A Geohydrologic Data Visualization Framework with an Extendable User Interface Design

Yanfu Zhou, Jieting Wu, Lina Yu, Hongfeng Yu, Zhenghong Tang
University of Nebraska-Lincoln

***Abstract*—We present a novel geohydrologic data visualization framework and apply the interface automata theory in support of time-varying multivariate data visualization tasks. The framework tackles heterogeneous geohydrologic data that has unique and complex data structures. The interface automata can generate a series of interactions and interfaces that are adapted to user selection and provide an intuitive method for visualizing and analyzing geohydrologic data. The interface automata can not only clearly guide user exploration, but also enhance user experience by eliminating automation surprises. In addition, our design can significantly reduce the entire system maintenance overhead, and enhance the system extendability for new datasets and data types. Our framework has been applied to a scientific geohydrologic visualization and analysis system, named INSIGHT, for the Nebraska Department of Natural Resources (NDNR). The new framework has brought many advantages that do not exist in the previous approaches, and is more efficient and extendable for visualizing geohydrologic data.**

***Keywords*-geohydrologic data; visualization; interface automata**

I. Introduction

Along with the technology boost, geoscience and its related fields have been generating a large amount of data at an unprecedented growth rate. Big geoscience datasets have been leveraged in various research areas, such as geography and urban planning, to benefit human society. Compared to the early stage of web 2.0, data storage and access paradigms have changed tremendously. The host of data has moved from personal computers to data centers that become more publicly accessible through the emerging cloud storage and computing services (for example, Amazon EC2, Google Earth Engine, and so on). Visualization becomes an increasingly attractive and feasible means for researchers to effectively and efficiently access and explore data to gain new understandings and discoveries.

However, it is challenging to develop viable visualization solutions for geoscience science. First, the size of a dataset can range from several Gigabytes to Petabytes [1]. Without suitable system and data supports, it is difficult to visualize and analyze these datasets [2], [3]. For example, a typical geoscience dataset has a hierarchical structure, which makes it hard for indexing and workload balancing [4]. Second, geoscience datasets often present diverse data types, which can be vector, raster, relation or binary. Examples include shapefiles, images, tables, and point clouds. Each data type can have its own unique data structures and require specific domain knowledge to process and interpret, incurring complex visualization and user interface operations. Users cannot effectively inspect and understand a large amount of data without appropriate visualization and user interface supports. However, this user experience problem has not been fully investigated in existing big geoscience data research.

We aim to develop viable visualization techniques for large-scale heterogeneous geoscience data (specifically, geohydrologic data), and focus on addressing the problems of visualization systems to support interactive and intuitive user interface and interaction design.

First, we develop a unified visualization framework to tackle the heterogeneity of geohydrologic data. Given that different research domains can have unique knowledge for data processing, we study their commonality from a data modeling perspective, and explore the feasibility to design a visualization framework adaptable to a wide spectrum of data types in different research domains.

Second, we investigate an automata design for user interface to address complex visualization operations. The automata theory provides a unique method to model the mathematical logics built in all the modern computer systems. The use of an automata-based language can possibly capture temporal aspects of interfaces, which could be hard to implement using other traditional techniques [5]. For a graphical user interface (GUI) design, the output of the interface automata is usually a graphic view of data (for example, a chart showing time-series), and interface automata can be a great tool for connecting data visualization and user interaction.

With our data visualization and interface automata approaches, a user can gain a better understanding of big geoscience data with a more intuitive method. In addition, a joint study of visualization system and interface automata can promote user feelings of ownership and enthusiasm about their data, and thereby enhance user experience.

II. Related Work

A. Geoscience Data Visualization Applications

Geography study covers a wide range of topics from climate change to urban planning. Geographic information system (GIS) has been used in these research areas intensively for their analysis and visualization tasks. GIS research domains are very wide, and no universal frameworks or common standards exist for visualizing geoscience data. However, nearly all the geoscience datasets either contain location information (coordinates or spatial regions) or have a spatial hierarchical structure.

In practice, there are a variety of geoscience data visualization examples in existing geography research domains. These studies have focused on visualization techniques (such as, visual cluttering [6] and visual association [7]), and analysis methods (such as, spatial-temporal interpolation [8] and similarity search [9]) for various topics (for example, disaster [9], urban planning [10], urban traffic [11], and hydrology [12]).

With the movement of data hosting services from personal computers to data centers, desktop GIS (such as, ArcMap, Quantum GIS, and uDig, etc.) is no longer the only option for geoscience data visualization and analysis. Web-GIS and Mobile-GIS have become another option for data scientists and researchers [13]. There is not a clear boundary between Web-GIS and Mobile-GIS: in general, Web-GIS refers to browser-based GIS applications, and Mobile-GIS refers to mobile-based GIS applications. They both target internet users, and share many common data visualization techniques, although they have a significant difference in screen size.

B. Interface Automata Applications

Using interface automata for generating data visualization and its interaction is not a novel topic. The interface automata have been applied to various application areas:

UI verification and validation. These applications are closely related with software testing studies. Examples include ADAutomation [14] for testing mobile apps, non-invasive UI automation [15] for simulation and characterization of mobile apps, and CosyVerif for verifying GUI on clusters [16].

UI layout generation and template creation. This type of application covers a wide rang of areas from web design to architecture design. In general, they share similar design ideas, either to create a feasible 2D layout in a room or create an interactive layout in a web page, in order to fill a certain kind of personalization process. Examples include the study of the algorithm for a room layout planning support system [17], and the variability models based on Linear Temporal Logic (LTL) for mobile app template personalization [18].

UI design. Interface automata has been widely used in different applications with various inputs. Apart from traditional computer input devices, such as mouse and keyboard, the inputs of interface automata could also be voices, videos, human gestures, 3D objects [19], mobile multi-touch screen movements [20], binary code [21], or even human brain and nervous system [22]. Different inputs may have different outputs that can be considered as a dependent upon user requirements. In today's mobile internet age, not only are devices having more input options, but also does a user have a higher expectation of interaction and various outputs. For instance, a smart phone may support not only numerous multi-touch movements, but also human body gestures and voice input. In addition, it can be connected with other devices with more input choices. For example, it can be used to control a game station, television, or can be used as a virtual reality (VR) embedded head screen. UI input/output is not limited to text and graphics any more, but also could be voices, videos, robot movements, and so on, thereby providing us unprecedentedly flexible UI choices.

Data visualization. There are a few studies that use automation design in data visualization techniques and show the advantages for visualizing time-series data and executing pattern searching task, such as using regular expressions and a nondeterministic finite automaton (NFA) to visualize the pattern of personal history [23], and using a deterministic finite automaton (DFA) to visualize the design of a discrete event system [24].

III. Driven Application

Our framework has been driven by a requirement to renovate the INSIGHT (Integrated Network of Scientific Information and GeoHydrologic Tools) system at the Nebraska Department of Natural Resources (NDNR). Scientists at NDNR collect and analyze geohydrologic data across Nebraska. The dataset mainly includes a spatial map data and a time-varying multivariate hydrologic data. The map contains two levels of geographical regions, including 12 basins and 42 subbasins. Hydrologic information is available at each level for most regions. The scientists desire that the INSIGHT system can consist of four levels of map visualization: the first level shows the country and the location of Nebraska in the U.S.; the second level shows the water summary information at all basins and Natural Resource Districts (NRDs); the third level shows the hydrologic details for each basin; and the fourth level shows the water details of each subbasin.

The original INSIGHT system consists of a set of static webpages corresponding to the combination of geographical regions and hydrologic data. The data is mainly presented as static images on a webpage that are generated in a preprocessing step. The transitions between different regions, data, and chart types are implemented via static links. Therefore, there is a lack of interactive visualization and analysis capability provided for users. In addition, for new regions, data, or chart types, new images and webpages are required to be manually processed and added into the system, incurring an unsustainable maintenance and development overhead.

A. Challenges

In general, a visualization process of geoscience data typically follows the paradigm of overview first, zoom and

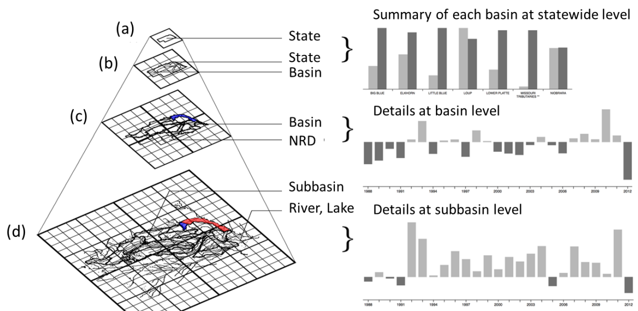


Figure 1. The hierarchical structure in our geoscience data visual analytic framework. (a) The country level only has the state boundary that shows the location of the state. (b) The state level has the basin boundaries and summarizes the essential information at the basin level. (c) The basin level shows the details of each basin and Natural Resource District (NDR) boundaries. (d) The subbasin level shows the details of each subbasin and other details such as rivers and lakes as references. The detailed data is available in the blue area, but not the red area of one particular basin. There is a data availability mismatching between the basin level and the subbasin level for that basin, and our visual analytic framework is designed to handle this situation.

filter, then details-on-demand [25]. Spatial information is often used as a constraint for querying the information in this process. To this end, a geoscience data is often orchestrated in a hierarchical structure in order to have multiple levels for user to navigate, as shown in the left image of Fig. 1. Apart from map information, each geographic region can be also associated with other geoscience data that can be time-varying and have multiple variables, as shown in the right image of Fig. 1.

However, a real-world dataset can contain multiple complex time-varying variables that may not have a hierarchical structure or may not have the same hierarchical levels as the map, as shown in the red area of Fig. 1. In this situation, it causes a disparity between the hierarchical structure of the map and the availability of geoscience data at each hierarchical level. This can incur a discontinuity in visualization when a user explores the map and data along the hierarchy, and thus disrupt the smoothness of user exploration. This user experience problem can become more severe with an increasing number and size of variables and a high possibility of data incompleteness.

IV. Geohydrologic Visual Analytics Framework

Our visual analytics framework is designed to bridge the disparity between different data. It will dynamically detect or predict the missing information caused by the gap of mismatching between the map hierarchical structure and the geoscience data and derive them according to domain knowledge. Although the design of this framework targets geohydrologic data visualization, it could also be applied to other applications (e.g., raster or LiDAR data exploration and visualization).

A. Data Management

Our solution to address data disparity is a two-fold means. First, in our design, a flag is added to the map to indicate whether a region has any hydrologic information or not. This design is flexible because the visual data representation of each region can be enabled or disabled based on the availability of the corresponding hydrologic information. Second, hydrologic data was originally represented as a relation table. It has a time-series column (Year), a unique identifier (ID), and other related information (for example, precipitation, water supply, water demand, and water balance) for each geographical region. In order to match the map hierarchical structure, our design has divided the hydrologic data into two tables according to the basin level and the subbasin level, respectively. In addition, the hydrologic data covers the information from the past 25 years, and in each year each table is further divided into 3 periods (i.e., annual, peak period, and non-peak period) according to application requirements.

Similarly, other types of geoscience data could also be integrated in this way using our framework design. For example, raster datasets (i.e., satellite imagery) and LiDAR point clouds are also time-varying and multivariate data commonly used in geoscience studies. However, due to the expensive and time consuming data collection procedure [2] and the limited budget of a government agency, these datasets are not always available for each geographical region. When a user is exploring on a map to find the availability of these geoscience data, our framework provides an intuitive way for showing the overall availability of these data in different levels. Besides, large-scale raster datasets and LiDAR point clouds are often stored using quad-tree structures. It could have a complexity of $O(n^2)$ in the worst case to build a quad-tree, and thus the algorithms for clipping, indexing, and summary of these datasets are generally expensive [4]. As a result, it can be costly to directly visualize and analyze these data. By building the indexing tables in a preprocessing step, a user can efficiently query and obtain their target areas using our framework design, which could be considered as a compensation for the high cost of the above algorithms at runtime.

B. Visual Exploration

The design of geoscience data visual exploration depends on domain application requirements. In our framework design, map visualization requires to support users to interactively select a geographical region for querying a series of charts that are categorized by hydrologic domain knowledge. During the navigation, the selected geographical region will be highlighted in order to give users an intuitive expression. When a user navigates out to a geographical region that has no data available, the visualization system will force the user back to the geographical region that has data available, which can prevent interface automation surprises [26].

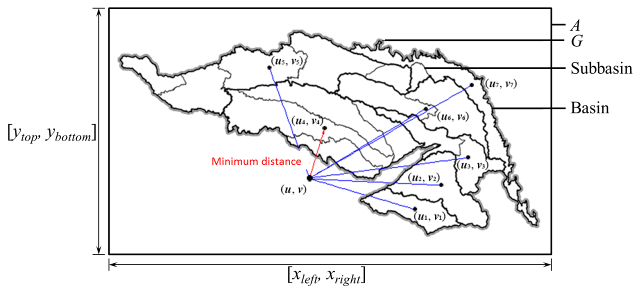


Figure 2. Visual query constraint model. Compared with the blue lines, the red line is the minimum distance between (u, v) and (u_4, v_4), denoting that (u_4, v_4) is the nearest neighbor.

Apart from the traditional visual exploration answering where, when and what questions [27], we also focus on the comparison analysis, such as the visual differences of data values between two geographical regions, the overall visual differences for each geographical region across the state, and the time-varying visual differences within a geographical region (e.g., comparison between precipitation in different seasons for a basin).

C. Visual Query Constraint Model

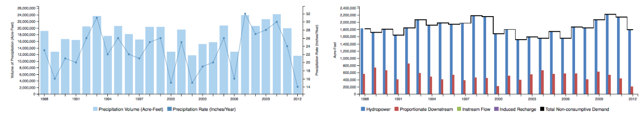
For geoscience data visualization, it is important for a map to have a visual query constraint to prevent a user from navigating out the study area. This is because if a query returns null (i.e., indicating no data available), it can result in automation surprises. Thus, we have designed three visual query constraints.

First, the map is organized using three hierarchical levels (i.e., the state level, the basin level, and the subbasin level) in a coarse-to-fine manner. Each level has a certain level of detail (LOD), and the detail is only visible at its corresponding level. This constraint gives users a helpful visual reference when they explore the map.

Second, we denote A as the bounding box of a geographical region G (i.e., the study area), and denote the horizontal and vertical ranges of A as $[x_{left}, x_{right}]$ and $[y_{top}, y_{bottom}]$, respectively. Then, a visual query is only allowed when $u \in [x_{left}, x_{right}]$ and $v \in [y_{top}, y_{bottom}]$, where (u, v) represents the center of A (See Fig. 2). This defines the range of the area for a user exploration.

Third, let $g_1, g_2, , g_n$ be a set of geographical sub-regions within G, and $g_i \cap g_j = \emptyset$ for any pair of g_i and $g_j (i \neq j, 1 \leq i, j \leq n)$. Let $T = \{(u_1, v_1), (u_2, v_2), ..., (u_n, v_n)\}$ be a set of centroids of these geographical sub-regions. If a $(u, v) \in g_i$, then let $f(x, y)$ be a function that returns the data availability for g_i. When $f(u, v) = \emptyset$, let $f(u, v) = f(u_i, v_j)$ and (u_i, v_j) is the nearest neighbor of (u, v) (See Fig. 2). Thus, this function can always returns data from the nearest g_i from a (u, v).

Because this model only shows the essential details at each level and always returns available data, it can create an intuitive visual link between the spatial map and the geoscience data, and effectively prevent automation surprises.



(a) (b)

Figure 3. Selected visualization methods in our framework: (a) combination chart and (b) step chart.

D. Visual Query Results

Visual query results vary in different domain fields. For geoscience data visualization, the visual query result can be charts, a set of images, tables, geographical regions data links, or other UI components. In our framework, specifically for visualizing hydrologic data, different types of charts are applied to visualize the data. Our framework also provides real-time interactions with each chart. Except traditional charts (for example, bar chart, pie chart and line chart, and so on), some advanced data visualization techniques are also used in our framework design, such as combination chart, and step chart, as shown in Figure 3.

Chart interaction is another important visualization feature in our framework design. Users could interactively transform data into different types of charts dynamically, such as changing a combination chart to a traditional chart. The extra visual information can be visually filtered out during chart transformation, which is helpful for users to select information for their different domain problems and gain discoveries. Embedding with different visualizations and interactions together, our visual analytic framework can show the data variation over time and provide a multiple perspective view of data.

E. Visual Expressiveness

The power of the visual expressiveness of geoscience data visualization is that it can lead users to gradually explore spatial-temporal data and gain an incrementally deeper understanding. This is very helpful to users who may not have sufficient domain knowledge about geoscience studies. In addition, domain experts could also quickly memorize essential data patterns through visual expressiveness and then apply domain knowledge on these patterns to testify their hypotheses underneath the data.

Data visual repetition is an important factor that impacts the visual expressiveness. For instance, a chart has more visual expressiveness than a table, and an interactive map has more visual expressiveness than the address list. To enhance the geoscience data visualization, we adopted the linked-view design [28] to create a linkage between geospatial information and geoscience data in our framework. When a user interactively selects a geographical region, the framework can dynamically generate the visualization results of data associated with the region for user exploration. This can give users not only an overall summary of the data of a geographical region, but also a strong visual expressiveness of each data property within the geographical region.

V. Automata Design for Geohydrologic Data Visualization and User Interaction

A. Background

Geoscience data visualization relies on user interaction extensively because of the explorative nature of geographical data. Traditionally, geographical information, such as on a paper map, conveys both direction and distance information. However, the limited size of paper makes it hard for a user to reach a region that is out of the map. In addition, if users explore a map that covers a large area, they need to continue change their visual extent on the map frequently. Along with the invention of map, such kinds of interactions already exist thousand years ago. Apart from conquered several limitations of the traditional paper map, modern technology also brought new types of interactions to the map navigation, such as zoom and rotate, and so on. Additionally, new hardware such as motion sensors and VR devices also add new types of interactions to the map navigation. Thus, map interactions have become increasingly complex, and the needs for having a universal methodology to design map interactions have become more concrete. The theory of automata provides a common mathematical model for designing such kinds of user interactions. We propose a unique design method for generating user interactions with diverse explorations for geoscience data visualization. Even if there would be a data discontinuity along a geographical hierarchy, our method could still produce a suitable user interface for geoscience data visualization and exploration.

B. Visual I/O

Visual input and output are the most two important components in geoscience data interaction. They determine the design of the corresponding interface automata. For geoscience data interactions, the visual input could be mouse hover, click, zoom, pan, drag, rotate, and so on. The visual output could be charts, download links, analysis results, and so on. Fig. 4(a) shows the visual I/O procedure in our visual analytic framework. First, for each visual input that can be a map interaction and/or a chart interaction, it is encoded to a sequence by an encoder. The sequence represents the region and/or the chart triggered by the interaction. The encoder can be broken down to a set of sub-encoder for different analysis tasks. Each sub-encoder represents a series of charts in a tree structure with three levels, as shown in Fig. 4(b). Then, there is an interpreter to interpret the sequence and retrieve the corresponding data from the database. After each data is retrieved from the database, a validator is used to check the data continuity. If there is no data discontinuity, the data will be loaded and presented to the user through an interactive

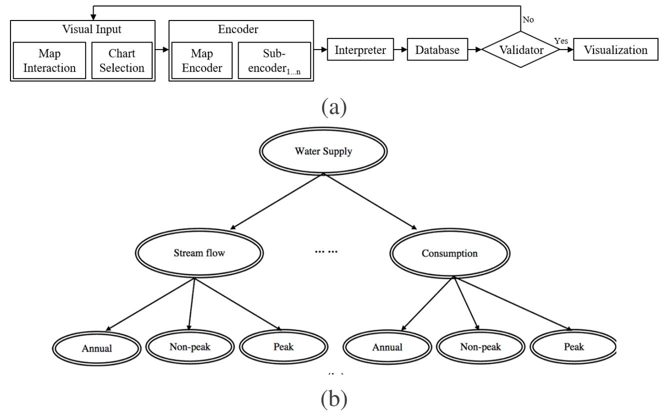


Figure 4. Visual I/O. (a) The procedure to process visual I/O in our framework. (b) An example of sub-encoder that represents a series of charts in a tree structure.

visualization; otherwise, the validator will select the nearest geographical area (based on the center of each geographical area) that has available data (See Section IV-C), and then start the retrieving process again from the encoder.

The encoded sequence is made of two parts, as shown in Fig. 5. The first part encodes the selected geographical region information. We use the level bits to indicate the hierarchical level and use the region bits to indicate the basin or subbasin region. The second part encodes the visualization chart selected by the interaction. As we organized the charts in a tree structure with three levels (See Fig. 4(b)), we use the group bits to indicate the selected second-level tree node (i.e., the chart group) and use the chart bits to indicate the selected leaf node (i.e., the specific chart). The two parts are concatenated together, which is the key step to link the map interaction and data visualization in our framework design.

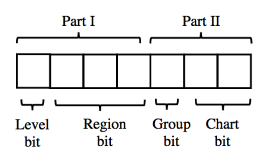


Figure 5. The encoded sequence structure.

Because there are a finite number of graphical regions and a finite number of chart types for each input stage, the length of the sequence can be fixed, which yields the complexity of $O(1)$ for encoding and decoding.

C. User Interface Components

As we need to dynamically visualize different charts associated with different geographical regions during user exploration, we design an interface automaton to control the dynamic display of user interface. The automaton has two components: a primary control component for capturing user map interactions, and a secondary control component for

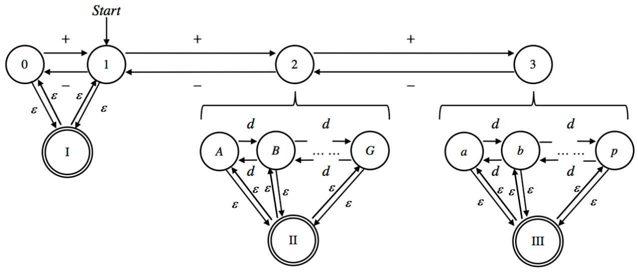


Figure 6. Primary control component.

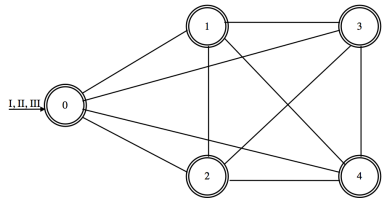


Figure 7. An automaton generated by the secondary control component.

generating a set of automata for each group of charts using the result from the primary control component.

Fig. 6 shows the primary control component, where 0, 1, 2, and 3 represent different map levels. In this study, 0 and 1 correspond to the state level, 2 corresponds to the basin level, and 3 corresponds to the subbasin level. +/- represents zoom in/out interaction, d represents drag interaction, and ε represents internal actions [5]. $A \sim G$ denotes different basins, and $a \sim p$ denotes different subbasins. I, II, III represent the states to load data for different chart groups at the state level, the basin level, and the subbasin level, respectively.

The transitions in the primary control component allow us to smoothly assess data across different geographical regions and different hierarchical levels. For the data of a specific region at a specific level (i.e., at the state I, II, or III), there may be multiple chart groups and types for visualizing them. A secondary control component is designed to dynamically generate automata according to the states output by the primary control component. The transitions in such an automaton allow us to smoothly display and compare different charts according to user interactions. Fig. 7 shows an automaton generated by the secondary control component. It contains a complete graph, where I, II, and III represent the data for different chart groups at different map levels, and 0-4 represent different chart groups.

As the secondary component is required to provide advanced visual analysis such as dynamic data summaries and comparisons, a complete graph is used to capture all the possible transitions between charts during user exploration (See Fig. 7). With this model, users can smoothly switch between different chart groups, different time-periods, and different variables of the geoscience data, which provides a unique method for users to examine the data from different aspects.

By leveraging interface automata theory and graph theory, the UI components in our framework design can guide users to explore hierarchical geographical data efficiently and effectively, and non-spatial time-varying and multivariate geoscience data can be dynamically linked with the spatial data at a lower computational cost. This makes a key contribution to geoscience data visualization in this research.

D. Visual Query Specification

The visual query model behind our geoscience data visualization framework is an interface automaton $M =< V_M, v', A_M^I, A_M^O, A_M^H, \delta_M >$ that consists of the following elements:

- V_M is a set of hierarchy levels.
- v' is the initial level, and $v' \in V_M$, and $v' \neq \emptyset$.
- A_M^I, A_M^O and A_M^H are mutually disjoint sets of visual input, visual output, and internal actions, respectively [5]. For example, A_M^I could be $\{+, -, d, r\}$ (d denotes a drag operation, and r denotes a rotate operation), A_M^O could be $\{G, L, D\}$ (G denotes a graph, L denotes a link, and D denotes the result), and A_M^H can be $\{\varepsilon\}$ (ε denotes internal actions). Let $A_M = A_M^I \cup A_M^O \cup A_M^H$ be the set of all actions.
- $\delta_M \subseteq V_M \times A_M \times V_M$ denotes a set of transitions.

The initial configuration of geoscience data visualization interface automaton (See Fig. 6) is $(v', \alpha, \varepsilon)$, whereas $\alpha \in A_M^I$, and $\varepsilon \in A_M^H$. The size of M is defined as $|M| = |V_M| + |\delta_M|$.

Besides a visual exploration on the map, there is an alternative mode to explore geoscience data via links. In this mode, A_M^I becomes a set of symbols that denotes each geographical regions, and each symbol is a unique link to a set of charts for corresponding geoscience data.

With the above interface automaton, a user can generate a sequence that represents a specific query, and then the returned data will be loaded in to a complete graph in order to enable visualization functionalities (See Fig. 7). The complete graph varies at different map levels, while the size of the graph is constant at each level. Because each node has linked all other nodes in the complete graph, a user is allowed to smoothly switch among the different chart types for the same set of the data.

We will present an implementation of our user interface design in Section VI-A. We note that in some circumstances, the transition between two chart types is prohibited. In this case, we can easily remove the link between the corresponding two nodes in our model to disable such a transition.

VI. Results

A. Implementation

We implemented our geoscience visual analytics framework for INSIGHT. The architecture of INSIGHT has two components, the front end and the back end, as shown in Fig. 8.

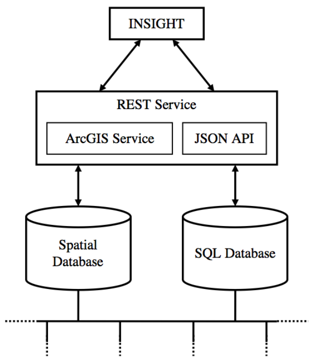


Figure 8. INSIGHT architecture.

The front end is a web application for data visualization, where D3.js (a JavaScript library for visualizing data with HTML, SVG, and CSS) [29] is used to render data chats dynamically.

The back end consists of a spatial database, a SQL relation database, and a REST (representational state transfer) service, which has implemented our data management solution. Because the daily hydrologic modeling and visualization workflow in NDNR relies on ArcGIS (a commercial product of ESRI), ArcGIS is chosen for storing and visualizing spatial data in our framework in order to make NDNR's hydrologic modeling data become more reusable. We note that ArcGIS is not the only option, and our framework could also work with other open sourced map database and visualization systems (such as, GeoServer and MapServer). The INSIGHT hydrologic data is stored using Microsoft SQL Server (a product of Microsoft) . The new INSIGHT system was running on a Windows Server with IIS web server enabled. The REST service contains a JSON API (Application Programming Interface) that works as a data viewer in the MVC (model-view-controller) framework, and only communicates with the SQL database. The ArcGIS server provides an embedded API to communicate with the spatial database. Both the SQL database and the spatial database are published using the REST service to communicate with the front end. A user can retrieve the geographical object identity (GEOID) from the spatial database through the front end, and then use the GEOID as a SQL constraint for later queries in the SQL database.

B. Hydrologic Data Visualization and Analysis

Compared to the previous solution, the newly developed INSIGHT system has achieved a goal of providing users with a diverse visualization and analysis tasks using our framework design. For instance, the previous solution used a static image to represent each hydrologically connected area. It cannot provide detail information either inside or outside the region, and resulted in a high maintenance overhead. It also lacked user interactions with hydrologic data. For example, users cannot interactively get detailed data from a chart.

Because of our new design, users can not only retrieve both the spatial data and the data table, but also get detailed values from the new INSIGHT system. The system also enabled some new analysis capabilities, such as displaying water non-consumption use, which could be hard to implement using the previous solution. In addition, the new INSIGHT system can allow users to interactively observe a group of time-varying and multivariate data. For example, users can select different members within a group of data to generate a new set of data visualization results that can contain many new features, such as showing how many portions a member could have impact to the group value.

The user interface of the new system can be dynamically adjusted by the interface automata according to different regions and levels selected by users. For example, as shown in Fig. 11 (a), the chart groups at the state level include Supply, Demand, and Nature and Extent of Use. Each group contains different charts. For example, the group of Supply contains two charts, Precipitation Rates and Volumes by Basin and Average Basin Water Supply. Both are visualized using bar charts. For each chart, a user also can select one for the three time periods, Annual, June-August, and September-May.

When zooming into a basin, the chart groups can be automatically adjusted. As shown in Fig.11 (b), the additional groups, Basin Overview, Big Picture, and Balance have been displayed for a particular basin LOUP. Within the group of Supply, the charts will change to Basin Water Supply, Streamflow, Surface Water Consumption, Groundwater Depletion, and Required Inflow. There are still three time periods available for user selection.

As shown in Fig.11 (c), when a user navigates to a subbasin, the chart groups and the chart within a group will change accordingly. Based on different charts and different visualization methods, certain options may be enabled or disabled on the user interface. For example, the data of Average Total Demand by Category for the three time periods are visualized as three side-by-side pie charts, and therefore no time period option appears on the user interface.

C. Evaluation Results

The dynamic control of interface and visualization enabled by the interface automata can significantly reduce the implementation cost and improve user experience and system performance. Our evaluation investigates the implementation cost and the effectiveness of visualization of

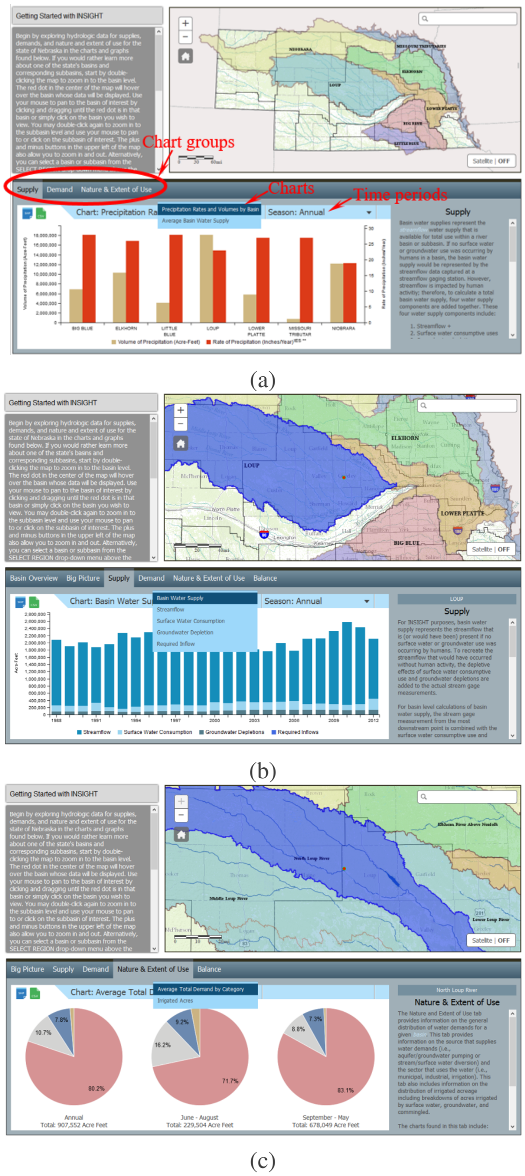


Figure 9. Different chart groups and charts for different regions.

our framework through the INSIGHT case study. We also evaluate how it can enhance the user experience and the performance compared to the previous INSIGHT system.

1) Implementation cost: One of the advantages of our new framework is that it can significantly reduce implementation cost during the system development and maintenance. In our approach, the interface automaton works as a controller that can dynamically load corresponding visualizations and interactions to the user interface. Because certain visualizations and interactions share same contents, interface automaton could reuse them in its corresponding states. This implies that when a developer tries to update the user interface, he/she only needs to update the shared contents once, while the previous INSIGHT solution needed to maintain several duplications of the static contents for each user interface. With the application of the interface automata theory, the parts of the system need to be replaced or maintained are reduced significantly. In fact, a developer needs to maintain 54 user interfaces in the previous solution, and the number can be increased considerably with new regions, data, and chart types. In our new INSIGHT solution, a developer mainly needs to maintain one user interface because the major components are generated dynamically by the interface automaton.

Apart from the implementation cost, the interface automaton also makes the new INSIGHT solution become more scalable. It is relatively easy to add new regions, data, and chart types into the system. For example, when adding a new basin or a new chart, we can simply add a new state to the interface automaton, and the rest of system can mostly remain the same. After receiving a new state, the interface automaton will dynamically create the transactions between the new state and the other corresponding states to support essential data visualizations and user interactions.

2) Geoscience data visualization: We first compared our framework with the previous solution to evaluate the effectiveness of the new data visualization. We examined how much interaction improvements were made and whether it reduced some unnecessary interactions and complexities. The previous INSIGHT solution has several disadvantages for user navigation and exploration data through the system. First, users need to navigate across multiple webpages to find the needed data. This enforces users to click back and forth frequently if they want to compare data in different geographical regions. In this situation, unnecessary interactions are imposed. Second, the temporal data is visualized at an abstract level on each interface. It lacks a mechanism to break the visualization down to smaller details. The previous system attempted to address this issue by adding a separate visualization and using additional navigation to show the details of the data, which however is troublesome and less efficient for users to search the information. In the new INSIGHT system, users do not need to click back and forth to compare the data, as all the unnecessary interactions and complexities have been significantly reduced using the hierarchical structure in map visualization. The separate visualization for the detailed geoscience data is triggered by an interaction with the UI, where users can search a specific time value through the interaction. Compared to the existing solution, the new INSIGHT system is more efficient for users to visualize and analyze the data.

In order to keep the consistency of UI design, the new INSIGHT system has a similar UI layout as the previous one, but there are marginal differences. The new INSIGHT system provides a freedom for users to zoom to any level of the map (i.e., jump between arbitrary two hierarchical levels). This is impossible in the existing solution where the user could only zoom from the state level to the basin level.

In the new solution, user could jump from the state level directly to the subbasin level. This is helpful for a user to search information quickly through different levels, which can improve the efficiency in geoscience data visualization.

3) System performance: Due to the increasing size of geoscience data, the data transferring between the back end and the front end has simply become the major performance bottleneck. For visualizing each chart, the previous system needs to send data requests to the back end for data retrieving. Thus, if a geographical region is associated with n charts, there are totally n data requests for drawing these charts, thus causing sluggishness when a user navigates through the charts. The running time is proportional to the number of charts. Because of our automata based design, the new INSIGHT system can reuse the shared data among the charts of a geographical region, and dynamically generate the charts at the front end. Therefore, only one data request is needed for visualizing the charts of one region, which can make the running time not depend on the number of charts, and thus significantly improve the performance.

By integrating our geoscience data visualization framework, the new INSIGHT system has significantly enhanced user experience by dynamically generating UI interactions, and demonstrated the performance improvement on data analysis and visualization compared to the previous solution. The new INSIGHT system has also proved and evaluated the extendibility and the effectiveness of our framework for visualizing time-varying multivariate geoscience data.

VII. Conclusion

In this paper, we present a new data visualization and analytic framework for geohydrologic data. The novelty of our study is providing an example of how to apply the interface automata theory to geoscience data visualization. Although our study tackles geohydrologic data, it has three major contributions to general geoscience data visualization applications.

First, geoscience data has its unique hierarchy data structure and complex formats, and therefore it is relatively easy for users to get lost or confused during their exploration of the data. By applying interface automata model to the UI design, users can be clearly guided to find the exact visualization and analysis that they want. In addition, from a development perspective, interface automaton is also easier to understand than conditional statements, which can simplify the development process.

Second, it is common that geoscience data has discontinuity in its hierarchy structure. The application of interface automata can prevent users from suffering automation surprises, and enhance user experience.

Third, for supporting a variety of different data visualization and analysis, our design with interface automata could also make applications become extendable in that a new visualization function or a new data group could be easily added to an existing application, which reduces the overhead of maintenance significantly.

The new framework has been applied to the implementation of a real-world application – the INSIGHT visualization system at NDNR. Our framework has significantly reduced the implementation and maintenance cost of the system, improved the user experience and the runtime performance, and thereby provided an extendible and scalable solution to visualize time-varying multivariate geoscience data. The new INSIGHT system has been operated on a daily basis by NDNR to provide an effective exploration of hydrologic data at Nebraska.

Our study of using interface automata for geoscience data visualization not only tackles complex geoscience data structures, but also provides an extensible solution in support of various interactions for users to better understand the data. In the future, we plan to continue to improve this framework and consider more data visualization techniques, such as raster data visualization and geometry data visualization. We will study integration strategies and enable a holistic user experience across different visualization techniques.

Acknowledgment

This research has been sponsored by the National Science Foundation's IIS Core program and EarthCube program, the Nebraska Department of Natural Resources, and Hyde Professorship from University of Nebraska-Lincoln. The contents do not necessarily reflect the views and policies of the funding agencies, and do not mention of trade names or commercial products constitute endorsement or recommendation for use.

References

[1] M. P. Peterson, *Online maps with APIs and WebServices*. Springer Science & Business Media, 2012.

[2] M. Sarwat, "Interactive and scalable exploration of big spatial data–a data management perspective," in *2015 16th IEEE International Conference on Mobile Data Management*, vol. 1. IEEE, 2015, pp. 263–270.

[3] A. Eldawy, M. F. Mokbel, S. Alharthi, A. Alzaidy, K. Tarek, and S. Ghani, "Shahed: A mapreduce-based system for querying and visualizing spatio-temporal satellite data," in *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 2015, pp. 1585–1596.

[4] J. Zhang, S. You, and L. Gruenwald, "Data parallel quadtree indexing and spatial query processing of complex polygon data on GPUs." in *ADMS@ VLDB*, 2014, pp. 13–24.

[5] L. De Alfaro and T. A. Henzinger, "Interface automata," in *ACM SIGSOFT Software Engineering Notes*, vol. 26, no. 5. ACM, 2001, pp. 109–120.

[6] A. Shrestha, Y. Zhu, and Y. Zhu, "Visual cluttering reduction for visualizing large spatio-temporal data sets," in *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 137–143.

[7] S. Wada, Y. Yaguchi, R. Ogata, Y. Wadanobe, K. Naruse, and R. Oka, "Associated keyword analysis for temporal data with spatial visualization," in *Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), 2013 International Joint Conference on*. IEEE, 2013, pp. 243–249.

[8] D. Kannangara, N. Fernando, and D. Dias, "A web based methodology for visualizing time-varying spatial information," in *2009 International Conference on Industrial and Information Systems (ICIIS)*. IEEE, 2009, pp. 233–238.

[9] H. Hayashi, A. Asahara, N. Sugaya, Y. Ogawa, and H. Tomita, "Spatio-temporal similarity search method for disaster estimation," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2462–2469.

[10] P. Shanbhag, P. Rheingans *et al.*, "Temporal visualization of planning polygons for efficient partitioning of geo-spatial data," in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* IEEE, 2005, pp. 211–218.

[11] I. Felde, L. Nádai, M. Mezei, and G. Bognár, "Characterization of urban traffic by using mobile phone traces," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2014, pp. 3360–3363.

[12] N. Ravindran and Y. Liang, "HIDE – A web-based hydrological integrated data environment," in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, vol. 3. IEEE, 2007, pp. 143–148.

[13] A. Oliveira, M. Urbieta, J. Araujo, A. Rodrigues, A. Moreira, S. Gordillo, and G. Rossi, "Improving the quality of webgis modularity using aspects," in *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the*. IEEE, 2010, pp. 132–141.

[14] A. Li, Z. Qin, M. Chen, and J. Liu, "ADAutomation: An activity diagram based automated GUI testing framework for smartphone applications," in *Software Security and Reliability (SERE), 2014 Eighth International Conference on*. IEEE, 2014, pp. 68–77.

[15] D. Sunwoo, W. Wang, M. Ghosh, C. Sudanthi, G. Blake, C. D. Emmons, and N. C. Paver, "A structured approach to the simulation, analysis and characterization of smartphone applications," in *Workload Characterization (IISWC), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 113–122.

[16] É. André, Y. Lembachar, L. Petrucci, F. Hulin-Hubard, A. Linard, L. Hillah, and F. Kordon, "CosyVerif: An open source extensible verification environment," in *Engineering of Complex Computer Systems (ICECCS), 2013 18th International Conference on*. IEEE, 2013, pp. 33–36.

[17] M. Inoue and H. Takagi, "Layout algorithm for an EC-based room layout planning support system," in *Soft Computing in Industrial Applications, 2008. SMCia'08. IEEE Conference on*. IEEE, 2008, pp. 165–170.

[18] R. Alcarria, T. Robles, A. Morales, S. Gonzalez-Miranda, and D. Caballero, "A variability model for template personalization in mobile prosumer environments," in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*. IEEE, 2013, pp. 1368–1373.

[19] Y. Masunaga and C. Watanabe, "Design and implementation of a multi-modal user interface of the virtual world database system (VWDB)," in *Database Systems for Advanced Applications, 2001. Proceedings. Seventh International Conference on*. IEEE, 2001, pp. 294–301.

[20] A.-C. Olteanu, G.-D. Oprina, N. Tapus, and S. Zeisberg, "Enabling mobile devices for home automation using zigbee," in *2013 19th International Conference on Control Systems and Computer Science*. IEEE, 2013, pp. 189–195.

[21] E. R. Rodrigues, A. J. Preto, and S. Stephany, "A parallel engine for graphical interactive molecular dynamics simulations," in *Computer Architecture and High Performance Computing, 2004. SBAC-PAD 2004. 16th Symposium on*. IEEE, 2004, pp. 150–157.

[22] C. I. Penaloza, Y. Mae, F. F. Cuellar, M. Kojima, and T. Arai, "Brain machine interface system automation considering user preferences and error perception feedback," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1275–1281, 2014.

[23] T. D. Wang, A. Deshpande, and B. Shneiderman, "A temporal pattern search algorithm for personal history event visualization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 799–812, 2012.

[24] L. Ricker, S. Lafortune, and S. Genc, "Desuma: A tool integrating giddes and umdes," in *Software tools, 8th international workshop on discrete-event systems*, 2006.

[25] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE, 1996, pp. 336–343.

[26] D. Ishii and T. Ushio, "A bisimulation-based design of user interface with alerts avoiding automation surprises," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 2, pp. 317–323, 2016.

[27] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva, "Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2149–2158, 2013.

[28] J. Heer and B. Shneiderman, "Interactive dynamics for visual analysis," *Queue*, vol. 10, no. 2, p. 30, 2012.

[29] D3.js: Data-Driven Documents. [Online]. Available: https://d3js.org/