IBIMA Publishing
Journal of Eastern Europe Research in Business and Economics
http://ibimapublishing.com/articles/JEERBE/2022/458047/
Vol. 2022 (2022), Article ID 458047, 10 pages, ISSN: 2169-0367
DOI: 10.5171/2022.458047

Research Article

OCL Queries of Historical Geographic Information

Piotr KOSIUCZENKO

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

Received date:23 March 2022; Accepted date:23 June 2022; Published date: 7 July 2022

Academic Editor: Artur Arciuch

Abstract

Searching for precise information contained in large textual files is very time consuming and error prone. Investigating past and historical data concerning geography and economy can be facilitated a lot when the data is appropriately stored in a data base or and the corresponding schema has a proper form. Therefore, it is convenient to transform textual data to an ontology and query it in such a form. Some research has been done on the application software engineering methods for construction of ontologies. Ontologies constructed in this way need a proper querying counterpart. In this paper, the usefulness the problem of querying of geographic and economic information from a object-oriented data base is studied. The querying language is Object Constraint Language (OCL) and software engineering techniques are applied. OCL allows one to query information stored in oo-database which schemata expressed by UML class diagrams. An evaluation of the method shows that it is very powerful and allows for very fine querying. The querying is facilitated by the fact that considered UNL ontology is close to the reality. However, the queries tend to be complicated and an inexperienced user of OCL may have problems with formulating them. This research may help in assessing regional development, qualitative and structural changes in the regional economy, assessing ownership, infrastructure and living standard transformation.

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

Introduction

Querying heterogeneous information in historical textual data is a very complex problem, since geographical and economical information systems can be very complex and one has to take into account the change in time and the heterogeneity of the information. Even more challenging is the querying of such information in historic books. Such queries can concern regional development,

Cite this Article as: Piotr KOSIUCZENKO (2022)," OCL Queries of Historical Geographic Information ", Journal of Eastern Europe Research in Business and Economics Vol. 2022 (2022), Article ID 458047, DOI: 10.5171/2022.458047

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, standard of living for residents, etc.

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

Ontologies are a very well-studied research topic. Ontology learning 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 in (Wong et al. (2012)). 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 exist also methods for semiautomatic 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).

A closely related topics are the Geographical Information Systems. GIS concern of geographic data such as objective events, entities occurring on, above and below the Earth surface. They are aimed at storing, processing, representing, analysis, visualisation 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, spatial relationships, etc. There exist also approaches to modelling and extraction of GIS (cf., e.g., Zhu et al. 2018, Khurshid et al. (2005)). In a previous paper, we examined the possibility of extracting an object-oriented ontology from historical lexicons using methods developed for and successfully used in software engineering (Kosiuczenko (2020)).

In this paper we investigate the usability of the Object Constraint Language (OCL, OMG (2014)) and the methods of software engineering for modelling and querying GIS information stored in oo-databases. We assume that these databases are specified with schemata corresponding to UML class diagrams as demonstrated in (Kosiuczenko (2020)). The key application is the historical geographic data contained in a relatively well structured and complete Lexicon called "Lexicon of the Kingdom of Poland" by Sulimierski et al. (1880-1902) containing geographic and economic information about the Kingdom of Poland in XIX century. We assess the suitability of OCL for extracting ontology from historical resources using the example of the "Lexicon of the Kingdom of Poland" (LKP) and ontology extracted from LKP. It should be noted, the Lexicon is used for various historical studies including ownership structure, economic development, industrialization, infrastructure, local development etc.

We decided to use UML and OCL due to the fact software engineering provides well-developed and mature methods of system specification extraction and development. It allowed us to apply Design Patterns proposed first in (Gamma et al. (1994)) (see also Bafandeh et al. (2017)). Such patterns provide well-structured model and the Refactoring method aimed at redesign of models and code developed by Fowler (1999) (see also Baqais (2020)). Both have proved very useful in model construction and modification.

The choice of OCL and UML class diagrams (cf. OMG (2017) is due also to their popularity and broad use in software engineering practice. On the other hand, the model must accommodate various kinds of information and meta-information and must allow for sophisticated querying

concerning various aspects of the historical development. In our opinion, the best suited query language is OCL due to its very fine querying capacities and its good integration with UML class diagrams. This combination ensures very expressive querying capabilities and supports various data types. Actually, the querying of oo-databases was studied in a number of papers (cf., e.g., Savnik et al. (1999)). However, we need a query language allowing for the use of UML based ontologies with fine modelling and querying possibilities well integrated with software engineering methods. Up to date, only OCL and its variants meet these requirements.

This paper is a journal version of the paper (Kosiuczenko, P. (2022)). We discuss the Lexicon in more detail presenting its exemplary entries. We refine the ontology presented in (Kosiuczenko, P. (2022)) by specifying in more detail the roles geographic units may play. In consequence, the structure of roles is more detailed and allows us for an easier querying. The paper is structured in the following way: In the second Section we present the Lexicon and describe briefly problems with its structure and its processing. In the third Section, we present the class diagram based ontology which is the refined version of the ontology presented in the previous paper. In the fourth Section, we discuss the form of OCL queries and the way they can be used. In the fifth Section, we evaluate usability of OCL queries for querying the Lexicon. The last Section concludes the paper.

Lexicon of the Kingdom of Poland

In this Section we present briefly the Lexicon of the Kingdom of Poland (Sulimierski, F. et. al., (1880—1902)). We describe its form, structure and content. The Lexicon is devoted to the area of Kingship of Poland before its partitions, but also contains information of some areas where Polish influences were strong. It was published in the period of 22 years by 3 main authors, but had several other contributors. It contains geographical, economic, demographic, historical and even biographical informations. The entries describe regions, cities, villages and settlements, rivers, lakes and mountain peaks. Some entries include include geographic coordinates with accuracy to degrees and minutes. Due to many contributors and lack of rigid form, the entries do not have a uniform shape and the amount of information varies; e.g., sometimes there is some historical information, sometimes not, sometimes there geographic coordinates are specified, but in most cases not, etc. To illustrate the content of the Lexicon, we quote some of its entries (Sulimierski, F. et. al., (1880—1902)):

Bąków, 1). Government village and farmhouse, opoczyński county, municipality of Russin, parish of Nieznamirowice. In 1827 there were 12 houses and 105 inhabitants. 2). B. or Bonkow, a village and manor farm, the county of Koñskie, Szydłowiec commune, parish of Wysoka, land owned by the proprietor. 201 morgs, 964 m of manor land. In 1827, there were 20 houses and 123 sq. m.; currently, 19 houses and 139 residents. A manor mill with a l wheel on an unnamed river (abandoned at Zinberg).

Bodzanów, a village in the Wieliczka county, has 1047 morgas in extent, including 777 morgas of arable land, 97 houses, 507 residents, the parish is in place; the wooden church, extremely ancient, dedicated to St. Peter and Paul, existed before 1229, as it was subsidized by Michał, the abbot of Tyniec, in the latter year. There is a tradition that the former church was consecrated by St. Stanislaus, Bishop of Krakow. The location is hilly, the soil is rye. Azarycze, or Ozarycze, is a town with a municipal government, in the Minsk province, on the southeastern edge of the Bobruisk district, near the border of the Tvoretsk district, on the Ochowka river, among the forests and marshes of Polesie. There is a municipal school and a parish church. The Azaryk community consists of 24 villages and has 1,374 male residents, while the town itself has 900 residents.

Balice, a village in the Stopnica county, Gnojno municipality, Janna parish, on the right side of the road from Stopnica to

Chmielnik, a communal school. In 1827 there were 49 houses and 342 residents.

Balice, with hamlets: Werychów, Szczyglicki, and Podkamycz, a village in the Cracow county, 4 kilometers south of Zabierzów, has 2312 n. a. morgas of land, 100 houses, 753 residents, parish in Morawica, one-classroom folk school, distillery, and an American mill, hilly location, with a beautiful view of Cracow.

Dobrojewo, domin., Szamotulski County, 11,618 morgas of area; 8 places: 1) D., 2) manors: Bielejewo, 3) Binino, 4) Nosalewo, 5) Spibieda, 6) Stefanowo, 7) forestry: Klemensowo, 8) Forestowo; 54 houses, 950 inhabitants.; 74 Evagelicals, 876 Catholics, 365 illiterate, postal station Ostroróg (Scharfenort), iron railroad station. Wronki is 7 kilometres away. The owner was count Stefan Kwilecki of Szreniawa coat of arms. In this property there is a sheepfold, famous since 1864. Since 1866 they have been producing pressed peat here, up to 4 million bricks a year, i.e., about 28,000 cents.

As the reader may see, the entries are far from being uniform and sometimes contain more sometimes less detailed information. The terminology used is also far from being uniform. We resolved the abbreviations to make the reading easier: All in all, the original form of the Lexicon is far from being ready for the application of natural language processing methods and tools.

The Model of the Geography of the Kingship

In this section, we present in a simplified form an extract of the ontology corresponding to the "Lexicon of the Kingdom of Poland" which is a refined version of the ontology presented in (Kosiuczenko (2022)), (see also Kosiuczenko (2020)). The ontology is generic, uses single inheritance and delegation. In the model we have 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). The class BuildingRole is here split into subclass BuildingRole allowing one to specify roles of buildings in a more accurate way. It contains also some additional classes, such as River, Parish, etc.

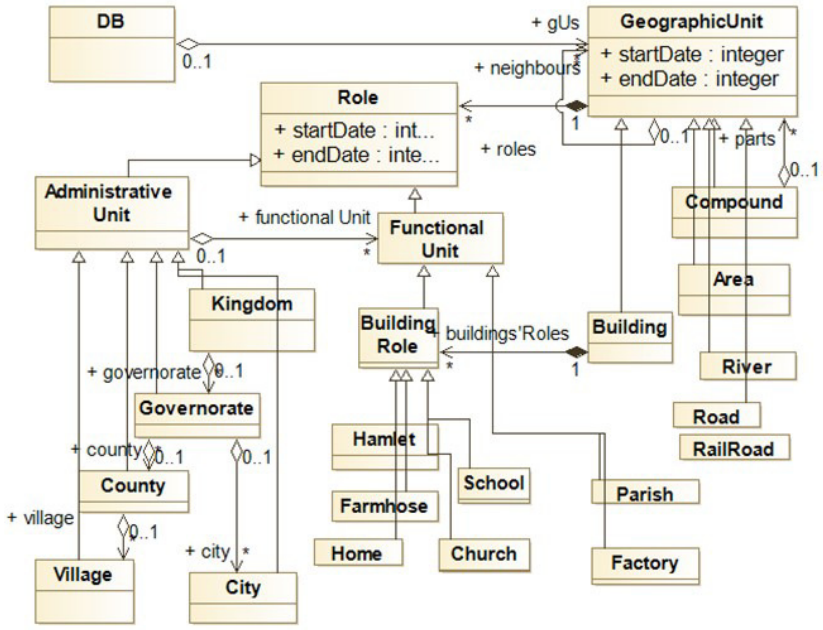

Fig 1. The Class Diagram representing the basic scheme of the historical data

The class diagram on Fig 1 (it is a modified and extended version of the diagram on Fig. 1 presented in (Kosiuczenko, P. (2022))) shows five fundamental classes: GeographicUnit, Location (not shown on the diagram), Role, AdministrativeUnit and FunctionalUnit. The class DB stands for a data base.the basic units are geographical objects which correspond to the root class GeographicUnit. Objects of the root class are characterized by start and end date, geographic coordinates (not shown on the diagram) and can be farther specialized. Objects may play different roles, e.g., a building can play the role of a school and of an office. There are essentially two categories of roles: AdministrativeUnit and FunctionalUnit. This results in a dichotomous division of roles. All other roles are subcategories of these two classes. The geographic units can play different roles. The roles are objects of the class Role. This information can be specified in greater detail in appropriate subclasses.

The above model is based on inheritances, but also takes 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. A geographical object can play many roles at once. Various roles played by the same object at the same time can be modelled by multiple inheritance. However, this kind of modelling restricts the expressiveness and flexibility of the ontology. Roles can be treated as subclasses, this requires the use of multiple inheritance, or reified by a dedicated class as in the "State Pattern" (cf. Gamma et al. (1994)). When roles are used, each of them can be characterized by a separate object of the appropriate category, e.g., a farm, a hamlet, a post office (reification of roles). We chose the second option due to its flexibility. Since there can be many such roles at the same time, in the diagram we specify the appropriate multiplicity at the end of the association (in the diagram we use also *). 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); however for the sake of simplicity in this paper, we model time by integers. Time stamps allow us for a modelling of the fact that a given geographical object starts and ends to play a given role at certain moments in time. It should be noticed that we apply here the "Replace inheritance by delegation" refactoring pattern (cf. Fowler (1999) and Baqais (2020)).

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 certain area. We use here association of type aggregation, when the parts do not depend existentially on the whole, e.g., Area and GeographicUnit, and composition to express existential dependences, e.g., roles depend existentially on their bearer. Geographic units can have multiple parts being also geographic units. This is made possible due to the aggregation association parts and to inheritance. Here we apply the "Composite Pattern", see Gamma et al. (1994). One geographic unit can play several roles, one after another or in parallel what is specified by the corresponding association; in this case we use also composition to indicate the fact that roles are existentially dependent on the object playing them. Geographic units may have their neighbours, e.g., a village may be located on a bank of a river or be crossed by a road, or a railroad. This is modelled by the aggregation neighbours relating geographic units.

In order to make the role reification model even more flexible, we divide roles into two groups: administrative and functional. Thus, the class Role is subclassed by AdministrativeUnit and FunctionalUnit dividing it in two separate categories. Administrative units are modelled by the class AdministrativeUnit and its respective subclasses which describe the administrative role of the geographical features. This allows for uniform treatment of all roles, such as "governorate" and 'county' on the one hand, and 'school' and 'factory' on the other. As mentioned above, we allow for a finer specification of buildings' roles by introducing the class BuildingRole. This allows us for an easier querying of the roles of buildings.

We introduce also dates for spatial objects and roles, since both can have their start and end date. For example, a building can play a role of school in 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.

The class diagram requires some additional constraints, which can be specified in OCL (not shown on 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 expressive and flexible. However, for a person who is not familiar with object-oriented modelling, it is more difficult to understand.

OCL Queries

In this section, we show how the OCL queries can be defined relatively to the above defined ontology and how they can be used to extract relevant information. Queries are functions on a domain returning some values without modifying the domain in any away. In OCL, queries are not defined explicitly, but by a special attribute on the meta level (OMG (2014), OMG (2017)). They can be modelled by properties, which are similar to derived attributes. Queries can be specified in the same way as methods are specified. Methods and defined by pre- and post-conditions. In this case, an explicit requirement assuring that a method is a query must be added. This requirement is a kind of frame axioms prohibiting any changes to the system state (see Kosiuczenko (2020)).

OCL formulas include the keyword context followed by a class name, followed by keyword inv or pre, or post indicating that the formula is an invariant or a pre-

condition or a post- condition. Then, such a keyword is followed by a Boolean condition in an navigational form. The form is called navigational because it navigates, so to say, trough the graph of objects using their associations and attributes. Thus, it allows to go from one object to another if the other is a value of the attribute of the first one or is contained in the set of its values.

OCL formulas, invariants in particular, are always formulated in a context of a UML class diagram. This diagram provides a typing of the terms contained in the formulas. Without such a diagram, and the resulting typing, the formulas have no sense. For example, in case of the diagram on Fig 1, we can formulate the following invariant:

context DB :

inv : DB.gUs.size() > 5

The context of the above invariant is the DB class shown in the diagram above. The formula contains the navigational term DB.gUs.size(). The term starts with the class name DB, standing for a data base, followed by dot, followed by association-end named gUs, which is the static attribute of DB and contains all geographic units in the data base. The function size() returns number of elements of a given collection; the invariant requires that the returned number must be larger than 5.

We will use contexts to specify queries without using the keywords inv, pre and post. Thus, the formulas we use are not complete OCL formulas, in the sense of being OCL invariants, but they fragments. They are not describing the system by boolean formulas, but only the navigational terms defining the values returned by queries. For example above, we consider term DB.gUs.size(), which returns a number, not a boolean value.

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

The above mentioned “State Patter” allows one to use roles instead of multiple inheritance. Roles allow 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. For example, using select and roles, we can for example query the number of schools in a given region established before date d for at least n years:

context DB :

DB.gUs.select(s | s.roles.exists(r : Role | r.isKindOf(School) and r.startDate <= d and n <= r.endDate – x.startDate))

The navigational terms selects all geographic units which played the role of school created before d for at least n years. To check if an object r is of a class School we use the OCL primitive r.isKindOf(School), it returns true if r is of this class and false otherwise. Thus, a geographic object s plays the role of school if there exist a role r of s such that the role is of class School. Similarly, the beginning of the role r must be before time d (r.startDate <= d) and the time between the beginning of this role and its end must be at least n (n <= r.endDate – x.startDate). For the sake of simplicity, we assume here that the dates are modelled by numbers.

Now, we can refine the above query and ask for geographic units which played the role of schools in that period and were transformed to offices afterwords:

context DB :

DB.gUs.select(s | s.buildings'Roles.exists(r | r.isKindOf(School) and r.startDate <+ d and n <= r.endDate – x.startDate and s.roles.exists(r1 | r1.isKindOf(Office) and r.endDate <= r1.startDate)))

This query selects all geographic units s which play a role of school, i.e., in the set of their roles there exists a role of school r, such that the start date of the role r.startDay is before d and, moreover, s has another role r1, which is a role of an office and its start date is after the end date of r, i.e., r.endDate < r1.startDate. It should be noted that it uses the subclass BuildingRoles described above.

We can also formulate fine queries concerning spacial properties. In particular query if a geographic unit is composed of other units of a certain kind. For example, we can ask about all factories composed of 5 at least units and created before date d:

context DB :

DB.gUs.select(c | c.isKindOf(Compound) and c.buildingsroles.exists(f | f.isKindOf(Factory)) and r.startDate <= d and 5 <= c.parts.size())

The navigational term selects all compounds c which play the role of a factory such that their start date is before d and they have 5 parts.

Evaluation

Historical geographic data are characterised by diversity, heterogeneity, complexity and change in time. This requires adequate ontologies and querying capabilities. We analysed the adequacy of the proposed ontology in terms of structure, consistency, adequacy and usability. It is also important that it may be relatively easy to construct. In the previous paper (Kosiuczenko (2020)), we demonstrated a method based on software engineering methods for a relatively simple, iterative construction of the ontology. The ontology proved to model the geographic data adequately. In general, there are several approaches to 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 constructed ontology proved to be consistent with the Lexicon of Kingship of Poland and cover the dictionary terms suitably (see also Kosiuczenko (2020)). After relatively few cycles of the Lexicon processing, the ontology proved to be stable, in the sense of not requiring changes, but only addition of new model elements, such as classes. We did not have many problems with equivocal terms, as they are easy to resolve and relatively small in number. It is also adequate in the sense of allowing complex and detailed queries, which proved to be the most challenging aspect.

The querying capabilities provided by OCL proved to be very powerful. It is possible to formulate queries concerning very complex questions such as those relating to evolving structure and development in time, as needed in the case of historical geographic data. The average path length is relatively small. The so-called fan-outness, i.e., the number of outgoing edges, equals one apart 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 we did not encounter any serious obstacles of this kind when processing the lexicon.

An important advantage of this ontology is the fact that it closely depicts the corresponding reality. This is the advantage of object-oriented modelling in general. In consequence, the associate queries can be more easily formulated than in the case of relational data bases. Thus, we can formulate them with the reality in mind. Another advantage is the flexibility of the ontology, the fact that it can express

very complex interdependences. In consequence, the coupling of the relevant and powerful ontology with OCL proved to be a good solution.

However, it may be hard find all relevant information. OCL requires a certain degree of skills in reading class diagrams, understanding queries and formulating them. In particular, the selection of objects may be sophisticated. Nonetheless, we think that the problems are due mostly to the fact that it is not easy to ask a precise question concerning intrinsically complex data models like the historical GIS. The questions we ask in a natural language may seem simple, but are usually imprecise. In consequence, it they have a simple form and are easy to formulae. However, making them precise may require considering a lot of details which the asking person may be unaware of or do not care about, and their precise formulation is not simpler than their formalisation in languages such as OCL. Thus, the problem is caused primarily not by the ontology as such nor concerns strictly the querying language used, but by the sophistication of the problem. The fine ontology requires fine way of querying, which may be hard for an inexperienced user. Moreover, the information stored in the object-oriented data based may be incomplete,. Thus, when querying the model, one has to take into account the potential incompleteness of the information. It makes the querying more complicated.

Conclusion

In this paper, we discussed an ontology corresponding to a historical Lexicon of the Kingship of Poland and the corresponding querying capabilities provided by OCL. To construct the ontology it is enough to apply standard methods of software engineering, in particular, an iterative approach to class diagram design when processing the dictionary. The ontology applies well-known design patterns. In effect, we obtained a stable and adequate class diagram which allows for complex querying capabilities. 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.

OCL proved to be adequate for querying the historical GIS system. The queries can be statically type checked, i.e., before their execution, due to the fact that they are formulated in context formed by a class diagram. This capability helps the users since type errors in queries can be highlighted at the time of writing and eliminated. This is especially important in case of complex ontologies, such as those used for historical GIS; it also helps users inexperienced in UML and OCL. Most historians are of this category. Out findings may help professionals in the areas of regional economy, assessing ownership, infrastructure and living standard transformation.

References

- Arabameria, A., Rezaeib, K., Cerdac-Luigi, A., Lombardod, L., 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, Vol. 658, 160-177
- Bafandeh M., B. Rasoolzadegan, A., Yazdib, Z., H., (2017) The state of the art on design patterns: A systematic mapping of the literature Author, Journal of Systems and Software, Vol. 125, 93-118
- Baqais, B. A., Alshayeb, M., (2020) Automatic software refactoring: a systematic literature review, Software Quality Journal, Vol. 28, 459-502
- Fawei, B., Pan, J.Z., Kollingbaum, M. et al., (2019) A Semi-automated Ontology Construction for Legal Question Answering, New Gener. Comput. 37, 453-478
- Fowler, M., (1999) Refactoring: Improving the Design of Existing Code, Addison-Wesley Professional

- Gamma, E., Helm, R., Johnson, Vlissides, J., (1994) Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley
- Gangemi, A., Catenacci, C., Ciaramita, M., 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, Vol. 14-15-16
- Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J., (2006) Modelling Ontology Evaluation and Validation, ESWC, LNCS, Vol. 4011, 140-154
- Khurshid, A., Lee, G, (2005) Automatic Ontology Extraction from Unstructured Texts, ODBASE 2005, LNCS, Vol. 3761, 1330-1346
- Kosiuczenko, P. (2020) UML Based Ontology for the Extraction of Historical Geographic Information, Proceedings of the 36th International Business Information Management Association (IBIMA), 2020
- Kosiuczenko, P. (2022) Querying UML Based Ontology of Historical Geographic Information, Proceedings of the 39th International Business Information Management Association (IBIMA), 2022
- 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, Vol. 15(1)
- OMG, (2014) Object Constraint Language, Spec. ver. 2.4, January
- OMG, (2017) Unified Modeling Language, Spec. ver. 2.5.1, December
- Savnik, I., Stefan, J., Tahir, Tari., (1997) Querying Conceptual Schemata of Object-Oriented Databases, Proceedings of Seventh International Workshop on Database and Expert Systems Applications, IEEE Explore, DOI:10.1109/DEXA.1996.558349
- Sulimierski, F., Chlebowski, B., Walewski, W., et. al., (1880—1902) Slownik geograficzny Krolestwa Polskiego, 1880-1902, Warsaw
- Wong, W., Liu, W., Bennamoun, M., (2012) Ontology Learning from Text: A Look Back and into the Future, ACM Computing Surveys, Vol. 44(4),