API Issues in Linked Geo Data

Rob Warren
Big Data Institute, Dalhousie University
Halifax, Canada
rhwarren@dal.ca

David Evans
School of Computing and Mathematics
University of Derby, UK
d.f.evans@derby.ac.uk

ABSTRACT

This paper reports on the experiences of building a linked geo data coordinates translation API and some of the issues that arose in the process. Beyond the basic capacities of SPARQL, a specialized API was constructed to translate obsolete British Trench Map coordinates from the Great War into modern WGS84 reference systems. Concerns over current methods of recording geographic information along with accuracy and precision of information are discussed. Open questions about managing the opportunistic enrichment of geographical instances are discussed as well as the scalability pitfalls therein.

Keywords

LOD, Linked Geo Data, Semantic Web, British Trench Coordinates

1. INTRODUCTION

This paper reports on the British Trench Map coordinate system API[1] in use by the Muninn Project[2] for translating the coordinate system used by British Empire troops on the Western Front during the Great War. With the start of the centenary of the war there is renewed interest in the locations represented by these coordinates both to locate historical events and to cross-link the contents of online archives about the war.

We begin with a short background about the unique challenges of the coordinate system and building an API to translate them into a modern equivalent. Specific use cases for this API are reviewed and several ongoing issues identified during its creation and initial use are expanded upon. We close with specific questions that need further inquiry.

[1]The actual API is at `http://rdf.muninn-project.org/api/TrenchCoordinates`, but a human readable application is at `http://rdf.muninn-project.org/TrenchCoordinates.html`.
[2]`http://www.muninn-project.org/`

LGT14 '14 London, UK

2. BRITISH TRENCH MAP API

With the invasion of Belgium by the Germans in 1914, the Belgian plates for the base country maps were evacuated to England where the Ordnance Survey used them as the basis for a new series of small scale maps. Using the Bonne projection and a Delambre ellipsoid the projection of the trench maps was metric, while the gridding system used for coordinates was in yards. While the reason for this unit mismatch are lost in history, it meant that in several cases duplicate trench coordinates exist and others overlap.

The specifics of the coordinate systems are reviewed in other documents [1, 14, 3] but it consists of an alphanumeric string read left to right with increasing accuracy. Most recorded coordinates result in a 50 yard sided square, through rectangular coordinates occur for less precise coordinates. As an example: the location of a trench coordinate such as 27.L.22.d.6.3 would be a 50 yard sided box with centroid 50.8300, 2.7005.

At maximum accuracy, a limit of 5 yard sided squares was possible and occasionally recorded. Depending on the sources used to create new map sheets or update the original Belgian ones, the accuracy of the maps would vary dramatically. A complete update of a map by a ordnance survey team would be expected to be accurate within 20 yards.

The origin of the projection is supposed to be the old Brussels observatory, which moved several times and the exact coordinates of the origin remains a point of contention. Positions officially recorded by Mugnier [10] as 50°25'0.0006", 4°22'12.6978" and Winterbotham [14] (see also Close [3]) as 51°10'06.895", 4°22'05.89" are both kilometres off. A recalculation from the original Belgium Triangulation (1867, [2]) by the Belgium Geographical Institute yielded an origin of of 50°24', 4°22'5.89" and after adjustment using several referenced church steeples, 50°23'57.2418", 4°22'10.0518" currently yields results that are satisfactory.

The origin of the projection is important because it is used to calculate a conversion between the Bonne projection and the WGS84 datum. One on the interesting elements that has caused not a little amount of frustration on the part of the authors is the uneven precision of the maps and the difficulty in obtaining precise location information for referenced land marks within the maps. The angle of observation of the overhead imagery provided by Bing and GoogleEarth tends to induce errors when trying to locate church steeples precisely and makes make the resolution of the origin difficult.

All of the different uncertainties with the trench coordinates make for a conversion process that is at times tedious.

It is interesting to note that the original users had not such problem since officers would reference their maps visually and whatever error would have existed would be cancelled through the use of the same printed map.

3. USE CASES

Several use cases has been identified as ancillary to the API that are presented here.

3.1 Coordinate Translation

Coordinate translation is the primary objective of the API. The trench coordinates have not been used in over a century and lost to the average person that has come to expect a GPS receiver to be used with any navigation. The ability to move from one coordinate system to another is an obvious benefit of a geo API.

The API is paired with an ontology[3] containing the different instances of all map sheets used within the coordinates system, the relationships that bind them and the underlying organization of the coordinate system. The ontological structures borrow heavily from the modern British Ordnance Ontologies [4] as well as from other ontologies.

A side benefit of publishing an ontology capable of handling native trench map coordinates is that the locations can be referenced without committing to a longitude/latitude translation. This allows authors of semantic web datasets to use coordinates as a means of locating additional information at that exact location, nearby or within a greater geographic areas.

3.2 Grid Conversion and Overlap

While end-users expect their navigation to be something resembling that a GPS will understand, geographers have been working for a long time at creating various reference grids for both navigation and location purposes in specialised application.

Each country has its own gridding and datum system that is optimized for its geography, demographics and culture. In many cases, this same gridding system is what is used as sheet lines for official government maps of the areas in question. The API provides the numbered sheets of the topographical maps of Belgium as currently published by the National Geographic Institute of Belgium. This lets the end user locate the additional representation of the location for further research.

Other gridding systems include the one used by the Kriegsmarine in the Second World War for communicating operating areas. While dedicated to sea locations the grid touches on coastal areas and is of interest as any shore activities are recorded in this manner. Reporting overlap in between grids from different temporal eras is a useful means of identifying areas of mutual interest across topics.

3.3 Working Set Retrieval

In a number of applications such as mash-ups, explorations or those requiring a statistical datum a complete set of features from the area is required. Currently, this is most exemplified by the creation of non-linked open data APIs such as the OSM Overpass[5] that allow clients to retrieve all features within a bounding box.

In the case of the trench map API, only one coordinate is converted at a time. However, as the ontological structure used by the API record both the coordinate and the respective longitude and latitude. This allows for the locations of the Great War to be used as well as modern coordinates systems.

The advantage of these APIs is that all of the required data is retrieved in one transaction. Usually when a client is faced with a server whose query engine is limited a strategy of flooding the server with multiple queries is used which can results in an overload of the server. If the client interleaves the requests, then the time delay cost due to latency makes the query impractical.

This is not an indication of the need for more complex query languages beyond that of SPARQL as much as the need for a streaming mode for SPARQL queries that allows for very large queries that retrieve a "working set" of information.

4. ONGOING ISSUES

In this section, we review the issues involved with running a linked geo API given current approaches and standards.

4.1 Precision and Accuracy

A trench map is made from several different sources of mapping information: on site surveys, larger scale maps (1:100,000), fire direction maps and the original 1:40,000 Belgium grid plates. The precision one can expect of the map varies wildly depending on the source used to create it. The re-projection of large scale maps (1:100,000) down to smaller (1:40,000) scale was performed often at the beginning of the war and in these cases one could expect errors of about 200 yards. A survey units making maps from sightings or aerial photographs would achieve a precision of about 20 yards.

The position of a grid square is communicated using a series of geo:Point instances, one for each of the vertices of the shape, as well as an additional point at the centroid of the shape which can be used for placing labels. Calculating the theoretical precision of a transformation from a longitude, latitude point to a grid square is straightforward because the error can be determined from both significant figures and the physical size of the grid square. Documenting the precision is still problematic; there currently exists no standardized way of reporting precision information beyond the terms provided by the Semantic Sensor Ontology[6]. Currently precision information is reported through it and the Provenance Ontology[7]. A means of standardizing the addition of precision information to the specific geo:Point instances without referencing either the upstream processes or sensor would be beneficial for data sharing purposes.

As previously explained, the trench maps were constructed from many different sources of information and calculating the accuracy of any coordinate translation is non-trivial. Trench coordinates were reported by Army officers from the paper map in front of them, but the registration of the feature would vary across each series of a released map. We can mathematically transform a coordinate to a longitude, latitude pair and keep track of the error due to significant

[3] http://rdf.muninn-project.org/ontologies/btmaps
[4] http://data.ordnancesurvey.co.uk/
[5] http://wiki.openstreetmap.org/wiki/Overpass_API
[6] http://www.w3.org/2005/Incubator/ssn/wiki/SSN
[7] http://www.w3.org/TR/prov-o/

figures. But the error due to the registration error of features on the map is harder to quantify since it can also vary across map series.

The irony is that this problem is brought upon by the use of technology; officers using these maps during the war would never have a problem since map series would be the same across organizational units and registration errors would thus be ignored. Recomputing the actual location that they were referencing depends as much on what feature was drawn on the map where as it does on the mathematical transformation.

A method that is used to resolve this issue is the reuse of the reference points used to derive the origin of the Bonne projection. By tracking the accuracy of the computed coordinate transformation against the actual position of the reference points we can get an estimate of the map registration error in the area of a trench coordinate. As with precision information above, reporting this information to API user is still not standardized from a liked geo perspective.

Currently, this information is reported using the ssn:Accuracy term from the Semantic Sensor ontology which is still in the incubator stage. In some cases, there is sufficient information about the coordinate systems and maps series that a heat-map of the different probability areas can be reported. This style of data reporting is useful in risk analysis applications, such as located forgotten ammunition depots and an appropriate means of reporting it using linked data is still an open question.

4.2 Revisiting OpenGIS

An ongoing concern with linked geo data is the use of the OpenGIS[8] standard which simply envelopes previous geo encodings using "well known string" properties. This was done initially to enable the reuse of existing code-bases as is without taking in into account the possibilities of RDF and OWL.

We believe that this is a lost opportunity to get the most out of the Semantic Eeb in that the current standards are burying the building blocks of geo shapes (point, linestring, etc...) in un-referenceable literals. This also means that massive duplication is needed to make the data as useful for different users of the data. The individual points that make up a track, polygon or other shape are not accessible as instances in RDF or OWL and must be parsed out of the encoded string outside of linked open data space. This prevents the reuse of the basic point information for multiple uses. One can easily imagine that the geo:Point that makes up a polygon also has accuracy, time-stamp and provenance information attached to it.

The sensors that report on the individual way-points of a track might not be the same and the ability to separate them while keeping them within an aggregated object has obvious value. This separation of the geographic components from the main body of data dates back to relational database extensions and while there were reasons for this at the time, we have an opportunity to move beyond this.

Given the semantic and reasoning hooks that are being built-into the semantic, we believe that this is a mistake and a lost opportunity to get the most out of the semantic web. OWL provides facilities for basic reasoning and it would be unfortunate not to align geographical reasoning with it. A geo:Point should be partOf a Track or a Line or within a closed Polygon. This opportunity has been written about and partially implemented in both the modern Ordnance Survey Ontology [7] and the current Linked Geo Data [13] exports from the Open Street Map. To our mind a sophisticated SPARQL engine should be able to infer contains and within triples through an inference of both typing information and shape information.

[8] http://www.opengis.net/

4.3 Data enrichment - How much is too much?

While the API provides the numbers and references to official topographical maps, other enhancements are possible. Locating different depictions of the "lost locations" being reported on in the API are useful as it points to further resources that can be used to study a feature in a very specialized field where finding additional information is likely to be very labour intensive.

The API currently concerns itself with the battlefields of the Western Front exclusively. Deriving the country in which a coordinate is within is straightforward and this opens the possibility of developing serendipitous data enrichment from the country's national library.

The French National Library publishes extensive cataloguing data as part its experimental data portal[9]. This permits us to perform a concurrent search within its catalogue for maps matching the area of interest to the client before, during and after the war. Pointers to the instances of maps and works that depict that area of France are thus returned.

A similar process can be used to retrieve images of the requested coordinates, using the Flickr^{TM}wrapper[10] developed by by Chris Bizer and Christian Becker. Appropriate coordinates and radius for the search are derived from the trench coordinates and the derived precision and accuracy at that location. The usefulness of such add-on are obvious for mash-ups that wish to immediately display images of the location for aesthetic purposes or to provide visual confirmation to the client user that the location is the appropriate one.

Given the amount of potential linkages that can be serendipitously created, even obviously relevant linkages, what is an appropriate number of serendipitous linkages to return? One can let the client decide with a parameter, through a self-tuning method would be preferable.

4.3.1 Ideal packet size

Network links have a *maximum transmission unit* (MTU) size which reflects the largest packet that the link can carry. Messages larger than this must be broken into multiple packets, network protocols being required to effect this so-called fragmentation and reassembly. IPv4's approach is described by Postel [11]. If data must traverse several network links en route from sender to receiver, the smallest MTU along the path determines the size of a message that may be sent without being fragmented along the way. Given that such fragmentation and reassembly by routers consumes resources, can cause problems for higher-level protocols and is forbidden in implementations of IPv6 [4], Internet hosts are encouraged to use "Path MTU discovery" (PMTU discovery) [9] to determine the maximum size of network packets to be sent. The PMTU has the expected effect on higher-level network protocols by limiting the amount of applica-

[9] http://data.bnf.fr/

[10] http://wifo5-03.informatik.uni-mannheim.de/flickrwrappr/

tion data they can transmit in one packet. In the Internet, TCP is often used and IPv4 and IPv6 header sizes mean that TCP's Maximum Segment Size (MSS) tends to be 40 bytes smaller than the PMTU for an IPv4 connection and 60 foe IPv6 [12, 11, 4].

We expect the network stack in the server's operating system to perform PMTU discovery to ensure correct operation and therefore set TCP's MSS to a workable value. However, the geo server software can play its part in effective use of resources. There is potential for the server to select a representation for the response and to control the volume of meta-data that is attached. Through these two processes the server can adjust the size, in bytes, of the response. Scalability concerns or the cost of resources to the server operator may make it wise, whenever possible, to fit API call replies into one network packet. In other words, the server should adapt its responses to the PMTU that it observes and the consequent MSS.

Luckie and Stasiewicz explain why, in the Internet, PMTU discovery is not always possible [8]. This means that the server cannot depend on it to guide its behaviour, though it should be used if available. At other times the server will have little idea of the actual PMTU. The experiments from 2010 by Luckie and Stasiewicz found an MSS of 1460 bytes for about 86% of IPv4 paths (and 1440 foe 85% of IPv6 paths). Assuming that the server operates over HTTP and considering a minimal response overhead of 18 bytes [6, §6], the server can with some confidence arrange for the response to be at most 1442 bytes for IPv4 (and 1422 for IPv6). However, this is very much a rough guide as PMTU discovery should be used when possible and HTTP/1.1's use of one TCP connection for multiple requests and responses means that there is no guarantee of the response stating its own TCP packet.

The Trench Map API will report about 3500 bytes for a basic grid reference, which can be compressed to about 962 bytes - easily contained within a single packet. Optimally, a single request from the API would return a single packet. But this would not be sufficiently large for enriched data to be available. Would it be possible to select the amount of serendipitous enrichment created based on network traffic? Given this, and the obvious benefit in providing linkages, can we build a self tuning system that reacts to operating conditions?

4.4 Mass Retrieval Over an Area

One of the current uses of the Trench Map API and its related data-sets is its use to create terrain for ultra realistic simulations of the Great War. Within an ancillary data-set, the used the data set from Shuttle Radar Topography Mission [5] to create a series of geo:Point's reflecting the elevation sensed at that location. Through the use of SPARQL, it is possible to only retrieve the subset of the topography wanted while averaging out the measurements to desired scale factor. Ordering of the information in a sequence that lends itself to building the base plane of the world within the simulation is also accomplished through SPARQL commands.

The ability to construct the data behind the landscape at a scale and within the bounding box of the area of interest is valuable. It allows the client to focus on its simulation while asking for exactly the information needed in a format that is usable. It may be useful to look at extending the SPARQL functions to create packed binary fields from query results on the fly. This would allow clients to specific the binary encoding expected by their engine without requiring specialized API for this purposes.

5. CONCLUSION

In this paper we presented an API used for the translation of obsolete military coordinates from the Great War into a modern coordinate system. The issues is processing these requests using a linked open data approach and the issues encountered where reported on. Additional problems are being encountered in the binarization of Great War Trench Maps using the system that deal with the identity of features when depicted in multiple maps. In order for use to be in a position to tackle these problems, the items highlighted within this position paper must be resolved.

Acknowledgements

The authors wish to thank Pierre Voet for his help with the Belgian triangulation and Bill Frost for access to his database of church steeples.

6. REFERENCES

[1] Howard Anderson. *How to find a point on a Great War trench map*. The Western Front Association.

[2] Institut cartographique militaire. *Triangulation du Royaume de Belgique*. Number v. 2-3 in Triangulation du Royaume de Belgique. Cnophs, 1867.

[3] Charles Close, Lieut.-Colonel Jack Keeling, Lieut.-Colonel Salmon, Colonel McCaw, Captain, Captain Jones, C. B. Simonds, and H. S. L. Winterbotham. British survey on the western front: Discussion. *The Geographical Journal*, 53(4):pp. 271–276, 1919.

[4] S. Deering and R. Hinden. Internet Protocol, version 6 (IPv6) specification. RFC 2460, December 1998.

[5] Tom G. Farr, Paul A. Rosen, and et al. The shuttle radar topography mission. *Reviews of Geophysics*, 45(2):n/a–n/a, 2007.

[6] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol—HTTP/1.1. RFC 2616, June 1999.

[7] John Goodwin. Experiences of using owl at the ordnance survey. In *Proceedings of the OWLED*05 Workshop on OWL*, volume 188. CEUR-WS.org, 2005.

[8] Matthew Luckie and Ben Stasiewicz. Measuring path mtu discovery behaviour. In *10th Conference on Internet Measurement*, pages 102–108, 2010.

[9] J. Mogul and S. Deering. Path MTU discovery. RFC 1191, November 1990.

[10] Clifford J. Mugnier. The kingdom of belgium. *Photogrammetric Engineering & Remote Sensing*, pages 956–957, October 1998.

[11] J. Postel. Internet protocol. RFC 791, September 1981.

[12] J. Postel. Tcp. RFC 793, September 1981.

[13] Claus Stadler, Jens Lehmann, and et al. Linkedgeodata: A core for a web of spatial open data. *Semantic Web Journal*, 3(4):333–354, 2012.

[14] H. S. L. Winterbotham. British survey on the western front. *The Geographical Journal*, 53(4):pp. 253–271, 1919.