

Stefan Rozmus<sup>1</sup>, Maciej Kiedrowicz<sup>2</sup>

## **DATA WAREHOUSES FOR HGIS – IDENTIFYING USER REQUIREMENTS AND CREATING DATABASES**

**Abstract:** Historical geographic information systems (HGIS) are the key focus of a relatively new branch of science that deals with analyzing spatial data in a historical context. One of the approaches that may be taken when performing such studies relies on data warehouses. While the rules for building data warehouses for the purpose of conducting business analyses are well known and thoroughly described, this issue has not yet been widely reflected in the subject literature in connection with HGIS. This paper attempts to take into consideration HGIS-related requirements in the first two phases of the process of building a data warehouse. In the first phase, during which user requirements are identified, the use of a well-recognized point model has been proposed, with its geographic portion expanded to offer the ability to distinguish many-to-many relationships between various levels of the dimension hierarchy. The proposed solution allows to intuitively model a multidimensional space on a plane, without requiring the user to be familiar with the technical details involved. Technical issues are delegated to the designer in the form of guidelines that define the rules to be followed while transforming the point model into the structure of the database that has been adopted by the designer. As far as the second stage is concerned, the rules for creating a relational data model based on the previously-developed point model have been specified. A solution has been proposed that relies on bi-temporal tables to store data change history, including spatial data, and to create the so-called bridges that connect many-to-many relationships with database triggers, responsible for maintaining the correct chronology of historic facts. Furthermore, select elements of practical examples have been included in the work as well.

**Keywords:** HGIS, point model, dimension, slowly changing dimension, bi-temporal table

Received: 19 May 2025; accepted: 19 June 2025

© 2025 Authors. This is an open access publication, which can be used, distributed and reproduced in any medium according to the Creative Commons CC-BY 4.0 License.

---

<sup>1</sup> Military University of Technology, Faculty of Cybernetics, Warsaw, Poland, ORCID ID: <https://orcid.org/0000-0003-1816-927X>, email: [stefan.rozmus@wat.edu.pl](mailto:stefan.rozmus@wat.edu.pl)

<sup>2</sup> Military University of Technology, Faculty of Cybernetics, Warsaw, Poland, ORCID ID: <https://orcid.org/0000-0002-4389-0774>, email: [maciej.kiedrowicz@wat.edu.pl](mailto:maciej.kiedrowicz@wat.edu.pl)

## Introduction

Historical GIS (HGIS) are the product of an interdisciplinary field of science that integrates geographic information tools and methodologies with historical research. Historical research aims to provide us with a better understanding of the past by interpreting data originating from primary sources (originating directly from the period in which they were created) and secondary sources (created by verifying underlying evidence, e.g. monographs, biographies, scientific and popular science articles). The subject of historical research is extremely broad and covers various aspects of human activity as well as various social, political, economic and cultural phenomena. Historical facts (such as, for instance, changes affecting national borders, newly erected cities or newly adopted clothing styles) may be interpreted and analyzed in many different ways. Geographic Information Systems (GIS) are a tool that is perfectly suited for reconstructing such facts. They offer the techniques and tools for creating, managing, displaying, and analyzing historical data. History researchers have, at their disposal, numerous GIS functionalities (e.g., digitization) allowing them to convert spatial features presented on historical paper maps or found in other archival documents into digital maps. These, supplemented with descriptive data, serve as a basis for building spatial-temporal databases that combine location and spatial data with non-spatial information. Databases created using this methodology may be used for various types of studies, as they are capable of taking into account the location as well as temporal-spatial relationships of historical facts. Typical solutions used in this field are based on database structures that are adapted to transaction processing. It may be difficult, however, to conduct advanced historical research based on such solutions.

The comparative method is one of the most valuable tools in the work of any historian. It is used to establish historical facts which are not directly identified in the sources as well as to substantiate hypotheses concerning cause-effect relationships and historical generalizations. By building an HGIS system with the use of a data warehouse, we obtain an excellent tool facilitating the comparative method. The advantages of a tool constructed using such an approach stem directly from the features of the data warehouse, fitting very well with the needs of the comparative method. The following observations may be highlighted:

- Data warehouses allow for the collection and integration, in one place, of data from various sources – a characteristic that is crucial in the case of HGIS, where data often originates from archives, historical maps, administrative documents, population registers, etc.
- In HGIS systems, it is crucial to track changes (for instance in national borders, population distribution, or infrastructure) taking place over time. Slowly changing dimensions are used for this specific purpose, as these allow to store historical states of individual objects and analyze their evolution over time (Inmon, 2005; Kimball & Ross, 2013).
- Data warehouses facilitate multidimensional analyses as well. Facts may be analyzed according to various dimensions describing the multidimensional space, such as time,

location, historical events, or people. The placement of numerous facts in a multidimensional space allows to search for potential relationships between facts and facilitates identification of their mutual impacts.

- Data warehouses may be integrated with GIS systems, which enables to conduct spatial analyses, such as examining the impact of neighborhoods or distances between objects on the level of importance of a given object, compared with other objects of the same type.
- Data warehouses offer a solid foundation for creating interactive reports and visualizations. Thanks to their integration with such tools as Power BI (Business Intelligence class toolkit used for analyzing data) or ArcGIS, one may create thematic maps and border change animations. Population density may be analyzed using this feature as well.
- Thanks to access to historical data presented in a multidimensional manner, one may create comparative models, simulate alternative scenarios and analyze the causes and effects of political or economic decisions.

As one may see, the use of data warehouses in HGIS systems allows for the effective collection, analysis and visualization of complex historical data in spatial and temporal contexts. Data centralization, the ability to perform a multidimensional analysis and support of spatial analyses – these are the features that open up new prospects for historical researchers, allowing them to better understand the processes and events that have shaped our past.

This paper aims to present a specific approach to building a data warehouse for the needs of HGIS. Due to the complex and broad character of the issues that need to be dealt with in the course of an HGIS system's production cycle, this work tackles only problems encountered during the first two phases of building a data warehouse, i.e.:

- Identification and specification of user needs – phase 1.
- Building a database for a data warehouse – phase 2.

In the first phase, during which user requirements are identified, the use of a well-recognized point-based model has been proposed, with its geographic portion expanded to offer the ability to distinguish many-to-many relationships between various levels of the dimension hierarchy. The proposed solution allows to intuitively model a multidimensional space on a plane, without requiring the user to be familiar with the technical details involved. Technical issues are delegated to the designer in the form of guidelines that define the rules to be followed while transforming the point model into the structure of the database that has been adopted by the designer. As far as the second stage is concerned, the rules for creating a relational model based on the previously-developed point model have been specified. Specifically, a solution has been proposed that relies on bi-temporal tables to store data change history, including spatial data, and to create the so-called bridges that connect many-to-many relationships, and database triggers.

## Material and methods

**Identification of HGIS requirements.** Identification of the requirements of an HGIS system is a key stage in the process of designing the complete solution. It is not an easy task, as HGIS takes into consideration, by its nature, the spatial and temporal variability of historical objects, as well as the socio-political context of a given period. Therefore, the process of identifying specific requirements calls for an interdisciplinary approach that combines knowledge from the fields of history, geography, computer science and data science. Requirements applicable to HGIS systems may be identified based on various sources, such as historical documentation, field-specific experts, end users, but also GIS standards. Standards applicable to geospatial data and services developed by OGC (*Open Geospatial Consortium*) need to be taken into consideration here, along with the guidelines worked out in compliance with the INSPIRE (*Infrastructure for Spatial Information in Europe*) directive. Identification of functional requirements applicable to an HGIS system that relies on a data warehouse boils down to drawing up a list of reports that will serve as a basis for building a point model (Chodkowska-Gyurics, 2014; Todman, 2011). The point model may be treated as a simplified model of relationships between entities that constitutes a point of departure for building an analytical database.

The point model comprises two parts: graphical and descriptive. The graphical part depicts the multidimensional space and is developed at the initial requirement identification stage. The descriptive part contains additional details concerning the point model objects. These are supplemented, on an on-going basis, both during the user requirement identification phase and during the subsequent stages of building the data warehouse. The meaning of the term “object” is derived from the context in which it has been used. It may be a historical object or an object of the point model. To discuss the point model, it is convenient to use a simple example. Let us assume that we are interested the regions of a given country which were the locations of the highest number of battles over the years. Therefore, the HGIS system should enable, among other things, generating reports presenting, for example, the number of battles fought in individual regions of countries in a given period, along with the number of casualties. From this sample report (let us call it “Battles in regions”) it follows that the battle engagement event is the analyzed parameter. It is measured by the number of battles fought and casualties suffered and deals with a two-dimensional space, defined by the dimensions of time and region. After agreeing on the hierarchy of dimensions, a graphic part of the point model was developed for each of the dimensions (Fig. 1).

The diagram shows all the components required for creating the graphical part of the point model, i.e. a circle, a line and specific captions. The circle represents a fact and the caption located next to the circle (in this specific case *Battle Engagement Event*) is the name of the fact. The captions that are connected by lines indicate the names of the dimensional hierarchy levels defining the multidimensional space in which the fact in question may be analyzed. The lowest hierarchy level of each dimension must always be associated with a fact. The lines are the equivalent of one-to-many relationships, where the quantity of *one* refers to a parent level and the quantity of *many* to a sub-level. A level

that is related to the fact is a parent level. It needs to be noted that this part of the model is not highly formalized. In particular, it lacks specifically defined rules concerning the naming of the hierarchy of dimensions. It may therefore be assumed that the name of the hierarchy is identical to the name of the lowest level of the hierarchy. When the dimension has only one level (in this scenario, we are dealing with a single-level hierarchy or no hierarchy), such an approach is definitely adequate. However, in the case of a multi-level hierarchy comprising a group of objects, the name of the lowest level of the hierarchy does not always reflect the essence of the entire hierarchy. The multi-level hierarchy of the time dimension may serve as a typical example here. Therefore, a symbol of a group may be entered and may be assigned a name that reflects the essence of the grouped elements. Here, it has been decided that a rectangle drawn using a dashed line, containing the dimension hierarchy, will serve as the symbol of the group. More than one hierarchy may be present within a single dimension. In the example considered (Fig. 1), the following hierarchies are present:

- Time, grouping Year, Decade and Century levels,
- Location, grouping Battle, Place and Country levels.

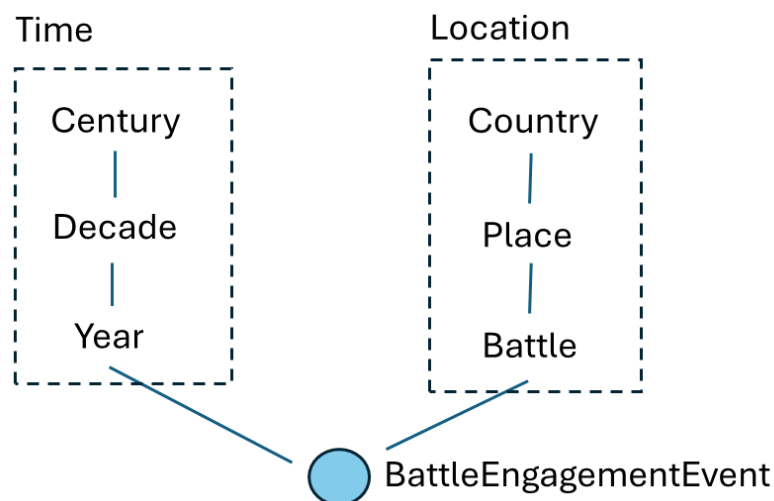


Fig. 1. Graphical part of the point model developed based on an example of the “Battles in regions” report

Source: own compilation

Having defined the preliminary requirements and having developed the graphical part of the point model, the next step consists in creating the descriptive part that details and supplements the information already collected. The descriptive part may be presented, depending on the needs, in the form of one or more spreadsheets containing tables whose columns describe various aspects of the requirements in relation to each object specified in the graphical part of the point model. The number of columns will increase as the work progresses during the subsequent phases of the process of building the data warehouse. In the requirement identification phase in which it is the business analyst who plays the key role, it is necessary to determine the name and type of each object and the method used to approach changes in the object taking place over time.

A description, along with justification of the assumptions made and limitations applied, needs to be provided as well. Therefore, in the first approach to identifying requirements, the spreadsheets comprising the descriptive part should contain four columns: Object name, Object type, Retrospection, and Description. In the next phase in which the database for the data warehouse is created, taking into account the ability of the business analyst to contact the end user, it is worth supplementing the spreadsheets with two more columns: Data type and Mandatory (Table 1). Before presenting the descriptive part of the graphical portion of the point model that has already been shown (Fig. 1), we shall discuss the meaning of the individual columns of the spreadsheet, which are important at this stage of the process of identifying applicable requirements.

Table 1. Columns of the spreadsheet required in the descriptive part of the requirement identification phase

| Object name | Object type | Retrospection | Description | Data type | Mandatory |
|-------------|-------------|---------------|-------------|-----------|-----------|
|             |             |               |             |           |           |

Source: own compilation

The individual columns have the following meanings:

- Object name. Its meaning is not always obvious and usually requires further clarification in the *Description* column.
- Object type. Three types of objects may be distinguished: entity, attribute and relationship. An entity is an abstract object representing a specific class of objects (e.g. persons, employees, events, notions) sharing identical features called attributes. In a point mode, facts and dimension levels are considered to be entities. A relationship is the link between entities. Relationships may exist between specific dimension levels or between a dimension level and a fact.
- Retrospection (literally: “taking a look back”) – defines whether a given object undergoes changes over time. If so, it also states whether the history of its changes need to be stored, from the analytical point of view, or not. A retrospection may be True, False or Permanent. The meanings of these values in relation to specific object types are presented in Table 2.
- Description – any crucial description, assumption or restriction explaining the meaning of an object.
- Data type – type of data specified for an attribute (does not apply to the remaining object types).
- Mandatory – specifies whether a given attribute is mandatory (it must not assume the NULL value) or optional (may assume a NULL value). It may assume one of the following values: Yes (mandatory attribute) or No (optional attribute).

As the work progresses, the spreadsheets will usually be supplemented with additional columns identifying, for instance, sources of data for the individual attributes, transformations of source data output forms and identification of attribute values, object access rights, etc. However, as these are not crucial from the point of view of the overall goal, they will not be taken into consideration in this paper.

Table 2. Handling changes that affect various object types over time

| Object type  | Retrospection | Explanation                                                                                                                                  |
|--------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Entity       | True          | During the life time of an entity, periods of discontinuity may be encountered and it is necessary to remember the history of those periods. |
|              | False         | During the life time of an entity, periods of discontinuity may be encountered, but only the current status of the entity is remembered.     |
|              | Permanent     | The life time of an entity never discontinues (periods of discontinuity will never occur).                                                   |
| Relationship | True          | The duration of each relationship must be recorded and stored.                                                                               |
|              | False         | Only the current relationship must be recorded and stored.                                                                                   |
|              | Permanent     | No changes to an established relationship are envisaged.                                                                                     |
| Attribute    | True          | Values of the attribute may change, but each of them needs to be recorded and remembered.                                                    |
|              | False         | Values of the attribute may change, but only the current value must be remembered.                                                           |
|              | Permanent     | No changes to an established attribute are envisaged.                                                                                        |

Source: own compilation

With the explanations provided above taken into consideration, the descriptive part of the point model, supplementing the graphical portion presented in Fig. 1, has been created. Due the low number of objects, the descriptive part occupies one spreadsheet only (Table 3).

Table 3. Descriptive part of the point model developed based on an example of the “Battles in regions” report

| Object name                   | Object type  | Retrospection | Description                                                                          | Data type | Mandatory |
|-------------------------------|--------------|---------------|--------------------------------------------------------------------------------------|-----------|-----------|
| BattleEngagement Event        | Entity       | Permanent     | Battle engagement event involving armed forces                                       | n/a       | n/a       |
| Century                       | Entity       | Permanent     | Century, according to the Gregorian calendar                                         | n/a       | n/a       |
| Decade                        | Entity       | Permanent     | Decade, according to the Gregorian calendar                                          | n/a       | n/a       |
| Year                          | Entity       | Permanent     | Year, according to the Gregorian calendar                                            | n/a       | n/a       |
| Country                       | Entity       | True          | Country – periods of discontinuity may occur and must be remembered                  | n/a       | n/a       |
| Place                         | Entity       | Permanent     | Location at which the battle was fought                                              | n/a       | n/a       |
| Battle                        | Entity       | False         | The battle may have been fought with interruptions, but these are not important here | n/a       | n/a       |
| BattleEngagement Event → Year | Relationship | Permanent     | The battle engagement event must be associated with a specific year                  | n/a       | Yes       |
| Year → Decade                 | Relationship | Permanent     | Relationship established based on the Gregorian calendar                             | n/a       | Yes       |

|                                            |              |           |                                                                                                                                      |                         |     |
|--------------------------------------------|--------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------|-------------------------|-----|
| Decade → Century                           | Relationship | Permanent | Relationship established based on the Gregorian calendar                                                                             | n/a                     | Yes |
| BattleEngagement Event → Battle            | Relationship | Permanent | The battle engagement event must be associated with a specific battle                                                                | n/a                     | Yes |
| Battle → Place                             | Relationship | True      | The battle may have been fought at one or several locations (concurrently or with interruptions)                                     | n/a                     | Yes |
| Place → Country                            | Relationship | True      | The battles fought may have resulted in territorial affiliation of a given location. The history of those changes must be remembered | n/a                     | Yes |
| BattleEngagement Event.<br>NumOfBattles    | Attribute    | Permanent | Number of battle engagement events                                                                                                   | integer                 | Yes |
| BattleEngagement Event.<br>NumOfCasualties | Attribute    | Permanent | Number of casualties suffered in the battles                                                                                         | integer                 | Yes |
| Year.Number                                | Attribute    | Permanent | Year (number) according to the Gregorian calendar                                                                                    | integer                 | Yes |
| Decade.Number                              | Attribute    | Permanent | Number of the decade according to the Gregorian calendar                                                                             | integer                 | Yes |
| Century.Id                                 | Attribute    | Permanent | Century identifier                                                                                                                   | integer                 | Yes |
| Century.Name                               | Attribute    | Permanent | Number of the century, expressed using Roman numerals, with a BCE suffix for the “before Christ” era                                 | string                  | Yes |
| Place.Id                                   | Attribute    | Permanent | Identifier of the location at which one or more battles have been fought                                                             | integer                 | Yes |
| Place.Name                                 | Attribute    | True      | Name of the location at which one or more battles have been fought. This name may change and such a change must be remembered        | string                  | Yes |
| Place.Geo                                  | Attribute    | Permanent | Geographic coordinates of a battlefield (point or area)                                                                              | depends on the database | Yes |
| Country.Id                                 | Attribute    | Permanent | Country identifier                                                                                                                   | integer                 | Yes |
| Country.Name                               | Attribute    | True      | Name of the country. This name may change and such a change must be remembered.                                                      | string                  | Yes |
| Battle.Id                                  | Attribute    | Permanent | Battle identifier                                                                                                                    | integer                 | Yes |
| Battle.Name                                | Attribute    | Permanent | Name of the battle                                                                                                                   | string                  | Yes |

Source: own compilation

In the general scenario, when a point model is being built, a need may arise to distinguish more than one fact, with some (or all) of them sharing the predetermined dimensions (Fig. 2).



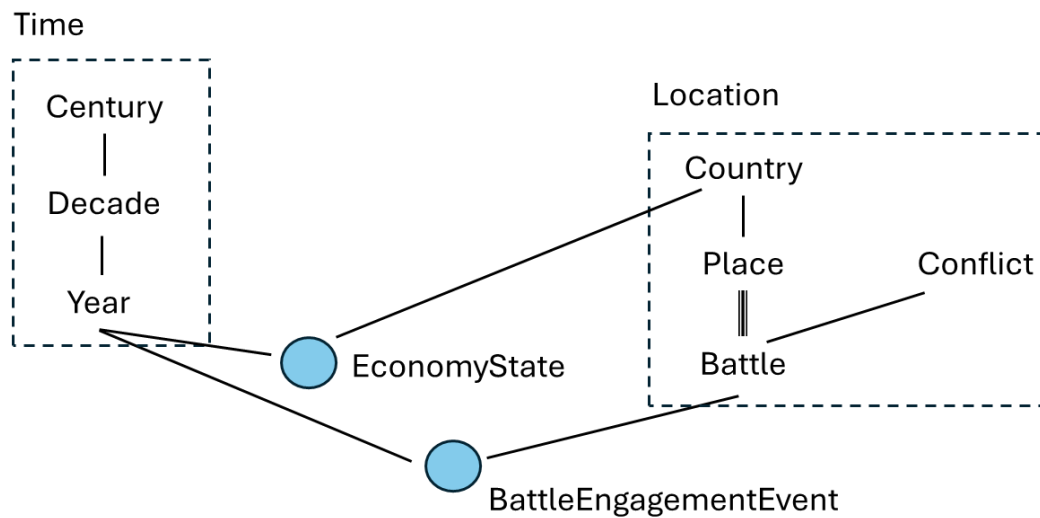


Fig. 2. Example of a graphical part of a point model with two facts

Source: own compilation

The above diagram supplements the from Fig. 1 with a new fact (EconomyState) describing the state of the economy at a given time and in a given country. Typical EconomyState-related metrics include population, gross domestic product (GDP), debt, or agricultural production. Because BattleEngagementEvent and EconomyState facts are linked by Time and Location dimensions, it is possible to analyze the mutual influence of the economy on the initiation of conflicts, and also to assess the impact of the results of existing conflicts on the economy. For example, a persistent downward GDP trend observed over a longer period of time may be the root cause of increased social tensions and, consequently, may lead to a conflict with another country. Conversely, losses suffered in decisive battles fought in the course of the resulting conflict may lead to a decrease in GDP of the defeated country. This only briefly shows the enormous potential of discovering the mutual dependencies between facts that is offered by HGIS systems based on data warehouses.

The point model serves as a foundation for designing a structure of a database for a data warehouse. In the case of relational databases (the ones we are discussing in this paper), the designer will generally use the fact constellation structure. Depending on the needs, the dimension may have the form of:

- a single table (de-normalized form of the dimension) which contains columns corresponding to the attributes of all entities that make up all hierarchies of the dimension (in the examples given, dimensions, for simplicity, have only one hierarchy each),
- a set of related tables (normalized form), with each of them containing columns that correspond to individual entities in the hierarchies of the dimension.

It should be mentioned that 1-to-many is the typical relationship between the parent and child entity in the dimension hierarchy. However, *many-to-many* relationships may occur as well. Such relationships are not distinguished in the point models used currently.

The authors propose to introduce a distinction of such relationships in the form of a triple line (Fig. 2, Battle → Place relationship). The proposed approach allows to intuitively model a multidimensional space on a plane and still does not require the user to possess knowledge about technical details. Technical issues are delegated to the designer in the form of guidelines that define the rules to be followed while transforming the point model into the structure of the database that has been adopted by the designer.

**Requirements applicable to databases for HGIS systems.** Due to the nature of historical data, the databases in which they will be stored needs to satisfy specific requirements. In particular:

- historical data are related, by definition, to the time of existence of the historical objects they describe. A historical object is understood as a unit of data representing a specific entity, having a unique identity and a set of attributes describing its properties,
- historical data may originate from different sources. Hence, data concerning the same historical object may differ, both in terms of the value of its features and the time of its existence (for example, different maps related to the same historical period may present different outlines of country borders),
- in historical research focusing on a given object, there are usually periodic gaps in data concerning the object being studied. Therefore, after obtaining additional data, these gaps must be filled,
- upon obtaining further data describing a historical object that is already recorded in the database, a need may arise to correct the data of such an object. Both the values of the object's features and the time during which they remained valid in relation to the previous period may be subject to such a correction,
- data concerning a given object, obtained from several sources or at different times, may overlap in terms of their validity intervals, i.e. the time in which they were true.

It follows from the above that when considering historical data in the context of time, one should take into account both the time in which such data were true in the past (validity time) and the time that has passed since they were saved in the database (transaction time). This requirement may be fully satisfied by using a relational database offering temporal extensions – the data base management systems by main database vendors (IBM, Oracle, Microsoft, Teradata) offer temporal extensions based on the SQL 2011 standard. Although issues related to temporal databases have been repeatedly raised in the literature on the subject (Johston, 2014; Snodgrass, 2010; Saracco et al., 2012), the authors are of the opinion that several of importance from the point of view of the construction of a database still need to be explained. The state of the object under consideration is a key concept here. The state of an object is a set of values assigned to all of its attributes at a given point in time. A change to the value of any attribute modifies the state of the object. Let us trace how the states of an object change during its life cycle, assuming that each state of the object is remembered in the database. It is therefore required to assign two time stamps to each of the object states, i.e. a validity time stamp and a transaction time stamp. Interval markers used in temporal databases have been relied upon in the project. A graphical presentation relying on bi-temporal terms, i.e.

a two-dimensional time space defined by the validity time axis VT and orthogonally positioned transaction time axis TT is a convenient form of presenting changes in object states. The object appears in the database at the moment at which its first state is recorded in the database (Fig. 3a). This state is the current state marked as  $S_1-A$ . The validity time stamp is defined by the  $\langle vt_1, Now \rangle$  pair, the transaction time stamp is defined by the  $\langle tt_1, Now \rangle$  pair, where arrows  $\uparrow \rightarrow$  symbolize the passage of time (of validity and the transaction, respectively). The *Now* is a symbolic temporal value used in time stamps to indicate that a given state remains current. This state lasts continuously until the next, new state is recorded in the database (Fig. 3b). Saving a new state, in the data base, at time  $tt_2$ , with its validity beginning at time  $vt_2$ , renders the previous state a historical state  $S_1-H$  and the new state a current state  $S_2-A$ . The change from the current state to a historical state is expressed by replacing the *Now* value in the validity time stamp with the time at which the validity of the new state commences. Therefore, timestamps for the historical state are defined by the  $\langle vt_1, vt_2 \rangle$ ,  $\langle tt_1, Now \rangle$  pairs, and for the new current state, by the  $\langle vt_2, Now \rangle$ ,  $\langle tt_2, Now \rangle$  pairs. There may be many such consecutive states.

If the object ends its life (Fig. 3c), the current state will be changed to the historical  $S_2-H$  state with  $\langle vt_2, vt_3 \rangle$  and  $\langle tt_2, tt_3 \rangle$  validity time stamps. The new  $S_3-D$  state means that the object is “dead”. An object is considered “dead” in a bi-temporal data base if its last known state has a specific end moment of the validity time, after which there is no further information about the object. Time stamps corresponding to the validity and transaction of the new state are represented by the  $\langle vt_2, vt_3 \rangle$  and  $\langle tt_3, Now \rangle$  pairs, respectively. As one may notice, this state is equivalent to the previous state in terms of the validity time. If a new state of the “dead” object appears (Fig. 3d) with a validity time equal to or greater than the validity time of the end of its life (here  $vt_4 > vt_3$ ), the object is considered to have been brought back to life. The new  $S_4-A$  state becomes the current state. The validity and transaction time stamps of the new state will be determined in the same manner as the state of the object saved in the data base for the first time (here  $\langle vt_4, Now \rangle$  and  $\langle tt_4, Now \rangle$ ). Meanwhile, in the transaction time stamp of the  $S_3-D$  state, the *Now* value is changed to  $tt_3$ .

The presented descriptions of changes in the object’s states may be helpful, when deciding about the type of data base to be used, in assessing the capabilities of the temporal mechanisms inherent in the data base’s management system. This applies to the method of storing the history of and operations performed on historical data.

As already mentioned, after obtaining subsequent data describing a historical object already saved in the data base, a need may arise to correct the details of such an object. This may mean, in general, the need to correct the indicated state, but also to divide a state or merge several states. Due to the limited scope of this work, the description of corrections has been limited to the correction of the specified state of the object. A detailed description of such corrections is presented in (Rozmus, 2021).

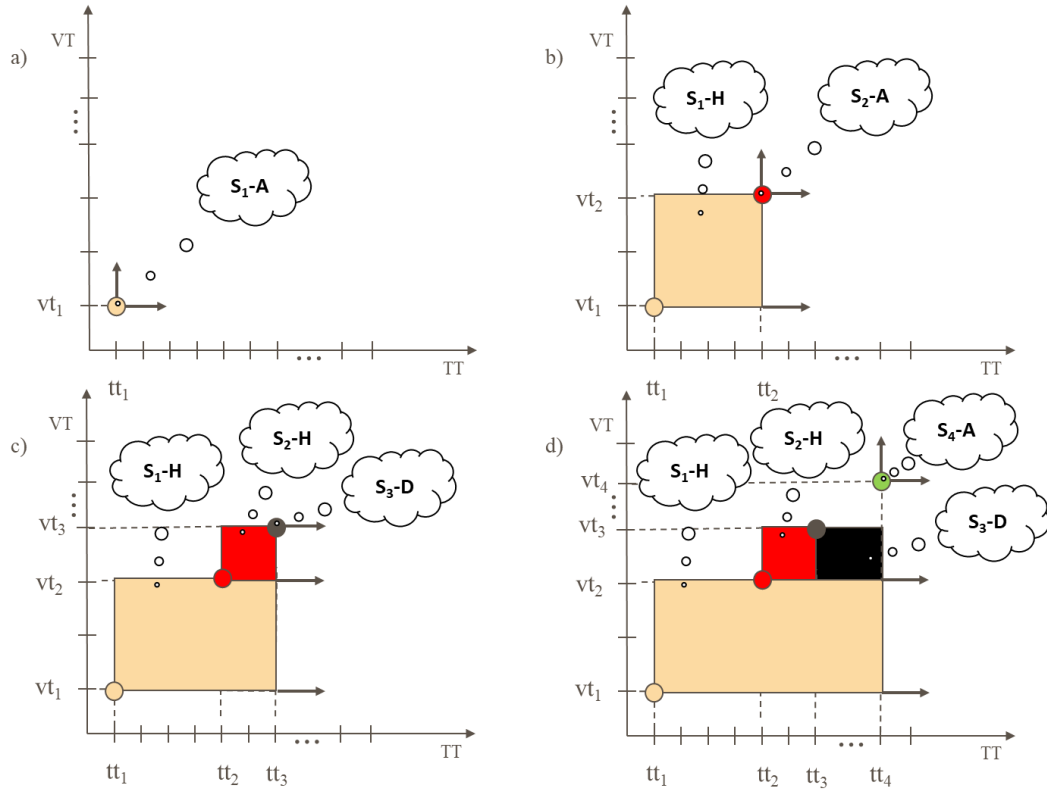


Fig. 3. Graphical illustration of object status changes in the bi-temporal approach  
Source: own compilation

Correction of the object state means changing the value of one or more attributes of this state. As a result of a correction of the state of an object state (corrected state), a new state of the object is created with changed values of corrected attributes (state after correction) and the state is marked as corrected (Fig. 4). In the corrected state (Fig. 4a), the validity time stamp, regardless of the number of corrections, does not change (here  $\langle vt_1, vt_2 \rangle$ ), while the Now value in the transaction time stamp is replaced by the time of the correction. Hence, in the case of the first correction (Fig. 4b), the transaction stamp is specified by the  $\langle tt_1, tt_4 \rangle$  pair, in the case of the second correction (Fig. 4c), the transaction stamp is specified by the  $\langle tt_4, tt_5 \rangle$  pair. The expiration time stamp in the new state is the same as the expiration time stamp of the corrected state (here  $\langle vt_1, vt_2 \rangle$ ), while the transaction time stamp assumes the  $\langle tt_{cor}, Now \rangle$  values, where  $tt_{cor}$  denotes the time of the correction in the data base (here  $tt_4$  – the first correction, and  $tt_5$  – the second correction). It should be noted that the correction may concern both the historical and the present state.

The descriptions presented refer to changes in the states of individual objects. The limited scope of the work does not allow for a broader description of changes in the states of related objects. Such a description has been presented in (Rozmus, 2021).

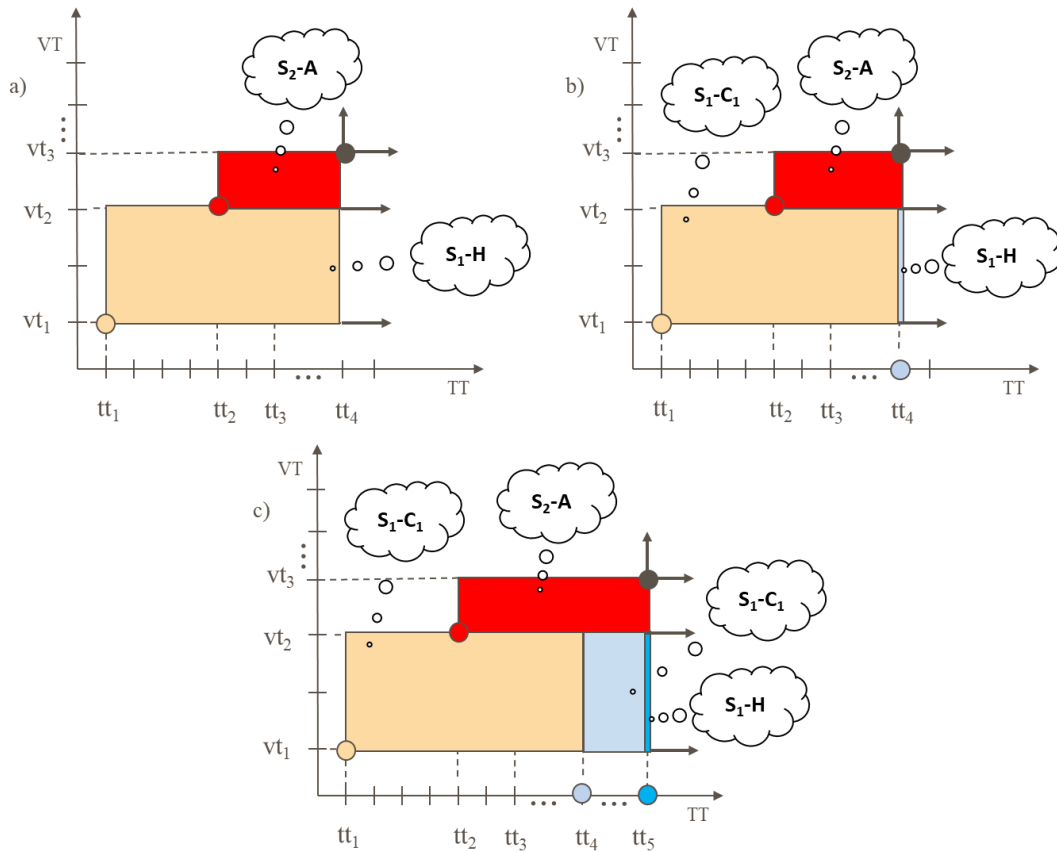


Fig. 4. Graphical illustration of the correction of an object's state in the bi-temporal approach  
Source: own compilation

As far as the preservation of the chronology of historical events in relation to the validity time is concerned, it is important to make sure the individual states of the object do not overlap. This means that the time intervals designated by the validity time stamps of any two object states must meet one of the following conditions:

- they share not common point (Fig. 5a),
- they share a single common point (Fig. 5b),
- they share two common points (Fig. 5c).

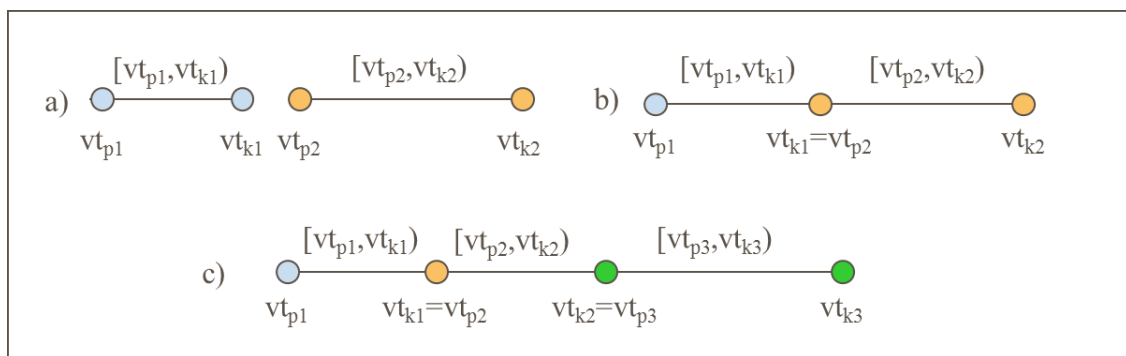


Fig. 5. Conditions of non-overlapping historical states of an object  
Source: Rozmus & Kiedrowicz, 2019

Fulfillment of the above conditions can be ensured by associating an appropriate trigger (Rozmus & Kiedrowicz, 2019) with each temporal table, which, when trying to save a new state, returns an exception if none of the conditions are met. In the general case, an attempt to record a state that overlaps with other states does not have to result in an error if that state is considered to be of the corrective variety. It would be outside the framework of this paper to describe such cases.

## Results and discussion

To illustrate the ability of a data warehouse-based HGIS system to create analytical reports, a simplified data warehouse was built (based on the previously presented point model) and supplied with data on 13 randomly selected battles fought between 1991 and 2014. Microsoft tools were used to build the data warehouse: the SQL Server relational data base management system for storing and manipulating data, SSAS (SQL Server Analysis Services - Microsoft's data analytics tool) for generating a multidimensional cube constituting a data source for the reports, and Power BI for generating sample, interactive reports (Fig. 6 – Fig. 9).

The locations of the battles are shown against the background of the political map of the world (Fig. 6). The locations of battle engagements are represented by bubbles, with their size depending on the number of casualties suffered in the battle by both sides. Additional information about the battle may be obtained in the form of a tooltip by pointing the cursor at the location of a given battle.

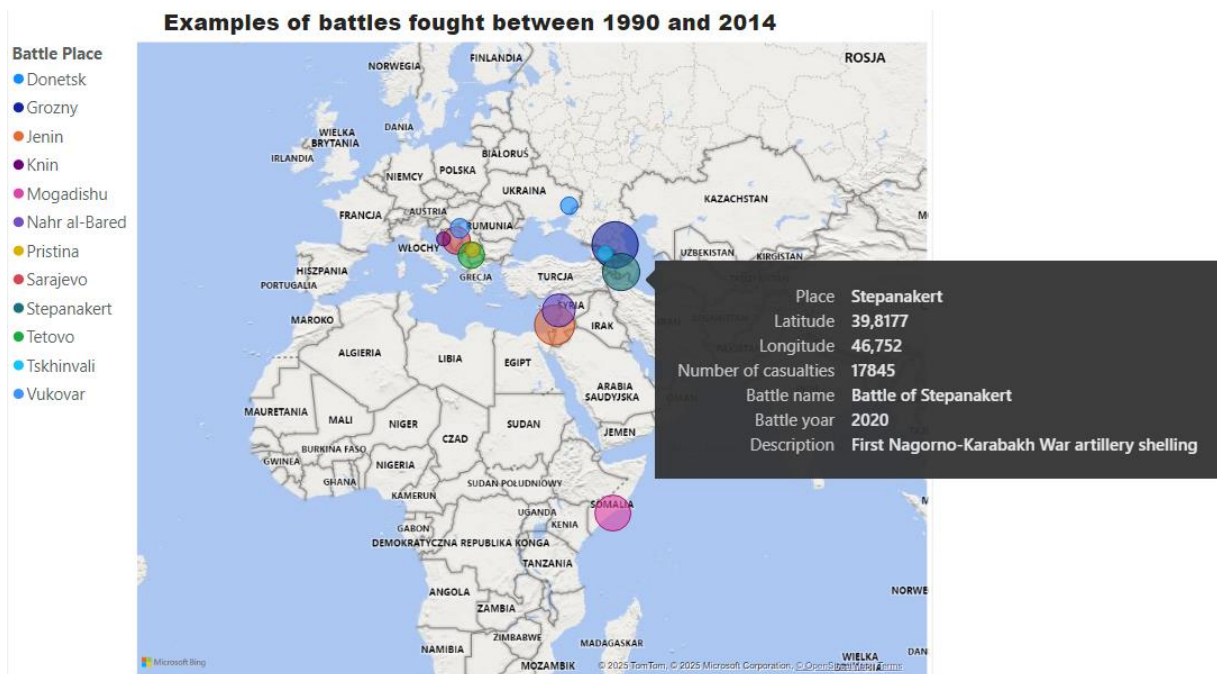


Fig. 6. Examples of battles fought between 1999 and 2014

Source: own compilation

The total number of casualties in each decade, along with additional information about these battles (here: name of the battle, year of the battle, number of casualties suffered in the battle and the country in which the battle was fought) is presented in Fig. 7.

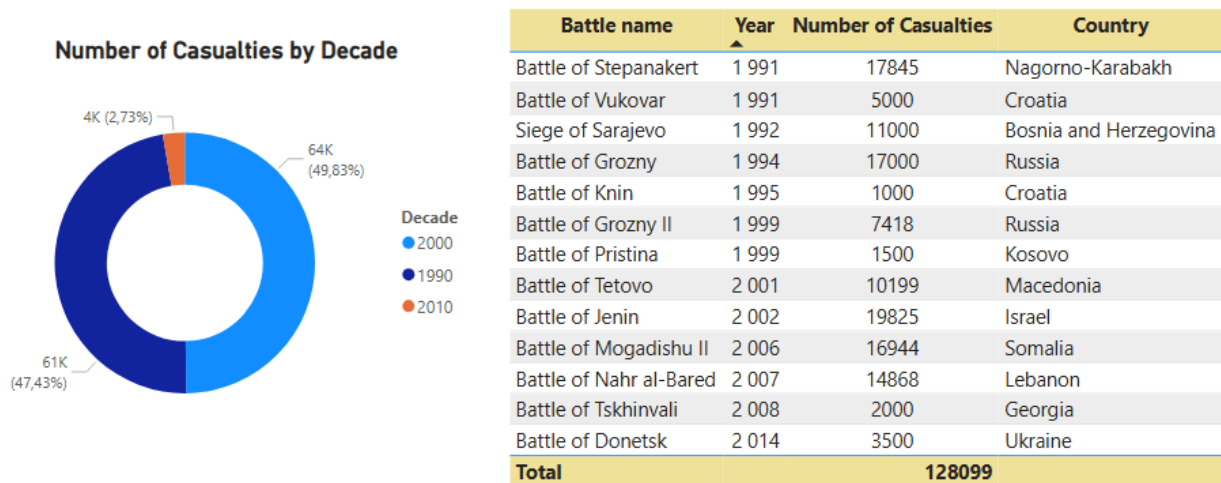


Fig. 7. The number of casualties in specific decades and additional information concerning the battles

Source: own compilation

Since the generated multidimensional cube was based on the previously described constellation of facts, it was possible to examine the impact of the battles fought on the GDP and population of the country in which they took place. For example, by analyzing the demographic trend (Fig. 8) and the GDP trend (Fig. 9) in the periods spanning 3 years before and 3 years after the Battle of Pristina, it can be assumed that the conflict had a short-term negative impact on the economy, which was manifested by a noticeable drop in GDP in the year of the battle, but in the following years the economy started to recover. At the same time, demographic data may indicate a halt or slowdown in population growth, potentially resulting from war losses, migration or deterioration of living conditions in the region affected by the conflict. It should be emphasized, however, that this is only a hypothesis that may be the subject of additional research based on a wider set of data and supplemented by an analysis of historical sources.

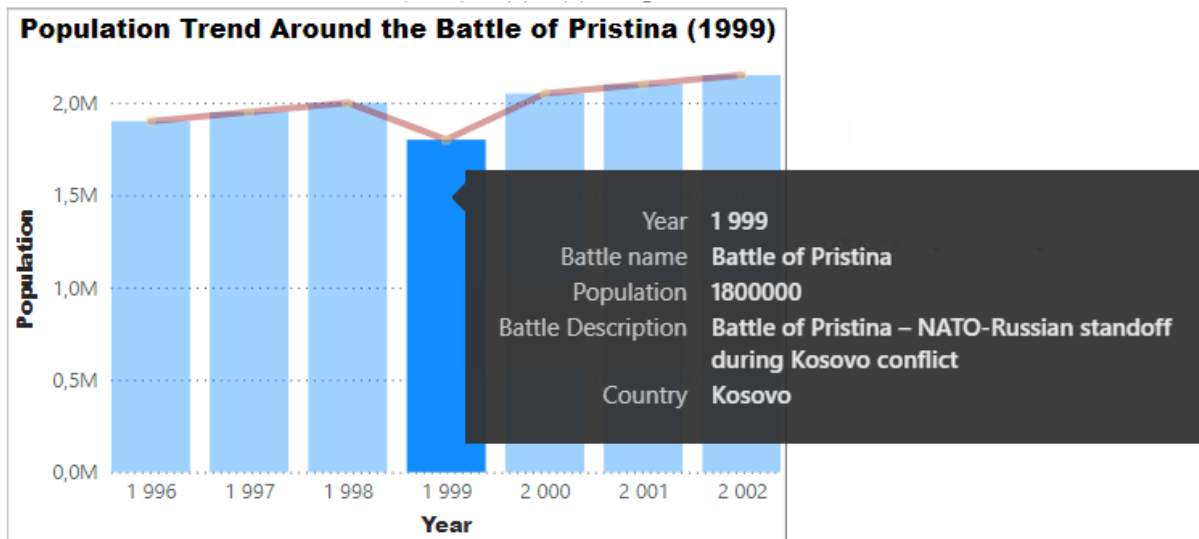


Fig. 8. Demographic trend around the battle of Pristina (1999)

Source: own compilation

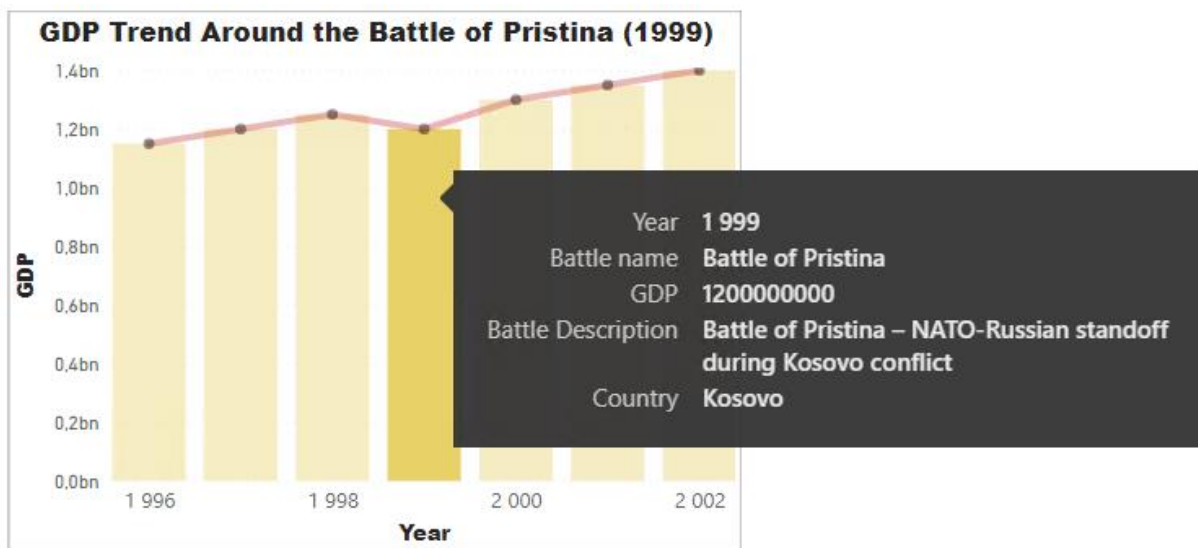


Fig. 9. GDP trend around the battle of Pristina (1999)

Source: own compilation

## Conclusions

The approach to designing data warehouses for historical geographic information systems presented in this paper indicates that existing modeling methods can be successfully adapted to the specific nature of historical and spatial data. Extension of the point model to include the ability to represent many-to-many relationships allows for more precise representation of complex dependencies observed in HGIS data, while simultaneously maintaining its intuitive character. The proposed principles based on which the point model may be transformed into a relational data model – taking into account bi-temporal mechanisms and logical bridges – form a solid foundation for future



development of data warehouse design methods intended for historical-spatial applications.

The practical part of the work shows that such a data model facilitates advanced spatial and temporal analyses. For example, it was possible to mark the locations of battles on a political map of the world, taking into account the number of casualties suffered, present war losses using a decade-by-decade approach, and link these events with economic and demographic data. The analysis of the Battle of Pristina in which the number of casualties was compared with the GDP and population change trends occurring in the three years prior to and after the battle, shows the potential the proposed solution offers for examining cause-effect relationships in historical data.

The potential areas of further research include the principles relied upon while handling data corrections and situations in which overlapping object state validity periods exist. Selecting the right state in such scenarios may require resolving temporal conflicts, and sometimes also updating neighboring states – by separating or combining them. Introduction of consistent rules for resolving such situations is an important element in the process of improving the quality of historical data and may contribute to increasing the reliability of analyses conducted within historical geographic information systems.

## **Acknowledgment**

This work was financed/co-financed by Military University of Technology under research project UGB 531-000023-W500-22.

## **References**

- Chodkowska-Gyurics A. (2014). *Hurtownie danych – Teoria i praktyka (Data Warehousing – Theory and Practice)*. PWN, Warszawa.
- Johnston T. (2014). *Bi-temporal Data. Theory and Practice*. Elsevier Inc., Waltham, USA.
- Inmon W.H. (2005). *Building the Data Warehouse, Fourth Edition*. Wiley Publishing, Inc., Indianapolis, Indiana, USA.
- Kimball R., Ross M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition*. Wiley Publishing, Inc., Indianapolis, Indiana, USA.
- Rozmus S.J. (2021). Databases with Full History of Changes. New Opportunities, Old Problems. Proceedings of the 37th International Business Information Management Association Conference (IBIMA), 30–31 May 2021, Cordoba, Spain. Innovation Management and Information Technology Impact on Global Economy in the Era of Pandemic. IBIMA Publishing, 4151–4164.
- Rozmus S.J. (2021). Persistent Stored Modules as Support for Temporal Joins. Proceedings of the 37th International Business Information Management Association Conference (IBIMA), 30–31 May 2021, Cordoba, Spain. Innovation Management and Information Technology Impact on Global Economy in the Era of Pandemic. IBIMA Publishing, 9973–9978.

- Rozmus S.J., Kiedrowicz M.K. (2019). Maintenance of chronology of facts in the Historical Geographic Information System (HGIS). 26th Geographic Information Systems Conference and Exhibition GIS ODYSSEY 2019: Conference proceedings, 231–240.
- Saracco C., Nicola M., Gandhi L. (2012). A matter of time: Temporal data management in DB2 for z/OS.  
[https://cs.ulb.ac.be/public/\\_media/teaching/infoh415/a\\_matter\\_of\\_time.pdf](https://cs.ulb.ac.be/public/_media/teaching/infoh415/a_matter_of_time.pdf)  
[access: 15.03.2025].
- Snodgrass R.T. (2010). A Case Study of Temporal Data. Teradata Corporation.  
[http://cs.ulb.ac.be/public/\\_media/teaching/infoh415/teradata\\_temporal\\_case\\_study.pdf](http://cs.ulb.ac.be/public/_media/teaching/infoh415/teradata_temporal_case_study.pdf) [access: 11.04.2025].
- Todman C. (2011). Projektowanie hurtowni danych. Wspomaganie zarządzania relacjami z klientami (*Data warehouse design. Customer relationship management support*). Helion, Gliwice.