Data; Managing Versions of the Same data. (Locher & Thermens, 2012)

The NGDA architecture at the UCSB (Erwin & Sweetkind-Singer, 2009)
2. Literature Review and Theoretical Background

2.2 Status Quo and Challenges
2. Literature Review and Theoretical Background

2.3 Evaluation of Geospatial Data Format for Archiving Purpose

2.3.1 Vector Data Format – Shapefile

- Most commonly used vector data format for archiving and distributing geospatial data
- Non-topological binary format
- Multi-File format
- 4GB size limit
- Poor support for attribute data type
- Metadata in separated data file
- No support for symbology and feature labels
- Data compression issue with data dissemination
2. Literature Review and Theoretical Background

2.3 Evaluation of Geospatial Data Format for Archiving Purpose

2.3.2 Raster Data Format – GeoTIFF

- Most commonly used raster data format for archiving and distributing geospatial data
- 32bit binary format
- Multi-File format
- 4GB size limit
- Metadata in separated data file
- No support for symbology
- Data compression and size issue with data dissemination
2. Literature Review and Theoretical Background
2.3 Evaluation of Geospatial Data Format for Archiving Purpose
2.3.2 Alternative Data Format – GeoPackage


• 64bit ASCII format
• SQLite database file – single file format
• Support for storing both vector and raster data
• 150TB storage size
• Metadata support by default with ISO 19115
• Style support through extension
• “Interoperable file between SDI and the mobile system” - Bogossian et al.,(2014)
3. Methodology and Case Study

3.1 Case Study Background and Scenario

3.2 Overview of Approach
   3.2.1 Original Dataset Review
   3.2.2 Geospatial Data Format Conversion
   3.2.3 Extracting Metadata from PDF
   3.2.4 Inserting Metadata
   3.2.5 Editing Style in QGIS
   3.2.6 Adding Style into GeoPackage
   3.2.7 Uploading GeoPackage to Geoserver
   3.2.8 Importing Style and Digital Maps Distribution
3. Methodology

3.1 Case Study Background and Scenario

Scenario 1
GIS technician: “I want to archive the national maps, but I only have the shapefile from BEV. Should I create the map from scratch?”

Scenario 2
Regular Austrian Citizen: “I want to have the city map with national style as a digital collection, do I have to buy it from BEV?”

Scenario 3
Emergency Response Event such as Etna Volcano eruption in Sicily, Italy, “Is it possible to make and share the maps in a short time period”

Case Study
The cartographic model (KM1000-V) and national map style from Austrian Mapping Agency (BEV)
3. Methodology

3.1 Overview of Approach

Original Dataset Review → Geospatial Data Format Conversion (SHP to GeoPackage) → Extracting Metadata from PDF

Inserting Metadata → Editing Style in QGIS → Adding Style into Geopackage

Uploading GeoPackage to GeoServer → Importing Style and Digital Maps Distribution

Workflow Overview
3. Methodology

3.2 Overview of Approach

<table>
<thead>
<tr>
<th><strong>WORKFLOW</strong></th>
<th><strong>SOFTWARE/PROGRAMMING LANGUAGE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Dataset Review</td>
<td>PDF Reader</td>
</tr>
<tr>
<td>Geospatial Data Format Conversion</td>
<td>QGIS 3.4.1</td>
</tr>
<tr>
<td>Extracting Metadata From PDF</td>
<td>PDF Reader/Python 3 IDLE/SQLite</td>
</tr>
<tr>
<td>Inserting Metadata</td>
<td>Python 3 IDLE/SQLite</td>
</tr>
<tr>
<td>Editing Style in QGIS</td>
<td>QGIS 3.4.1/Color Picker</td>
</tr>
<tr>
<td>Adding Style into GeoPackage</td>
<td>QGIS 3.4.1</td>
</tr>
<tr>
<td>Uploading GeoPackage</td>
<td>GeoServer 2.15</td>
</tr>
<tr>
<td>Importing Style and Digital Maps Distribution</td>
<td>GeoServer 2.15</td>
</tr>
</tbody>
</table>

Table 1. Software and Programming Language Requirement
3. Methodology

3.2 Overview of Approach

3.2.1 Original Dataset Review

Original Dataset Overview

18 SHPs, 2 PDF files, 3 DBF files, 1 TXT file, 78 files in total
3. Methodology

3.2 Overview of Approach

3.2.2 Geospatial Data Format Conversion

Package Layer Tool Box in QIS

`gpkg_contents` table in output GeoPackage using DB Browser for SQLite
3. Methodology

3.2 Overview of Approach

3.2.3 Extracting Metadata from PDF

```python
1. #Convert PDF to Text
2. #Source Code: http://stanford.edu/~mgorkove/cgi-bin/python_tutorials/Using%20Python%20to%20Convert%20PDFs%20to%20Text%20Files.php
3. def convert(fname, pages=None):
4.     if not pages:
5.         pagenums = set()
6.     else:
7.         pagenums = set(pages)
8.     codec = 'utf-8'
9.     output = StringIO()
10.    manager = PDFResourceManager()
11.    converter = TextConverter(manager, output, laparams=LAParams())
12.    interpreter = PDFPageInterpreter(manager, converter)
13.    infile = open(fname, 'rb')
14.    for page in PDFPage.get_pages(infile, pagenums):
15.       interpreter.process_page(page)
16.    infile.close()
17.    converter.close()
18.    text = output.getvalue()
19.    output.close()
20.    return text
21.
22. output = convert(BEVPDF, pages=[21])
23. print (output)
24. file = open("PDFPage22.txt", "w", encoding='utf-8')
25. file.write(output)
26. file.close()
27. print ("FINISHED Converting pdf to txt")
```

```python
1. #Convert tables in PDF to CSV
2. tabula.convert_into(BEVPDF, "DataFormatFileTable.csv", encoding='utf-8', multiple_tables=True, output_format='data_format', pages="22")
3. tabula.convert_into(BEVPDF, "tablepage23.csv", encoding='utf-8', guess=False, multiple_tables=True, output_format='data_format', pages="23")
4. tabula.convert_into(BEVPDF, "tablepage24.csv", encoding='utf-8', guess=True, lattice=True, multiple_tables=True, output_format='data_format', pages="24")
5. tabula.convert_into(BEVPDF, "tablepage25.csv", encoding='utf-8', guess=True, lattice=True, multiple_tables=True, output_format='data_format', pages="25")
6. tabula.convert_into(BEVPDF, "tablepage2627.csv", encoding='utf-8', guess=True, lattice=True, multiple_tables=True, output_format='data_format', pages="26,27")
7. tabula.convert_into(BEVPDF, "tablepage28.csv", encoding='utf-8', guess=True, lattice=True, multiple_tables=True, output_format='data_format', pages="28")
8. tabula.convert_into(BEVPDF, "tablepage2930.csv", encoding='utf-8', guess=True, lattice=True, multiple_tables=True, output_format='data_format', pages="29,30")
9. tabula.convert_into(BEVPDF, "tablepage31.csv", encoding='utf-8', guess=True, lattice=True, multiple_tables=True, output_format='data_format', pages="31")
10. tabula.convert_into(BEVPDF, "tablepage32.csv", encoding='utf-8', guess=True, lattice=True, multiple_tables=True, output_format='data_format', pages="32")
11. tabula.convert_into(BEVPDF, "tablepage33.csv", encoding='utf-8', guess=True, lattice=True, multiple_tables=True, output_format='data_format', pages="33")
12. print ("FINISHED EXTRACT Table from PDF")
```
3. Methodology

3.2 Overview of Approach

3.2.3 Extracting Metadata from PDF

4. Kartographisches Modell 1:1 Million – Vektor (K1000-V)

4.1 General structure

4.1.1 About K1000-V

Data of K1000-V is the Austrian part of EuroGlobalMap (EGM) the pan-European vector dataset at small scale. EGM Database is intended to be used in map scale 1:800 000. Detailed specifications are described in EGMDoc3. This document is a summary of the most relevant specifications.

4.1.2 Data format and file table

Table of K1000-V is stored in these files:

<table>
<thead>
<tr>
<th>Shape Files</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>KM1000_AIRFLD_POINT.shp</td>
<td>Airport / Airfield</td>
<td>Point</td>
</tr>
<tr>
<td>KM1000_DAM_LINE.shp</td>
<td>Dam/ Wair</td>
<td>Line</td>
</tr>
<tr>
<td>KM1000_ELEV_POINT.shp</td>
<td>Height point</td>
<td>Point</td>
</tr>
<tr>
<td>KM1000_GLACIER_AREA.shp</td>
<td>Glacier</td>
<td>Area</td>
</tr>
<tr>
<td>KM1000_ISLAND_AREA.shp</td>
<td>Island</td>
<td>Area</td>
</tr>
<tr>
<td>KM1000_LAKE_AREA.shp</td>
<td>Lake</td>
<td>Area</td>
</tr>
<tr>
<td>KM1000_NAME_POINT.shp</td>
<td>Named location</td>
<td>Point</td>
</tr>
<tr>
<td>KM1000_POLBND_LINE.shp</td>
<td>Administrative boundary</td>
<td>Line</td>
</tr>
<tr>
<td>KM1000_POLBND_LINE.shp</td>
<td>Administrative boundary</td>
<td>Area</td>
</tr>
<tr>
<td>KM1000_RAILRD_NODE.shp</td>
<td>Railway station</td>
<td>Point (Node)</td>
</tr>
<tr>
<td>KM1000_RAILRD_LINE.shp</td>
<td>Railway</td>
<td>Line</td>
</tr>
<tr>
<td>KM1000_RESERVOIR_AREA.shp</td>
<td>Reservoir</td>
<td>Area</td>
</tr>
<tr>
<td>KM1000_ROAD_LINE.shp</td>
<td>Road</td>
<td>Line</td>
</tr>
<tr>
<td>KM1000_BUILTUP_AREA.shp</td>
<td>Built-up area</td>
<td>Area</td>
</tr>
<tr>
<td>KM1000_BUILTUP_POINT.shp</td>
<td>Built-up point</td>
<td>Point</td>
</tr>
<tr>
<td>KM1000_SPRING_NODE.shp</td>
<td>Spring/ Water hole (connected)</td>
<td>Point (Node)</td>
</tr>
<tr>
<td>KM1000_WATRCRS_AREA.shp</td>
<td>Watercourse</td>
<td>Area</td>
</tr>
<tr>
<td>KM1000_WATRCRS_LINE.shp</td>
<td>Watercourse</td>
<td>Line</td>
</tr>
</tbody>
</table>

Info Tables

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMIN_ISN.dbf</td>
<td>This table includes the names of the administrative hierarchy levels</td>
</tr>
<tr>
<td>EGM_CHR.dbf</td>
<td>This table describes the national character sets used for each language</td>
</tr>
<tr>
<td>SHN_NAM.dbf</td>
<td>The table includes the names of the units of all administrative levels</td>
</tr>
</tbody>
</table>
3. Methodology
3.2 Overview of Approach
3.2.4 Inserting Metadata

Set up initial connection with Geopackage

Create metadata tables based on OGC GeoPackage Encoding Standard
3. Methodology

3.2 Overview of Approach

3.2.4 Inserting Metadata

- Inserting general metadata information of KM-1000 to GeoPackage
3. Methodology

3.2 Overview of Approach

3.2.4 Inserting Metadata

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Add description of layer to gpkg for KM1000_polbnd_area</td>
</tr>
<tr>
<td>2.</td>
<td>FeatureClassName = &quot;KM1000_polbnd_area&quot;</td>
</tr>
<tr>
<td>3.</td>
<td>Definition = &quot;An area controlled by administrative authority.&quot;</td>
</tr>
<tr>
<td>4.</td>
<td>EGM_Feature_Class = &quot;PolbndA&quot;</td>
</tr>
<tr>
<td>5.</td>
<td>FeatureType = &quot;Area&quot;</td>
</tr>
<tr>
<td>6.</td>
<td>PrimitiveType = &quot;Face&quot;</td>
</tr>
<tr>
<td>7.</td>
<td>PortrayalCriteria = &quot;Each administrative unit consists of one main area and occasionally of one main area with exclave(s). Exclaves bigger than 3 km² included. If a country has national administrative levels below a country level, then the lowest level in EU-countries is a level equivalent to NUTS3 level and in other countries the lowest level is comparable to this level.&quot;</td>
</tr>
<tr>
<td>8.</td>
<td>attributeTable = &quot;KM1000_polbnd_area_07_2018_description_of_attributes&quot;</td>
</tr>
<tr>
<td>9.</td>
<td>table = [FeatureClassName, Definition, EGM_Feature_Class, FeatureType, PrimitiveType, PortrayalCriteria]</td>
</tr>
<tr>
<td>10.</td>
<td>cursor.execute(DROP TABLE IF EXISTS KM1000_polbnd_area_07_2018_description)</td>
</tr>
<tr>
<td>11.</td>
<td>CREATE TABLE KM1000_polbnd_area_07_2018_description (id INTEGER CONSTRAINT m_pk PRIMARY KEY ASC NOT NULL, FeatureClassName TEXT NOT NULL, definition TEXT NOT NULL, EGM_Feature_Class TEXT NOT NULL, FeatureType TEXT NOT NULL, PrimitiveType TEXT NOT NULL, PortrayalCriteria TEXT NOT NULL)</td>
</tr>
<tr>
<td>12.</td>
<td>cursor.execute('INSERT INTO KM1000_polbnd_area_07_2018_description VALUES (?, ?, ?, ?, ?, ?)', table)</td>
</tr>
<tr>
<td>13.</td>
<td>conn.commit()</td>
</tr>
</tbody>
</table>

---

**Layer Description**

**KM1000_polbnd_area**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Definition</th>
<th>Value/Code or Example Value description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FO CODE</td>
<td>FACO feature code</td>
<td>F-AG1 Administrative area</td>
</tr>
<tr>
<td>TAA</td>
<td>Type of the administrative area</td>
<td>0 Unknown, 1 Mainland, 2 Exclave or island, 3 Condominium, 4 Water area</td>
</tr>
<tr>
<td>SHN0</td>
<td>Id-code of country-level (ISO 3166 Nation Code = number of zeros, so that fields SHN0 - SHN4 have equal width)</td>
<td>F1000000 (Example), 3XYX000 (Example) (For in dispute areas between countries XX and YY)</td>
</tr>
<tr>
<td>SHN1</td>
<td>Id code of 1st order administrative unit</td>
<td>N100000 (Example), N_A Not applicable if country has no more than the country level is (E000)</td>
</tr>
</tbody>
</table>

**Description of Attribute Table**

**BEV - Bundesamt für Eich- und Vermessungswesen**

---

**Metadata Overview of KM1000_polbnd_area Layer in BEV PDF**
3. Methodology

3.2 Overview of Approach

3.2.4 Inserting Metadata
3. Methodology

3.2 Overview of Approach

3.2.4 Inserting Metadata – INSPIRE Metadata Support

```python
#insert KM1000_polbl_line XML to gpkg
polbl_line_xml = "KM1000_POLBL_LINE_20080507-757c-4d4b-b220-ca0ef422197d.xml"

if latestInfo != "":
    sg.Popup(polbl_line_xml, 'Import Successful')
else:
    sg.Popup(polbl_line_xml, 'Import Failed', 'Program Stopped')
sys.exit('Error')

polbl_line_xml_copy = "polbl_line_xml_copy.xml"
copyfile(polbl_line_xml, polbl_line_xml_copy)
xmlformatpreserved = xml.dom.minidom.parse(polbl_line_xml_copy)
xmlformatpreserved = xmlformatpreserved.toprettyxml()
xmlformatpreserved = str(xmlformatpreserved)

cursor.execute("insert into gpkg_metadata values (?, ?, ?, ?, ?)", metadata)
conn.commit()
print("Finishing inserting xml to GeoPackage")
timestamp = datetime.datetime.now()
metadata_reference = [table, 'KM1000_POLBL_LINE_07_2018', '', '',
timestamp, 2, 1]
cursor.execute("insert into gpkg_metadata_reference values (?, ?, ?, ?, ?, ?)", metadata_reference)
conn.commit()
def replace_in_config(old, new):
    with open(polbl_line_xml_copy, 'r') as f:
        text = f.read()
    with open(polbl_line_xml_copy, 'w') as f:
        f.write(text.replace(old, new))
    return
replace_in_config('gmd:', '')
replace_in_config('gco:', '')

for child in root.iter():
    tagstring = str(child.tag)
    textstring = str(child.text)
    extractxml = str(child.tag + ' ' + textstring)
    extractxml = str(extractxml)
cursor.execute("insert into gpkg_metadata values (?, ?, ?, ?, ?)", metadata)
conn.commit()
timestamp = datetime.datetime.datetime.now()
metadata_reference = [table, 'KM1000_POLBL_LINE_07_2018', '', '',
timestamp, 3, 1]
cursor.execute("insert into gpkg_metadata_reference values (?, ?, ?, ?, ?, ?)", metadata_reference)
conn.commit()
if os.path.exists("polbl_line_xml_copy.xml"):
    os.remove("polbl_line_xml_copy.xml")
else:
    print("The file does not exist")
```
3. Methodology

3.2 Overview of Approach

3.2.4 Inserting Metadata – INSPIRE Metadata Support
3. Methodology

3.2 Overview of Approach

3.2.4 Inserting Metadata – INSPIRE Metadata Support
3. Methodology

3.2 Overview of Approach

3.2.5 Editing Style in QGIS
3. Methodology

3.2 Overview of Approach

3.2.5 Editing Style in QGIS

Additional Cartographic Representations created to match the National Map Style
3. Methodology

3.2 Overview of Approach

3.2.6 Adding Style into GeoPackage

Add Layer Style to GeoPackage in QGIS
3. Methodology

3.2 Overview of Approach

3.2.7 Uploading GeoPackage to Geoserver

1. Austria_Boundary_Buffer_Effect
2. HillShade50m
3. KM500_R_WALD_07_2018
4. KM1000_glacier_area_07_2018
5. KM1000_reservoir_area_07_2018
6. KM1000_builtup_area_07_2018
7. KM1000_watrcrs_line_07_2018
8. KM1000_polbnd_line_07_2018
9. KM1000_watrcrs_area_07_2018
10. KM1000_road_line_07_2018
11. KM1000_lake_area_07_2018
12. KM1000_builtup_point_07_2018

Layer Overlay Order
3. Methodology

3.2 Overview of Approach

3.2.8 Importing Style and Digital Maps Distribution
3. Methodology

3.2 Overview of Approach

3.2.8 Importing Style and Digital Maps Distribution
4. Results

• Archiving Digital Maps with GeoPackage

![Data Size Comparison]
4. Results

- Archiving Digital Maps with GeoPackage
4. Results

- Archiving Digital Maps with GeoPackage
4. Results

- Vector Tiles Dissemination
4. Results

- Python Script for Automatically Embedding Metadata
4. Results

- Python Script for Automatically Embedding Metadata
5. Discussion and Future Work

• Discussion

• Limited research concentrates on the disadvantage of existing geospatial data formats or proposes an alternative format for archiving digital maps.

• Limited research related to GeoPackage format.

• The default metadata table provided in GeoPackage Encoding Standard was not sufficient.

• The method designed for the case study provides a completely free, open-source, and cross-platform solution for archiving and dissemination of digital maps.

• This approach uses SLD format to store style information of feature layers and GeoServer to disseminate digital maps.

• The Python script developed for this approach is exclusively designed to extract the metadata in the PDF file of BEV KM1000-V datasets.
5. Discussion and Future Work

- Character Encoding Issue

German Umlauts Encoding Issue

KM1000-V datasets are encoded in ISO 8859-1 (Latin -1)
5. Discussion and Future Work

- Map Labeling and Symbology Issue
- Geodata Quality Issue
5. Discussion and Future Work

• Future Work

• More detailed research is needed to study benefit and drawback of different geospatial data formats used in archiving digital maps.

• Further research is needed to study metadata extension and set new standard for metadata tables in Geopackage.

• Further research is needed to examine how well proprietary GIS software and other open-source GIS software support a Geopackage that contains metadata and style information.

• It is important to study the compatibility of SLD format with other web map platforms and the conversion capability between SLD format and other style formats.

• Future work should focus on the proper method to export and share the label placement in QGIS.
6. Conclusion

- This case study, archiving the 1:1 million Austria Cartographic Model with national map style, demonstrates the possibility of using GeoPackage to embed relevant metadata from different data formats that the digital maps requires, such as general information for the whole dataset, layer description, description of attribute table, symbology, and reference system information.

- Through the last three stages in this research approach, the goal of distributing digital maps through vector-tile format has been achieved by uploading the embedded SLD style from GeoPackage to Geoserver.

- The Python script has significantly reduced the workload of embedding metadata into a GeoPackage.

In sum, this thesis proposes an innovative approach to embed relevant metadata for future usage in digital maps using GeoPackage and to embed styling from GeoPackage when distributing digital maps that use vector-tile format. It also shows the possibility of using Python scripts to automate the procedure of embedding relevant information into GeoPackage.
7. Reference


Thank you very much for your time!