Summer 8-2016

# AN EXTENDABLE VISUALIZATION AND USER INTERFACE DESIGN FOR TIME-VARYING MULTIVARIATE GEOSCIENCE DATA

Yanfu Zhou
*University of Nebraska - Lincoln*, zyfbta@hotmail.com

Zhou, Yanfu, "AN EXTENDABLE VISUALIZATION AND USER INTERFACE DESIGN FOR TIME-VARYING MULTIVARIATE GEOSCIENCE DATA" (2016). *Computer Science and Engineering: Theses, Dissertations, and Student Research*. 102.
http://digitalcommons.unl.edu/computerscidiss/102

# AN EXTENDABLE VISUALIZATION AND USER INTERFACE DESIGN FOR TIME-VARYING MULTIVARIATE GEOSCIENCE DATA

by

Yanfu Zhou

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Hongfeng Yu

Lincoln, Nebraska

August, 2016

# AN EXTENDABLE VISUALIZATION AND USER INTERFACE DESIGN FOR TIME-VARYING MULTIVARIATE GEOSCIENCE DATA

Yanfu Zhou, M.S.

University of Nebraska, 2016

Adviser: Hongfeng Yu

Geoscience data has unique and complex data structures, and its visualization has been challenging due to a lack of effective data models and visual representations to tackle the heterogeneity of geoscience data. In today’s big data era, the needs of visualizing geoscience data become urgent, especially driven by its potential value to human societies, such as environmental disaster prediction, urban growth simulation, and so on. In this thesis, I created a novel geoscience data visualization framework and applied interface automata theory to geoscience data visualization tasks. The framework can support heterogeneous geoscience data and facilitate data operations. The interface automata can generate a series of interactions that can efficiently impress users, which also provides an intuitive method for visualizing and analysis geoscience data. Except clearly guided users to the specific visualization, interface automata can also enhance user experience by eliminating automation surprising, and the maintenance overhead is also reduced. The new framework was applied to INSIGHT, a scientific hydrology visualization and analysis system that was developed by the Nebraska Department of Natural Resources (NDNR). Compared to the existing INSIGHT solution, the new framework has brought many advantages that do not exist in the existing solution, which proved that the framework is efficient and extendable for visualizing geoscience data.

# COPYRIGHT NOTICE

# ACKNOWLEDGEMENTS

Professor Hongfeng Yu, my advisor, is on top of my appreciation and acknowledgement list. Through his endless support, guidance, feedback and critique throughout my graduate work, I have achieved many of my goals and also developed a broad set of skills ranging from algorithm, to architecture and hands on experience of implementations, as well as effectively updating apps monthly and server in his lab.

I would appreciate the constructive comments and valuable guidance from my committee members: Professor Zhenghong Tang and Professor Lisong Xu.

I dedicate my thesis work to my family and many friends. A special feeling of gratitude to my loving parents, Hong Wang and Yongkai Zhou whose words of encouragement and push for tenacity ring in my ears.

I also dedicate this thesis to my many friends who work at the Nebraska Department of Natural Resources. I really appreciate all they have done with me, especially Jeff Hogan, Steve Rathje, and Jessie Winter.

At the last, I would like to give special thanks to my best friend Clark Sintek who helped me to conquer my weaknesses, and did many hours proofreading for me with great patience.

TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# Chapter 1

# Introduction

Along with the technology boost of post-web 2.0, environmental science and its related fields have been generating a large amount of data at an unprecedented growth rate, which 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 desktops to data centers that become more publicly accessible, and provide as 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, there are several critical challenges to realize viable visualization solutions for environmental science.

First, the size of a dataset can range from several Gigabytes to Petabytes [1], which becomes a major challenge. For example, in November 2013, NASA released hundreds of Terabytes of remote sensing data. Without suitable system and data support, it is difficult to visualize and analysis these datasets [2, 3]. For example, a typical geography dataset has a hierarchical structure, which makes it hard for indexing and workload balancing [4, 5]. Advanced algorithms and computer architectures are required to support scalable data storage and retrieval.

Second, complex algorithms, such as polygon clipping (for example, Greiner-Hormann [6], Vatti's [7] and Karinthi's [8]), space filling curves (for example, Z-curves [3]), and clustering (for example, K-means [9]), are commonly required in environmental studies. However, it is a nontrivial task to carry out these algorithms using advanced architecture techniques, such as graphics processing unit (GPU), solid state drive (SSD), and cloud computing (for example, Hadoop and Apache Spark [2, 3, 4, 10]).

Third, environment datasets (for example, geography datasets) present diverse data types, which can be vector, raster, relation or binary. Examples include Esri's Shapefiles, images, tables, and LiDAR 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.

In this study, I aim to develop viable visualization techniques for large-scale and heterogeneous environment data (specifically, geography data) to provide scientists with an interactive visual analysis capability. In particular, I focus on addressing the problems of visualization systems to support user interface and interaction design.

First, I develop a unified visualization framework to tackle the heterogeneity of geography data. Although different research domains can have unique knowledge for data processing, I study their commonality from the data modeling perspective, and explore the feasibility to designing a visualization framework that is adaptable to different data types and can be possibly portable to different research domains.

Second, I investigate an automata design for user interface to tackle the complexity of visualization operations. In this work, I focus on graphical user interface (GUI) design, because it is closely related to the creation of user interaction, and is an essential component of data visualization in order to providing users with an interactive visual analysis capability in corresponding research domains. 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 realize using other traditional techniques [12]. Automata-based languages are also used in pervasive system designs [13] and user interaction designs [14]. A user interaction design for specific applications highly depends on domain knowledge, because it needs to guide a user to the correct output of the interface automata. For a graphical user interaction 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 data visualization and interface automata techniques, a user can be equipped with a more intuitive method to gain a better understanding of big data. 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.

In this thesis, I first review several existing visualization system designs and automata applications for user interface designs. Then I introduce a new visualization framework for presenting geoscience data. I have applied the framework to the hydrology data from the Nebraska Department of Natural Resources (NDNR), and developed a new hydrology data visualization system. Finally, I evaluate and demonstrate the effectiveness of the new system. The result shows that our approach can deliver an intuitive interactive visual analysis capability with a more featured user interface, which is a remarkable improvement and enhancement compared to the existing solutions.

# Chapter 2

# Related Work

## 2.1 Geoscience data visualization applications

Geography study covers a wide range of topics from climate change to urban planning, and geographic information system (GIS) has been used in these research areas for many years for their analysis and visualization tasks. Under the pressure driven by emerging big data applications, although GIS is not a novel idea, there is still a strong need for visualizing and analyzing geoscience data (geospatial big data) in both industry and academic fields. Even in other research domains (such as, biology), GIS is used for some special research purposes (for example, used as genomic islands detection tool [15]). As GIS research domains are so wide, no universal frameworks or common standards exist for visualizing geoscience data. However, all the geoscience data (specifically, geography data), either contain location information (coordinates or spatial regions) or have a spatial hierarchical structure. Thus, it would be possible to develop a common framework for geoscience data visualization independent of specific GIS related research domains.

In practice, there are a variety of geoscience data visualization examples from existing geography research domains, and the study of these applications mostly focus on visualization techniques (such as, visual cluttering [16] and visual association [17]) and analysis methods (such as spatial-temporal interpolation [18] and similarity search [19]) for specific spatial-temporal topics (for example, disaster [19], urban planning [20], urban traffic [21], hydrology [22], and tweets [23]). While in my study, I focus on creating a common visual framework for geoscience data visualization in different domains using Web-GIS approaches.

With the movement of data hosting services (from personal desktops 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 [24, 25]. In general, there is not a clear boundary between Web-GIS and Mobile-GIS: Web-GIS refers to browser based GIS applications, and Mobile-GIS refers to mobile app based GIS applications. They both target internet users, and share many common data visualization techniques, although they have only an obvious difference in screen size. In this study, I focus on the development of a new Web-GIS visual analytic framework that could also be extended to Mobile-GIS applications that can be considered as an enrichment of existing geoscience data visualization frameworks.

## 2.2 Interface automata applications

Using interface automata for generating data visualization and its interaction is not a novel topic. We investigate the examples in the current studies and examine the input and output for those automata to develop our framework. The interface automata have been applied the following main application areas:

A. *Automata applied to UI verification and validation*. Those applications are closely related with software testing studies. Examples include ADAutomation (An activity diagram based automated GUI testing framework) [26] for testing mobile apps, Non-invasive UI automation [27] for simulation and characterization of mobile apps, CosyVerif - an open source tool for verify GUI on clusters [28], Paul's framework for identifying key graphical elements in the screen and automated interaction [29], White's algorithm of building a finite state machine (FSM) to specify the GUI testing [30, 31], and Actuator automation in Waseem's UPPAAL verifier for model-checking [32].

B. *Automata applied to UI layout generation and template creation*. This type of application covers a wide study area from web design to even building interior 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 [33], and the variability models based on Linear Temporal Logic (LTL) for mobile app template personalization [34].

C. *Automata applied to rule generations of a game*. Automata have also been applied to generate game rules. For example, a modular cellular automaton based on a variation of Conway's Game of Life has been used to automatically generating chess variants in a computer game [35].

D. *Automata applied to UI design*. This is the most widely used area of interface automata with various inputs in different applications. 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 [36, 37, 38], mobile multi-touch screen movements [39, 40, 41], binary code [42], or even human brain and nervous system [43], and so on. Different inputs may have different outputs that can be considered as a dependent upon user requirements. In today's market driven mobile internet age, not only are devices having more input options, but also does a user has 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 voices 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 only 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. In this study, we aim to leverage automata theory to support complex input/output choices for visualization tasks.

E. *Automata applied to data visualization tasks*. 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 [44], and using a deterministic finite automaton (DFA) to visualize the design of a discrete event system [45].

F. *Automata applied to communication simulation and design*. It has been a long history that automata have been used in network communication testing and simulation, and state diagrams have been used to simulate the transitions between different nodes, and design the protocol of a network [46, 47, 48].

G. *Automata applied to architecture design*. Automata could also be used to simulate the logics between different processes/tasks in a system, which is similar to network communication simulations. In architecture design, an automaton often works as a controller to solve control issues such as in a human-robot interaction system [49]. Existing applications of interface automata and their inputs and outputs are summarized in Table I.

TABLE I EXISTING APPLICATIONS OF INTERFACE AUTOMATA

| Applied Area | Type of Input | Type of Output |
|---|---|---|
| UI verification & validation | UI component, GUI, API, machine interface, models | Boolean values |
| UI layout generation & template creation | Expectations, state diagram, graphics, models | A completely UI, GUI, API |
| Game rule generation | Possibilities, game results | A completely rule for the game |
| UI design | Multimedia, gestures, movements, objects, sequences, human bodies | Graphics, text, formula, rules, animation, interaction, queries |
| Data visualization | Datasets, queries, sequences, graphics, models | Data interactions, Boolean values |
| Network communication simulation & design | Network diagram, protocols, nodes, models | Network simulation, network design, Boolean values |

# Chapter 3

# Geoscience Visual Analytics Framework

## 3.1 Background

A visualization process of geoscience data typically follows the paradigm of overview first, zoom and filter, then details-on-demand [52]. Spatial information is often used as a constraint for querying the information in this process. To this end, a geoscience data is typically arranged in a hierarchical structure in order to have multiple levels for user to navigate, as shown in the left image of Figure 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. However, these time-varying and multivariate datasets may not have a hierarchical structure or may not have the same hierarchical levels as the map. 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 cause a discontinuity in visualization when a user explores the map and the data along the hierarchy, and thus disrupt the smoothness of user exploration. My visual analytics framework is designed to bridge the gap. It will dynamically detect or predict the missing information caused by the gap of mismatching between the map hierarchical structure and the geoscience data (it is hydrology data in this study) and derive them according to domain knowledge, as shown in Fig. 1. In addition, the design of this framework could also be applied to other scenarios (for example, exploring and visualizing a raster dataset).

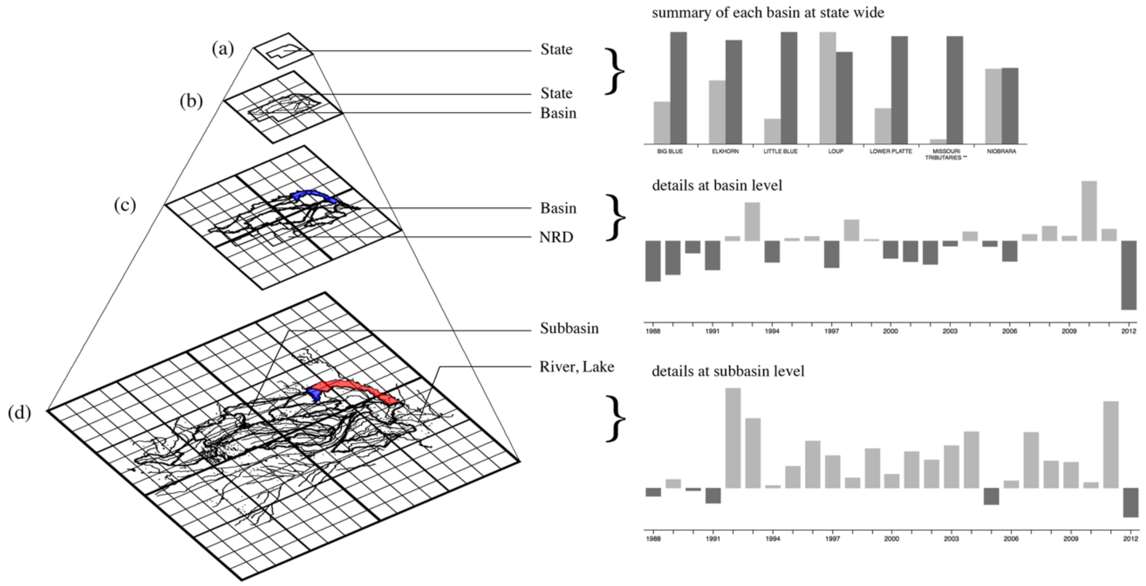


Fig. 1 The hierarchical structure in *our* geoscience data visual analytic framework. (a) The country level only has the state boundary that shows the location of the state. (b) The state level has the basin boundaries and summarizes the essential information at the basin level. (c) The basin level shows the details of each basin and NRD (Natural Resources District) 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 my visual analytic framework is designed to handle this situation.

### 3.1.1 Data management requirements

The data used in my study is provided by NDNR, which mainly includes a spatial dataset (provided as a map) and a time-varying multivariate dataset (provided as a relation table). The map contains two levels of geographical regions, including12 basins and 42 subbasins. However, not every region has its corresponding hydrology information available at each level.  My solution to address this issue is two-fold. First, in my design, a flag is added to the map to indicate whether a region has any hydrology information or not. This design is flexible because the data visual representation of each region can be enabled or disabled based on the availability of the corresponding hydrology information. Second, hydrology 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, my design has divided the hydrology data into two tables according to the basin level and the subbasin level, respectively. In addition, the hydrology data covers the information from 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 my 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, my 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 highly and 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 my framework design, which could be considered as a compensation for the high cost of above algorithms at runtime.

### 3.1.2 Visual exploration requirements

The design of geoscience data visual exploration depends on domain application requirements. In my framework design, map visualization requires to support users to interactively select a geographical region for querying a series of charts that are categorized by hydrology 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 [50].

Apart from traditional visual exploration requirements that visually answering where, when and what questions [51], I also focus on answering the comparison analysis between them, such as (1) data value visual differences between two geographical regions, (2) overall visual differences for each geographical region across the state, and (3) time-varying visual differences within a geographical region (e.g., comparison between precipitation in different seasons for a basin).

## 3.2 Data visualization design

### 3.2.1 Visual query constraint model

For geoscience data visualization, it is important for a map to have a visual query constraint to prevent a user navigate out the study area. This is because if the query returns null (i.e., indicating no data available), it will result in automation surprise. Therefore, in my visual analytic framework, I 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 navigate on the map.

Second, we denote $A$ as the bounding box of a geographical region $G$ (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 $\{g_1, g_2, \dots, g_n\}$ 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 $T = \{(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)\}$ 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.

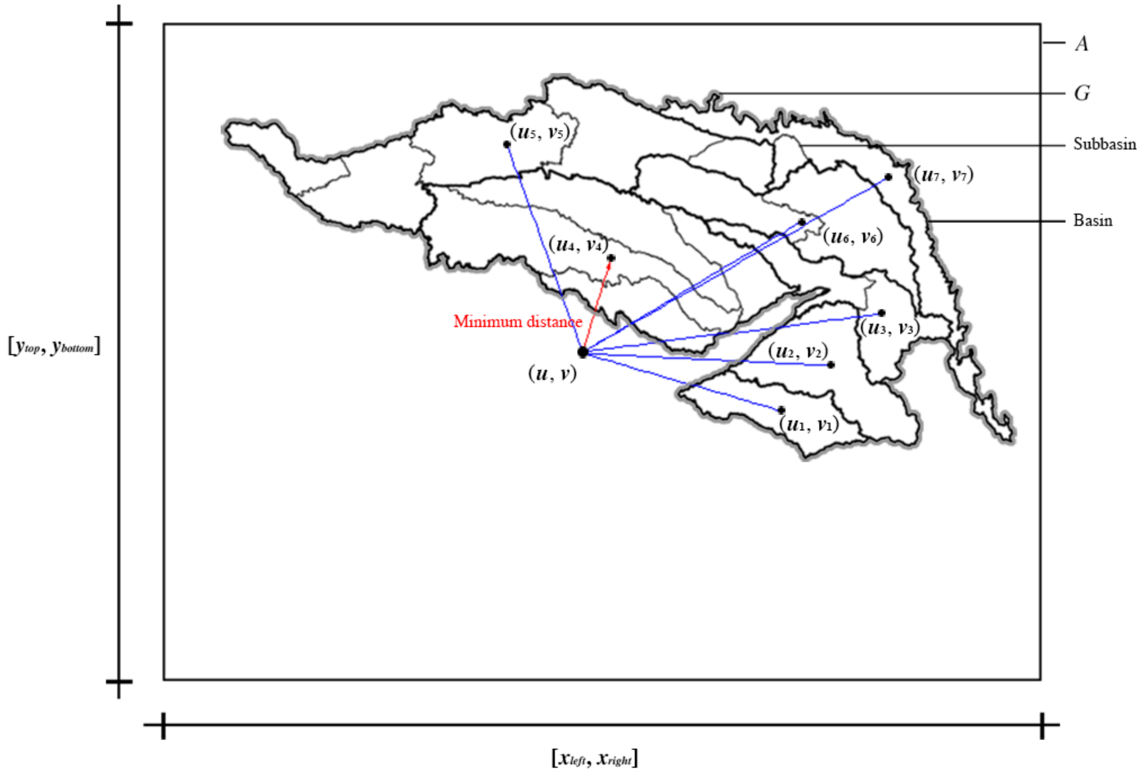


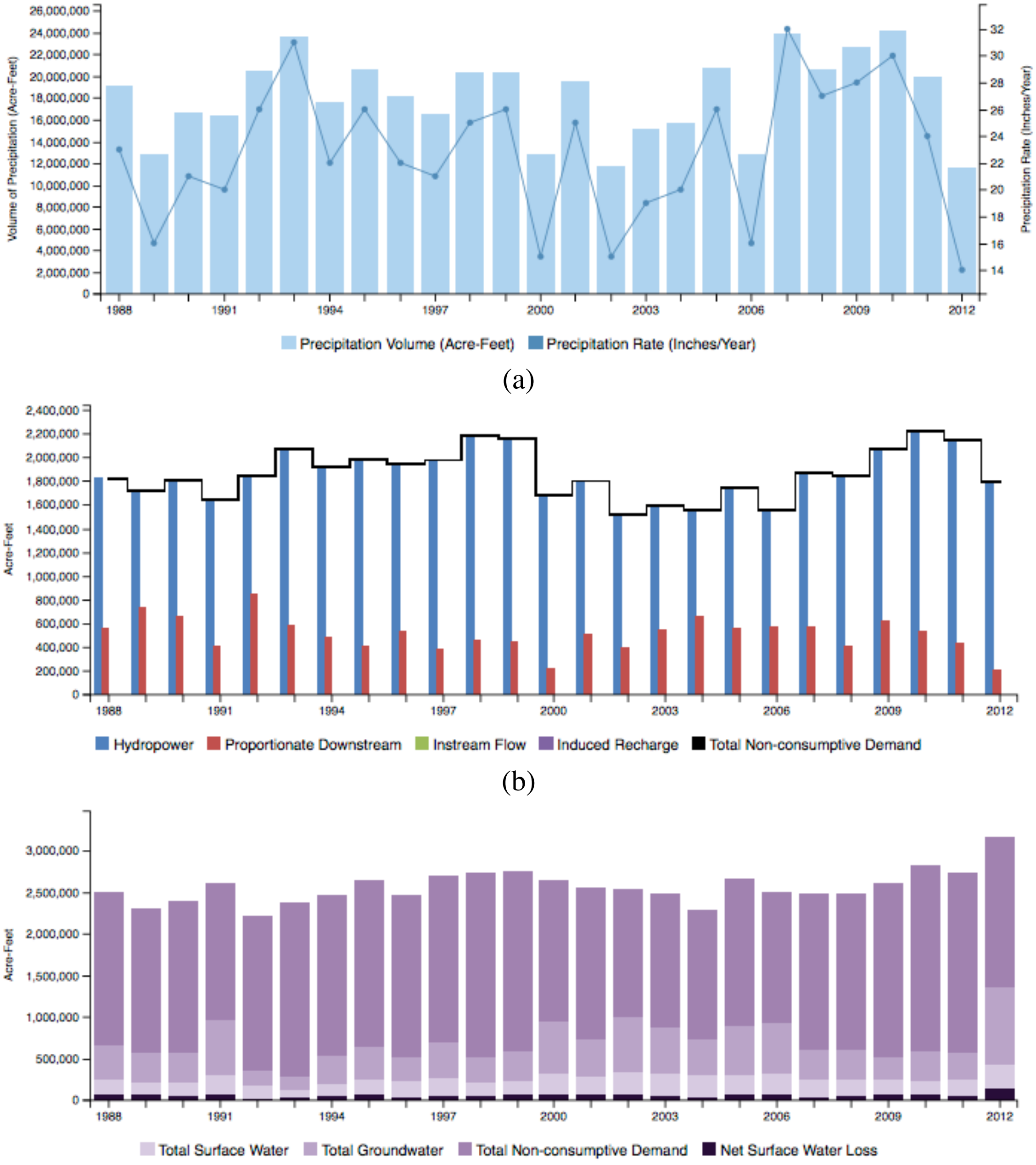
Fig. 2 Visual query constraint model. Compared with the blue lines, the red line is the minimum distance between $(u, v)$ and $(u_4, v_4)$, denoting that $(u_4, v_4)$ is the nearest neighbor.

### 3.2.2 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 and data links, and other UI components. In my framework design, specifically for visualizing hydrology data, different types of charts are applied to visualize the data. The 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 my framework design, such as combination chart, step chart, and stacked chart. Combination chart is a combination of traditional charts. One of its main advantages is that it can show data in many different ways or visualize different data together, which can enable visual analysis such as comparison and discovering data patterns (See Fig. 3 (a)). On the other hand, a combination chart may have too much visual information for a user when representing a large amount of data in visualization. Step chart is similar to the traditional bar chart, but has an advantage to show non-continuous value changes, such as temperature jumps or stock price changes, while the disadvantages is that it is less self-explanatory and it cannot show continuous value changes. Therefore, in my framework design, I combined step chart and bar chart together, in order to show the maximum value of each set of bars within a same time period (See Fig. 3 (b)). Stacked bar chart is another option for visualizing different data together, but it cannot visualize continuous values and consecutive values together. As it is more like a combination of pie chart and bar chart, a stacked bar chart has the merits of both pie chart and bar chart (for example, being able to show percentages and scalar values together) (See Fig. 3 (c)).

Chart interaction is another important visualization feature in my framework design. It allows a user to visually explore geoscience data to gain a better understanding of the data. 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.

By equipped with different visualizations and interactions together, our visual analytic framework can show the variations of data and provide a multiple perspective view of data.



(a)

(b)

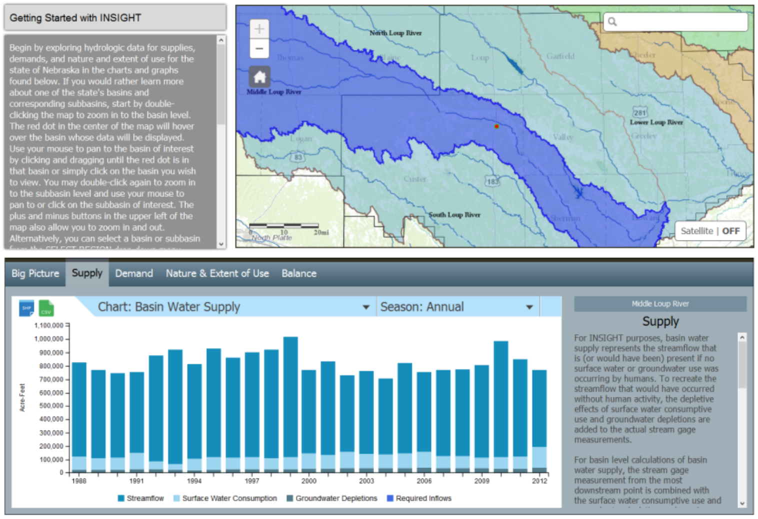
(c)

Fig. 3 Selected visual query results in my framework. (a) Combination chart; (b) step chart; and (c) stacked bar chart.

### 3.2.3 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 knowledges about geoscience studies. In addition, domain experts could also quickly memorize essential data patterns through visual expressiveness and then apply domain knowledges on these patterns to testify their hypotheses underneath the data.

Data visual repetition is an important factor that impacts on 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, I adopted the linked-view design [53] to create a linkage between geospatial information and geoscience data in my framework, as shown in Fig. 4. 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 not only give users an overall data summary of a geographical region, but also leave a strong visual expressiveness of each property of the geographical region in users' mind.



Begin by exploring hydrologic data for supplies, demands, and nature and extent of use for the state of Nebraska in the charts and graphs found below. If you would rather learn more about one of the state's basins and corresponding subbasins, start by double-clicking the map to zoom in to the basin level. The red dot in the center of the map will hover over the basin whose data will be displayed. Use your mouse to pan to the basin of interest by clicking and dragging until the red dot is in that basin or simply click on the basin you wish to view. You may double-click again to zoom in to the subbasin level and use your mouse to pan to or click on the subbasin of interest. The plus and minus buttons in the upper left of the map also allow you to zoom in and out. Alternatively, you can select a basin or subbasin

For INSIGHT purposes, basin water supply represents the streamflow that is (or would have been) present if no surface water or groundwater use was occurring by humans. To recreate the streamflow that would have occurred without human activity, the depletive effects of surface water consumptive use and groundwater depletions are added to the actual stream gage measurements.

For basin level calculations of basin water supply, the stream gage measurement from the most downstream point is combined with the surface water consumptive use and

**Map View**

**Chart View**

Fig. 4 The linked-view design. The map view allows users to interactively select different regions at different levels of detail. The chart view can dynamically generate the chart according to the user selected region and chart type.

# Chapter 4

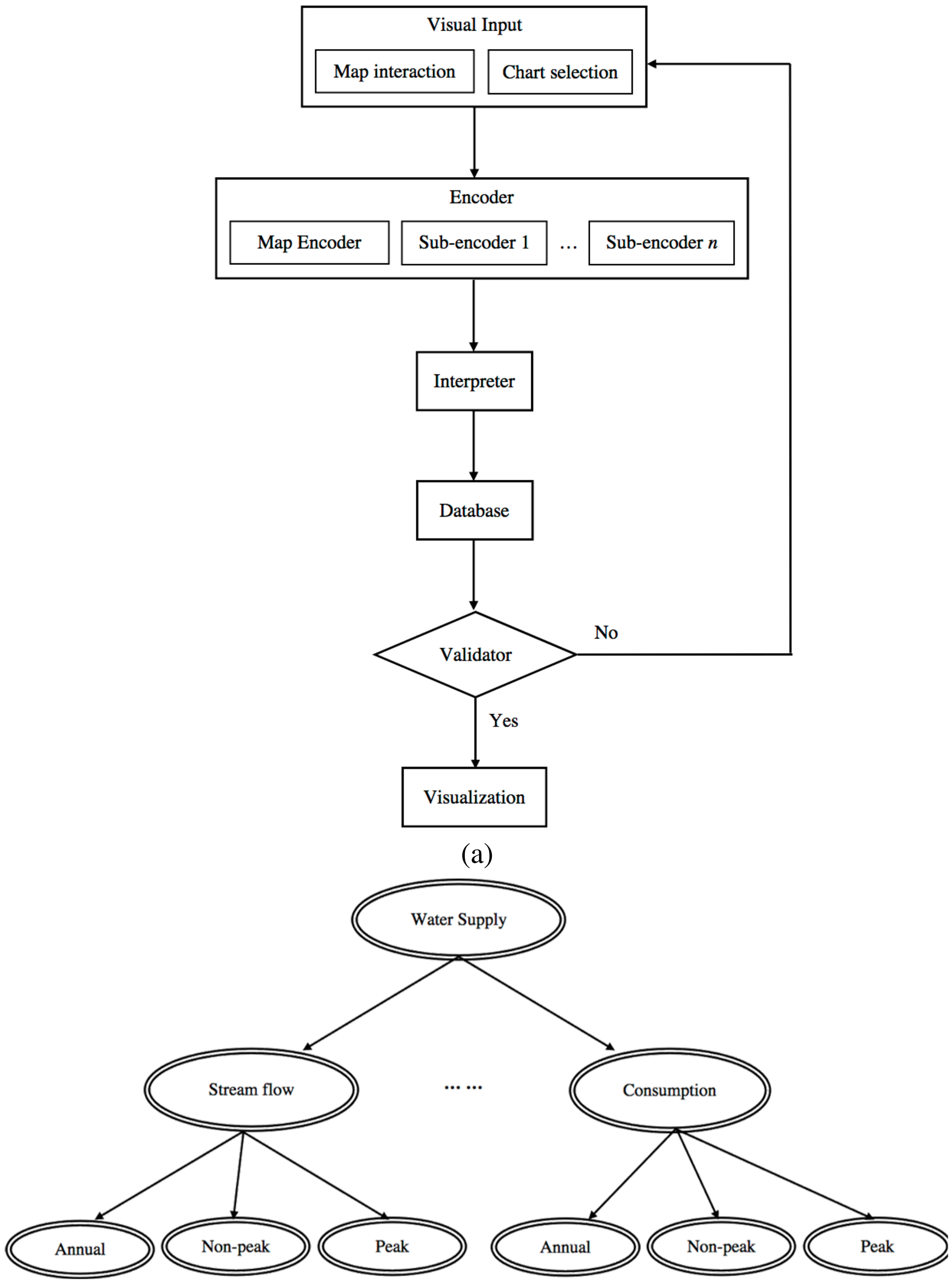
# Automata Design for Geoscience Data Visualization and User Interaction

## 4.1 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, gives scientists both direction and distance information, but the limited size of paper makes it hard for a scientist to reach a region that is out of the map. In addition, if a scientist travels on a map that covers a large area, he/she needs to continue change their visual extent on the map frequently (when traveling with a paper map, travelers focus on a small area that is reachable within a time period, such as in one day. If travelers reached the destination, then they will move on to next target area on the map). 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 added 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. In my research, I 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, my method could still produce a suitable user interface for geoscience data visualization and exploration.

## 4.2 Visual I/O

Visual input and output are the most two important components in geoscience data interaction, and decide the design of the corresponding interface automata. For geoscience data interactions, the visual input could be mouse hover, mouse click, zoom, pan, drag, rotate, and so on. The visual output could be charts, download links, analysis results, and so on. Fig. 5 (a) shows the visual I/O procedure in my 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. 5 (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 visualization; otherwise, the validator will select the nearest geographical area (based on the center of each geographical area) that has available data (See Section 3.2.1), and then start the retrieving process again from the encoder.



(a)

(b)

Fig. 5 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.

The encoded sequence is made of two parts, as shown in Fig. 6. 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. 5 (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 my framework design.

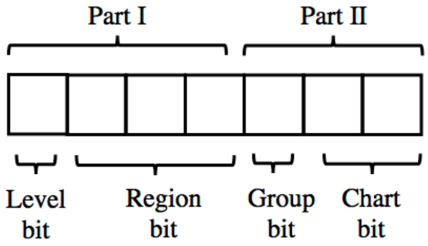


Fig. 6 The encoded sequence structure.

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.

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

Fig. 7 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 the subbasin level. +/- represents zoom in/out interaction, *d* represents drag interaction, and *ε* represents internal actions [12]. *A* ~ *G* denotes different basins, and *a* ~ *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.

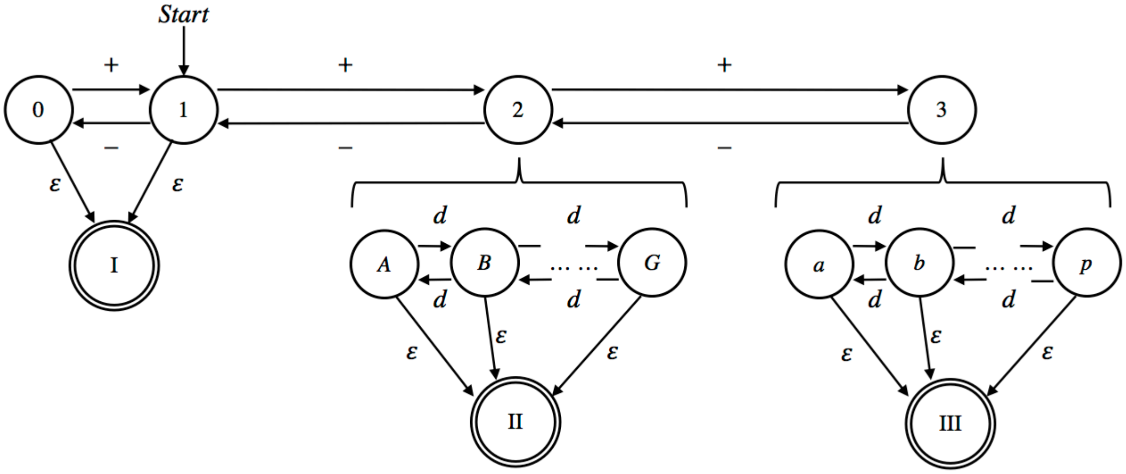


Fig. 7 Primary control component.

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 states 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 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. 8 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.

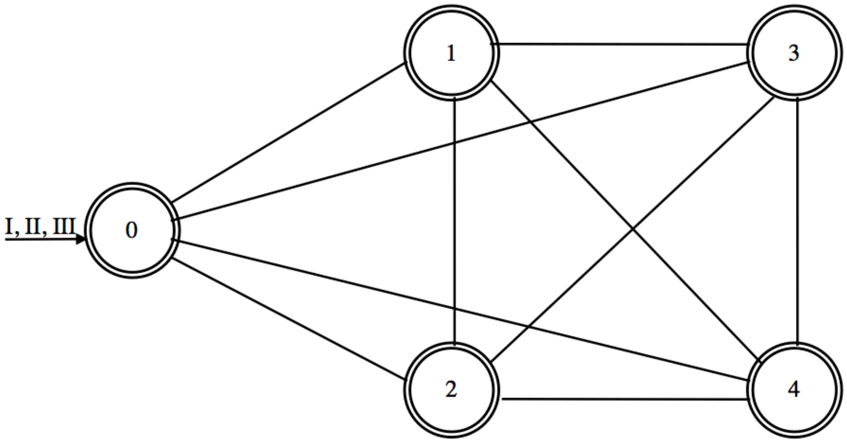


Fig. 8 An automaton generated by the secondary control component.

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. 8). 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 my 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 algorithm complexity. This makes a key contribution to geoscience data visualization in this research.

## 4.4 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 [12]. For example, $A_M^I$ could be $\{+, \text{-}, d, r\}$ ($d$ denotes a drag operation, $r$ 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 $A_M^H$ could be $\{\varepsilon\}$ ($\varepsilon$ 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. 7) is $(v', a, \varepsilon)$, whereas $a \in \ A_M^I$, 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. 8). 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 data.

In Chapter 5, we will present a detailed realization of our user interface design. We note that in some circumstance, the transition between two chart types may be prohibited. In this case, we can easily remove the link between the corresponding two nodes in our model to disable such a transition.

# Chapter 5

# Case Study

## 5.1 Background

Our framework has been applied to renovate the INSIGHT (An Integrated Network of Scientific Information & GeoHydrologic Tools) hydrology visualization system at NDNR. The users at NDNR desired 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 resources district (NRD); the third level shows the hydrology details for each basin; and the fourth level shows the water details of each subbasin.

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

## 5.2 New INSIGHT architecture

We realized our geoscience visual analytics framework (Chapter 3) for INSIGHT. The architecture of INSIGHT has two components, the frontend and the backend, as shown in Fig. 9.

The frontend is a web application for data visualization, where D3.js (a JavaScript library for visualizing data with HTML, SVG, and CSS) [54] is used to render data chats dynamically.

The backend has consisted 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 hydrology 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 hydrology 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 hydrology data is stored using Microsoft SQL Server (a product of Microsoft)[1]. 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

---

[1] This is the only option for NDNR because it is required to use Microsoft product by the CIO (Nebraska Chief Information Office).

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.

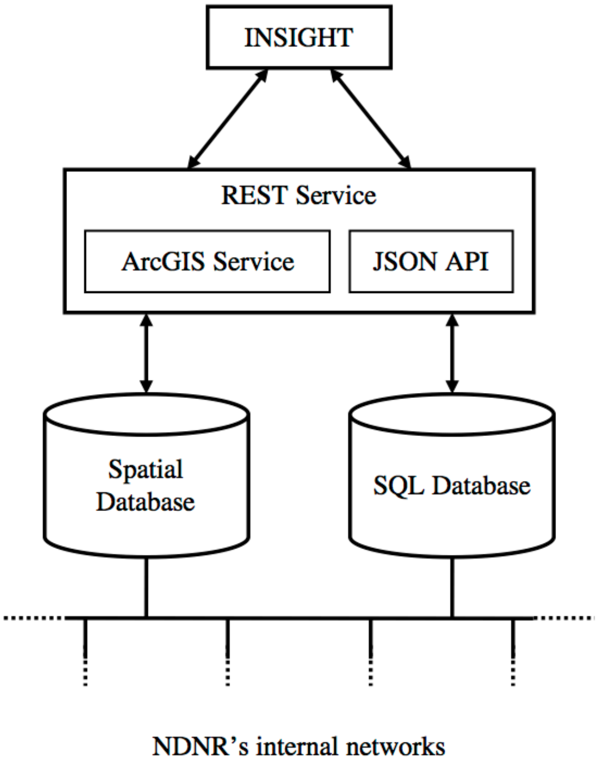
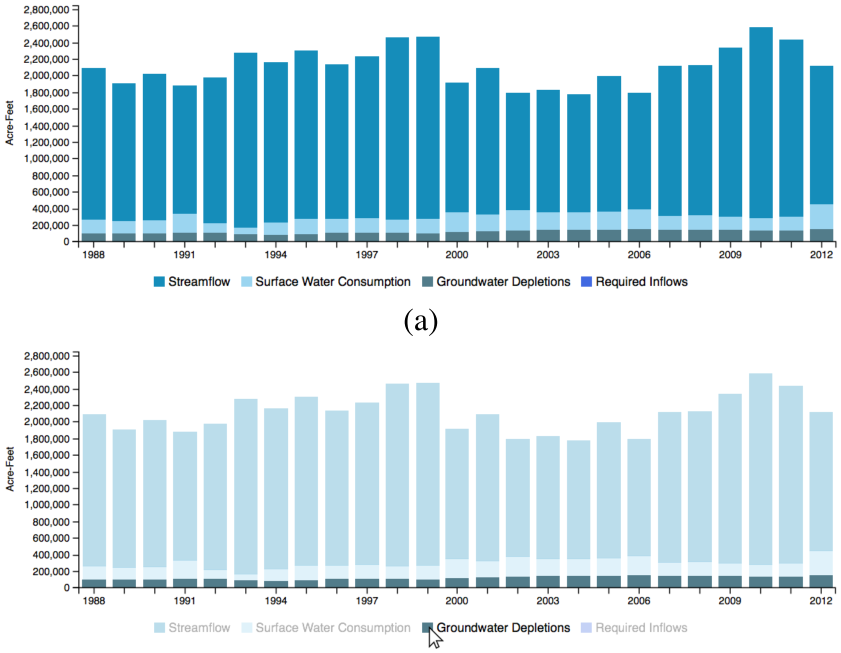


Fig. 9 INSIGHT architecture.

## 5.3 Hydrology 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 my framework design. For instance, the previous solution used a static image to represent each hydrology geographical region, cannot provide detail information either inside or outside the region, and also resulted in a high maintenance overhead. It also lacked of user interactions with hydrology data. For example, users cannot interactively get detailed data from a chart.

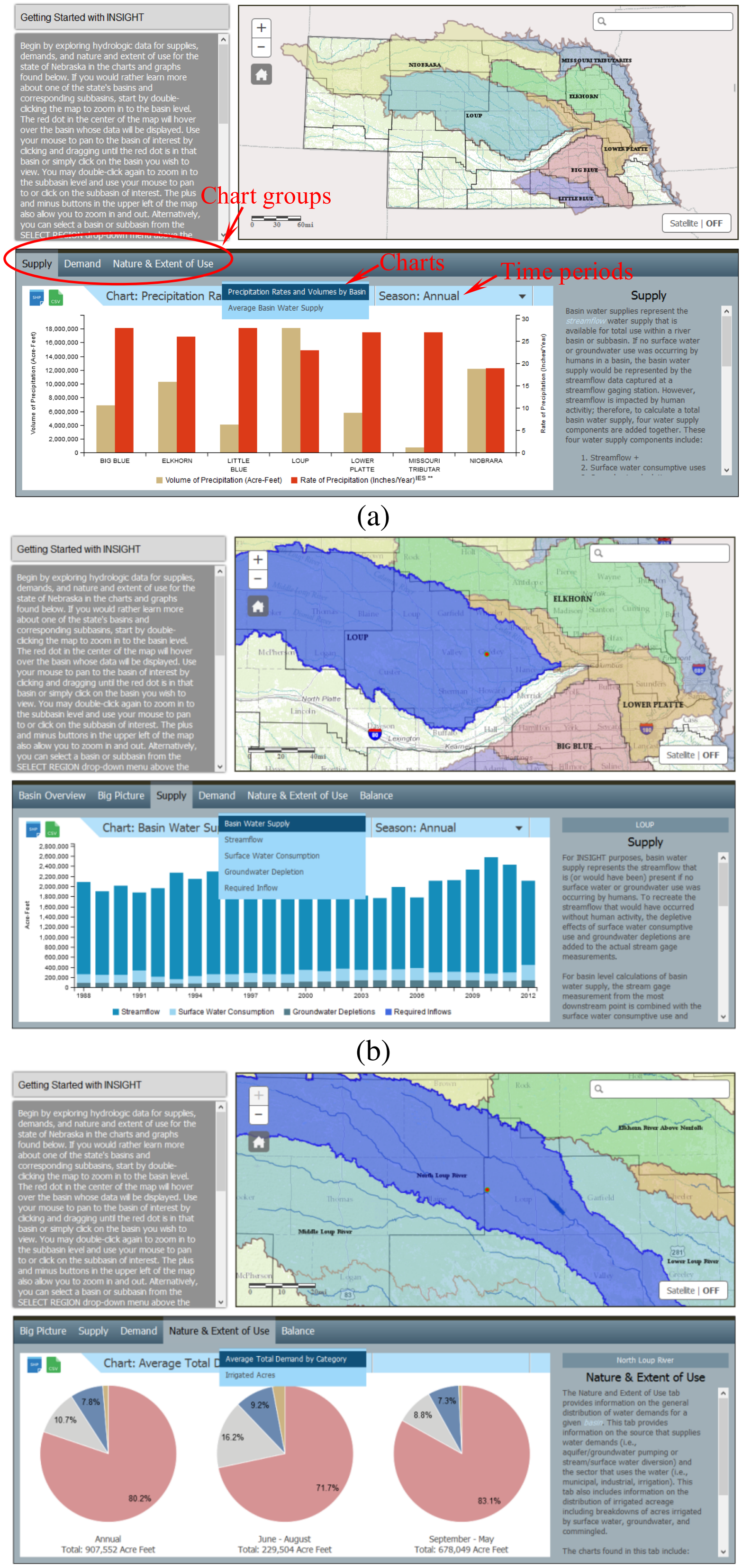
Because of our new design, users can not only retrieve both the spatial data and the data table, but also get detailed values from the new INSIGHT system. The system also enabled some new analysis capabilities, such as displaying water non-consumption use, which could be hard to realize using the previous 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 that can contain many new features, such as showing how many portions a member could have impact to the group value. For instance, Fig. 10 (a) shows a group of time-varying data visualized in stacked bar chart. Fig. 10 (b) shows a user can use the mouse to point to the legend to highlight a member of the group data, which gives the user a clear idea of how many portions of that member could impact to the group data.



(a)

(b)

Fig. 10 User generated visualization.



Getting Started with INSIGHT

Begin by exploring hydrologic data for supplies, demands, and nature and extent of use for the state of Nebraska in the charts and graphs found below. If you would rather learn more about one of the state's basins and corresponding subbasins, start by double-clicking the map to zoom in to the basin level. The red dot in the center of the map will hover over the basin whose data will be displayed. Use your mouse to pan to the basin of interest by clicking and dragging until the red dot is in that basin or simply click on the basin you wish to view. You may double-click again to zoom in to the subbasin level and use your mouse to pan to or click on the subbasin of interest. The plus and minus buttons in the upper left of the map also allow you to zoom in and out. Alternatively, you can select a basin or subbasin from the SELECT REGION drop-down menu above the

Supply

Basin water supplies represent the *streamflow* water supply that is available for total use within a river basin or subbasin. If no surface water or groundwater use was occurring by humans in a basin, the basin water supply would be represented by the streamflow data captured at a streamflow gaging station. However, streamflow is impacted by human activitiy; therefore, to calculate a total basin water supply, four water supply components are added together. These four water supply components include:

1. Streamflow +
2. Surface water consumptive uses

(a)

Getting Started with INSIGHT

Begin by exploring hydrologic data for supplies, demands, and nature and extent of use for the state of Nebraska in the charts and graphs found below. If you would rather learn more about one of the state's basins and corresponding subbasins, start by double-clicking the map to zoom in to the basin level. The red dot in the center of the map will hover over the basin whose data will be displayed. Use your mouse to pan to the basin of interest by clicking and dragging until the red dot is in that basin or simply click on the basin you wish to view. You may double-click again to zoom in to the subbasin level and use your mouse to pan to or click on the subbasin of interest. The plus and minus buttons in the upper left of the map also allow you to zoom in and out. Alternatively, you can select a basin or subbasin from the SELECT REGION drop-down menu above the

LOUP

Supply

For INSIGHT purposes, basin water supply represents the streamflow that is (or would have been) present if no surface water or groundwater use was occurring by humans. To recreate the streamflow that would have occurred without human activity, the depletive effects of surface water consumptive use and groundwater depletions are added to the actual stream gage measurements.

For basin level calculations of basin water supply, the stream gage measurement from the most downstream point is combined with the surface water consumptive use and

(b)

Getting Started with INSIGHT

Begin by exploring hydrologic data for supplies, demands, and nature and extent of use for the state of Nebraska in the charts and graphs found below. If you would rather learn more about one of the state's basins and corresponding subbasins, start by double-clicking the map to zoom in to the basin level. The red dot in the center of the map will hover over the basin whose data will be displayed. Use your mouse to pan to the basin of interest by clicking and dragging until the red dot is in that basin or simply click on the basin you wish to view. You may double-click again to zoom in to the subbasin level and use your mouse to pan to or click on the subbasin of interest. The plus and minus buttons in the upper left of the map also allow you to zoom in and out. Alternatively, you can select a basin or subbasin from the SELECT REGION drop-down menu above the

North Loup River

Nature & Extent of Use

The Nature and Extent of Use tab provides information on the general distribution of water demands for a given *basin*. This tab provides information on the source that supplies water demands (i.e., aquifer/groundwater pumping or stream/surface water diversion) and the sector that uses the water (i.e., municipal, industrial, irrigation). This tab also includes information on the distribution of irrigated acreage including breakdowns of acres irrigated by surface water, groundwater, and commingled.

The charts found in this tab include:

(c)

Fig. 11 Different chart groups and charts for different regions.

The user interface of the new 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 & 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 zoomed 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 have been changed 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 are changed accordingly. According to 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.

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, which will be discussed in Chapter 6.

# Chapter 6

# Evaluation

My evaluation investigates the implementation cost and the effectiveness of visualization of my framework through the INSIGHT case study. I also evaluate how it can enhance the user experience and the performance compared to the previous INSIGHT system.

## 6.1 Implementation cost

One of the advantages of the new framework is that it can significantly reduce the amount of 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.

## 6.2 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 on the interface. 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 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 analysis 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.

## 6.3 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 backend 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 1 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.

In general, by integrating my 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 my framework for visualizing time-varying multivariate geoscience data.

# Chapter 7

# Conclusion

In this thesis, I proposed a new geoscience data visualization and analytic framework. The novelty of my study is providing an example of how to apply the interface automata theory to geoscience data visualization. My study has three major contributions to the 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 (http://data.dnr.ne.gov/insight) has been operated on a daily basis by NDNR to provide an effective exploration of hydrologic data at Nebraska.

My 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, I plan to continue to improve this framework and consider more data visualization techniques, such as raster data visualization and geometry data visualization. I will study integration strategies and enable a holistic user experience across different visualization techniques.

# Bibliography

[1] Michael P. Peterson. *Online Maps with APIs and WebServices*. pp. 3-12, Heidelberg, Germany, 2012. Springer.

[2] Mohamed Sarwat. Interactive and Scalable Exploration of Big Spatial Data - A Data Management Perspective. *IEEE International Conference on Mobile Data Management*, pp. 263-270, Pittsburgh, PA, USA, 2015. IEEE.

[3] Ahmed Eldawy, Mohamed F. Mokbel. SHAHED: A MapReduce-based System for Querying and Visualizing Spatio-temporal Satellite Data. *International Conference on Data Engineering*, pp. 1585-1596, Seoul, Korea, 2015. IEEE.

[4] Jianting Zhang, Simin You, Le Gruenwald. Data Parallel Quadtree Indexing and Spatial Query Processing of Complex Polygon Data on GPUs. *International Conference on Very Large Data Bases*, Hangzhou, China, 2014.

[5] Simin You, Jianting Zhang, Le Gruenwald. Parallel Spatial Query Processing on GPUs Using R-Trees. *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, pp. 23-31, New York, NY, USA, 2013. ACM.

[6] Satish Puri, Sushil K. Prasad. A Parallel Algorithm for Clipping Polygons with improved bounds and A Distributed Overlay Processing System using MPI. *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Shenzhen, China, 2015. IEEE.

[7] Bala R. Vatti, A Generic Solution to Polygon Clipping. *Communications of the ACM*, vol. 35, no. 7, pp. 56-63, 1992.

[8] Raghu R. Karinthi, Kankanahalli Srinivas, and George Almasi, A parallel algorithm for computing polygon set operations. *Journal of Parallel and Distributed Computing*, vol. 26, no. 1, pp. 85-98, 1995.

[9] Ilias K. Savvas. Clustering EU's Countries According to I. Th. Mazi's Systemic Geopolitical Theory Using K-means and MPI. *International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 78-83, Larnaca, Cyprus, 2015. IEEE.

[10] Alexandre Sorokine. Implementation of a parallel high-performance visualization technique in GRASS GIS. *Computers & Geosciences*, vol. 33, issue 5, pp. 685-695, 2007.

[11] David Pastor-Escuredo, Alfredo Morales-Guzmán, Yolanda Torres-Fernández. Flooding through the Lens of Mobile Phone Activity. *IEEE Global Humanitarian Technologies Conference*, pp. 279-286, San Jose, CA, USA, 2014. IEEE.

[12] Luca de Alfaro, Thomas A. Henzinger. Interface Automata. *Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering*, pp. 109-120, New York, NY, USA, 2001. ACM.

[13] Muffy Calder, Philip Gray, Alice Miller. An Introduction to Pervasive Interface Automata. *Formal Aspects of Component Software*, pp. 71-87, Heidelberg, Germany, 2012. Springer.

[14] Gerhard Viehstaedt, Mark Minas. Interaction in Really Graphical User Interfaces. *Proceedings of IEEE Symposium on Visual Languages*, pp. 270-277, St. Louis, MO, USA, 1994. IEEE.

[15] Haswanee D. Goodur, Vyasanand Ramtohul ; Shakuntala Baichoo. GIDT — A tool for the identification and visualization of genomic islands in prokaryotic organisms. *IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE)*, pp. 58-63, Larnaca, Cyprus, 2012, IEEE.

[16] Ayush Shrestha, Ying Zhu, Yan Zhu. Visual cluttering reduction for visualizing large spatio-temporal data sets. *IEEE 17th International Conference on Computational Science and Engineering*, pp. 137-143, Chengdu, China, 2014. IEEE.

[17] Shunsuke Wada, Yuichi Yaguchi, Ryo Ogata. Associated Keyword Analysis for Temporal Data with Spatial Visualization. *International Joint Conference on Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA)*, pp, 243-249, Aizuwakamatsu, Japan, 2013. IEEE.

[18] Dasuni Kannangara, Nimalika Fernando. A web based methodology for Visualizing Time-varying Spatial Information. *4th International Conference on Industrial and Information Systems*, pp. 233-238, Sri Lanka, 2009. IEEE.

[19] Hideki Hayashi, Akinori Asahara. Spatio-Temporal Similarity Search Method for Disaster Estimation. *IEEE International Conference on Big Data*, pp. 2463-2469, Santa Clara, CA, USA, 2015. IEEE.

[20] Poonam Shanbhag, Penny Rheingans. Temporal Visualization of Planning Polygons for E_cient Partitioning of Geo-Spatial Data. *IEEE Symposium on Information Visualization*, pp. 211-218, 2005. IEEE.

[21] Imre Felde, László Nádai, Miklós Mezei, Gábor Bognár. Characterization of of urban traffic by using mobile phone traces. *IEEE International Conference on*

*Systems, Man, and Cybernetics*, pp. 3360-3363, San Diego, CA, USA, 2014. IEEE.

[22] Nimmy Ravindran, Yao Liang. HIDE – A Web-based Hydrological Integrated Data Environment. *8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pp. 143-148, Qingdao, China, 2007. IEEE.

[23] Vineeth Rakesh, Chandan K. Reddy, Dilpreet Singh. Location-Specific Tweet Detection and Topic Summarization in Twitter. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 1441-1444, Niagara, Ontario, CAN, 2013. IEEE.

[24] Ana Oliveira, João Araújo, Armanda Rodrigues, Ana Moreira. Improving the Quality of Web-GIS Modularity with Aspects. *7th International Conference on the Quality of Information and Communications Technology*, pp. 132-141, Porto, Portugal, 2010, IEEE.

[25] Zeqian Shen, Kwan-Liu Ma. MobiVis: A Visualization System for Exploring Mobile Data. *IEEE Pacific Visualization Symposium*, pp. 175-182, Kyoto, Japan, 2008, IEEE.

[26] Ang Li, Zishan Qin, Mingsong Chen. ADAutomation: An Activity Diagram Based Automated GUI Testing Framework for Smartphone Applications. *International Conference on Software Security and Reliability*, pp. 68-77, San Francisco, CA, USA, 2014. IEEE.

[27] Dam Sunwoo, William Wang, Mrinmoy Ghosh. A Structured Approach to the Simulation, Analysis and Characterization of Smartphone Applications. *IEEE International Symposium on Workload Characterization (IISWC)*, pp. 113-122, Portland, OR, USA, 2013. IEEE.

[28] Étienne André, Yousra Lembachar, Laure Petrucci. *CosyVerif* : an Open Source Extensible Verification Environment. *International Conference on Engineering of Complex Computer Systems*, pp. 33-36, Singapore, 2013. IEEE.

[29] Paul Givens, Aleksandar Chakarov, Sriram Sankaranarayanan. *Exploring the Internal State of User Interfaces by Combining Computer Vision Techniques with Grammatical Inference*, pp. 1165-1168, San Francisco, CA, USA, 2013. IEEE.

[30] Renyi Zhao, V Lakshmi Narasimhad. Algebraic Semantics for Complete Interaction Sequence. *TENCON - IEEE Region 10 Conference*, pp. 1-10, Melbourne, Qld, Australia, 2005. IEEE.

[31] Lee White, Husain Almezen. Generating Test Cases for GUI Responsibilities Using Complete Interaction Sequences. *Proceedings 11th International Symposium on Software Reliability Engineering*, pp. 110-121, San Jose, CA, USA, 2000. IEEE.

[32] Waseem A. Mufti, Dimitri C. Tcherukine. Integration of Model-Checking Tools: from Discrete to Hybrid Models. *IEEE International Multitopic Conference (INMIC)*, pp. 1-4, Lahore, Pakistan, 2007. IEEE.

[33] Makoto Inoue, Hideyuki Takagi. Layout Algorithm for an EC-based Room Layout Planning Support System. *IEEE Conference on Soft Computing in Industrial Applications (SMCia)*, pp. 165-170, Muroran, Japan, 2008. IEEE.

[34] Ramón Alcarria, Tomás Robles, Augusto Morales. A Variability Model for Template Personalization in Mobile Prosumer Environments. *International Conference on Advanced Information Networking and Applications Workshops*, pp. 1368-1373, Barcelona, Spain, 2013. IEEE.

[35] Mikael Fridenfalk. Application of Cellular Automata for Generation of Chess Variants. *IEEE International Games Innovation Conference (IGIC)*, pp. 57-63, Vancouver, BC, Canada, 2013. IEEE.

[36] Yoshifumi Masunaga, Chiemi Watanabe. Design and Implementation of a Multi-modal User Interface of the Virtual World Database System (VWDB). *Proceedings 7th International Conference on Database Systems for Advanced Applications*, pp. 294-301, Hong Kong, China, 2001. IEEE.

[37] Ali Asghar Nazari Shirehjini, Shervin Shirmohammadi. A Mobile 3D User Interface for Interaction with Ambient Audio Visual Environments. *IEEE International Workshop on Haptic Audio Visual Environments and Games*, pp. 186-191, Lecco, Italy, 2009. IEEE.

[38] Dávid Vincze, Szilveszter Kovács. A simple interface to the Virtual Collaboration Arena for MATLAB applications. *12th IEEE International Symposium on Computational Intelligence and Informatics*, pp. 357-360, Budapest, Hungary, 2011. IEEE.

[39] Alexandru-Corneliu Olteanu, George-Daniel Oprina. Enabling mobile devices for home automation using ZigBee. *19th International Conference on Control Systems and Computer Science*, pp. 189-195, Bucharest, Romania, 2013. IEEE.

[40] Achilleas Achilleos, Nearchos Paspallis, George A. Papadopoulos. Automating the Development of Device-Aware Web Services: A Model-Driven Approach. *35th IEEE Annual Computer Software and Applications Conference*, pp. 535-540, Munich, Germany, 2011. IEEE.

[41] Mattia Gamba, Alessandro Gonella, Claudio E. Palazzi. Design Issues and Solutions in a Modern Home Automation System. *International Workshop on Networking Issues in Multimedia Entertainment*, pp. 1111-1115, Garden Grove, CA, USA, 2015. IEEE.

[42] Eduardo Rocha Rodrigues, Airam Jonatas Preto, Stephan Stephany. A Parallel Engine for Graphical Interactive Molecular Dynamics Simulations. *Proceedings of the 16th Symposium on Computer Architecture and High Performance Computing*, pp. 150-157, 2004. IEEE.

[43] Christian I. Penaloza, Yasushi Mae, Francisco F. Cuellar. Brain Machine Interface System Automation Considering User Preferences and Error Perception Feedback. *IEEE Transactions on Automation Science and Engineering*, pp. 1275-1281, 2014. IEEE.

[44] Taowei David Wang, Amol Deshpande, Ben Shneiderman. A Temporal Pattern Search Algorithm for Personal History Event Visualization. *IEEE Transactions on Knowledge and Data Engineering*, pp. 799-812, 2012. IEEE.

[45] Laurie Ricker, Stephane Lafortune. DESUMA: A Tool Integrating GIDDES and UMDES. *Proceedings of the 8th International Workshop on Discrete Event Systems*, pp. 392-393, Ann Arbor, Michigan, USA, 2006. IEEE.

[46] Valeri Ayvazyan, Kay Rehlich, Stefan Simrock. Finite State Machine Implementation to Automate RF Operation at the TESLA Test Facility. *Proceedings of the 2001 Particle Accelerator Conference*, pp. 286-288, Chicago, IL, USA, 2001. IEEE.

[47] Patrick Wacht, Thomas Eichelmann, Armin Lehmann. A New Approach to Design Graphically Functional Tests for Communication Services. *4th IFIP International Conference on New Technologies, Mobility and Security*, pp. 1-5, Paris, French, 2011. IEEE.

[48] YoungJoon Byun. A Tool Support for Design and Validation of Communication Protocol using State Transition Diagrams. *International Conference on Information Technology*, pp, 628-636, Las Vegas, NV, USA, 2007. IEEE.

[49] Daniel Simon, Bernard Espiau. Computer-Aided Design of a Generic Robot Controller Handling Reactivity and Real-Time Control Issues. *IEEE Transactions on Control Systems Technology*, pp. 213-229, 2002. IEEE.

[50] Daiki Ishii, Toshimitsu Ushio. A Bisimulation-Based Design of User Interface with Alerts Avoiding Automation Surprises. *IEEE Transactions on Human-Machine Systems*, pp. 317-323, 2016. IEEE.

[51] Nivan Ferreira, Jorge Poco, Huy T. Vo. Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips. *IEEE Transactions on Visualization and Computer Graphics*, pp. 2149-2158, 2013.IEEE.

[52] Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *Visual Languages,* Proceedings., IEEE Symposium on,

Boulder, CO, 1996, pp. 336-343. 1996

[53] Jeffrey Heer and Ben Shneiderman. Interactive Dynamics for Visual Analysis. Commun. ACM 55, 4 (April 2012), pp.45-54. 2012.

[54] D3.js: Data-Driven Documents. https://d3js.org/.