Description of Application based on the Alexandria Digital Gazetteer Protocol*

Martin Doerr [1] and Manos Papagelis[1, 2]

[1] Institute of Computer Science, Foundation for Research and Technology - Hellas
P.O. Box 1385, GR-71110, Heraklion, Greece
{martin, papaggel}@ics.forth.gr
[2] Department of Computer Science, University of Crete
P.O. Box 2208, GR-71409, Heraklion, Greece

Introduction

Most of the information available digitally, as well as, information in library catalogues, bibliographic indexes, and museums has some reference to locations or objects in geographic space. However, only a fraction of it is properly *georeferenced*, i.e. indexed using a geographic coordinate system. In most cases, simple place names and other non-formalized descriptions are used. Besides, between data sources different place names (variant place names) are used to describe identical places. The key to merging these data sources with the geospatial data and thus toward the promise of the Digital Earth metaphor for information management is the *digital georeferenced gazetteers*.

Digital georeferenced gazetteers link *place names* to *geographic footprints*. Current digital library implementations (for example, the Alexandria Digital Library and the Digital Library Project of the University of Berkeley) have demonstrated the use of such gazetteers to provide *indirect* georeferencing to geospatial datasets. Through gazetteer translations, a place name can be converted to a geographic footprint that can be used to find relevant geospatial datasets (Indirect georeferencing).

However, there are instances of geographic names that are *non-unique* within geographic domains. For example, the geographic name "Athens" is not identified by its name in the global geographic domain, so if we query a digital georeferenced gazetteer using the geographic name "Athens" the response will consist of more than one results, however only one of them is correct according to what we had in mind. It is obvious that when integrating georeferenced information we need a *one-to-one correct mapping* between a *vague geographic name* of a datasource and a *recognized and identified geographic name* in a digital gazetteer, otherwise the integrated information will suffer from essential semantic problems and the benefits of the unified information will be reduced.

Application Description

Data cleaning, precision and recall, identification and verification are some of the aspects that are encountered throughout this work and the challenge is the automation of the mapping process surpassing the *misspelling*, *incompleteness* and *duplicate detection* problems that are involved, so that *human intervention* can be systematically reduced.

Our main objective is to analyse aspects of correctness and completeness of digital gazetteers and to stress the impact that they have to the mapping process. Based on our study, we propose methods that automate the mapping process. We aim at the maximum accuracy of the mappings so as to eliminate the human control afterwards. Ideally, we would like our algorithms to be able to decide with one hundred percent probability (100%) that an identification is correct. The goal is that our algorithms provide considerably smaller amount of the total data to the user that need to be manually controlled. We believe that our work demonstrates a relevant way in which digital gazetteers could be used in order to add value in information integration projects and especially where geographic names serve as critical point of properties linkage between heterogeneous resources.

* This work has been inspired and supported by "ubi-erat-lupa" project (www.ubi-erat-lupa.org). The objectives of the project are to interlink archaeological research on the Roman era systematically and transnationally and to exchange expert knowledge; to compare primary sources on a broad scale and gain new insights into the history and the cultural heritage of the Roman era; to save the information from the Roman stones in the form of digital memory and to preserve it according to a uniform standard to be improved by the project; to intensify the co-operation; to make sources available to the public.

In our application, we try to identify the place names that exist in diverse datasources by using a third-party digital georeferenced gazetteer, the Alexandria Digital Gazetteer. For each single place name that is described in a datasource, we try to form a query that follows a standard format. This query includes information about the place name, wider place names, if available, that include this place name and finally, the country in which this place name falls within. Each query evaluation results in an answer string that follows a standard format. The application takes as input a set of query strings and forms the answer strings by querying the Alexandria Digital Gazetteer using the Gazetteer Protocol. We use a portion of the ADL query language types such as *identifier-query*, *name-query*, *class-query*, *relationship-query*, *footprint-query* with a combination of available operators in order to *generalize* or to *specialize* our query. Below, there are more formal descriptions, as well as, examples of query and answer strings.

Query string format

Format
Place/WiderArea/…/WiderArea@Country
where,
Place = PlaceName(PlaceAlternative1;…;PlaceAlternativeN)
WiderArea = WiderAreaName(WiderAreaAlternative1;…;WiderAreaAlternativeN)
Country = CountryName

Explanation
PlaceName: This is the name of the place
PlaceAlternative(1..N): Alternative names of the PlaceName
WiderAreaName: Wider area on which the PlaceName falls within.
CountryAlternativeName(1…N): Alternative name of the WiderAreaName

Description
To form a query string we first write a PlaceName. If we have knowledge of alternative names of this PlaceName then we include in parenthesis [()] the PlaceAlternative names. PlaceAlternative names are separated by semi colon [;]. As you can notice the WiderArea is Recursive. We can have as many as possible WiderAreas of a place. We call them levels. Levels are separated by the “slash” symbol [/]. When we have described the Place and the WiderAreas of it we write the “at” symbol [@] and then write the CountryName.

Examples:
1. Edessa(Edhessa;Vodena)/Makedonia(Makedhonia)@Greece
2. Melje/Maribor@Slovenia
3. St. Urban/Klagenfurt-Land@Austria
4. Radlje (Mahrenberg)/Slovenj Gradec@Slovenia
5. Purbach am Neusiedlersee/Eisenstadt@Austria

Answer String Format

Format
PlaceKeyword;CountryKeyword;Level:PlaceADLPrimaryName;PlaceADLID;PlaceADLClass;CountryADLPrimaryName;CountryADLID;CountryADLClass

Explanation
PlaceKeyword: The place keyword for which the answer was formed
CountryKeyword: The country keyword for which the answer was formed
Level: Level represents whether the answer formed according to place found in the 1st, 2nd, etc level of the query string.
PlaceADLPrimaryName: The primary name of the PlaceKeyword according to ADL
PlaceADLID: The adl id of the PlaceKeyword
PlaceADLClass: The class name that the PlaceKeyword belongs to according to ADL.
CountryADLPrimaryName: The primary name of the CountryKeyword according to ADL

CountryADLID: The adl id of the CountryKeyword
CountryADLClass: The class name that the CountryKeyword belongs to according to ADL.

Examples

1. Edessa;Greece;L0:Edhessa;adlgaz-1-2245758-40;populated places;Greece;adlgaz-1-58-36;countries
2. Maribor;Slovenia;L1:Maribor;adlgaz-1-3797492-06;populated places;Slovenia;adlgaz-1-132-23;countries
3. Klagenfurt-Land;Austria;L1:Klagenfurt Land, Politischer Bezirk;adlgaz-1-1221248-18;administrative areas;Austria;adlgaz-1-9-02;countries
4. Slovenj Gradec;Slovenia;L1:Slovenj Gradec;adlgaz-1-3798678-71;populated places;Slovenia;adlgaz-1-132-23;countries
5. Purbach am Neusiedlersee;Austria;L0:Purbach am Neusiedlersee;adlgaz-1-1227617-61;populated places;Austria;adlgaz-1-9-02;countries

Transformations of Special Characters

Our evidence shows that Alexandra Digital Gazetteer (ADL) makes a kind of transformation/normalization on special characters of place names (e.g. national symbols). This normalization leads to misspelling problems. In order to eliminate the problems that come up by special characters we transform all place names in ANSI format. We make the transformation by using the "Save As …" option of the Notepad application and then by selecting "ANSI" in the "Encoding" drop list. Although, this is not an optimal solution, it is proven to be the one that cover the most of the transformations made by ADL. In the table below, we try to figure out some of the effects of this transformation.

In Original Text (Coming from a datasource)	**In ANSI text (transformed)**
ß	sz
ü	u
À	A
ä	a
ö	o
Ñ	N
É	E
ï	i
…	…

Since the normalization in the Alexandria Gazetteer does not follow any standard transformation, we recommend users to use the correct national spelling in Unicode, and use as patch an automatic transformation to the form used in Alexandria. E.g. the German "Umlaut" is not an accent mark, but a different character. Following national standards, the transformation of "ü" to ANSI would be "ue" rather than "u", the transformation of "ß" to ANSI would be "ss" rather than "sz", the transformation of "ä" to ANSI would be "ae" rather than "a", the transformation of "ö" to ANSI would be "oe" rather than "o", etc.