VisSnippets: A Web-Based System for Impromptu Collaborative Data Exploration on Large Displays

Andrew Burks
aburks3@uic.edu
Electronic Visualization Laboratory
University of Illinois at Chicago
Chicago, Illinois

Luc Renambot
renambot@uic.edu
Electronic Visualization Laboratory
University of Illinois at Chicago
Chicago, Illinois

Andrew Johnson
ajohnson@uic.edu
Electronic Visualization Laboratory
University of Illinois at Chicago
Chicago, Illinois

Figure 1: An example of a VisSnippets collaborative session, one user is standing near the large display wall and examining financial data trends. The seated user on the left is interacting with the display content using a personal computer, while the seated user on the right has the source code for the visualization open to make changes and advance the exploration.

ABSTRACT

The VisSnippets system is designed to facilitate effective collaborative data exploration. VisSnippets leverages SAGE2 middleware that enables users to manage the display of digital media content on large displays, thereby providing collaborators with a high-resolution common workspace. Based in JavaScript, VisSnippets provides users with the flexibility to implement and/or select visualization packages and to quickly access data in the cloud. By simplifying the development process, VisSnippets removes the need to scaffold and integrate interactive visualization applications by hand. Users write reusable blocks of code called "snippets" for data retrieval, transformation, and visualization. By composing dataflows from the group's collective snippet pool, users can quickly execute and explore complementary or contrasting analyses. By giving users the ability to explore alternative scenarios, VisSnippets facilitates parallel work for collaborative data exploration leveraging large-scale displays. We describe the system, its design and implementation, and showcase its flexibility through two example applications.

CCS CONCEPTS

• **Human-centered computing** → **Visualization systems and tools**; *Collaborative and social computing systems and tools.*

KEYWORDS

information visualization, visual data science, collaborative visual analytics

ACM Reference Format:
Andrew Burks, Luc Renambot, and Andrew Johnson. 2020. VisSnippets: A Web-Based System for Impromptu Collaborative Data Exploration on Large Displays. In *Practice and Experience in Advanced Research Computing (PEARC '20), July 26–30, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3311790.3396666

PEARC '20, July 26–30, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-6689-2/20/07…$15.00
https://doi.org/10.1145/3311790.3396666

1 INTRODUCTION

In the age of data-intensive scientific discovery [13], the influx of information across scientific domains requires interdisciplinary efforts to gain a holistic and actionable understanding of the data. This interdisciplinary collaboration necessitates systems which

allow experts to apply their soft-knowledge and semantic understanding of these data-intensive problems and efficiently collaborate with one another, regardless of the scientific domain [40]. Recently, computational notebooks [18, 28] have become widely adopted across scientific domains as effective tools for exploration and explanation [19, 35], with Jupyter [18] becoming the standard. To improve the usability of notebooks, supplementary meta-visualization of notebook structure [43] has been studied and some notebooks [9, 28] support collaboration through collaborative editing. Nevertheless, the representation and organization of these notebooks remains a hindrance to collaborative analysis and the simultaneous exploration and comparison of alternative hypotheses.

With the scale of data constantly growing, visualization techniques must be adapted or introduced to handle the number of entries, multidimensionality, increasing levels of detail, and complexity or heterogeneity of data [2]. In this context, large, high-resolution displays provide "space to think" [1] and space for collaborators to work together or in parallel with high-resolution information. SAGE2: The Scalable Amplified Group Environment [26] is an open-source middleware which leverages any size display, from 65" high-resolution televisions to 30' tiled display walls, to create a collaborative workspace in which users can share, organize, and interact with large amounts of diverse content [22]. SAGE2 has a large international user community across science domains of 100+ sites worldwide[1]. However, SAGE2 and other free-form collaborative systems lack generalized tooling supporting complex, data-intensive work.

VisSnippets provides teams with an exploratory programming environment complemented by a reactive dataflow system to allow users to quickly and simultaneously incorporate diverse data sources and scaffold exploratory analysis/visualization within their collaborative sessions, supported by the windowing and content manipulation capabilities of SAGE2. VisSnippets takes advantage of large, high-resolution displays, affording users space to collaborate while performing parallel and subjunctive analyses [25]. Abstractions built into the system allow users to quickly create and connect highly interactive linked visualization applications. Additionally, the web-based implementation of the VisSnippets system makes it compatible with most popular web visualization toolkits, including D3 [6] and Vega-Lite [37].

The design of VisSnippets is motivated by two primary aspects: supporting *Mixed-focus Collaboration* coupled with *Expressive Analysis*. During exploratory analysis, **Mixed-focus Collaboration** allows for of transient states of individual and shared, group activity with an emphasis on enabling the *Low-cost Integration* of parallel streams of work. In order to support this collaborative problem-solving across multiple domains, **Expressive Analysis** allows users to integrate their data- and domain- specific semantic understanding into analyses.

In this paper, we motivate and discuss the design and implementation of our VisSnippets system in the context of these two main design principles. To illustrate the flexibility and expressiveness of VisSnippets, we discuss two example applications in *section 4*. Finally, we discuss the limitations of VisSnippets and propose directions for future research.

[1]VisSnippets is open-source and released with SAGE2, available at https://sage2.sagecommons.org/

2 RELATED WORK

The two aspects which underpin our design of VisSnippets are motivated by the following research in collaborative visual analytics, particularly in the context of large displays, and visualization authoring during exploratory analysis. Additionally, we note three existing applications at the intersection of these areas.

Collaborative Visual Analytics. Within group work, "mixed-focus collaboration" denotes shifts between individual **loosely coupled** work and shared **closely coupled** collaboration [11, 39]. *Loosely coupled* work may occur when two users are working quietly and independently on different sections of a problem, moving later to *closely coupled* collaboration while discussing the results of each user and synthesizing new information. Research has outlined sets of overlapping design considerations and challenges in the creation of new collaborative visual analytics tools to support all levels of collaborative coupling [12, 14]. While studies have shown the effectiveness of closely-coupled collaboration [8, 15], both levels of coupling are effective for different analytical tasks, assuming that collaborators can efficiently integrate parallel streams of work and establish of common ground [16]. Methods of establishing this common ground include creating fused views from parallel knowledge [7] or parallel, subjunctive analysis scenarios [25].

Large High-resolution Displays. During collaborative analysis and problem solving, large high-resolution displays provide a shared workspace which supports collaborative sensemaking for analytical tasks in high-fidelity. These display systems provide users a sense of ownership of their content and workspace partition during collaborative work [23]. For single users and groups, software like SAGE [33], DisplayCluster [17], and SAGE2 [26] transform large high-resolution displays into an enhanced workspace which supports diverse collaborative work. Large high-resolution displays provide "space to think," supporting a user's sensemaking by outsourcing memory to the display and applying a semantic

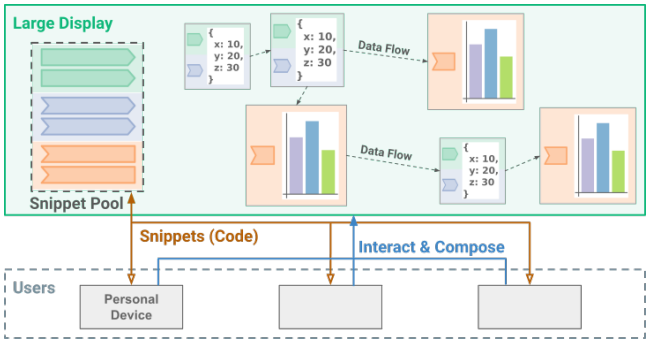

Figure 2: *The VisSnippets workflow.* The shared Snippet Pool can be added to or edited by users on their laptops. Next, users compose the modular analyses through direct manipulation to scaffold data flows organized across the collaborative workspace. Users may also interact with the interfaces and visualizations produced.

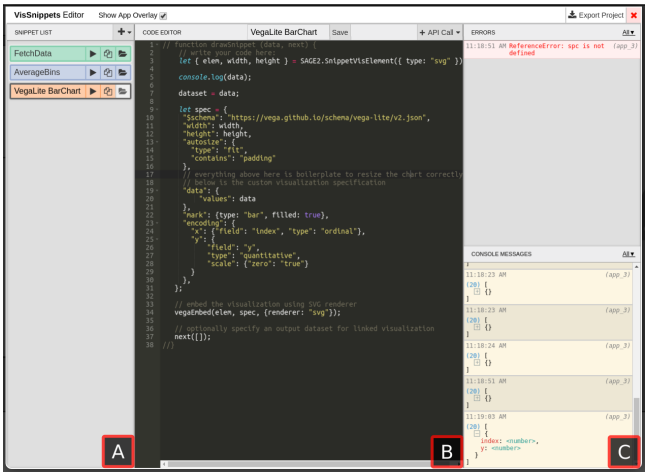

Figure 3: *The VisSnippets editor.* A The Snippet List displays a group's snippets and allows users to create new snippets by type. Three icons on each snippet (left to right) allow users to run, clone, or load and edit a snippet. B The Code Editor lets users edit, rename, and save snippets. Additionally, the editor provides functionality to insert VisSnippets API calls by type. C The Debugging Panel redirects any errors or logging output captured during a snippet's execution.

layer through content's spatial organization [1]. Research has found these displays to be more effective than traditional displays for the basic pan-zoom visualization task [4] and also for insight acquisition in more complex visual data exploration tasks [31]

Visualization Authoring. To maximize their applicability across domains, collaborative visual analytics systems must afford users flexibility in the visual representation of data and functionality of the visualizations while addressing the trade-off between expressiveness and efficiency. Software like Microsoft Excel and Tableau (formerly Polaris) [38] provide users with lower expressiveness in favor of declarative efficiency. Toolkits for custom visualization authoring such as Protovis [5] and D3 [6] provide designers with a high level of control while minimizing the loss of efficiency though careful abstraction. High-level visualization grammars Vega [42] and Vega-Lite [37] maximize efficiency while maintaining their expressiveness and compatibility with low-level implementation. These grammars are supplemented by tools for data exploration through specification recommendation [44, 45] and automated visualization creation using a constraint solver [27]. In contrast, designer-oriented authoring tools maximize efficiency through direct manipulation [32, 36] or vector drawing [24]. Nevertheless, the recent growth in popularity and adoption of computational notebooks [18, 28] has shown that many analysts value programming and its expressiveness for exploratory data analysis, favoring small scripts for incremental analysis [19] and live programming environments [10].

Few applications sit at the intersection of these three background areas. Lark [41] supports mixed-focus collaboration around a touch-enabled table through the use of linked visualizations coupled with a data pipeline meta visualization. Langner et al. [21] built a collaborative analysis tool and found that the coordinated views application (45+ views) was effective for collaborative data analysis. However, each was limited by the included data types, operations, or visualization choices. Vistrates [3] uses the paradigm of "sharable dynamic media" [20, 30] to support expressive visualization authoring across devices and user expertise. VisSnippets, in contrast, focuses on collaborative *exploration* by supporting rapid prototyping, mixed-focus collaboration, and parallel hypothesis testing through subjunctive analytical workflows. Additionally, the freedom through which users can define and compose new analyses in VisSnippets mitigates the problem of limited included functionality.

In summary, *Mixed-focus Collaboration* is critical in facilitating diverse analytical tasks and the challenges associated with each form of collaborative coupling can be mitigated through the workspace partitioning and individual ownership afforded by large displays. Furthermore, *Low-cost Integration* of parallel work is critical when transitioning between individual and group activity. A user's ability to externalize working memory and semantically organize information on large displays coupled with the flexibility through which users can interact with data afforded by the exploratory programming paradigm together support *Expressive Analysis*.

3 DESIGN

VisSnippets is designed to allow groups of users to bring diverse datasets and their preferred analysis methods to a collaborative data exploration workspace in which they can quickly scaffold refined interactive visualizations which support their ad-hoc exploratory analysis. The system supports *Mixed-focus Collaboration* through its large display implementation which affords groups "space to think" and allows parallel work streams to coexist. Parallel exploration is welcomed by providing users freedom to choose analytical methods and libraries aligned with their experience and preference. The shared workspace and the transparent dataflow architecture provided by VisSnippets elucidate collaborative analyses and provide data provenance to establish common ground between user viewpoints to facilitate the *Low-cost Integration* of parallel work. The flexibility in collaborative organization in addition to the freedom during implementation which VisSnippets support allows users to perform *Expressive Analysis*.

3.1 Expressive Analysis

The foundation of the VisSnippets system is JavaScript code written in the form of small "snippets." Each snippet implements a simple, modular analysis at the data-pipeline level in the form of a function which utilizes a callback to hand off its result to any subsequent snippets. The VisSnippets workflow (Figure 2) is structured to allow collaborative development to occur in parallel on each user's own device and provides the freedom for each collaborator to utilize their experience to choose their preferred analysis methods and complement them with their choice of toolkits within the large JavaScript ecosystem.

To begin using VisSnippets, a user navigates to the SAGE2 User Interface through their web browser and opens the VisSnippets editor (Figure 3). The editor is designed to simplify development and debugging during exploration. When a user chooses to create a

Figure 4: A set of modular, reusable snippets which, when composed, produce a simple data pipeline. A A *Generator* snippet asynchronously retrieves real-time sensor data from our internal REST API [34], inserting as a URL parameter a chosen sensor's name provided through a *SnippetsInput* API call, creating a text field on the SAGE2 display. *SnippetTimeout* periodically executes this snippet to construct a time-series dataset at an interval of 5s. B A critical *Processing* snippet reshapes data from the API's format into the format required by our visualization toolkit. C A *Visualization* snippet creates a line chart with Plotly [29] using `time` and `temp` values produced by *B*.

snippet of a desired type, the editor provides the user with skeleton code in the basic format of that snippet category. Each type of snippet is similar in implementation, but the role of each within the analytical workflow varies.

When created, a Snippet is defined to be one of three types including data retrieval/creation, processing, and visualization operations and their functionality is as follows:

`Generator` snippets are the root of a data flow tree. They are used to create or fetch datasets to be later used by other snippets, supporting tasks including loading local data from within SAGE2 and performing asynchronous calls to fetch real-time data sources from the cloud (Figure 4a).

`Processing` snippets are incremental transformations of input data which allow users to lead analysis from raw data to clean or restructured data for visualization (Figure 4b).

`Visualization` snippets take data as an input to produce a visualization. These snippets request a specific drawing element to use and can be interspersed throughout the pipeline to produce linked applications which are dynamic and react to user input. (Figure 4c).

Once created, each snippet can be composed one or more times to build complex, branching dataflows to retrieve, manipulate, and visualize data (Figure 5). These pipelines are constructed through direct manipulation from a user's own computer. To compose a snippet onto the existing analysis, a user chooses a snippet to execute from the the editor's Snippet List (Figure 3a) and selects a target onto which the snippet will be added. The dataflow topology constructed by a group is stored centrally in the system's server and used to produce the pipeline meta-visualization (Figure 5a). The system automatically executes the analysis blocks and propagates updates downstream when snippets are modified. The freedom to explore alternative analysis paths by composing multiple snippets anywhere onto an existing pipeline allows users to create complex branching data topologies for exploratory analysis, seen in section 4.

During the development of a traditional web-based linked visualization, there is significant effort required to scaffold the application by coordinating the data source, interface structure and multiple interactive visualizations to support a user's analysis. VisSnippets simplifies for users the task of scaffolding complex applications and provides further abstraction to aid in development and usage. VisSnippets implements and exposes to the user a simple, declarative JavaScript API to aid in analysis. First, this declarative API allows users to quickly and easily control the periodicity of a snippet's execution, for example, to retrieve real-time data at a user-defined interval. Additionally, it allows a user to request a specific output element by type onto which they may visualize data. Lastly, to improve the reusability of code written, we provide to users the functionality to define a basic set of input elements (text, radio buttons, checkboxes, sliders) whose current values will be injected into the snippet when executed (Figure 7c, d).

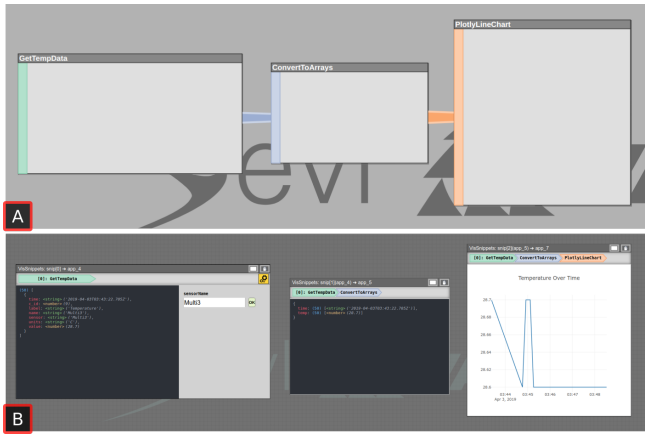

Figure 5: An example of a simple data pipeline constructed from three snippets (Figure 4). It is represented with A a pipeline meta-visualization on the user interface and B the pipeline output displayed on the large display. For each view, the pipeline proceeding the result is displayed across the top of the view on the large display.

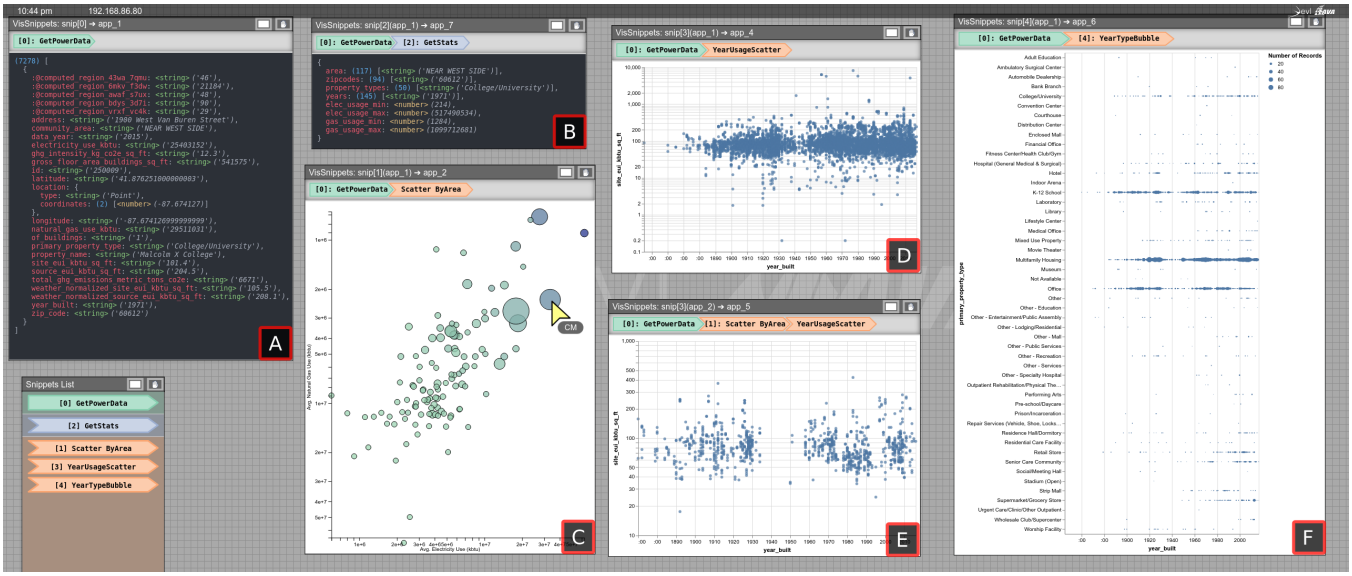

Figure 6: An exploratory analysis of Chicago Energy Benchmarking data. A The initial dataset when fetched from the Chicago Data Portal API. B Additional summary information produced through processing the data further, yielding unique or min and max values for attributes in the initial dataset. C A custom D3 bubble chart plotting mean electricity vs mean natural gas usage per community area. The bubble size and color represent the number of buildings and their average square footage, respectively. Additionally, this view is interactive — a selected bubble will produce a dataset of buildings within that community area. D, E Vega-Lite scatterplots which encode the energy usage per square foot by year, where *D* shows all buildings and *E* shows buildings from the community area selected in *C*. Note that a single snippet is used to create both *D* and *E*, and is used twice in the branching pipeline to support *comparison*. F A Vega-Lite bubble chart encoding the number of buildings constructed per year by property type.

3.2 Mixed-focus Collaboration

The large-display implementation of VisSnippets affords groups "space to think" and allows parallel work streams to coexist within the shared workspace, facilitating *Mixed-focus Collaboration* during exploratory analysis. Execution is performed in the shared workspace to maintain data provenance and facilitate "show and tell." Each user in a collaborative session uses their personal device to write and edit snippets. However, consistent with Mixed-focus Collaboration, VisSnippets supports multiple styles of **collaborative coupling** [39] during group work.

First, users can work completely independently, each exploring separate sub-problems which may or may not originate from the same data source. In our system, the *Different problems* coupling style can be observed when two users are exploring separate scenarios branching off of a common data ancestor. Second, users can work on on the same sub-problems in different areas of the display and/or analytical pipeline. This *Same problem, different area* style can occur when an upstream user is editing snippets or interacting with a visualization while their downstream collaborator is working with the result of their interaction. Lastly, two users can be working together at the same location in the analysis. The *Same problem same area* or *View engaged* style presents itself when multiple collaborators discuss the same stage of the analysis pipeline, interacting with that element of the analysis simultaneously either through the visualization directly or by editing the snippet which produces that output.

To simplify user interaction with and communication regarding the data elements within an analysis, we implement a standalone, open-source, zero-dependency JSON summarization utility[2] used in the VisSnippets editor and on the shared display. The benefits of this utility are twofold: for single-user interaction, this representational efficiency can communicate the data structure and semantics without displaying the entire data object; for collaboration, it reduces the overhead required to negotiate a "data contract" between collaborators — an agreement on a consistent data format connecting two snippets created by different users — by presenting pertinent information about the data directly in the shared workspace.

4 EXAMPLE APPLICATIONS

To demonstrate the expressiveness and flexibility of VisSnippets we present two example applications built within the system. These examples illustrate two very different use-cases of the system at a small scale. For each example, we discuss the functionality implemented, deconstruct the user-defined information topology of the application, and discuss the efficiency of creating and scaffolding such an analysis within VisSnippets in terms of the total Source

[2]The `json-summary` package, examples, and documentation can be found at https://github.com/AndrewTBurks/json-summary

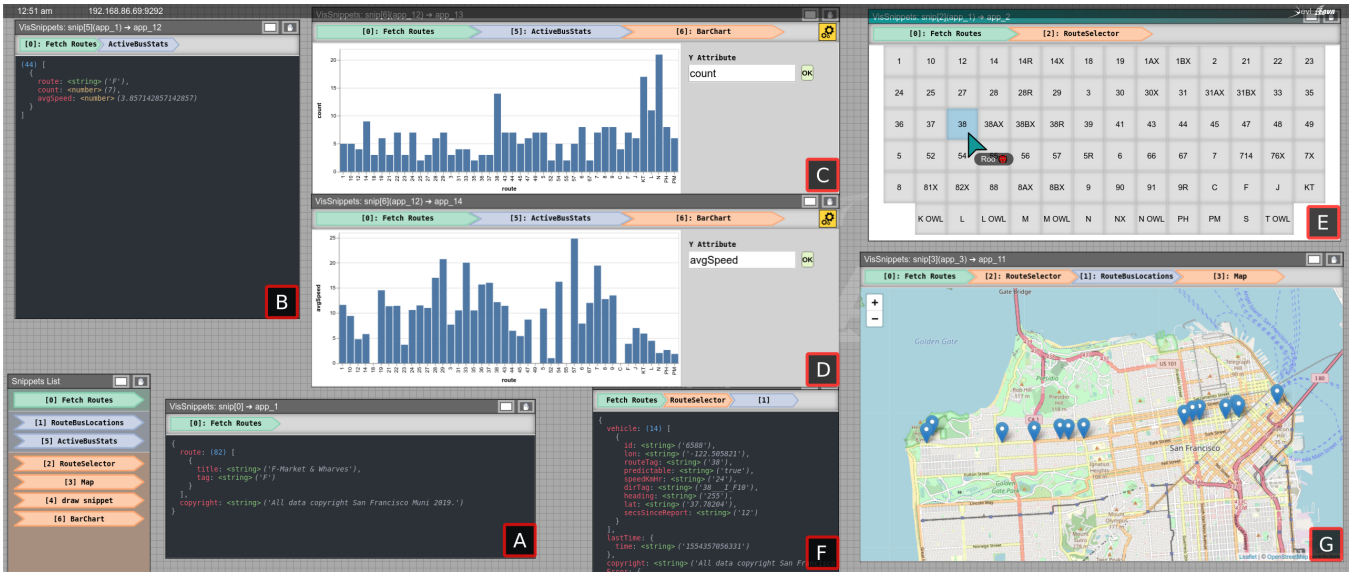

Figure 7: An interactive transportation dashboard for SF Muni bus data. A The list of all bus routes fetched from the NextBus API. B A per-route summary of the number of busses and their average speed for all routes from *A*. This summary periodically retrieves new information every 15s. C, D Vega-Lite bar charts plotting an interactively specified Y-attribute against bus route. The same snippet is used for *C* and *D*, allowing users to enter the desired attribute through the text input. E A route selector interface built in HTML, populated by all routes from *A*. This interactive view produces an object with details of the selected route. F Vehicle locations for all busses, fetched using information provided form the selected route in *E*. These locations for the selected route are retrieved and update periodically every 5s. G A Leaflet map plotting the current location of all busses for the selected route retrieved in *F*.

Lines of Code (SLOC) required from users to produce the linked visualization applications.

4.1 Power Consumption Data Exploration

Our first example application illustrates an exploratory analysis of power consumption data (Fig. 6). The dataset used is provided by the City of Chicago Data Portal[3]. This dataset contains energy consumption information for 1% of the buildings in Chicago which produce an aggregate 20% of the citywide carbon emissions as per the data description. The goal of this example application is to identify trends and outliers in the data and guide analysis in search of the main contributing factors.

The application first retrieves the dataset from the REST API provided. A summary view complements the raw data object representation to help a developer understand the shape and characteristics of the data and its attributes. From here, the custom D3 bubble chart is used to plot average energy consumption for buildings in each community area, also encoding the number of buildings and square footage for context. This bubble chart also allows users to interactively select a community area, updating a linked visualization of energy consumption versus year built. A scatterplot for the selected community area is juxtaposed with an identical representation for all community areas to support comparison. Lastly, the Vega-Lite bubble chart on the right-hand side of the interface provides supplementary information as to the distribution of building types by displaying the count of each type of building constructed by year. Each view takes as input the raw dataset initially fetched from the data portal aside from the scatterplot which is linked to the D3 bubble chart.

When using the application, a user would likely start at the D3 bubble chart. This is an effective representation which allows users to identify outliers as these will be at the top-right corner of the view due to an above-average electricity and natural gas consumption. From there, a user will select that community area and view the distribution of power usage across buildings in comparison to the rest of the city. In these two views, plotting the year constructed can point to changes in the energy efficiency of newer structures, as well.

To produce this application using the VisSnippets system, the group would need to write a total of ~170 SLOC including comments and whitespace, with fewer required Logical Lines of Code (LLOC). The majority of the SLOC (104 of ~170) were written to produce the custom D3 Bubble Chart (Fig. 6c).

4.2 Interactive Transportation Dashboard

The second example application illustrates an interactive transportation data dashboard (Fig. 7). The dataset showcased is provided by the San Francisco Municipal Transportation Agency through

[3]Chicago Energy Benchmarking dataset at https://data.cityofchicago.org/Environment-Sustainable-Development/Chicago-Energy-Benchmarking/xq83-jr8c

NextBus[4]. The data used in this application is retrieved from one NextBus API to retrieve all route information, and a second which provides all busses for a queried route. The goal of this application is to provide a real-time dashboard which would allow users to monitor their fleet of busses and to identify and address potential issues. Detailed information about the functionality and usage of each snippet can be found in Figure 7.

When using the application to manage a fleet of busses, the user may start by identifying a route of interest using the two summary bar charts. These allow users to identify anomalous behavior such as a decreased number of vehicles in service or an abnormally low speed along a route. For further investigation, the user may select this route to populate the map of bus locations. The bus locations may show that many busses are caught in a high-traffic area but could alternatively alert the user to an unforeseen event such as a car accident.

To produce this interactive application within the VisSnippets system, each user can each work to their own strengths. For example, a first user familiar with the data API can write the snippets to handle the fetching operations, a second user proficient in VegaLite can produce the bar charts, and a third experienced user of Leaflet can create the map. Through the abstraction provided by VisSnippets, its API, and its compatibility with external visualization toolkits, this interactive application is created using a mere 145 Source Lines of Code (SLOC), including comments and whitespace, with the required Logical Lines of Code (LLOC) being even fewer.

5 DISCUSSION

Through the two illustrative example applications, we see the diversity of analyses supported by VisSnippets. In the first example, Power Consumption Data Exploration, we see an exploratory analysis which attempts to identify patterns in Energy Benchmarking data across building age, classification, and community area. In the second example, Interactive Transportation Dashboard, we see a real-time dashboard built to help monitor a bus fleet's location and status. The applications begin with two contrasting datasets, one in a tabular form and the other a unique JSON structure defined by the specific APIs. In both applications, the design of VisSnippets allowed differing datasets to become manageable to a user. Between applications, the processing and reshaping operations on the data were also very unique to a specific dataset. Built-in processing functionality would likely not be able to manage the SF Muni bus dataset due to its unique structure.

Since one application was focused on the task of data exploration while the other is an interactive dashboard visualization, the tasks were decidedly different. Again, the flexibility of the system supported both. Underscoring the power of the system and abstraction through which complex, interactive dataflows can be created, these two contrasting applications were implemented in a total of ~315 SLOC.

5.1 Limitations

While VisSnippets provides abstraction to streamline development, there remains a requirement to have a visualization or data analysis expert user present to support most collaborative analyses. This is true of the data acquisition, data processing, and view specification phases. VisSnippets provides a limited set of built-in snippets for basic operations like fetching, filtering/aggregating, and visualizing. The branching capability of VisSnippets is helpful for exploratory and subjunctive analysis. However, this branching only supports juxtaposition and basic faceting through child views downstream in the data pipeline. VisSnippets can be improved in terms of the flexibility of data movement through the pipelines by supporting a merging operation to join together results. Lastly, while the design of VisSnippets is motivated by background research towards supporting *Low-cost Integration* and *Mixed-focus Collaboration*, we have not yet performed an extensive study of user behavior to support the efficacy of the design instantiation in these areas.

6 CONCLUSION AND FUTURE WORK

In this paper, we presented our design for a collaborative data exploration system and our prototype implementation of the VisSnippets system in SAGE2[5]. We discussed our implementation which provides general-purpose functionality through a low-level JavaScript-based programming interface. Abstracting the runtime of the visualization data pipelines by introducing reactive and periodic updates removes the need to scaffold visualization applications by hand. Additionally, declarative specification of drawing elements and parametric inputs allow users to easily build generalized and reusable snippets after which users can export their collaborative session as a standalone web page. We demonstrated the flexibility and versatility of the system by outlining two example applications and discussed the implementation of each to support its respective analytical workflow.

For the future of VisSnippets, we plan to address the limitations of the system by first extending the functionality of the system to increase the speed and simplicity of snippet and pipeline creation and also support analysis workflows which involve merging multiple data flows as input to a single snippet. Additionally, in the context of the Model for Ubiquitous Analytics [3], VisSnippets could benefit from further support for diverse workflows across users of varying expertise and display devices. Lastly, we would like to study how users perform collaborative data exploration tasks in a large-display environment. The users' "collaborative" distance, measured by the user's position based on content location on the large display and content relation attained by analyzing the dataflow topology, can then be used to better understand how display position corresponds to work relatedness at different states of collaborative coupling during *Mixed-focus Collaboration*.

ACKNOWLEDGMENTS

The authors thank their collaborators at the Electronic Visualization Laboratory and the Laboratory for Advanced Visualizaton and Applications. This work was supported in part by NSF award ACI-1441963.

[4]The San Francisco (SF) Muni NextBus dataset is found at http://www.nextbus.com/xmlFeedDocs/NextBusXMLFeed.pdf

[5]A video demonstrating the use of the VisSnippets can be seen at https://www.youtube.com/watch?v=8fC6FwrD20g

REFERENCES

[1] Christopher Andrews, Alex Endert, and Chris North. 2010. Space to think: large high-resolution displays for sensemaking. In *Proc. CHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM Press, Atlanta, Georgia, USA, 55. https://doi.org/10.1145/1753326.1753336

[2] Christopher Andrews, Alex Endert, Beth Yost, and Chris North. 2011. Information visualization on large, high-resolution displays: Issues, challenges, and opportunities. *Information Visualization* 10, 4 (2011), 341–355.

[3] Sriram Karthik Badam, Andreas Mathisen, Roman Rädle, Clemens N Klokmose, and Niklas Elmqvist. 2019. Vistrates: A Component Model for Ubiquitous Analytics. *IEEE Trans. Visualization and Computer Graphics* 25, 1 (2019), 586–596.

[4] Robert Ball and Chris North. 2005. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *Extended Abstracts on Human Factors in Computing Systems (CHI '05)*. ACM Press, Portland, OR, USA, 1196. https://doi.org/10.1145/1056808.1056875

[5] M. Bostock and J. Heer. 2009. Protovis: A Graphical Toolkit for Visualization. *IEEE Trans. Visualization and Computer Graphics* 15, 6 (Nov. 2009), 1121–1128. https://doi.org/10.1109/TVCG.2009.174

[6] M. Bostock, V. Ogievetsky, and J. Heer. 2011. D^3 Data-Driven Documents. *IEEE Trans. Visualization and Computer Graphics* 17, 12 (Dec. 2011), 2301–2309. https://doi.org/10.1109/TVCG.2011.185

[7] Susan Brennan, Klaus Mueller, Greg Zelinsky, Iv Ramakrishnan, David Warren, and Arie Kaufman. 2006. Toward a Multi-Analyst, Collaborative Framework for Visual Analytics. In *2006 IEEE Symposium On Visual Analytics And Technology*. IEEE, Baltimore, MD, USA, 129–136. https://doi.org/10.1109/VAST.2006.261439

[8] Joohee Choi and Yla Tausczik. 2017. Characteristics of Collaboration in the Emerging Practice of Open Data Analysis. In *In Proc. ACM Conference on Computer Supported Cooperative Work and Social Computing - CSCW'17*. ACM Press, Portland, Oregon, USA, 835–846. https://doi.org/10.1145/2998181.2998265

[9] Google Colaboratory. April 2019. https://colab.research.google.com/.

[10] Robert DeLine and Danyel Fisher. 2015. Supporting exploratory data analysis with live programming. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, Atlanta, GA, 111–119. https://doi.org/10.1109/VLHCC.2015.7357205

[11] Carl Gutwin and Saul Greenberg. 2002. A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work (CSCW)* 11, 3-4 (2002), 411–446.

[12] Jeffrey Heer and Maneesh Agrawala. 2008. Design considerations for collaborative visual analytics. *Information visualization* 7, 1 (2008), 49–62.

[13] Anthony JG Hey, Stewart Tansley, Kristin M Tolle, et al. 2009. *The fourth paradigm: data-intensive scientific discovery*. Vol. 1. Microsoft research Redmond, WA.

[14] Petra Isenberg, Niklas Elmqvist, Jean Scholtz, Daniel Cernea, Kwan-Liu Ma, and Hans Hagen. 2011. Collaborative visualization: Definition, challenges, and research agenda. *Information Visualization* 10, 4 (Oct. 2011), 310–326. https://doi.org/10.1177/1473871611412817

[15] Petra Isenberg, Danyel Fisher, Meredith Ringel Morris, Kori Inkpen, and Mary Czerwinski. 2010. An exploratory study of co-located collaborative visual analytics around a tabletop display. In *2010 IEEE Symposium on Visual Analytics Science and Technology*. IEEE, Salt Lake City, UT, USA, 179–186. https://doi.org/10.1109/VAST.2010.5652880

[16] Dong Jeong, Soo-Yeon Ji, Evan A Suma, Byunggu Yu, and Remco Chang. 2015. Designing a Collaborative Visual Analytics System to Support Users' Continuous Analytical Processes. *Human-centric Computing and Information Sciences* 5, 1 (2015), 5. https://doi.org/10.1186/s13673-015-0023-4

[17] Gregory P. Johnson, Gregory D. Abram, Brandt Westing, Paul Navr'til, and Kelly Gaither. 2012. DisplayCluster: An Interactive Visualization Environment for Tiled Displays. In *2012 IEEE International Conference on Cluster Computing*. IEEE, Beijing, China, 239–247. https://doi.org/10.1109/CLUSTER.2012.78

[18] Jupyter. April 2019. http://jupyter.org/.

[19] Mary Beth Kery, Amber Horvath, and Brad Myers. 2017. Variolite: Supporting Exploratory Programming by Data Scientists. In *Proc. CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM Press, Denver, Colorado, USA, 1265–1276. https://doi.org/10.1145/3025453.3025626

[20] Clemens N Klokmose, James R Eagan, Siemen Baader, Wendy Mackay, and Michel Beaudouin-Lafon. 2015. Webstrates: shareable dynamic media. In *In Proc. ACM Symposium on User Interface Software & Technology*. ACM, 280–290.

[21] Ricardo Langner, Ulrike Kister, and Raimund Dachselt. 2018. Multiple Coordinated Views at Large Displays for Multiple Users: Empirical Findings on User Behavior, Movements, and Distances. *IEEE Trans. Visualization and Computer Graphics* (2018), 1–1. https://doi.org/10.1109/TVCG.2018.2865235

[22] Jason Leigh, Mahdi Belcaid, Dylan Kobayashi, Nurit Kirshenbaum, Troy Wooton, Alberto Gonzalez, Luc Renambot, Andrew Johnson, Maxine Brown, Andrew Burks, et al. 2019. Usage Patterns of Wideband Display Environments In e-Science Research, Development and Training. *eScience 2019* (2019).

[23] Jason Leigh, Andrew Johnson, Luc Renambot, Tom Peterka, Byungil Jeong, Daniel J. Sandin, Jonas Talandis, Ratko Jagodic, Sungwon Nam, Hyejung Hur, and Yiwen Sun. 2013. Scalable Resolution Display Walls. *Proc. of the IEEE* 101, 1 (Jan. 2013), 115–129. https://doi.org/10.1109/JPROC.2012.2191609

[24] Zhicheng Liu, John Thompson, Alan Wilson, Mira Dontcheva, James Delorey, Sam Grigg, Bernard Kerr, and John Stasko. 2018. Data Illustrator: Augmenting Vector Design Tools with Lazy Data Binding for Expressive Visualization Authoring. In *Proc. CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM Press, Montreal QC, Canada, 1–13. https://doi.org/10.1145/3173574.3173697

[25] Aran Lunzer and Kasper Hornbæk. 2008. Subjunctive interfaces: Extending applications to support parallel setup, viewing and control of alternative scenarios. *ACM Trans. Computer-Human Interaction (TOCHI)* 14, 4 (2008), 17.

[26] Thomas Marrinan, Jillian Aurisano, Arthur Nishimoto, Krishna Bharadwaj, Victor Mateevitsi, Luc Renambot, Lance Long, Andrew Johnson, and Jason Leigh. 2014. SAGE2: A New Approach for Data Intensive Collaboration Using Scalable Resolution Shared Displays. In *In Proc. IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*. ICST, Miami, United States. https://doi.org/10.4108/icst.collaboratecom.2014.257337

[27] Dominik Moritz, Chenglong Wang, Greg L Nelson, Halden Lin, Adam M Smith, Bill Howe, and Jeffrey Heer. 2019. Formalizing Visualization Design Knowledge as Constraints: Actionable and Extensible Models in Draco. *IEEE Trans. Visualization and Computer Graphics* 25, 1 (2019), 438–448.

[28] Observable. April 2019. https://observablehq.com/.

[29] Plotly. April 2019. https://plot.ly/.

[30] Roman Rädle, Midas Nouwens, Kristian Antonsen, James R Eagan, and Clemens N Klokmose. 2017. Codestrates: Literate computing with webstrates. In *In Proc. ACM Symposium on User Interface Software & Technology*. ACM, 715–725.

[31] Khairi Reda, Andrew E Johnson, Michael E Papka, and Jason Leigh. 2015. Effects of Display Size and Resolution on User Behavior and Insight Acquisition in Visual Exploration. In *Proc. CHI Conference on Human Factors in Computing Systems (CHI '15)*. ACM, 2759–2768.

[32] Donghao Ren, Bongshin Lee, and Matthew Brehmer. 2019. Charticulator: Interactive Construction of Bespoke Chart Layouts. *IEEE Trans. Visualization and Computer Graphics* 25, 1 (2019), 789–799.

[33] Luc Renambot, Arun Rao, Rajvikram Singh, Byungil Jeong, Naveen Krishnaprasad, Venkatram Vishwanath, Vaidya Chandrasekhar, Nicholas Schwarz, Allan Spale, Charles Zhang, et al. 2004. Sage: The Scalable Adaptive Graphics Environment. In *Proc. of WACE*, Vol. 9. Citeseer, 2004–09.

[34] Leonard Richardson and Sam Ruby. 2008. *RESTful web services*. " O'Reilly Media, Inc.".

[35] Adam Rule, Aurélien Tabard, and James D. Hollan. 2018. Exploration and Explanation in Computational Notebooks. In *In Proc. CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM Press, Montreal QC, Canada, 1–12. https://doi.org/10.1145/3173574.3173606

[36] Arvind Satyanarayan and Jeffrey Heer. 2014. Lyra: An Interactive Visualization Design Environment. *Computer Graphics Forum (Proc. EuroVis)* (2014). http://idl.cs.washington.edu/papers/lyra

[37] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Trans. Visualization and Computer Graphics* 23, 1 (Jan. 2017), 341–350. https://doi.org/10.1109/TVCG.2016.2599030

[38] Chris Stolte, Diane Tang, and Pat Hanrahan. 2002. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Trans. Visualization and Computer Graphics* 8, 1 (2002), 52–65.

[39] Anthony Tang, Melanie Tory, Barry Po, Petra Neumann, and Sheelagh Carpendale. 2006. Collaborative Coupling over Tabletop Displays. In *Proc. CHI Conference on Human Factors in Computing Systems*. ACM, 1181–1190.

[40] James J. Thomas and Kristin A. Cook. 2006. A visual analytics agenda. *IEEE Computer Graphics and Applications* 26 (2006), 10–13.

[41] Matthew Tobiasz, Petra Isenberg, and Sheelagh Carpendale. 2009. Lark: coordinating co-located collaboration with information visualization. In *IEEE Trans. Visualization and Computer Graphics*. 1065–1072.

[42] Vega. April 2019. https://vega.github.io/vega/.

[43] John Wenskovitch, Jian Zhao, Scott Carter, Matthew Cooper, and Chris North. 2019. Albireo: An Interactive Tool for Visually Summarizing Computational Notebook Structure. In *2019 IEEE Visualization in Data Science (VDS)*. IEEE, 1–10.

[44] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Trans. Visualization and Computer Graphics* 22, 1 (Jan. 2016), 649–658. https://doi.org/10.1109/TVCG.2015.2467191

[45] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2017. Voyager 2: Augmenting Visual Analysis with Partial View Specifications. In *Proc. CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM Press, Denver, Colorado, USA, 2648–2659. https://doi.org/10.1145/3025453.3025768