TripleGeo: an ETL Tool for Transforming Geospatial Data into RDF Triples

Kostas Patroumpas[†,§] Michalis Alexakis[§] Giorgos Giannopoulos[§,†] Spiros Athanasiou[§]

[§]Institute for the Management of Information Systems
"Athena" Research Center, Hellas

[†]School of Electrical and Computer Engineering
National Technical University of Athens, Hellas

kpatro@dblab.ece.ntua.gr, {alexakis, giann, spathan}@imis.athena-innovation.gr

ABSTRACT

Integrating data from heterogeneous sources has led to the development of Extract-Transform-Load (ETL) systems and methodologies, as a means of addressing modern interoperability challenges. A few such tools have been available for converting between geospatial formats, but none specifically addressing the emerging needs of geospatially-enabled RDF stores. In this paper, we introduce TripleGeo, an open-source ETL utility that can extract geospatial features from various sources and transform them into triples for subsequent loading into RDF stores. TripleGeo can directly access both geometric representations and thematic attributes either from standard geographic formats or widely used DBMSs. It can also reproject input geometries on-the-fly into a different Coordinate Reference System, before exporting the resulting triples into a variety of notations. Most importantly, TripleGeo supports the recent GeoSPARQL standard endorsed by the Open GeoSpatial Consortium, although it can extract geometries into other vocabularies as well. This tool has been validated against OpenStreetMap layers with millions of geometries, opening up perspectives to add more functionality and to address much bigger data volumes.

1. INTRODUCTION

Nowadays, geospatial data is ubiquitous on the Web, either explicitly (through maps or satellite imagery) or implicitly (e.g., via addresses, geotagged photographs, or geolocation hashtags). Spatial information can be found in a variety of data formats, schemas, and heterogeneous platforms, systems, web services, etc. Most of this data still remains in proprietary databases and Geographic Information Systems (GIS) maintained by commercial vendors or governmental agencies. Standardization by the Open GeoSpatial Consortium (OGC) [24] or initiatives for building spatial data infrastructures under the EU INSPIRE Directive [11], pave the way towards geospatial data interoperability and dissemination. In parallel, crowdsourced geodata is rapidly emerging, and projects like OpenStreetMap [29], GeoNames [16], or Wikipedia [46] currently offer reliable, up-to-date geographic information for free.

Besides, knowledge representation and reasoning according to the Linked Data paradigm [5] is extremely useful in Semantic Web applications, such as online shopping platforms, personalized content delivery, etc. Crowdsourced initiatives like DBpedia [7] extract structured information from Wikipedia [46] and link it to other web resources. Thanks to knowledge representation models like RDF [35] and OWL [32], and query protocols such as SPARQL [39], much work is done towards transforming relational data into RDF, including standardization by the RDB2RDF W3C group [36].

Linked data technologies can also provide the means for semantic manipulation of spatial features, including interlinking, querying, reasoning, aggregation, fusion, and visualisation. With a handful of notable exceptions [2, 31, 43], only a small amount of such information has been published as *linked geospatial data* and associated with other resources in the Semantic Web. A major difficulty has to do with the inherent complexity of geospatial concepts. Apart from points, which can be simply abstracted as a pair of latitude/longitude coordinates, all other geometries require more robust representations to cope with irregular shapes (e.g., polygons with holes, or curves with multiple disconnected parts). It is also difficult to express implicit topological relationships between web resources, e.g., to provide an answer to the query "Find all subway stations within 1 km distance from my hotel". After several initial proposals, the recent OGC GeoSPARQL standard [25] suggests a unified approach for representing linked geospatial data as RDF triples and querying them through a SPARQL extension equipped with a variety of spatial operators and functions [4]. Yet, it is surprising the lack of tools for converting geospatial features from several sources into GeoSPARQL-compliant serializations.

Towards this goal, we introduce open-source utility TripleGeo [1]. Our aim is to bridge the gap between typical geographic representations from a variety of proprietary files, DBMSs, and georeference systems with the demands of geospatially-enabled RDF stores. Development was based on open-source geometry2rdf library [19], but with notable modifications and substantial enhancements to meet interoperability needs in RDF stores. In fact, TripleGeo is designed as a spatial ETL tool, enabling users to: (i) *Extract* spatial data from a source; (ii) *Transform* this data into a triple format and geometry vocabulary prescribed by the target RDF store; and (iii) *Load* resulting triples into the target RDF store. Therefore, TripleGeo always preserves data integrity and provides consistent, well-defined geospatial information to end users.

Among its distinctive features, we point out that TripleGeo can:

- Directly access de facto geographic formats (e.g., shapefiles [9]) or DBMSs (e.g., Oracle Spatial [30] or PostGIS [33]).
- Recognize many *geometric data types*, i.e., not only points, but (multi-)linestrings and (multi-)polygons as well.
- Extract *thematic attributes*, e.g., identifiers, names, or types, associated with each feature.

- Allow on-the-fly reprojection between *Coordinate Reference Systems* (CRS), e.g., transform geometries from GreekGrid87 (a local CRS) into WGS84 (used for GPS locations).

- Export triples into various *notations* (RDF/XML, TTL, etc.) and geometry vocabularies for swift loading into RDF stores.

The remainder of this paper proceeds as follows. In Section 2, we survey related work on specifications and tools for converting relational and geospatial data into RDF. In Section 3, we present TripleGeo's architecture, by examining its components and processing flow, along with its dependencies on third-party platforms and libraries. In Section 4, we discuss the current implementation status and planned extensions for future releases of TripleGeo.

2. RELATED WORK

Creating knowledge from structured (e.g., relational databases, XML) or unstructured sources (e.g., text, images) can be extremely valuable in the Semantic Web. The R2RML Recommendation [37] by W3C specifies an RDF notation for mapping relational tables, views or queries into the RDF data model. Among the thirty tools for knowledge extraction reviewed in [42], the majority are considered as proof-of-concept prototypes. Some of them are rather "mature" tools for transforming relational databases into RDF, such as Triplify [3], D2R Server [8], or Virtuoso's RDFizer Middleware (Sponger) [27]. During conversion, these tools allow reuse of existing vocabularies and ontologies. Although under development, the Google Refine RDF Extension [38] seems promising, and can reconcile against SPARQL endpoints and RDF dumps. However, none of the aforementioned methodologies and tools currently provides any particular support for geospatial data and operations.

On the other hand, several ETL tools can manage the unique characteristics of spatial data. Among them, *GDAL/OGR* [13] is an open-source translator library implementing the OGC vector model [24] and can handle proprietary storage models for many geospatial DBMSs and files. *GeoKettle* [40] is a metadata-driven spatial ETL tool dedicated to integration of different data sources for building and updating geospatial data warehouses. Finally, *FME Workbench* is included in ESRI's ArcGIS Data Interoperability extension [10] and enables transformation of geometric and thematic attributes along with schema redefinitions. Currently, such utilities are mainly used for data cleaning, merging, verification or conversion into various formats, but have absolutely no RDF support.

There have been several proposals for geospatial RDF data management such as [14, 17, 47], but none provided a solid framework for developing large-scale applications and services. Recently (2012), the OGC GeoSPARQL standard [25] suggests a concrete ontology for representing features and geometries in RDF as Well Known Text (WKT) or Geography Markup Language (GML) literals. GeoSPARQL defines a core set of classes, properties and data types that can be used to construct query patterns in an extension of SPARQL. To cope with incompatible methods for representing and querying spatial data, GeoSPARQL follows other OGC standards [24]. With such standardization, both vendors and users can achieve uniform, transparent, platform-independent access to geospatial RDF data with a rich collection of query operators. Currently, only few RDF stores like Parliament [34] or uSeekM [28] have partially implemented GeoSPARQL specifications. Instead, several geospatially-enabled triple stores prefer proprietary geometric representations (e.g., AllegroGraph [12]) or restrict their support to points only (such as OWLIM [23] or Virtuoso [26]).

To the best of our knowledge, there have been very few initiatives specifically for converting geospatial features into RDF resources. Data conversion into an appropriate RDF format using a selected ontology is among the functionalities supported by the generic *DataLift* platform [6]. Although geometries can be extracted as WKT strings under a custom namespace, there is currently no support for GeoSPARQL. *LinkedGeoData* [2] aims at adding a spatial dimension to the Semantic Web. It offers a flexible platform for mapping OpenStreetMap (OSM) data [29] to RDF, a SPARQL endpoint for making RDF data publicly available, as well as several tools for performing mappings and interlinking of geospatial semantic data. The resulting graph comprises more than 20 billion triples interlinked with DBpedia [7] and GeoNames [16]. Nevertheless, spatial operations deal strictly with OSM nodes and ways, ignoring any other geographic sources or data types.

In parallel, *Geo.LinkedData.es* is an open initiative to enrich the Web of Data with geospatial data for Spain. Among the tools they developed, geometry2rdf [19] enables extraction of geometries as RDF triples [44]. Geometries can be available in GML or WKT serializations and are manipulated with GeoTools [18], not only in order to retrieve features, but also to perform coordinate transformation. However, its RDF model is not compliant with the GeoSPARQL standard [25], and cannot handle attribute values (e.g., name literals) or export triples in various notations apart from RDF/XML. Concerning interaction with geospatial sources, it only supports geometry extraction from shapefiles [9] and Oracle Spatial [30]. Despite these important deficiencies, this open source library provided an initial base for developing our own utility TripleGeo. As we explain next, we particularly aim at integrating several external databases and providing support for GeoSPARQL data types.

3. CONVERTING GEOMETRIES TO RDF

In this Section, we present the architecture and capabilities of ETL tool TripleGeo for converting vector geospatial features into RDF triples. This process iterates through all features in the input dataset and emits a series of triples per record. Every geometric feature is converted into properly formatted triple(s), according to the specified RDF vocabulary. Most typically, geometries are turned into WKT serializations as prescribed by GeoSPARQL [25], but some legacy namespaces are supported as well. Thematic attributes can be extracted in tandem, such as identifiers, names, or classifications. Results are written into a file using standard triple notations, so that they can be readily loaded into an RDF store.

3.1 Integrated Libraries

TripleGeo inherits from geometry2rdf dependencies to various open-source tools and libraries, all of which are used "as is". The most significant of these libraries are:

- *Apache Jena* [21] is a widely used Java framework for developing Semantic Web applications, tools and servers.

- *GeoTools* [18] offer Java implementations of OGC specifications [24] for geospatial data management comparable to GIS desktop applications and web services. Its rich API supports feature access to many file formats (like CSV, DXF, GeoJSON, ESRI shapefiles, etc.) and spatial DBMSs, as well as coordinate transformations between CRS.

- *GDAL/OGR*. We actually make use of OGR Simple Features embedded in this Geospatial Data Abstraction Library [13]. This includes command-line tools for read access to a variety of *vector* formats (shapefiles, PostGIS, Oracle Spatial, etc.).

- *Java Topology Suite* (*JTS*) [45] is an open source API that provides support for 2-dimensional topological predicates and spatial functions conforming to OGC [24].

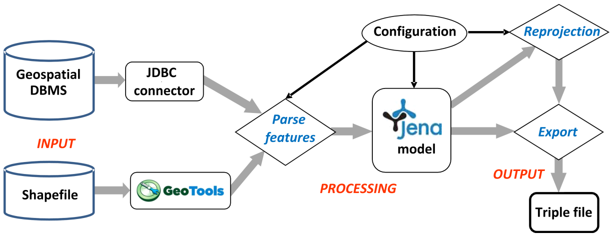

Figure 1: Processing flow diagram for ETL utility TripleGeo.

3.2 TripleGeo Components

TripleGeo has been implemented with several Java classes that perform specific tasks in a modular fashion. From a user's perspective, this command-line utility is entirely automated according to preconfigured settings. Figure 1 illustrates the flow diagram used for converting geospatial features into RDF triples. Next, we outline the basic components of the utility:

- *Input* data may be obtained either from geographic files or geospatially-enabled DBMSs, as explained in Section 3.3.
- *Connectors* to source data are required in order to access geometric features. In case of a DBMS, this is possible thanks to suitable JDBC drivers. With respect to file formats, the integrated GeoTools library provides all required functionality.
- A *configuration file* declares user preferences concerning all stages of the conversion: how input source will be accessed, which data is involved, what geometric representation should be used, whether geometries must be reprojected into another CRS, as well as the output triple notation.
- A *parser* iterates through each input record and converts geometries into a suitable representation according to user specifications. It also consumes thematic attribute values (e.g., types, names) and emits properly formatted literals.
- A *Jena model* is a main-memory data structure that is used to retain all state information consisting of the collection of generated triples. This model denotes an RDF graph, so called because it contains a collection of RDF nodes, attached to each other by labelled relations. In Java terms, this model acts as the primary container of RDF information in graph form. A significant benefit from using the Jena model is that it offers a rich API with many methods intended to make it easier to write RDF-based applications.
- Optionally, *reprojection* of geometries into another spatial reference system is possible. This transformation is carried out thanks to the integrated GeoTools library and according to user specifications for the source and target CRS.
- *Export* of generated triples into a single file is performed by the Jena API. This offers the possibility of writing the output into several triple formats, as discussed in Section 3.4.

3.3 Input

Current version 1.0 of TripleGeo can access geometries from:

- ESRI shapefiles [9], which is a well-known format for storing geospatial features in files.
- Geospatially-enabled DBMSs, such as: IBM DB2 with Spatial Extender [20], MySQL [22], Oracle Spatial and Graph [30], and PostGIS (spatial module for PostgreSQL) [33].

Geometric data must reside in a single table (in case of a database) or a file. Combining thematic information from multiple tables (e.g., via joins) from the same source is also available. However, it is not currently possible to concurrently process data from diverse sources or formats. Attributes (i.e., table columns) that can be extracted from a given data source include:

- The *geometry* itself (mandatory), expecting valid, georeferenced points, (multi-)linestrings, and (multi-)polygons according to the OGC specification [24].
- A unique *identifier* (mandatory) for each entity, which will be used to generate the IRI of the extracted resource.
- Optionally, a *name* value associated with an entity can be converted into a string literal.
- Optionally, a *type* value that characterizes an entity can be associated with an `rdf:type` predicate.

3.4 Output

In terms of output serializations, and according to the specifications of the Jena API [21] that exports the model, the triples can be obtained in one of the following notations: *RDF/XML* (default), *RDF/XML-ABBREV*, *N-TRIPLES*, *N3*, and *TURTLE* (*TTL*).

In terms of standardization, output triples are conformant to W3C standards, thanks to Jena API methods for creating resources, properties and literals, and statements linking them. Therefore, all output triples are compatible with the most commonly used standards, including RDF, RDFS, OWL, and SPARQL.

With respect to geospatial features, triples can be exported according to the GeoSPARQL standard [25]. TripleGeo also supports legacy namespaces, such as `pos:` of the W3C Basic Geo Vocabulary [47] or Virtuoso's `virtrdf:` for custom point geometries [26]. But note that such syntaxes are neither compliant to GeoSPARQL nor can they handle shapes other than points.

Basically, the output geometry serialization depends on the RDF store where triples will be loaded afterwards. Parliament [34] and uSeekM [28] only accept GeoSPARQL-compliant triples. Virtuoso [26] requires its own custom syntax and currently handles point features only. OWLIM [23] supports only points under the W3C Basic Geo Vocabulary. Other RDF stores like Oracle [30] or Strabon [41] are close, but not fully conformant to GeoSPARQL, mainly due to differing namespaces. In that case, geometries can be extracted into GeoSPARQL and then replace the necessary prefixes.

4. CURRENT STATUS AND OUTLOOK

TripleGeo is free software and its current version 1.0 is available from [1], including the Java source code and sample data. We provide distributions in both platform-neutral Java JARs and Debian-specific DEB packages. TripleGeo can be redistributed or modified under the terms of the GNU General Public License. This tool is also integrated into `stack.linkeddata.org`, which comprises many utilities for managing the lifecycle of Linked Data.

We have tested TripleGeo with diverse input formats, RDF stores, data sources, and geometric serializations. In one test case scenario, OpenStreetMap layers [29] for Great Britain were converted into more than 25 million triples, including 3.5 million geometries (points, polylines, and polygons). TripleGeo can readily accept shapefiles from OSM dumps as input and convert them into RDF triples. However, these original shapefiles were also imported into databases hosted in PostGIS [33] and Oracle Spatial [30]. Thus, we have also conducted ETL operations from these spatial DBMSs into triples, and we were able to verify that TripleGeo can also interact and access spatial features from major DBMSs. Apart from verifying its functionality, we also performed some more comprehensive tests by converting large datasets into triples; a detailed evaluation is available at [15]. Not only has such testing proven the robustness of the tool, but the differing geospatial specifications of each RDF store also guided development and progressive refinement towards handling as many cases as possible.

TripleGeo is still a work-in-progress. Thanks to its modular implementation, more utilities are under development without affecting existing functionality, including interaction with more geographic data sources (e.g., GML, KML, etc.) and DBMS platforms, as well as support for more complex geometric types (e.g., geometry collections [24]). We also plan to expose the full functionality of TripleGeo via a RESTful API (e.g. for web-accessible data), and also offer a web interface to upload, convert and download large datasets. As scalability with increasing data volumes is most challenging, a possible solution would be to automatically split the input into disjoint batches and use a parallelization scheme like MapReduce to generate triples. Last, but not least, ability to define mappings and vocabularies and export geometric and thematic values under a user-specified ontology would be noteworthy. As a test case, we have begun developing an ETL methodology for INSPIRE-compliant [11] data and metadata.

5. ACKNOWLEDGEMENTS

This work was partially supported by the European Commission under EU/FP7 grant #318159 for project *"GeoKnow: Making the Web an Exploratory Place for Geospatial Knowledge"*.

6. REFERENCES

[1] Athena R.C. TripleGeo open source utility. URL: https://github.com/GeoKnow/TripleGeo

[2] LinkedGeoData project. URL: http://linkedgeodata.org

[3] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller. Triplify: Light-weight linked data publication from relational databases. In *WWW*, pp. 621-630, April 2009.

[4] R. Battle and D. Kolas. GeoSPARQL: Enabling a Geospatial Semantic Web. *Semantic Web Journal*, 3(4):355-370, 2012.

[5] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *IJSWIS*, 5(3): 1-22, 2009.

[6] DataLift project. URL: http://datalift.org/

[7] DBpedia.URL: http://dbpedia.org

[8] D2R Server. URL: http://d2rq.org/d2r-server

[9] ESRI Inc. Shapefile Technical Description. URL: http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

[10] ESRI Inc. FME Workbench for ArcGIS Data Interoperability URL: http://www.esri.com/software/arcgis/extensions/datainteroperability/key-features/spatial-etl

[11] European Commission. INSPIRE Directive – Infrastructure for Spatial Information in the European Community. URL: http://inspire.jrc.ec.europa.eu/

[12] Franz Inc. AllegroGraph Triple Store. URL: http://www.franz.com/agraph/allegrograph/

[13] GDAL/OGR library. URL: http://www.gdal.org/

[14] GeoJSON 1.0. URL: http://geojson.org/

[15] GeoKnow Deliverable D2.2.1: Integration of External Geospatial Databases. URL: http://geoknow.eu/t2-2.html

[16] GeoNames database. URL: http://www.geonames.org/

[17] GeoRDF Profile. URL: http://www.w3.org/wiki/GeoRDF

[18] GeoTools library. URL: http://www.geotools.org/

[19] Geo.LinkedData.es Team. geometry2rdf Utility. URL: https://github.com/boricles/geometry2rdf

[20] IBM DB2 Spatial Extender. URL: http://www.ibm.com/software/products/us/en/db2spaext/

[21] Apache Jena project. URL: http://jena.sourceforge.net/

[22] MySQL Database. URL: http://www.mysql.com/

[23] Ontotext AD. OWLIM Semantic Repositories. URL: http://www.ontotext.com/owlim

[24] OGC Inc. Implementation Specification for Geographic Information - Simple Feature Access. URL: http://portal.opengeospatial.org/files/?artifact_id=25354

[25] OGC Inc. GeoSPARQL Standard - A Geographic Query Language for RDF Data. URL: https://portal.opengeospatial.org/files/?artifact_id=47664

[26] OpenLink Software. Virtuoso Universal Server. URL: http://virtuoso.openlinksw.com/

[27] OpenLink Software. Virtuoso's RDFizer Middleware (Sponger). URL: http://docs.openlinksw.com/virtuoso/virtuososponger.html

[28] OpenSahara uSeekM library. URL: https://dev.opensahara.com/projects/useekm/

[29] OpenStreetMap project. URL: http://www.openstreetmap.org/

[30] Oracle Inc. Oracle 12c Spatial and Graph. URL: http://www.oracle.com/technology/products/spatial

[31] Ordnance Survey. Linked Data Platform. URL: http://data.ordnancesurvey.co.uk/

[32] OWL Web Ontology Language: http://www.w3.org/TR/owl

[33] PostGIS - Spatial and Geographic Objects for PostgreSQL. URL: http://postgis.net/

[34] Raytheon BBN Technologies Inc. Parliament Triple Store. URL: http://parliament.semwebcentral.org/

[35] Resource Description Framework Primer. URL: http://www.w3.org/TR/rdf-primer/

[36] RDB2RDF Working Group. URL: http://www.w3.org/2001/sw/rdb2rdf/

[37] R2RML: RDB to RDF Mapping Language. URL: http://www.w3.org/TR/r2rml/

[38] RDF Refine: a Google Refine extension for exporting RDF. URL: http://refine.deri.ie/

[39] SPARQL 1.1 Query Language for RDF. URL: http://www.w3.org/TR/sparql11-query/

[40] Spatialytics.org. GeoKettle Spatial ETL tool. URL: http://www.spatialytics.org/projects/geokettle/

[41] TELEIOS EU/FP7 project. Strabon prototype. URL: http://strabon.di.uoa.gr/

[42] J. Unbehauen, S. Hellmann, S. Auer, and C. Stadler. Knowledge Extraction from Structured Sources. In *Search Computing III*, pp. 34-52, 2012.

[43] U.S. Geological Survey. Building Ontology for the National Map. http://cegis.usgs.gov/ontology_userguide.html

[44] L.M. Vilches-Blázquez, B. Villazón-Terrazas, V. Saquicela, A. de León, O. Corcho, and A. Gómez-Pérez. GeoLinked Data and INSPIRE through an Application Case. In *ACM SIGSPATIAL GIS*, pp. 446-449, November 2010.

[45] Vivid Solutions Inc. JTS Topology Suite. URL: http://www.vividsolutions.com/jts/JTSHome.htm

[46] Wikipedia. URL: http://wikipedia.org

[47] W3C Basic Geo (WGS84 lat/long) Vocabulary. URL: http://www.w3.org/2003/01/geo/