AutoConViz: automating the conversion and visualization of spatio-temporal query results in GIS

Sugam SHARMAa*, Udyaya Sunday TIMb and Shashi GADIAa

aDepartment of Computer Science, Iowa State University, Ames, IA 50010, USA; bDepartment of Agricultural and Bio-systems Engineering, Iowa State University, Ames, IA 50010, USA

(Received 22 February 2012; final version received 11 March 2012)

In recent years, large multifaceted spatial, temporal, and spatio-temporal databases have attained significant popularity and importance in the database community. In order to perform preliminary investigation, exploratory visual analysis of such data-sets is highly desirable. To facilitate the convenient and efficient visualization, scientists and practitioners often need to convert the spatial component of the data-set into a more usable format. Though among the various formats available today in spatial data science community, Geographical Markup Language (GML) adheres to its central position and is of our interest in this work. The development of a tool to satisfy the spatial format conversion needs tailored to every user’s needs from scratch is difficult, time-consuming, and requires skills not easy to possess. We developed AutoConViz, to solve the issue stated above. It accepts the spatial component in GML format, converts that into shapefile format, and facilitates informative and automated interaction with the data-sets. It supports basic query and geospatial analysis and visualization tasks and offers functionalities such as zooming, panning, and feature selection. Furthermore, our software leverages navigation to classical ArcGIS software interface for users interested in more intensive analysis. AutoConViz serves both the database and geographical information system communities to explore insights of spatio-temporal databases and will help to further geospatial research and development.

Keywords: GML; shapefile; AutoConViz; automation; visualization; GIS

1. Introduction

According to statistics available in the literature, (1) in the database world, about 80% of all stored data in corporate databases are spatial data. In recent years, large spatial, temporal, and spatio-temporal databases have gained considerable attention and momentum. Exploratory visual analysis of such large multifaceted data-sets offers a helping hand in preliminary investigation (2), (3), where the spatial artifacts of such data-sets is of paramount importance. With the rapid growth in geographical information system (GIS) use, data collection by various agencies resulted in the mammoth heterogeneous data available today.

Huge amount of potentially important data is stored at various distributed locations and in heterogeneous formats. These diverse formats do not facilitate the easy acquisition and sharing of the data using off-the-shelf GIS software systems such as ESRI ArcGIS, GeoMedia, etc. available commercially or in research community. It is very unlikely that all the software use the same format (4) as each vendor has its own proprietary design, data model, and storage technology. To ease the exchange and sharing of the information among various heterogeneous spatio-temporal database systems, conversion tools – from one format to another – play vital role. The development of a new custom tool that can accommodate the spatial format of the artifact is difficult, time-consuming, and requires skills not easy to possess, leaving the many data-sets consequently underused.

Often times, the scientific community and practitioners must convert the spatial artifact of the data-set into a more suitable format easily accommodated into the domain of a proprietary or nonproprietary software tools for exploratory visualization. Among the various spatial formats available today in spatial community, because of its open source standard nature, the Geographical Markup Language (GML) format overshadows the use of others and adheres to its central position and is of our interest in this work. It is an Extensible Markup Language (XML) format which stores the geographical information that includes both the spatial and nonspatial properties of geographic features (5) and promises to support variety of sources. The format of GML data is text, a universal format thus offers easy integration across a variety of resources and has the potential for real-time data sharing.

Due to the explosive growth, the analysis of large volumes of data is getting difficult, necessitating the extraction of data subsets.

*Corresponding author. Email: sugamsha@iastate.edu
The smaller subset facilitates the desired analysis efficiently. In this work, we are interested in spatial/spatio-temporal data-sets which may exist in various formats. Let us consider an example of spatio-temporal data-set NC94 (6).

This is a climate–crop–soil data for the duration 1971–2000, consisting of the spatial and temporal component for North Central Region of the USA. For simplicity, we focus on climatic data-set only. The spatial and temporal artifacts are segregated into two separate groups, linked through their location references – Federal Information Processing Standard (FIPS) code (7). FIPS uniquely identifies a region in the USA and was developed by the US federal government for use in computing environments.

The spatial artifact is in GML format (8) consisting of spatial and nonspatial attributes, whereas the temporal artifact is one of the attributes of the climatic tuples consisting with climatic attributes as – FIPS, state FIPS, county FIPS, year, day, radiation, maximum temperature (MaxTemp), minimum temperature (MinTemp), and precipitation. An example of attributes and a tuple is given in Table 1.

Spatial/spatio-temporal queries are required to facilitate the driving of climatic results – possessing temporal aspect – from spatial/spatio-temporal data-sets. The result set produced by the query may be a subset of the original data-set. In this work, we are more interested in special kind of spatio-temporal queries – parametric queries also known as ParaSQL queries (9). ParaSQL queries are useful to database community and consist of both the aspects – spatial and temporal. An example of ParaSQL query is given in Figure 1. This ParaSQL query, applied on NC94 climatic data-set, results those days as output that possess the average temperature less than 0. In this work, we assume, the post execution spatial component is in GML format again and GIS community (8) is more interested to know about the intricate spatial and nonspatial attributes in much finer detail.

We attempt to make strides towards the development of a medium, where both the database and GIS communities, can harness the capabilities of the system helping to bridge the gap between database and GIS communities. While the database community relies on the execution of the spatial or nonspatial queries to produce results, the GIS community on the other hand wants to provide visualization support for a better or full understanding of the resulting data, provided that the data is visualizable. We appreciate the need for visualization to explore and understand the intricate details about the geospatial component in any spatial or spatio-temporal artifact, therefore this is the main focus of this work.

We develop a software tool, called AutoConViz to cater to the spatial artifact of spatial/spatio-temporal data by converting the original data-set into different formats to assist in automatic visualization of it. AutoConViz takes spatial/spatio-temporal data as input in GML format and generates a shapefile format of the corresponding data as output and renders it automatically to visualize what the shape of the spatial artifact of the data is. The AutoConViz window is equipped with basic functionalities such as zooming, panning, selection, etc. If a user is further interested to do more intensive analysis on the data (shp format), the software leverages a way to open the shapefile in ArcGIS (10) from the current window with a bare minimum effort – on a single click. AutoConViz is a robust and efficient software tool and was successfully validated using various valid sample spatial data-sets (in GML format). Due to the space limitation in this paper, unfortunately, we could accommodate the validation only on two inputs – "places.gml" and "counties.gml" – which are explained in the validation section of this paper in significant detail.

The novelty of our software comes from the fact that as soon as the shapefile, from its counterpart gml file, is generated, a user can visualize its shape on-the-fly without a single bit of human intervention. Thus AutoConViz provides this additional visualization functionality along with conversion. Upon visualization, a user can perform a few basic operations such as zoom in and out, selection partially or fully, selection inversion, etc. on the rendered shapefile.

As far as functionality is concerned, the AutoConViz tool adopts a rather conservative approach and offers enough basic functionality that can be useful to perform on the shapefile for the mere visualization at the prima facie. If the user is further interested in more intricate details of that spatial data-set (in shapefile format), AutoConViz, offers a bridge to navigate to fully functional classical ArGIS, which has a reservoir of unlimited functionalities readily available.

ArcGIS (with its various flavors) from ESRI (10) is considered as one of the leading products for spatial visualization. We attempt to implement the visualization module of AutoConViz, using ArcGIS’s ArcObject as

```
SELECT *
RESTRICTED TO [(C.MaxTemp + C.MinTemp)/2 < 0]
FROM Climate C
```

Figure 1. ParaSQL query.
the underlying technology. This helps in seamless integration of new functionalities and incorporation with ArcGIS desktop for future development.

Thus the distinguishing features of the AutoConViz spatio-temporal query and geovisualization experiments can be summarized as follows:

- **On-the-fly visualization of spatial shapefile format after conversion from corresponding input spatial GML format.** This helps fully understanding the geometry of the spatial data.
- **Conservative approach in functionalities exploration.**
- **Unified technology – ArcObject – in development for seamless integration with ArcGIS and easy future functionality extension.**

We do not challenge the functionality of various visualization tools available today in the literature, research or commercial arenas but we believe that AutoConViz can be an additional contribution to database and GIS communities.

The rest of the paper is structured as follows. Section 2 is the literature review. In this section, we survey the literature from both – database and GIS perspectives. Section 3 elaborates the functionality of the software tool. This section is decomposed into subsections and each subsection is a self-contained module. Prototype implementation of the software and its validation are explained in Section 4. We show in Section 5, how time efficient AutoConViz is in term of visualization as compared to classical ArcGIS. Section 6 concludes the paper.

2. Literature review

Effective and informative presentation of spatial or spatio-temporal is a challenge and many research studies focus on that. In this section, we review the previous work on visualization such as for query construction, visualization for decision-making, etc. Sharma et al. (11), (12) attempt to bridge the gap between database and GIS communities by exploiting the ParaSQL queries on spatio-temporal data-sets, NC94 with a spatial granularity at the county level, to calculate the coldness in Iowa and visualize that coldness using shapefile format of spatial data. Colder counties are rendered using a darker shade of blue. This work is extended by Sharma et al. (13), where the potential of XML as a configurable input is harnessed. An XML configuration file is employed as input in the visualization of coldness and the number of shades to be rendered is controlled from there. An interesting visualization approach has been described in GeoDec (14). It offers an immersive environment to visualize geospatial data and helpful in decision-making. It relies on the fusion of geospatial data – vector data, satellite imagery, and raster maps – on-the-fly. In spatial databases, visualization sometimes turns out to be very helpful for analysis and decision-making, and in (15) Sharma et al. attempted to visualize a spatial data-set – census data – for decision-making about supplemental nutrition assistance program eligibility in Iowa. This helps the US Department of Agriculture and local government distribute food stamps. GeoVR (16) is a system that utilizes a hybrid approach that combines database-aware display and display aware database scenario. Huang et al. (17) demonstrate a hybrid application of Java applet and common gateway interface that advocates a relatively balanced application for both modeling and visualization for data exploration. The extensive modulation is computed on the server side whereas visualization is executed on client side. MacEachren et al. (2) advocate the utility of visualization in spatiotemporal databases and mention that exploratory visual analysis can be witnessed as an effective means for first stage analysis. They explore geographical visualization (GVIs) of spatio-temporal data-sets to construct the knowledge discovery in databases (KDD) and design an integrated environment GVis-KDD. Gahegan (3) argues that in geographic research community the innovative tools and systems developed over the years are easy neither to integrate nor to use together and vague about what scientific activities they serve. Gahegan (3) further suggests that the entire scientific investigation process should be understood first then the roles of tools and systems as they play in that process. A process may include various tasks a researcher may venture and wish to be visualized. Kraak (18) admits that geovisualization enhances the visual thinking about complex geospatial patterns, trends, and relationships and advocates visual support for intelligent decision-making. Takatsuka and Gahegan (19) introduce an environment called GeoVISTA studio which offers a programming-free software development environment for geocomputation and geovisualization applications. GeoVISTA studio hides lots of engineering, metadata, and conceptual details from the active user. Its programming interface (visual) allows users to assemble their application using data-flow paradigm.

Based on the above spatial/spatio-temporal literature, good numbers of tools are available that facilitate conversion and/or visualization of data. AutoConViz also takes up similar challenges but its collective approach – conversion followed by automated conserved visualization – distinguishes it from others.

3. Functional architecture

This section describes the functional architecture of the software. We decompose the architecture into subsections for better understanding.

3.1. Large spatio-temporal database

In the database world, the analysis of a huge amount of data (Figure 2(a)) is not advisable from an efficiency point of view, rather, an interesting subset of the vast data is extracted to do interesting analysis. The smaller subset facilitates the desired analysis efficiently. There are numerous methodologies available to derive the
smaller subset from the vast chunk of the data in the database world.

The explanation of those methodologies is out of the scope of this paper. In this work, we are interested in the spatio-temporal data-set – NC94, (6) with a spatial artifact in GML format. The NC94 data-set is a 30 year data-set consisting of climate, crop, and soil data for North Central Region of USA. The well-organized spatial and temporal components of this NC94 data-set drew our attention for further exploration. The NC94 data-set has the spatial and temporal granularities by county and day, respectively.

3.2. Spatio-temporal query
Spatio/spatio-temporal query (Figure 2(b)) facilitates the driving of results from spatial/spatio-temporal data-sets.

The result set produced by the query is a subset of the original data-set. In this work, we are more interested in special kind of spatio-temporal queries – parametric queries – also known as ParaSQL queries (9). ParaSQL queries are useful for the database community and consist of both the aspects – spatial and temporal.

Figure 3 is one such example of ParaSQL query applied to NC94 climatic data-set. (6) The spatial and
temporal footprints of the NC94 data-set are county and day, respectively. The data-set has various attributes (Table 1), MaxTemp and MinTemp are two attributes used in the query. This ParaSQL query results show those days as output that possess an average temperature less than 0, this happens only for those counties which are the neighbor of “Story” county – a county in Iowa (20). As stated earlier, the query results in a spatial artifact with a GML format, but the GIS community is more interested in investigating the intricacy of spatial and nonspatial attributes in much finer detail.

3.3. Spatial artifact (GML format)
This block consists of two artifacts: (1) spatial (in GML format) and (2) nonspatial. The latter is the adjacent peer small block (on the right) of this spatial artifact which is a set of tuples comprising the temporal aspects of the NC94 data-set. The further use of these tuples is beyond the scope of this paper rather the spatial artifact (in GML format) is considered for the further operations.

In order to start functioning, AutoConViz relies on a GML file as input. Any valid spatial artifact as GML file is supplied as input to AutoConViz (Figure 2(c)) which produces the corresponding shapefile format as output. The readers who are interested to know more about GML can find voluminous information in the literature such as Wikipedia on GML (8).

3.4. GML-shp converter
This is one of the two main import components of AutoConViz (Figure 2(d)). This piece of functionality takes any valid spatial artifact as GML file (e.g. xxy.gml) as input and produces the corresponding shapefile format.

```
SELECT *
RESTRICTED TO [[((C.MaxTemp + C.MinTemp)/2<0)]]
FROM Climate C
WHERE NEIGHBOR (C.Name, "Story");
```

Figure 3. ParaSQL query with neighbor clause.
(xyz.shp) along with the necessary other files – dBase format (dbf) and shx (21) version of xyz. These are briefly explained in following Section 3(c).

3.5. Spatial artifact (shapefile format)

While producing the shapefile format (e.g. xyz.shp) as output, AutoConViz produces two more files also – xyz.dbf and xyz.shx (21) – along with a shapefile (Figure 2(e)). The shapefile (shp) consists of the geometrical part of that GML file. The dbf file contains attributes for each geometry in the shapefile. The geometry in dbf is recognized by a unique identifier – FIPS code. (7) The shape index file (shx) preserves the positional index of the features in the geometry of the GML file, consisting of index for geometry in the shape file.

3.6. Display automate

As shown in Figure 2(f), this is the additional functionality, AutoConViz offers along with conversion. It exploits the produced shp as input and renders it in ArcGIS window with basic functionality (Figure 2(g)). The objective is to visualize and analyze the generated shapefile format graphically to have better understanding about the spatial data-set. If the visualization of that shapefile format inspires user curiosity, AutoConViz offers navigation to the classical ArcGIS (Figure 2(h)) with one single button click, provided ArcGIS is preinstalled in the machine on which AutoConViz is running.

4. Prototype implementation

The implementation of the software is divided into phases and all of them are explained below in sufficient detail.

4.1. Software and hardware requirements

Software and hardware resource requirements are the building block of any software development and are considered as one of the most critical aspects. An intensive literature study is required to garner the knowledge about the suitable resources – both software and hardware. Following the same pattern, we did a wide literature review of the resource documentation provided by ESRI (10) and found the following software resources needed in this research work.

- ArcGIS Runtime Engine 9.3
- ArcGIS Map 9.3
- ArcGIS Developer’s Kit 9.3
- Eclipse Helios integrated development environment
- JDK 1.6

Hardware requirements comprise a computer system equipped with CPU, monitor, and mouse. Though, operating system is not the limitation, we use Windows XP for this development.

4.2. Software development

To facilitate the software development in an organized way, we practiced the Waterfall Model (Figure 4) to a large extent in AutoConViz construction and succeeded in accommodating most of the phases of this model with a team size of three. The words phase and stage are interchangeably used in the context of Waterfall Model. The emphasis is only on those stages which are relevant to this work.

4.2.1. Requirements

In this phase, we broadly explore the regarding persistent visualization needs in the database and GIS communities which must be fulfilled soon. Intensive research was conducted on the existing software handling similar kind of issues, and a literature review and progressive brainstorming sessions were held to unify the direction about inputs and outputs expected from our proposed software.

In a nutshell, for AutoConViz, the input must be a spatial/spatio-temporal data in GML format. The software system must generate a shapefile format from the corresponding data as output and render it in a window automatically to visualize what the shape of the data is. The window must be equipped with the basic functionalities such as zoom-in, zoom-out, etc. If a user wants to do further analysis on the data (shp format), the software must leverage a way to open the shapefile in an ArcGIS map from the current window with a bare minimum effort by the user such as a single click. The software must save a copy of the generated shapefile. For more details, readers are advised to read Section 3.

4.2.2. Design

A good design leads to a good software product. After the successful completion of stage 1 and passing through a rigorous thought process, a design layout plan was formulated both from theoretical and technical perspectives. At the abstract level, functionality is classified into modules to be engineered independently. Based on

![Waterfall model used in development.](image)
the functionality sought from AutoConViz, it turns out that there are two modules: GML-Shape Converter and Display Automate (Section 3.1), which were developed independently and integrated later onto offer full functionality. All the three team members participated in this phase.

4.2.3. Implementation
The implementation (team size one) of AutoConViz is done in Java programming language. Both the modules are implemented independently. Underneath the implementation to develop the module, GML-Shape Converter, the java archive (jar files) available from http://www.deegree.org/ (22) was used. The implementation of visualization module: Display Automate was done using the java archive (arcObject.jar) from esri.com. (10).

4.2.4. Validation
The validation of AutoConViz is described in Section 4.4 in greater detail.

4.2.5. Maintenance
This phase is important and needed to ensure that the developed software system continues to perform with the same efficiency as desired. Unfortunately, we did not accommodate this phase in our work.

4.3. Algorithm
In this section, we inscribe the pseudocode for the functionality of AutoConViz. We tried to formulate each step as self-explanatory.
4.4 Validation description

In software development, a set of test beds (of size one or more) is required in order to validate the software functionality intensively. Good quality software should be generic enough to attend various test cases of a variety of inputs of the same kind. AutoConViz is a robust and efficient software tool. Though AutoConViz has been validated successfully using various valid sample gml files, due to the space limitation of this paper, snapshots for only two inputs – “places.gml” and “counties.gml” – are discussed below.

Figure 5a and 5b are the “geometry” property defined here as a point and a polygon, respectively.

Figure 6a and 6b are the automatic visualization of the converted shapefiles, “places.shp” and “counties.shp,” respectively. As shown in the figures, the generated shapefile for the former is a set of points whereas that for the latter is set of polygons.

Figure 6a. Interface for automatic visualization of places.shp.

Figure 6b. Interface for automatic visualization of counties.shp.
In the AutoConViz Visualization window, a user can perform a few operations on the rendered shapefile such as panning, zooming, selection, full extent, and navigation. The pan button – represented by a hand – helps to move the displayed file anywhere inside the whole window. Zoom-in and out are self-descriptive operations. The selection button helps to select the displayed file partially or holistically. This command offers the opportunity: (1) to switch the selection in case of partially selection of the shapefile image, (2) to zoom-in the selected features, and (3) to clear the selection. Full extent command helps to bring the zoomed in or out image to the original size. We introduce a new command – ArcMap Viz – in the display window that navigates from this window to a traditional (classical) ArcGIS window.

This feature facilities interested users to do more analysis with the same shapefile (layer). When clicking this button (ArcMap Viz), the classical ArcGIS window opens up, preloaded with the same shapefile (Figure 7). Reader is encouraged to visit ESRI website (10) to learn more about classical ArcGIS.

5. Efficiency

In software engineering, various factors are calculated to find out the efficiency of a software system, time is one of them and is of main emphasis in this section. The efficiency of AutoConViz is evaluated based on the time taken to visualize the spatial component (shapefile format). We compare the visualization time of AutoConViz with ESRI ArcGIS for both the sample spatial components (shapefiles) – “counties.shp” and “places.shp” – used in this research work. Both of the shapefiles are of relatively different sizes: the size of counties shapefile is 7.29 MB and that of “places.shp” is 16.0 KB. For AutoConViz, we exclude the conversion time of spatial component from gml format to shapefile format.

The bar graph in Figure 8 depicts the comparison between AutoConViz and ArcGIS in term of time efficiency. A time logger is used to capture the time consumed in visualization for both the shapefiles exploited in this research work. Table 2 represents the concrete values of visualization time taken by AutoConViz and ArcGIS to render the shapefiles. Based on the statistics in Table 2, it can be said that for these two sample shapefiles, AutoConViz takes almost 43% less time compared to ArcGIS in order to visualize a shapefile. In both

![Figure 7. Visualization of counties.shp in ArcGIS.](image)

![Figure 8. Time efficiency of AutoConViz vs. ArcGIS.](image)

| S. no. | Name of file   | Size of file | Visualization time AutoConViz | ArcGIS |
|-------|----------------|--------------|-------------------------------|--------|
| 1     | Places.shp     | 16.0 KB      | 1.15 s                        | 2.05 s |
| 2     | counties.shp  | 7.29 MB      | 2.09 s                        | 3.67 s |

Table 2. Visualization time representation.
the cases, it can be observed that AutoConViz turns out to be more time efficient.

6. Conclusion
This paper describes the development and implementation of AutoConViz, a software tool to automate the component of spatio-temporal databases. AutoConViz expects a spatial input in GML format and converts into corresponding shapefile format and subsequently visualizes the converted shapefiles in AutoConViz visualization window – equipped with basic functionality – without any human intervention. If someone is further interested in more analysis of the produced shapefile, AutoConViz has equipped with a button to navigate to the classical ArcGIS.

An intensive validation of AutoConViz has been successfully conducted using various sample spatial datasets – in GML format – such as “places.gml” and “counties.gml” as test beds. We expect that AutoConViz can be an additional contribution to database and GIS communities and helpful to practitioners and researchers thus opening new avenues for further research.

Acknowledgments
Authors are thankful to the Geographic Information System, Support and Research Facility at Iowa State University, which provided useful resources and help to make this work possible.

Notes on contributor
Sugam Sharma works in interdisciplinary research domains with the following research interests: databases-spatial databases which include climatic data, GIS and XML, wireless sensor networks, and smart home technology. He has worked as an assistant professor, a researcher at Trent Lott Center for Geospatial Research, USA, and J2EE developer in NATVTEQ as an assistant professor, a researcher at Trent Lott Center for sensor networks, and smart home technology. He has worked with the following research interests: databases-spatial databases which include climatic data, GIS and XML, wireless sensor networks, and smart home technology. He has worked as an assistant professor, a researcher at Trent Lott Center for Geospatial Research, USA, and J2EE developer in NATVTEQ as an assistant professor, a researcher at Trent Lott Center for sensor networks, and smart home technology. He has worked

References
(1) Franklin, C. An Introduction to Geographic Information Systems: Linking Maps to Databases. Database 1992, 15 (2), 2–21.
(2) MacEachren, A.M.; Wachowicz, M.; Edsall, R.; Haug, D. Constructing Knowledge from Multivariate Spatiotemporal Data: Integrating Geographical Visualization and Knowledge Discovery in Database Methods. Int. J. Geogr. Inf. Sci. 1999, 13 (4), 311–334.
(3) Gahegan, M. Beyond Tools: Visual Support for the Entire Process of GIScience. In Exploring Geovisualization: Dykes, J., MacEachren, A.M., Kraak, M.-J., Eds., Elsevier Ltd: Amsterdam, 2005; pp. 83–99.
(4) Tarov, P.J. Evolution or Control. Traffic Technol. Int. 1998, 4 (2), 32–35.
(5) Open GIS. Geographical Markup Language (GML) v3.0. 2003. http://www.opengis.org/techno/documents/02–023r4.pdf
(6) North Central Regional Association of State Agricultural Experiment Station Directors Expected Outcomes. NC094: Impact of Climate and Soils on Crop Selection and, Management. 2004. http://www.nimss.umd.edu/homepages/outline.cfm?trackID=474
(7) United States Federal Government Federal Information Processing Standard. 1994. http://en.wikipedia.org/wiki/FIPS
(8) Open Geospatial Consortium Geography Markup Language. 2000. http://en.wikipedia.org/wiki/GML
(9) Noh, S. Hybrid Storage Design for NC-94 Database Within the Parametric Data Model Framework. In Proceedings of the International Conference on Computational Science and Its Applications (ICCSA), Part II; Viji, O. Ed., Lecture Notes in Computer Science; Glasgow, 2006; Vol. 3981/2006, pp. 145–154.
(10) Dangermond, J. Environmental Systems Research Institute. 1969. http://www.esri.com/
(11) Sharma, S.; Gadia, S. On Analyzing the Degree of Coldness in Iowa, a North Central Region, United States: An XML Exploitation in Spatial Database (NC94). First International Workshop on Database Management Systems: India, 2010.
(12) Sharma, S.; Gadia, S. On the Calculation of Coldness in Iowa, a North Central Region, United States: A Summary on XML Based Scheme in Spatial Database (NC94). International Conference on Advances in Information and Communication Technologies, India, 2010.
(13) Sharma, S.; Gadia, S. An XML-Based Range Variation Approach to Render the Coldness in Iowa, a North Central Region, United States. Second International Conference on Information Management and Engineering, China, 2010.
(14) Shahabi, C.; Banaei-Kashani, F.; Khoshgozaran, A.; Nocera, L.; Xing, S. GeoDec: A Framework to Visualize and Query Geospatial Data for Decision-Making. Multimedia, IEEE 2010, 17 (3), 14–23.
(15) Sharma, S.; Tim, U.S.; Smith, P. Geo-Spatial Patterns Determination for SNAP Eligibility in Iowa [C]. International Conference on Advances in Computing and Communication, India, 2011.
(16) Huang, B.; Lin, H. GeoVR: A Web-Based Tool for Virtual Reality Presentation From 2D GIS Data. Comp. Geosci. 1999, 25 (10), 1167–1175.
(17) Huang, B.; Jiang, B.; Lin, H. An Integration of GIS, Virtual Reality and the Internet for Spatial Data Exploration. Int. J. Geogr. Inf. Sci. 2001, 15 (5), 439–456.
(18) Kraak, M.J. Beyond Geovisualization. IEEE Comput. Graph. Appl. 2006, 26 (4), 6–9.
(19) Takatsuka, M.; Gahegan, M. GeoVISTA Studio: A Codeless Visual Programming Environment for Geoscientific Data Analysis and Visualization. Comput. Geosci. 2002, 28 (10), 1131–1144.
(20) Sharma, S.; Gadia, S.; Kumar, N.; Narayanan V.; Zhao X. Climate Analysis in IOWA Using XML and Spatio-Temporal Dataset-NC94. Int. J. Database Manage. Syst. 2010, 2 (3), 82–93.
(21) ESRI. Shapefile. 1990. http://en.wikipedia.org/wiki/Shapefile
(22) Poth, A.; Greve, K. Deegree. 2001. http://www.giub.uni-bonn.de/deegree/