IBIMA Publishing
Journal of Eastern Europe Research in Business and Economics
http://ibimapublishing.com/articles/JEERBE/2021/265538/
Vol. 2021 (2021), Article ID 265538, 9 pages, ISSN : 2169-0367
DOI: 10.5171/2021.265538

Research Article

Software Engineering Techniques for the Extraction of Ontology of Historical Geographic Information

Piotr Kosiuczenko

Institute of Information Systems, Warsaw, Poland
piotr.kosiuczenko@wat.edu.pl

Received date: 25 September 2020; Accepted date: 25 Janauary 2021; Published date: 29 March 2021

Academic Editor: Milena Janakova

Abstract

Unorganized processing textual information in large files is time consuming and error-prone, especially if the texts are ununiform. Investigating past and historical data concerning geography and economics can be facilitated a lot when the data is stored in a data base or has a proper form. There was a lot of research concerning the transformation of textual data to an ontology in a methodical way. In this paper, a method is presented for the extraction of geographic and economic information from a historic Lexicon. The method is based on software engineering techniques and follows an iterative cycle. It results in a well-defined ontology specified using UML class diagrams which can be queried using Object Constraint Language. An evaluation of the model is presented. This research facilitates the comparison of various historical stages of development. Thus, it helps in assessing regional development, qualitative and structural changes in the regional economy, ownership, infrastructure as well as living standard transformation.

Keywords: Software Engineering, Geographic Information Systems, Ontology, UML.

Cite this Article as: Piotr Kosiuczenko (2021)," Software Engineering Techniques for the Extraction of Ontology of Historical Geographic Information ", Journal of Eastern Europe Research in Business and Economics Vol. 2021 (2021), Article ID 265538, DOI: 10.5171/2021.265538

Introduction

Querying heterogeneous properties in historical data is a complex problem. In case of geographical and economical information, queries can concern regional development, quantitative, qualitative and structural changes in the economy of a given region, ownership, transformation of economic units, change of living standards, economic structure, economic potential, infrastructural development, and standard of living for residents. The problems are exhorted if the data is in a textual form.

In order to facilitate such queries, one needs a well-developed data structure with an appropriate querying language and mechanisms. This requires a well-developed ontology and then a transformation of it into an appropriate data structure based on the ontology. Ontologies represent the intentional aspect of a domain for governing the way corresponding knowledge bases are populated (see Wong et al. (2012)).

Ontologies are a very well-studied research topic. Learning Ontology from text can be defined as "the process of identifying terms, concepts, relations, and optionally, axioms from textual information and using them to construct and maintain an ontology, as pointed out by Wong et al. (2012). It is also a well-studied topic. Various methods have been developed including statistical analysis (cf., e.g. Khurshid et al. (2005)), and classification techniques (cf., e.g. Larin et al (2011)). There are also methods for ontology construction (cf. e.g. Fawei et al. (2019) and Fawei et al. (2019)). In case of GIS systems, various methods can be applied ranging from the statistical ones, to data mining-based ones (vide, e.g. Arabameria et al. (2019) for a comparison).

Closely related topics are the Geographical Information Systems. GIS concern geographic data such as objective events and entities occurring on, above and below the Earth surface. They are aimed at storing, processing and representing analysis, visualization and retrieval of geographic data (cf., e.g. Larin (2011)). The typical problems in GIS construction are the conceptual modelling of the geographic objects and their relation, identification of objects, definition of attributes (e.g. object tree and object forest), as well as spatial relationships. There are also approaches to the modelling and extraction of GIS (cf., e.g. Zhu et al. 2018)).

This paper elaborates a method of extraction of ontology from historical geographic data proposed in Kosiuczenko (2020). The data being the subject of interest is contained in a relatively well structured and complete Lexicon called the "Lexicon of the Kingdom of Poland" by Sulimierski et al. (1880-1902). From its name, it contains geographic and economic information about the Kingdom of Poland in XIX century. The overall goal is to propose a general method for extracting ontology from Lexicon resources using the example of the "Lexicon of the Kingdom of Poland". The Lexicon is used for various historical studies including ownership structure, economic development, industrialization, infrastructure, local development, etc.

The author decided to base the ontology on an object-oriented model described using UML class diagrams (cf. OMG (2017). The reason is that the model must accommodate various kinds of information and meta-information and must allow for sophisticated querying concerning different aspects of the historical development. In the author's opinion, the best suited query language is the Object Constraint Language (OCL, OMG (2014)) because it is well integrated with UML class diagrams, possesses very expressive querying capabilities and supports various datatypes. The statistical methods were not used to extract the ontology (cf., e.g. Khurshid et al. (2005)) since the data processed is relatively precise

DOI:10.5171/2021.265538

and well-structured, and the corresponding class diagram is small and relatively easy to elicit. Iterative software engineering methods are used as well, since they proved very successful in engineering of software systems. It should be mentioned that conventional development methods used classical engineering, such as the waterfall model, do not work for the development of complex software system (cf., e.g., Gullo (2016)).

Model Construction

Software engeeneering is a very well developed dyscipline of engineering. It provides proven methods guiding the software engineering process, languages for modeling software systems, patterns for software construction and tools supporting software development. Among the widly used modeling and specification languages are UML and OCL. The author decided to use methods of software engineering due to their popularity, on the one hand, and the choice of UML and OCL, on the other. These languages are inherently embedded in SE methods and have very well developed support tools. Another fact is that software engineering provides well-developed, mature and proven in practice methods of system specification extraction and development. In particular, the author used Design Patterns proposed first by Gamma et al. (1994) (see also Bafandeh et al. (2017)) to obtain a well-structured model, and the Refactoring method aimed at redesigning models and code developed by Fowler (1999) (see also Baqais (2020)). Both have proved to be very useful in model construction and modification. Design Patterns and Refactoring are today widely used in the area of object-oriented programming. In addition, refactoring tools are a standard part of numeros software development tools, such es Eclipse (cf. Eclipse (2021)).

The ontology is defined in terms of:

- Collection of classes that form denotations of terms.
- Attributes that characterize the terms.
- A class inheritance hierarchy which relates terms to their generality.
- Relations between classes and their structural presentation.

One of the most fundamental features of SE methods is their iterativeness, as well as agility in terms of adaptability to existing needs. Thus, the method used is highly iterative and takes into account the need to continuously modify the ontology as new categories of geographical units and new types of dependencies between them emerge in the processing of successive parts of the Lexicon and subsequent information.

In this approach, the starting point was an a'priorical class diagram corresponding to the terms, concepts and relationships among geographical units and administrative structures. Class diagrams depict hierarchis of entities' types. They modell their properties using the so called attributes. For example there can be attribute colour for colours; the attribute has value red if the actual color of the entities is red. The diagrams allow one to model relations between entities' classes and to model the fact that one class extends another, which is called inheritance. Class diagrams are used as models of ontologies as they describe the kinds of entities that exist, they properties and mutual relations. In the next section, we present an example of a class diagram and discuss its use as an ontology.

The initialy chosen class diagram forms the initial framework in the construction of the required ontology. The diagram is then iteratively modified. It should be pointed out, that iterative approaches proved very successful in the ingeneering of complex software systems. In the iterative process, new model elements, such as classes, attributes and associations, are added, and the inappropriate model elements are modified or removed. For example, if it turns out that a type of entities have a color, then we add the corresponding attribute color.

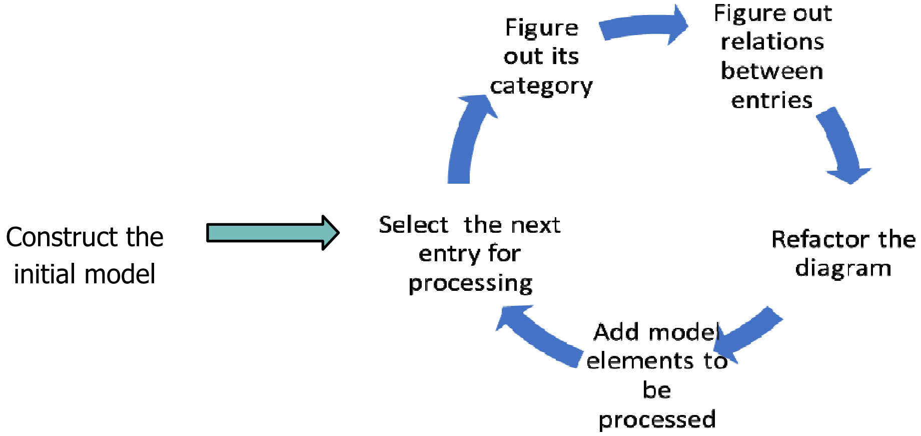
Construct the initial model
Select the next entry for processing
Figure out its category
Figure out relations between entries
Refactor the diagram
Add model elements to be processed

Fig. 1: The Ontology Creation Life Cycle

The approach to ontology extraction is shown in Fig. 1 (cf. Kosiuczenko 2020): starting with the afore-mentioned initial model, the author sulements and modifies it step by step. The consecutive step in the modification process corresponds to the next Lexicon entry. Each step has the following elements:

- Selection of the next Lexicon entry.
- Categorization of the items' corresponding terms: e.g. city, administrative unit, hamlet, post office, parish, ...
- Identification of the terms in the description of the selected entry, categorization of the corresponding terms and entries if not already performed, and the establishment of the structural relationships between them.
- Verification that the relationship between the term and the terms that appear in the description of the selected entry is appropriate. If not, then the class diagram is refactored: its elements are modified and replaced appropriately.
- Addition of new model elements such as classes, inheritance relations, associations and attributes.
- In the case of role change by an item corresponding to a term, replace inheritance with delegation.

It has turned out that after few iterations of the cycle, ca 40, on randomly selected entries, the resulting class structure remained stable and the refactoring was not needed anymore, but the author only needed to add new elements into the object diagram. Thus, the structure is adequate for the purpose, the subject of change being only the addition of a new element.

The Model

Below a simplified form of the ontological hierarchy is presented in the form of a class diagram. It is obtained by the application of the above-mentioned procedure. The model is generic, uses inheritances and delegations. In the model, there are essentially geographical objects and reified associations such as relative location, i.e. a so-called classifier that is both an association and a class at the same time. As a consequence, associations in this model are objects (it is a reification of relationships).

In this class diagram (see Fig. 2), the basic units are geographical objects which correspond to the Geographical Object class. There are essentially two subcategories here: Administrative Unit and Geographical Unit. This results in a dichotomous division of geographical objects. All other categories of geographical objects are details of these two classes, e.g. Governorate, County, Community etc. on the one hand and School, River, etc. on the other.

Administrative units are modelled by the class Administrative Unit and its respective subclasses which describe the administrative role of the geographical features. The generic Location class models the location of spatial objects, such as geographic coordinates. This information can be specified in greater details in appropriate subclasses.

The above-mentioned model is not only based on multiple inheritances but also has to take into account changes in roles over time, e.g. the fact that a hamlet can become a town, a private home can become a post office etc. Various roles played by the same object at the same time can be modelled by multiple inheritances. However, this kind of modelling restricts the expressiveness and flexibility of the ontology. A geographical object can play many roles at once. When roles are used, each of them is characterized by a separate object of the appropriate category, e.g. a farm, a hamlet, a post office (reification of roles). Roles can be treated as subclasses; this requires the use of multiple inheritances, or they can be reified by a dedicated class as in the "State Pattern" (cf. Gamma et al. (1994)). The author chose the second option due to its flexibility. Since there can be many such roles at the same time, in the diagram, the appropriate multiplicity at the end of the association is specified (in the diagram, the author also uses *). The time in which such roles are played is characterized by attributes startDate and endDate, whose values are objects modelling moments in time (with day, month and year attributes). This allows the static modelling of the fact that a given geographical object starts and ends to play a given role at certain moments in time.

The use of roles allows more sophisticated patterns of search for information about the condition of individuals at a given time and its evolution over time but makes the queries more sophisticated than in the case of multiple inheritances. It should be noticed that the author here applies the "Replace inheritance by delegation" refactoring pattern (cf. Fowler (1999) and Baqais (2020)).

The diagram in Fig. 2 shows the most fundamental part of the obtained ontology. It is a refined version of the diagram presented in Kosiuczenko 2020. There are five fundamental classes: Spatial Object, Location, Geographic Unit, Role, Administrative Unit and Functional Unit. The root class Spatial Object is characterized by its name, some other attributes as well as a set of locations. The class Location is characterized as geographically coordinated (not shown in the diagram) and can be farther specialized. The class Geographic Unit is a special case of the

Spatial Object. The geographic units can play different roles. The roles are objects of the class role; the class role is subclassed by Administrative Unit and Functional Unit, dividing it into two separate categories.

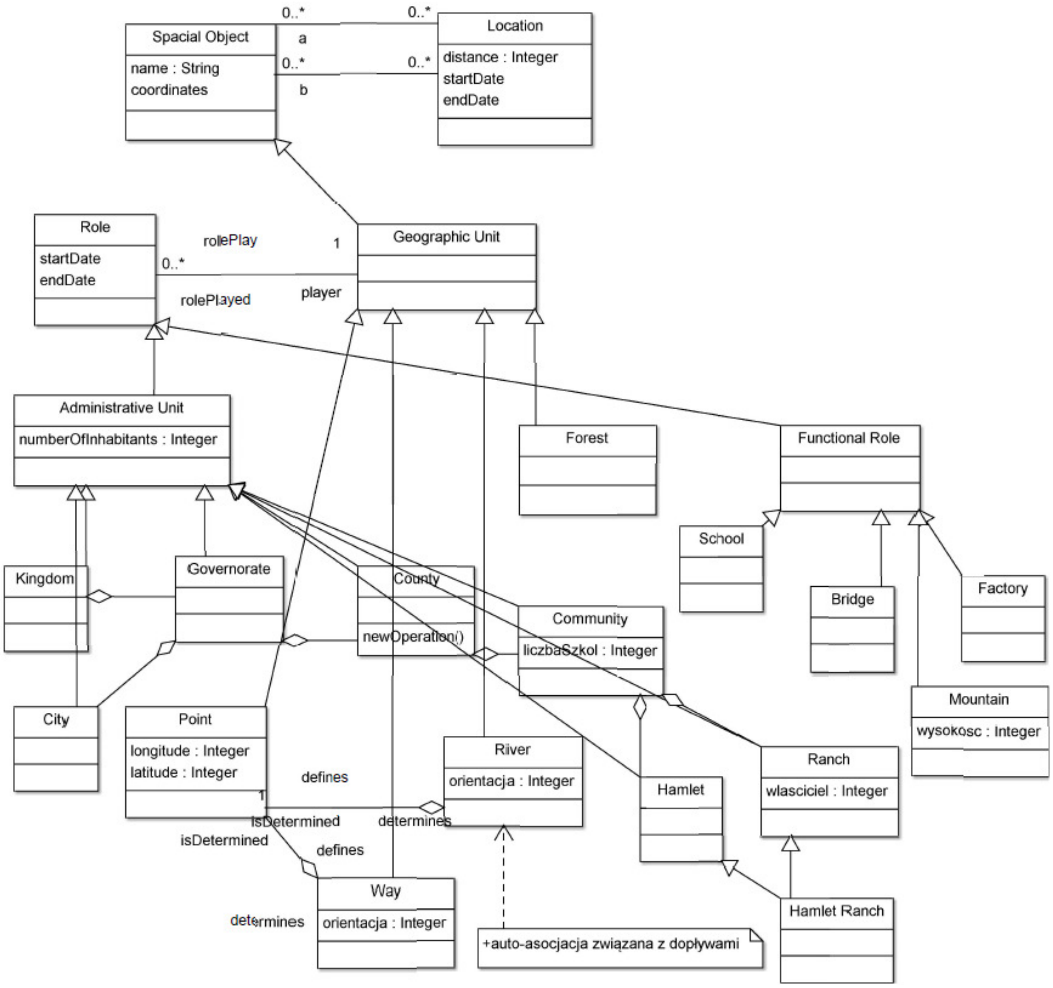

Fig. 2: The Class Diagram of the Ontology

The classes are related by associations. Spatial objects have their locations. There can be multiple location objects for one spatial object if the object is stretched over a certain area. The association of type composition is used here to express existential dependencies since the locations exist only to specify locations of the corresponding spatial objects. Similarly, one geographic unit can play several roles, one after another or in parallel, according to what is specified by the corresponding association. In this case, the composition is also used to indicate the fact that roles are existentially dependent on the object playing them.

The class diagram requires some additional constraints, which can be specified in OCL (not shown in the diagram). For example, there is a constraint saying that a point, i.e. an object of class Point, has one, and only one, location not several ones. Similarly, a river location can be specified by a sequence of location objects with certain consistency criteria imposed. The use of OCL to specify additional constraints makes this ontology very general and flexible. However, for a person who is not familiar with object-oriented modeling, it is more difficult to understand.

In order to make the role reification model more flexible, roles are divided into two groups: administrative and functional. This allows for the uniform treatment of all roles, such as “governorate” and 'county' on the one hand, and 'school' and 'factory' on the other. Dates for spatial objects and roles are also introduced, since both can have their start and end date. For example, a building can play a role of a school in a certain time-period. It should be noted that this does not prohibit playing other roles in other time-periods; the other time-periods can be disjoint, overlapping and the same as the first one.

The queries concerning the properties of the constructed model can be expressed in OCL. The major OCL-construct aimed for this purpose is select, which selects all objects with a given property. It has the form X->select(x : C | condition(x)), where X is a collection of objects of class C and condition(x) is a Boolean-valued condition concerning the objects of this class. The selection results in the set of objects x belonging to X and satisfying the condition. There are some other operators such as quantifiers, iterators, collectors etc. which make the language very expressive. This allows one to write very detailed queries concerning the investigated model.

Evaluation

The adequacy of the proposed ontology was analyzed in terms of structure, consistency, adequacy and usability. In general, there are several approaches to systematize the ontology evaluation (cf., e.g. Gangemi et al. (2005) and Gangemi et al. (2006) for a more systematic approach and references to further reading).

The structure of the ontology proved to be relatively simple with rather few model element kinds. The paths in the graph corresponding to the class diagram in Fig. 2 are relatively short, and the average path length is relatively small. The so-called fan-outness, i.e. the number of outgoing edges, equals one part of roles and locations. In general, the number of roles is also relatively small due to the fact that most spatial objects play one or few roles over time. In the case of locations, the number can be higher if the object has a complex shape.

One of the key issues is the modelling of time aspects. Objects of interest can appear in time, assume different roles and cease to exist. This can be adequately modelled using the proposed ontology, and the author did not encounter any counter example when processing the lexicon.

The ontology proved to be consistent with the Lexicon. As mentioned above, after relatively few cycles of the Lexicon processing, the ontology proved to be stable, in the sense of not requiring changes, except for the addition of new model elements, such as classes. There have not been many problems with equivocal terms, as they are easy to resolve and relatively small in number.

The constructed ontology proved to be adequate in the sense of covering the dictionary terms. It is also adequate in the sense of allowing complex and detailed queries. In respect to usability, the most complicated thing proved to be the querying. OCL requires a certain degree of skills in reading class diagrams and understanding the query language, in particular, the selection of objects. Nonetheless, the author thinks that the problems are mostly due to the fact that it is not easy to ask a precise question concerning complex models like this. Questions asked in a natural language are usually imprecise. Making them precise requires considering a lot of details of which the inquirer may be unaware or does not care about. Thus, the problem is not caused by the ontology as such.

The extracted ontology has also a practical meaning. This research facilitates the comparison of various historical stages of development. The queries can be performed using appropriate time parameters and time periods. The queries can be performed thanks to the startDate and endDate attributes of roles and spacial objects. It should be reminded that roles can change over time and one object can play different roles in time. Queries can take the corresponding values into account. Thus, the ontology helps in assessing the regional development, the qualitative and structural changes in the regional economy, the ownership, infrastructure and living standard transformation.

Conclusion

In this paper, the author presented a method for the construction of an ontology corresponding to a historical Lexicon of the Kingship of Poland. The Lexicon contains historical informations about cities, villages, cottages, etc. in the Kingship of Poland in XVIII, XIX until even the early XX century. It describes also the economic and cultural development of the regions. The data is only to some extend uniform and thus a flexible method for the ontology extraction was needed. The heterogeneity and large volume of the date pose a serious problem for historians and other researchers interested in the development. Thus, a tool support is needed for querying the data. However, the problems are also a real challenge for the automatic processing.

Given the complexity of the problems, on the one hand, and the apropeiatness of software engineering methods, on the other, the author applied standard methods developed in software engineering to construct the ontology. In particular, the iterative approach was used. This kind of approach is characteristic feature of SE and differenciates it from classical engineering approaches. The approach was used to design in consecutive steps the class diagram when processing the Lexicon. The final class diagram presents the ontology of the Lexicon.

Well-known design patterns and the refactoring approach were also applied. In effect, a stable and adequate class diagram was obtained which allows for complex querying. The processing procedure concerning Lexicon entries proved to be simple and natural, so that even a person untrained in software engineering can follow it. Thus, the use of software engineering methods proved to be successful in terms of the use and the quality of the results.

The proposed method of ontology extraction can be integrated with other methods, such as data mining, and applied to other areas such as GIS-information extraction.

References

- Arabameria, A., Rezaeib, K., Cerdac-Luigi, A., Lombardod, L. and Rodrigo-Cominoe, J., (2019) GIS-based groundwater potential mapping in Shahroud plain, Iran. A comparison among statistical (bivariate and multivariate), data mining and MCDM approaches, *Science of The Total Environment*, 658, 160-177
- Bafandeh M., B. Rasoolzadegan, A. and Yazdib, Z., H., (2017) The state of the art on design patterns: A systematic mapping of the literature Author, *Journal of Systems and Software*, 125, 93-118
- Baqais, B. A., Alshayeb, M., (2020) Automatic software refactoring: a systematic literature review, *Software Quality Journal*, 28, 459-502
- Fawei, B., Pan, J.Z. and Kollingbaum, M. et al., (2019) A Semi-automated Ontology Construction for Legal Question Answering, *New Gener. Comput*. 37, 453-478

- Eclipse (2021), Eclipse foundation, www.eclipse.org
- Fowler, M., (1999) Refactoring: Improving the Design of Existing Code, Addison-Wesley Professional
- Gamma, E., Helm, R., Johnson R. and Vlissides, J., (1994) Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley
- Gangemi, A., Catenacci, C., Ciaramita, M. and Lehmann, J., (2005) A theoretical framework for ontology evaluation and validation, CEUR Workshop Proceedings. 166, 2nd Italian Semantic Web Workshop, University of Trento, Trento, Italy, 14-15-16
- Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J., (2006) Modelling Ontology Evaluation and Validation, ESWC, LNCS, 4011, 140-154
- Gullo, D., (2016), Real World Agility. Publisher Pearson Education (US), ISBN10 0134191706
- Kosiuczenko, P., (2020) UML Based Ontology for the Extraction of Historical Geographic Information, 36th IBIMA Conference: 4-5 November 2020, Granada, Spain, ISBN: 978-0-9998551-5-7
- Khurshid, A., Lee, G, (2005) Automatic Ontology Extraction from Unstructured Texts, ODBASE 2005, LNCS, 3761, 1330-1346
- Larin R., Fonseca Garea-Llano, E., (2011) Automatic Representation of Geographical Data from a Semantic Point of View through a New Ontology and Classification Techniques, *Transactions in GIS*, 15(1)
- OMG, (2014) Object Constraint Language, Spec. ver. 2.4, January
- OMG, (2017) Unified Modeling Language, Spec. ver. 2.5.1, December
- Sulimierski, F., Chlebowski, B., Walewski, W., et. al., (1880—1902) Slownik geograficzny Krolestwa Polskiego, 1880-1902, Warsaw
- Wong, W., Liu, W. and Bennamoun, M., (2012) Ontology Learning from Text: A Look Back and into the Future, *ACM Computing Surveys*, Vol. 44(4)
- Zhu, J., Wright, G., Wang, J. and Wang, X., (2018) A Critical Review of the Integration of Geographic Information System and Building Information Modelling at the Data Level, ISPRS, *Int. J. Geo-Inf.*, 7(66).