Gravity++: A Graph-based Framework for Constructing Interactive Visualization Narratives

Humphrey O. Obie[a,*], Dac Thanh Chuong Ho[b,*], Iman Avazpour[c,*], John Grundy[a,*], Mohamed Abdelrazek[d,*], Tomasz Bednarz[e,f,*], Caslon Chua[b]

[a]*Monash University, Melbourne, Australia*
[b]*Swinburne University of Technology, Melbourne, Australia*
[c]*Enabling Technology Solutions, Melbourne, Australia*
[d]*Deakin University, Geelong, Australia*
[e]*CSIRO's Data61, Sydney, Australia*
[f]*University of New South Wales, Sydney, Australia*

Abstract

Interactive visualizations play a key part in the exploration and analysis of data, and in the creation of visual data stories. This paper describes a new graph-based framework for developing interactive visualizations for creating coherent visual data stories. We have realised our framework in a prototype tool named Gravity++. Gravity++ uses a novel graph architecture for modelling interaction, data navigation, and changes in visual representation to better communicate findings to an audience. The combination of these graph models provides better support and flexibility for designing interactive visualizations, data navigation, and visual data stories. We demonstrate the applicability of this framework by two example usage scenarios. We also report on an evaluation study conducted with representative participants. All participants successfully created meaningful visual data stories with a high level of complexity. Our results also show that Gravity++ is easy to use and supports the understanding and sense-making of the visual data story creation process.

Keywords: visualization, visualization systems, data storytelling, authoring tools

1. Introduction

With the ever-increasing importance of data in today's digital world, the software and information technology industries are growing more aware of the

*Corresponding author
Email addresses: humphrey.obie@monash.edu (Humphrey O. Obie), 101921623@student.swin.edu.au (Dac Thanh Chuong Ho), i.avazpour@gmail.com (Iman Avazpour), john.grundy@monash.edu (John Grundy), mohamed.abdelrazek@deakin.edu.au (Mohamed Abdelrazek), t.bednarz@unsw.edu.au (Tomasz Bednarz), cchua@swin.edu.au (Caslon Chua)

challenges of effectively using big data and information analysis for critical decision-making tasks. Consequently, the role of information visualization as a key part of complex data analysis and comprehension has become evident [1]. Presenting complex information for use in a visual format is more acceptable and effective for human beings in many circumstances, as visual representations use fuller capabilities of our powerful human visual system [2, 3, 4]. Visualizations therefore help users to explore and understand data through progressive and iterative interaction [5] and provide effective platform for decision makers [6].

Most current information visualization techniques follow the traditional visualization pipeline [7, 8], i.e., the visualization process starts by selecting a set of specific data; data transformations are then used to generate data tables; these data tables will be mapped to visual structures; finally one or more views will be generated from these visual structures. Using this framework, altering visualizations is arduous; users must reconstruct them from scratch. Consequently, new approaches are emerging to better incorporate user interactions with visualizations to support the creation of understandable visual narratives.

This paper presents a graph-based framework for generating flexible interactive visualizations to enable alternative data representations, support for new interaction paradigms, support for integration of newly acquired data into the visualizations, and support for the creation of coherent visual data stories. Our novel framework can be used stand-alone or as a backbone for current interactive visualization design systems. Our framework can help designers of visualizations utilize different interaction capabilities with a variety of complex visualizations and multiple data sources, and the sequencing of these visualizations into visual data stories to better communicate findings to their audience.

Although our graph-based framework can provide support for both the exploratory and communicative goals of information visualization, our focus in this work is on the *communicative* aspects of the framework. We implemented our framework in a stand-alone prototype tool named *Gravity++*. To evaluate our framework, we conducted an evaluation of Gravity++ with nine representative participants. All participants successfully created meaningful visual data stories with a high level of complexity. Our results also show that Gravity++ is easy to use and supports the visual data story creation process.

To summarise, our contributions are fourfold: (1) extensible plug and play of data sources and visualizations in a visual storytelling scenario; (2) efficient navigation and manipulation of data through various kinds of user interactions with the visualizations; (3) control of traceability and links between data and visualizations for multiple interaction and data manipulation scenarios; (4) meta-visualizations of the data, interaction, and visualization aspects of a visual narrative.

The rest of this paper is organized as follows. Section 2 provides a motivating scenario. Section 3 provides a brief background on related work. Sections 4 and 5 provide details of our architecture and interface design, followed by two usage scenarios in section 6, and a brief discussion of the implementation of the Gravity++ prototype in 7. Section 8 discusses our evaluation with representative users and section 9 provides a discussion and areas for future work. We conclude

the paper in section 10.

2. Motivating Scenario

Consider Wendy, a public health expert who wants to investigate the Covid-19 pandemic in Southeast Asian countries and the corresponding government responses. She has access to various related datasets, however, she does not have programming and data visualization experience. She instead asks Bob, who is a data expert with experience in data visualization to help her design a visual data story.

Wendy describes her process steps as follows: she starts by exploring the first dataset, which contains the total cumulative cases, new confirmed cases, total cumulative deaths, new deaths and the number of tests conducted in each day in each Southeast Asian country. Bob and Wendy both decide to use a line chart to compare the total case trajectories between countries over time. However, they also consider using a choropleth map which shows the total cases of each country by the color intensity.

Now, Wendy wants to investigate whether the number of covid tests is effective in revealing new cases in each country. She hypothesised that the more tests conducted, the higher number of new cases is revealed. She wants to choose an appropriate chart to verify her hypothesis. Bob decides to plot a line chart with two y axes, one for the number of new cases and another for the number of new tests conducted over time for each country.

Lastly, Wendy wants to use the governmental policy in response to new outbreaks, and aims to see the relationship between the governance strictness and the total number of covid cases. Bob decides to plot another scatterplot chart to display both the stringency index and the number of new cases. To make the comparison clear, Wendy suggests putting another pie chart on the side which shows the proportion of total cases in all Southeast Asian countries.

Wendy's description of the example scenario provides a typical case of "visual storytelling". Bob's contribution is to develop this visual storytelling scenario using a series of interactive visualizations to help Wendy perform her analysis and investigation. However, with current tools and approaches, generating arbitrary interactive visualizations for visual narratives is cumbersome and the developed visualizations may not be flexible enough. Implementing interactive visualization using programming codes, although very flexible, requires in-depth scripting knowledge.

The above is a typical use case in visual data storytelling, and Gravity++ and its underlying architecture aims to support these and other use cases.

3. Related Work

3.1. Information Visualization Architecture Approaches

The increasing importance of "big data" analysis and comprehension has highlighted the many benefits of incorporating powerful information visualizations. Visualizations help data users better understand and make sense of the

data [5, 9, 10]. However, most accepted and adopted visualization procedures today suffer from a classic problem: Certain characteristics of data are only revealed when it is visualized. Once these characteristics are revealed, to gain insight and analyze them, new more effective visualizations need to be developed from scratch. This has led to wide acceptance of the Visual Information Seeking Mantra proposed by Shneiderman [11]: Overview first, zoom and filter, then details on demand [12, 13]. Accordingly, to fully understand the data, it should be investigated through multiple angles and therefore multiple visualizations need to be developed.

A current approach for enabling visual information seeking is to incorporate multiple coordinated views to view the visualizations and the data from multiple perspectives [9]. Consider for example, multiple visualizations being demonstrated side by side on a visualization dashboard [14]. An example of such an approach is visualizations generated by tools such as Improvise [15] and Voyager [16]. Each perspective in this approach forms a view and the combinations of these views are then used side by side on a large display and therefore form a visualization dashboard. Interaction can also be coordinated on different views to some extent. For example, selecting an item in one view can trigger highlighting similar items in other views (brushing and linking). This approach can however only incorporate a limited number of views and requires an orchestrated attention to the views by the users. Too many views can clutter the display and reduce concentration of the users [9]. Views must also be defined before use, limiting exploratory visualization.

Different approaches have been proposed to help users cover a wide range of data fields and visualization views. For example, Voyager 2, a mixed-initiative system, employs manual and automated visualization view specification to help analysts engage in both open-ended exploration and targeted question answering [17].

Recently, using a meta-visualization approach, Obie et al. consolidated the different phases of the visual narrative creation process in a framework and prototype system named Gravity, which recommends and visualizes logically sequenced sets of visualization views to support the presentation of coherent visual narratives and data stories [18, 19]. Although both Gravity and Gravity++ fulfil the exploratory and communicative goals of information visualization, a major difference between Gravity and our novel Gravity++ is their underlying architectures and related capabilities and limitations. While Gravity supports the creation of logically sequenced visual data stories, it is based on the traditional visualization architecture which maps tabular/column-oriented data stores to visual structures and hence limiting it to the creation of **linear** visual data stories. In contrast, Gravity++ makes a fundamental shift into a complete graph-based architecture, supporting both **linear** and **non-linear** visual data stories. The Gravity++'s graph architecture enables custom interaction modes for visual data storytelling. The graph architecture underlying Gravity++ also has the potential to support data query prediction and traversal, and potentially serve as a metamodel for other data models.

Moreover, data cubes have been used to provide better navigation of the

data for visualization by capturing all possible aggregations. For example, Stolte et al. proposed to use data cubes to capture possible different aggregation levels of the data into a lattice [20]. As a result, zoom and pan interactions could be supported on visualizations and would trigger traversal of the data cube lattice. For example, if users zoom in on the visualization, their system would explore the data cube lattice and pass the required data to be represented by the visualization. Nanocubes extend data cubes by capturing, storing and querying spatio-temporal Role Up Cubes [21]. They provide better handling of large data sets for devices with modest memory such as a laptop computer.

Heer et al. proposed an approach for querying data by directly selecting visual elements in a visualization [22]. Users could select a set of visual representations (for example, markings on a scatter plot) and based on the selection, the system would generate a query to fetch data similar to the selected representations. Data cubes have also been shown to be effective, when queries to be prepared for interactive visualizations are incremental [20]. This way, new queries will be applied on the results of the previous query. This approach is used in Crossfilter [23]. Crossfilter's approach assumes that the data navigation space forms a tree structure. Our framework expands these concept by enabling queries to be executed in different modes and directions. For example, multiple data sources may become available at different stages of the visualization and interaction, i.e. a node in an assumed data navigation space can have multiple parents. Hence our framework is designed to support a graph-based data and visualization navigation space.

Furthermore, multilayer networks have been applied to the visualization of complex phenomena in various fields [24, 25] including life sciences [26, 27], sociology [28, 29], digital humanities [30, 31], and others [25]. Similar to other information visualization studies [32, 33, 34, 35, 36], Gravity++ also adopts the concept of a multi-layer graph. However, Gravity++'s approach to multilayer networks is the meta-visualization of the underlying graph architecture rather than the visual encoding of data, as data within Gravity++ is encoded using common visualization techniques.

3.2. Visualization Provenance

Visualization provenance is a body of work that Gravity++ leverages in the implementation of its underlying graph architecture. There have been several techniques and tools that have contributed to the understanding and utilization of visualization provenance [37]. Below we highlight the most relevant to our work.

Heer et al. proposed approaches for visualization provenance [38]. Their proposed guide provides methods for designing history interfaces to support the exploration and communication of information visualization. Dunne et al. introduced GraphTrail, an interactive visualization system for analysing large multivariate and heterogeneous networks while also displaying users' exploration history [39]. GraphTrail supports users in recalling the findings from their analysis and the exploration process that led them to arrive at those findings.

Similarly, CLUE is a model for capturing, labelling, understanding, and explaining provenance information of data-driven visual explorations [40]. CLUE allows users to create visualizations based on their data exploration history.

₁₈₅ With the aim of supporting the scientific discovery process and streamlining the visualization process, Callahan et al proposed VisTrails [41]. VisTrails manages the data and metadata of visualization products, and captures the provenance of visualization processes and the data they manipulate. Similarly, Bors, Gschwandtner, and Miksch [42] introduced DQProv Explorer, a system ₁₉₀ for capturing and visualising provenance from data wrangling operations.

Using a meta-visualization approach, Chart Constellation aids the meta-visualization and summarization of the results of prior analysts who have created data stories about a shared dataset [43]. The Chart Constellation system provides methods to help users derive insights about past data investigations and the ₁₉₅ exploration of new areas in a dataset.

InsideInsight is a tool that supports the organization of insights into information hierarchies linked to analytic provenance and analysis views, with the overall goal of bridging the gap between data analysis and reporting [44]. Segmentifier implements an interface for the interactive refinement of clickstream data [45]. ₂₀₀ The Segmentifier tool aims to support analysts in understanding the complexity of e-commerce consumer behaviour and features a glyph-based visual history of the recorded analysis process showing the provenance of action sequences organised into segments. Dabek and Caban applied a directed acyclic graph to model user interactions and provide suggestions to users during their data ₂₀₅ exploration session based on learned patterns from past usage histories [46].

Other studies such as [47, 48, 49, 50, 51] have also introduced approaches for tracking and maintaining history graphs with the aim of supporting users' analysis process.

As these meta-visualization and provenance histories have been shown to be ₂₁₀ effective in supporting visual understanding, thus Gravity++ also assumes this approach in our work.

3.3. Visual Data Story Authoring Tools

Segel and Heer's narrative visualization framework categorises visual data stories into 7 genres: magazine style, annotated charts, infographics, flow charts, ₂₁₅ comic strips, slideshows, and data videos, and different tools and techniques address one or more of these genres [52]. Moreover, Gravity++ addresses the slideshow and data video genre of the narrative visualizations framework. Below we discuss relevant visual data storytelling authoring tools.

Different studies have introduced techniques and tools for authoring visual ₂₂₀ data stories. Eccles et al. proposed GeoTime, a tool for integrating story narration with geotemporal patterns, with the aim of supporting spatiotemporal data storytelling [53]. GeoTime provides a mechanism for linking probable explanations to events in time and space. While GeoTime is specialised for spatiotemporal visual data stories, Gravity++ (with its underlying graph archi-₂₂₅ tecture) is a more generic system for authoring visual data stories using data that can be visualised with common visualization techniques.

6

Satyanarayan and Heer introduced a model of visual data storytelling that includes scene structures, dynamic annotations, and decoupled coordination of visualization components [54]. This model is realized in a prototype system named, Ellipsis. The Ellipsis system supports the import of already created visualizations and the use of textual and graphical annotations to create multiple story scenes [54]. Our Gravity++ consolidates the three processes of visual data storytelling creation: the creation of interactive visualizations, manual and automated sequencing of visualizations, and the presentation of the logically sequenced visualizations for the creation of coherent visual data stories.

Other studies have leveraged the medium of comics for visual data storytelling. For example, Kim et al. proposed DataToon, a data comic storyboarding tool that blends analysis and presentation of networks with pen and touch interaction [55]. Similarly, Zhao, Marr, and Elmqvist introduced Data Comics, a tool for capturing and editing online data visualizations to tell visual data stories using comic layouts of panels [56]. While both DataToon and Data Comics focus on visual data stories in the comic strip genre of the narrative visualization framework [52], Gravity++ addresses the slide show and data video genres.

DataClips is a tool aimed to lower the barriers in creating data videos by allowing non-experts to assemble data-driven clips to form longer sequences [57]. DataClips focuses on pictographs and has a large collection of clips to support the creation of data videos. However, it does not provide mechanisms for the creation of logical visualization sequences - a crucial feature for the creation of coherent visual data stories.

To support the creation of interactive articles, Conlen et al. introduced Idyll Studio, a structured editor for creating data-driven articles [58]. Idyll Studio supports the reification of text, components, state, and styles in its reactive document model. However, while Idyll Studio primarily targets the magazine style genre of the narrative visualization framework [52], Gravity++ focuses on different genres altogether - the slide show and data video genres of the framework.

Furthermore, the optimal sequence of visualization recommendation implemented in Gravity++ supports the slideshow and data video genres and consequently the visual data stories created by analysts; and thus lowers the cognitive effort required by the target audience to understand the created visual data story.

Table 1 summarises the similarities and differences between Gravity++ and existing tools and techniques based on key areas in [59, 52, 19].

4. The Gravity++ Approach

4.1. Design Methodology

Task descriptions and design requirements are typically used to guide the implementation of tools [43, 19]. In information visualization, several tasks taxonomies and classification systems exist as resources for analysing task requirements for visualization systems [60, 12]. For instance, Heer and Schneiderman

7

Table 1: Summary of key differences between Gravity++ and existing tools and techniques.*Ellipsis [54] only supports the import of already created visualizations and it does not handle data, hence it is not applicable (NA).

Parameters	GeoTime [53]	Ellipsis [54]	DataToon [55]	DataClips [57]	DataComics [56]	Idyll [58]	Gravity [19]	Gravity++
Visualization scope	Spatiotemporal	General	Network	General	General	General	General	General
Storytelling genre	Annotated chart	Slideshow	Comic strip	Data video	Comic strip	Magazine style	Slideshow, data video	Slideshow, data video
Audience participation level	High	High	High	Low	High	High	Low	Low
Visualization authoring support	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Creation of manual sequences	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Recommendation of effective sequences	No	No	No	No	No	No	Yes	Yes
Creation of insight notes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Graphical annotation	Yes	Yes	Yes	Yes	Yes	No	No	No
Creation of slideshows	No	Yes	No	No	No	Yes	Yes	Yes
Narration and recording of presentations	No	No	No	No	No	No	Yes	Yes
Collaboration support	Yes	Yes	No	No	No	No	Yes	Yes
Custom interaction modes	No	No	No	No	No	No	No	Yes
Narrative structure	Linear, nonlinear	Linear, nonlinear	Linear	Linear	Linear	Linear	Linear	Linear, nonlinear
Data model	Tabular	NA*	Tabular	Tabular	Tabular	Tabular	Tabular	Graph-based

8

[12] proposed a taxonomy of interactive dynamics for visual analysis; Tory and Moller [61] introduced a general high-level taxonomy for visualization algorithms; while Schulz et al. [62] presented a general design space of visualization tasks covering the main dimensions of task descriptions. Other taxonomies focus on specific visualization subfields, e.g., cartography [63], graph visualization [64], spatiotemporal data [65], network evolution [66], biological pathways [67].

Above all these, this paper adopts the Brehmer and Munzner [60] multi-level typology of abstract visualization tasks to identify the high-level tasks that our approach should satisfy. The rationales behind our choice of this typology over others are as follows; Brehmer and Munzner's typology offers abstract rather than domain-specific description of tasks. This distinctive characteristic supports useful comparisons between visualization systems targeted at either the same or different application domains; also, it aids the analysis necessary for the creation of new visual designs or system features, by informing the conversion of domain-specific problems into abstract tasks, and the qualitative evaluation of visualization (or tool) usage.

To specify the task descriptions for the system, we captured the domain-specific (i.e., visual data storytelling) requirements principally from the narrative visualization literature, e.g., Lee et al. [59], Kosara and Mackinlay [68], Segel and Heer [52], Chevalier et al. [69], Obie et al. [19] and also from the authors' experience in authoring visual data stories and tools, e,g, [70, 4, 19]. The abstract task analysis used in Gravity++ is also partly based on a *condensed* task analysis description detailed in our prior work (c.f. Gravity [19]). We highlight these tasks descriptions in the following subsection. We note that an extended discussion of the multi-level typology of abstract visualization tasks is beyond the scope of this paper (see [60]).

4.2. Task Analysis

4.2.1. Task 1: Prepare datasets and process data

Gravity++ should support **importing** raw datasets and provide the user the graphical interface to **explore** the dataset content as well as functionalities to **derive** an existing dataset into another. Users would be able to **select** to view both raw and derived datasets, and **record** (save) the datasets to be used later in the visualization domains.

4.2.2. Task 2: Create visualizations

The next task involves **producing** the charts or visualizations based on the data produced or imported from task 1. The process includes **selecting** the desired dataset, and selecting appropriate **encodings** to visualize the data. Gravity++ should provide the opportunity for users to customize their charts by providing their own **annotations** via an extensible format, e.g., Vega-lite JSON specifications.

4.2.3. Task 3: Create scenes and interactions

The last task is to **arrange** different visualizations in a desired sequence to **produce** a visual data story. Gravity++ should provide a presentation func-

tionality so that users can directly interact with the charts to navigate to other scenes. Users should also be able to view the **recorded** data story's configuration by downloading its metadata. This metadata can then be **imported** again by re-uploading if users wish to use it later.

4.3. Graph-based Architecture

Information visualization has two primary goals: exploration (including analysis) and communication (including visual data stories) [68, 19]. Furthermore, there are *three* main stages in the visual data story creation process: 1) analysis of data and creation of information visualizations; 2) sequencing of the created visualizations; 3) presenting and communicating the analysis results and visualizations [18, 19]. While Gravity++ provides some support for the *exploratory* goal of information visualization (stage 1), its primary focus is on the support of the *communicative goal* of information visualization (stages 2 and 3), i.e., aiding analysts in communicating findings to a target audience.

The Graph-based framework proposed by this paper expands current approaches by providing various user interactions and data processing with the visualizations in Gravity++. We incorporate a wide variety of interactions with visualizations, and at the same time, enable more complex data processing by direct manipulation of visualizations. Hence, our framework provides a more natural and effective platform for the Visual Information Seeking Mantra [11], through interaction with visualizations, and the creation of visual data stories.

Gravity++'s ultimate aim is to offer data experts an intuitive framework to construct complex visual narratives with direct manipulation of data, visualization and user interaction modes.

To design a visual narrative with interactive visualizations, we describe three aspects of Gravity++ as shown in Figure 1: a data domain, an interaction domain, and a visualization domain. These three aspects are loosely coupled i.e., updating or altering one domain does not necessarily affect other domains.

4.4. Data domain

The data domain in Figure 1 is a directed graph, modeling different datasets generated or introduced into the system at any time. The transition between datasets represents a parameterized data manipulation step of different types, e.g., filtering, aggregation, etc. For example, a dataset containing population: the population according to counties forms a data graph node and the population according to states forms another data node. If it is possible to generate a data query to transition from a data representation to another, an edge will be generated in the graph representing the corresponding data query. Accordingly the edges on data graph represent additions and updates to the data query. A path trace of the data graph to a certain node n would result in a complete query for processing the initial data to the dataset represented at node n. In this way, all data processing and navigation is captured with the graph and if new or additional data representations are required or made available, they can be added to the graph model; there is no need to change the whole design.
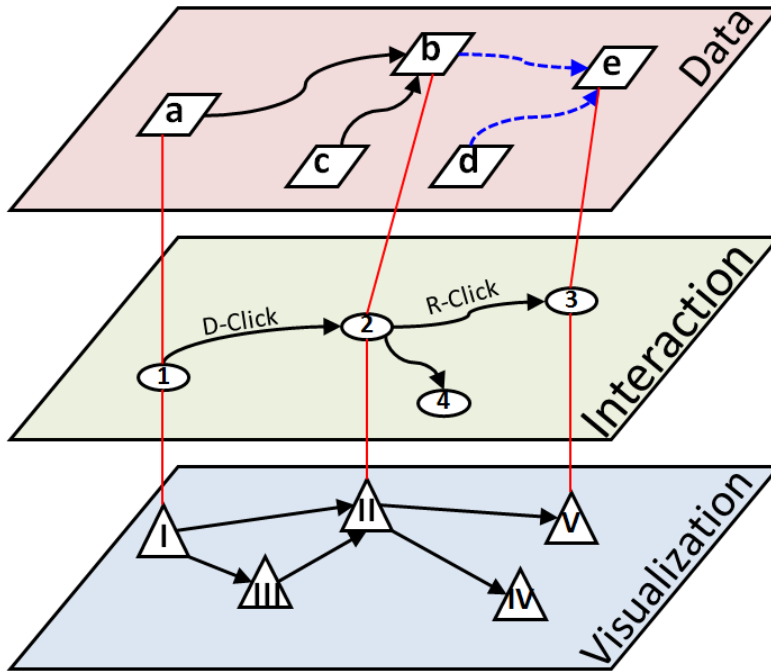
Figure 1: A high-level architecture of Gravity++. The architecture is a supergraph consisting of a data graph, interaction graph, and a visualization graph. The red lines are the traceability links that control the relationship between the data, interaction, and visualization graphs. For example, the rightmost red line shows that *node 3* in the interaction graph is associated with *node V* in the visualization graph and *node e* in the data graph.

4.5. Visualization domain

The visualization domain in Figure 1 is another directed graph that shows the flow and logical order of navigation between different available visualizations. It also helps to navigate backward between visualizations, e.g., to see the flow of changes in the visual representations and visual storytelling. Each node in this graph represents a visualization and the edges between them define if it is possible to navigate from a visualization to a neighboring visualization node. This transition will help visualization users trace the alteration and evolution of their visual representations.

A transition in the visualization graph might require additional data to be prepared and hence might trigger a transition in the data graph model. For example, to go from a scatter plot to a bar chart, data represented by different groups of the scatter plot need to be grouped to contribute to the bars' value of the bar chart. As a result, additional query scripts (a transition in data graph) would be provided to group the data based on the categories to be represented in the bar chart (new data node). However, data represented by a bar chart can also be represented by a pie chart, and therefore no data transition is required.

As another example, consider the population of United States depicted on a

map with each county colored according to its total population, hence forming a
Choropleth map; if the users would like to see population of a county according
to gender, then the information representing gender distributions could come
in another node in the data model and the corresponding visualization would
be associated with two nodes in the data graph, i.e., gender distribution and
population of counties. As a result, the two nodes will be merged to produce
a new node in the data graph. This new data node will be linked with the
visualization graph node using traceability links (red lines in Figure 1).

4.6. Interaction domain

The logic of user interactions and implicit system state is maintained in a
third directed graph - interaction graph (Figure 1). The interaction graph helps
navigate different visualization steps, datasets required or available at each step,
and user actions possible at each step. For example, assuming population of *states*
is being visualized as the first node, double clicking on population visualization
of a certain *state* could trigger transition from *state*-based visualization to the
more refined *county*-based visualization. Therefore, there is an edge connecting
node 1 in the interaction graph to node 2 with a double click event (See Figure 1).
Now if a user wants to see the gender-based population in *county*, a right click
event will transition the interaction to node 3, corresponding to gender-based
visualization. Given that the possible interactions are captured at each node
by the outgoing edges, the user interactions at each node are restricted to the
available set of outgoing edges. The designer of the visualization can decide
which interactions are available at each step. If a visualization is to be changed
(for example due to a user interaction with the system) requiring additional or
different data to be available, the additional data queries should be prepared to
make the data available.

The traceability links between these three graphs control the relationship
between the interaction, data, and visualization graphs. For example, the red
lines in Figure 1 show that node 3 in the interaction graph is associated with node
V in the visualization graph and node e in the data graph. If new interactions
became available, the visualization designer can add them to the set of edges
in the interaction graph without altering either the data graph or visualization
graph models.

With this architecture, the state of our system at each point in a visual
narrative can be represented by a set of tuples denoting the nodes of data graph
being used, a node in interaction graph, and nodes in visualization graph. Note
that a visualization might use multiple data nodes, or a visual display might use
multiple visualizations as a dashboard. Therefore the links between interaction
and data model, and interaction and visualizations could have one-to-many
relationships.

5. Gravity++ Interface Design

Gravity++ is a fully web-based application. It is first and foremost designed
for the authoring of logically sequenced visual data stories, and not primarily
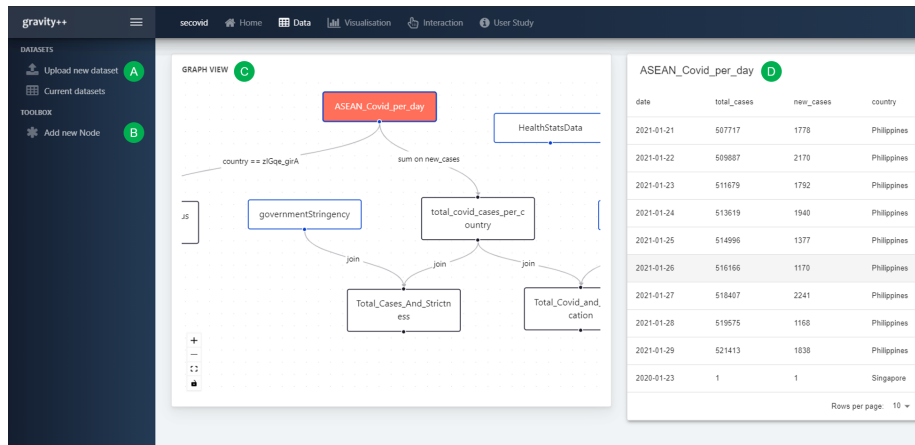
(a) Gravity++ interface: data domain overview.
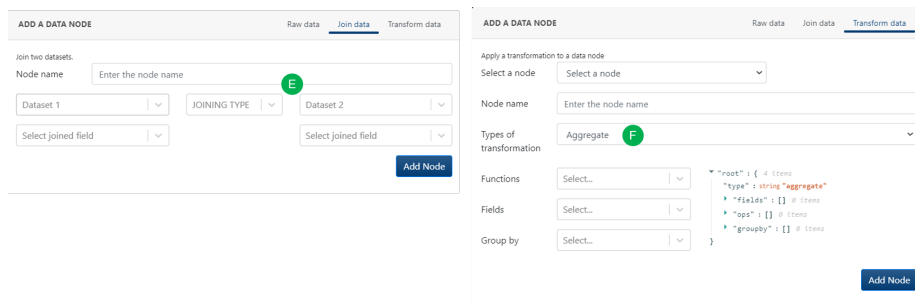
(b) Gravity++ interface: data domain modals.

Figure 2: Gravity++ interface: data domain. (A) Data upload for uploading data into Gravity++ (B) Data node creation: this allows a user to create a node in the data graph (C) Meta-visualization of the underlying data graph (D) Panel for navigating the data (E) Modal for joining datasets together (F) Data transformation modal for filtering and aggregating datasets.

on consumption. Hence, it is focused on larger screen sizes such as laptops and desktop monitors, rather than mobile devices with limited screen real-estate.

5.1. Data Domain Interface

⁴²⁰   The data domain view, shown in Figure 2a, provides a workspace for users to manage all current datasets used in data analysis, visualization views, and visual narratives. Users start by importing datasets into the workspace (2aA). Currently, Gravity++ accepts raw dataset sources in two formats, either in CSV or JSON, but can be extended to many others.

⁴²⁵   Once the datasets are imported, users can add a node which represents the dataset into the data domain graph (2aB). By clicking at any node, the data which the clicked node represents will be displayed in the data view. This view

13

enables users to perform data navigation steps, including sorting, projection and filtering. All data navigation steps are captured separately and therefore do not change the data at any node.

Data analysts can apply a data transformation to a node. For example, they can apply a filter or a data aggregation to the current dataset to create a new transformed dataset. Common filtering options include equal ($=$), greater than ($>$), less than ($<$), etc., while common aggregation methods are sum, average, max and min. The new dataset will be stored in a different node.

Gravity++'s data transformation options are not limited to single-source transformation. Gravity++ also enables data analysts to combine multiple datasets together via JOIN operations. Gravity++ currently supports LEFT, RIGHT and INNER JOINs.

A graph view is displayed to assist data analysts in tracing different datasets in the data domain and managing the relationships between them (2aC). Each node in the data graph contains meta-data of each dataset (dataset id, source, named label, etc.) while each edge contains data on how the transformation is applied. For instance, an edge that represents aggregation action will contain the aggregation type (mean, min, max, etc.), the field at which the aggregation method is applied, and the group-by fields.
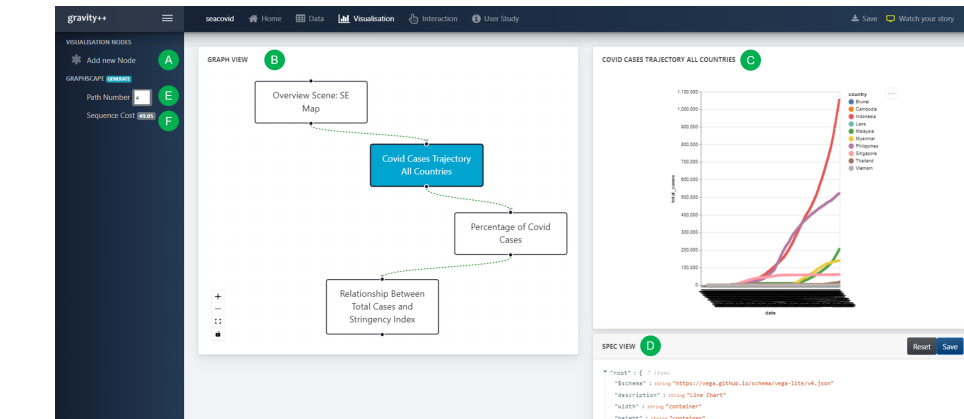

Figure 3: Gravity++ interface: visualization domain. (A) Visualization node creation: allows users to create visualizations in the visualization graph (B) Meta-visualization of the underlying visualization graph (C) Visualization preview panel for viewing created visualizations (D) Editable visualization specification panel where users can directly edit the Vega-lite JSON specifications (E) The position of the current active sequence out of the total number of recommended sequences (F) Sequence costs of recommmended visualization sequences [19].

5.2. Visualization Domain View

The visualization domain view, shown in Figure 3, provides a workspace for users to manage the visualization layer of the visual narrative process. To create a visualization node, users first select a dataset, the chart type from a given list of templates and specify the encoding channels (3A). The visualization

specification is stored in a Vega-lite [71] JSON object which can be edited and updated in real-time in the JSON specification view (3D).

Vega-Lite is a high-level grammar that enables rapid specification of inter-active data visualizations [71]. Vega-Lite combines a traditional grammar of graphics, providing visual encoding rules and a composition algebra for layered and multi-view displays, with a novel grammar of interaction. This high-level grammar of interactive graphics makes it relatively easy to generate web-based interactive visualizations, and is expressed in concise JSON formats [19, 71]. Futheremore, the Vega-Lite compiler synthesises a low-level Vega specification [72] with the requisite data flow, and default event handling logic that a user can override.

Moreover, the order in which visualizations are presented in a visual narrative has been shown to affect user understanding and interpretation of information [73, 74, 75]. Hence, Gravity++ supports the automatic and logical sequencing of visualizations to create coherent visual narratives.

For recommending the best sequences of visualizations, Gravity++ adapts and operationalizes Graphscape [73], a model to calculate the encoding distance between visualizations based on Vega-lite [71] specifications. Within the visualization domain view, users are able to inspect the automatically recommended optimal sequences (3B). Users are able to view and select the sequence rank (a higher rank indicates a more effective or optimal sequence) (3E). A sequence cost generated from the Graphscape model is also shown for each rank, aiding in evaluating between the different possible paths (3F).
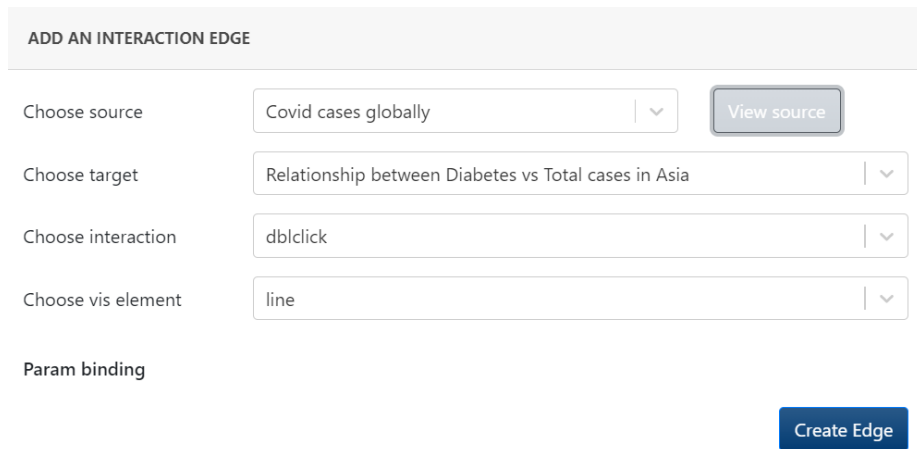
5.3. Interaction Domain View

The interaction domain view, shown in Figure 4a, provides a workspace for users to manage all scenes and transitions between scenes in a visual data story or narrative (4aC). Scenes are the essential building pieces of visual narratives. Each scene can contain one or more visualizations. Visual narratives are built by organising scenes in a sequence deemed appropriate by the visual designer. Users create a scene by simply appending a node on the workspace canvas (4aA).

Transitions between scenes are denoted as edges between scene nodes which are easily added on the workspace (4aB). A transition requires the following information: the scene where the transition is invoked (i.e., the source scene), the scene which the transition leads to (i.e., the target scene), type of triggers, the visualization element where the trigger is invoked and any data bindings (see modal in Figure 4b). Gravity++ supports reactive data binding. The idea is motivated from the use case that when a user interacts with a visualization element, the data associated with that element will be passed to the next scene. Unlike the traditional model of a linear narrative, Gravity++ supports non-linear visual narrative because it is based on a graph architecture; alternative paths are explicitly defined by scene transitions (4aC).

Gravity++ allows users to add insight notes for each scene for presentation

15
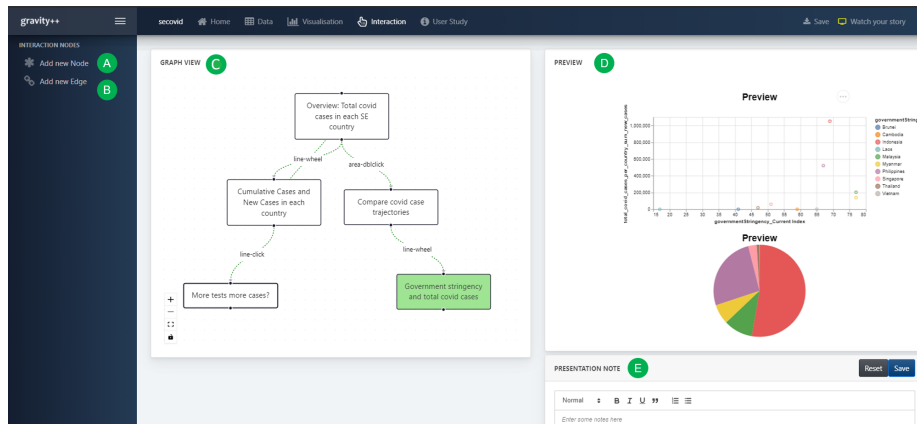
(a) Gravity++ interface: interaction domain overview.

(b) Gravity++ interface: interaction domain modal showing an example binding of an interaction event (double click) to a visualization element (line), from a source to a target visualization.

Figure 4: Gravity++ interface: interaction domain. (A) Interaction node creation: allows users to create nodes in the interaction graph (B) Interaction edge creation: allows users to create navigation edges between visualization scenes (C) Meta-visualization of the underlying interaction graph (D) Visualization scene panel for previewing visualizations(E) Editable presentation note for adding insight notes to visualization scenes.

purposes, akin to presenter notes in tools like Powerpoint[1] or Keynote[2]) (4aE).

The graph view provides a meta-visualization to help data analysts manage different scenes and trace the relationships between scenes (4aC). Each node

495

---

[1]https://www.office.com/launch/powerpoint
[2]https://www.apple.com/au/keynote/

16

stores scene meta-data (scene id, title and list of visualizations in each scene) while each edge contains the transition data (trigger type, triggering visual element).

A user session and state of the system can be saved in the graph database and can also be exported as a supergraph in a *gpp* file format and re-imported at a later time for collaboration purposes.

6. Usage Examples

In this section, we highlight two example usage scenarios showing Gravity++: the first example is based on the motivating scenario described in section 2 and the second a standalone example.

6.1. Example 1: Covid-19 dataset

Consider the motivating scenario described in section 2. Wendy and Bob can design a fully-working visual data story in the following tasks breakdown. A visual summary of the design (supergraph) is shown in Figure 5.

6.1.1. Task 1: Prepare datasets and process data

Having daily Covid statistics dataset, ASEAN governmental stringency index dataset, and Southeast Asia geographic location and boundary dataset (See C, D1, and A1 in Figure 5), Bob and Wendy can import them into Gravity++ using the data import feature within the data domain view. Then, Bob can use the data navigation feature in Gravity++ to help him to understand the different dimensions of each dataset. The data navigation feature also allows him to select and view different columns or apply different multi-criteria filters to retrieve the records that meet his target requirements. Once he has a good comprehension of the current datasets, he can then use the aggregation/filter/join features to derive new datasets from the original ones (See C1, A, B and D in Figure 5). In this case, he may use the filtering feature and apply it on the country field to select the covid statistics of a single country. Later on, Wendy also wants to know if the government strictness level affects the total number of positive cases per country, so she asks Bob to join the government stringency index dataset with the Covid-19 per day statistics together. Finally, as Wendy prefers to have a map of Southeast Asia with the total covid cases of every Southeast Asian country depicted in it, Bob joins the geographic location dataset with the total number of covid cases together.

6.1.2. Task 2: Create visualizations

After obtaining all necessary datasets, Wendy and Bob start to evaluate different visualizations. They decide to use a choropleth map with the color intensity to show the total covid cases of each country in a map. Bob and Wendy agree that using line charts will be best to depict the total case and new case growth trajectory for each country. Later on, they agree that a scatterplot will be appropriate to display the relationship between total covid cases and the
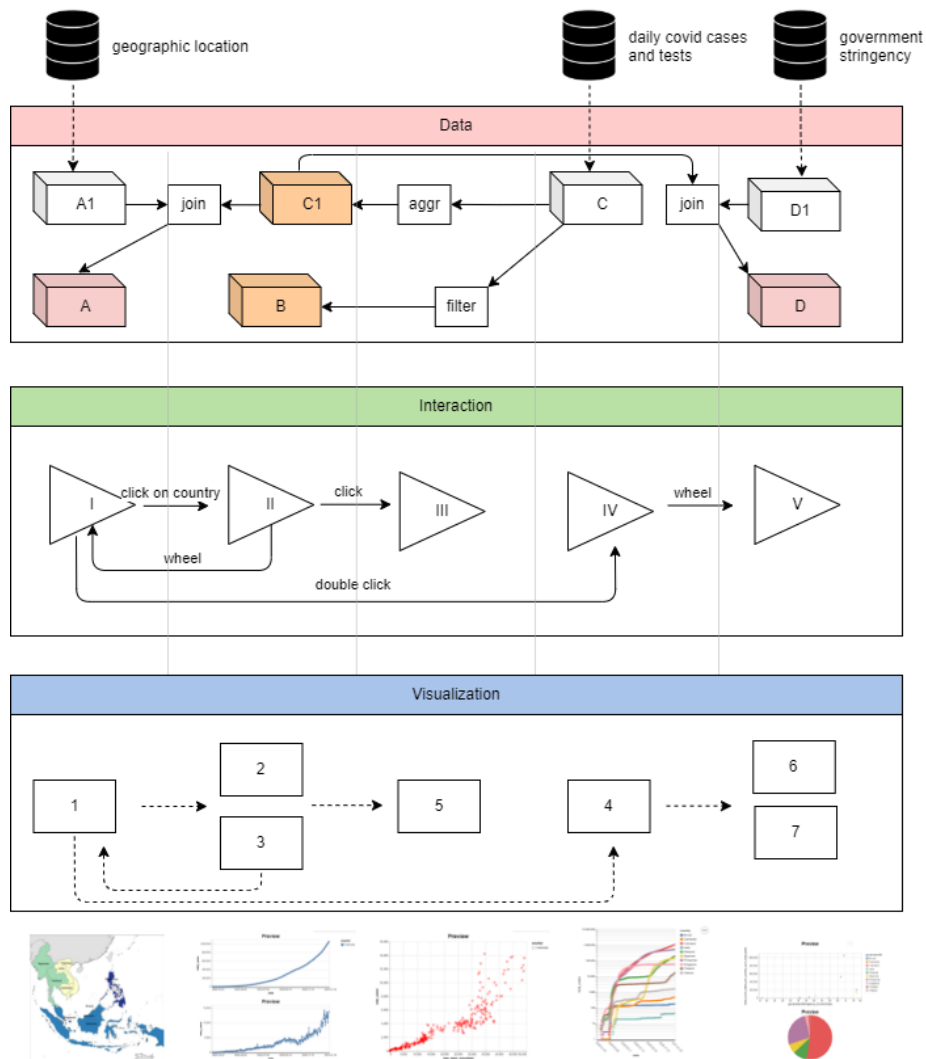
17

Figure 5: Bob's design for the analysis of covid cases in South East Asia. In the **Data** layer, A1, A, B, C, C1, D, and D1 represent raw data and derived datasets via transformation, e.g., datasets A1 and C1 are joined using a join operation to derive dataset A. In the **Interaction** layer, I to V are visualization story scenes and the edges show the events for navigation between scenes, e.g., a click event navigates from scene I to scene II while a wheel event navigates back from scene II to scene I. Each scene can have multiple vizualisations in it. In the **Visualization** layer, 1 - 7 represent individual visualizations and the edges capture their relationship in the interaction graph, e.g., visualizations 2 and 3 constitute scene II of the Interaction layer.

governmental stringency index. Scatterplots are also good candidates to verify if there is a correlation between the number of tests conducted and the number of new cases.

6.1.3. Task 3: Create scenes and add interactions

Once Bob and Wendy finish creating the necessary visualizations, they can start sequencing these visualizations into a fully functional visual narrative. Bob and Wendy agree that in the first scene, they would show the choropleth map of Southeast Asia, when the color intensity depicts the number of confirmed covid cases for every country. The second scene contains two line charts, the upper line chart displays the total case trajectory over time of a country and the lower line chart displays the new case trajectory over time of the same country.

From the overview map in the first scene, if users click at any country's territory, the second scene will show only the data of the selected country. From the second scene, if users single-click at any line, the third scene will appear, showing the relationship between number of tests conducted and the number of new covid cases, displayed in a chart with two corresponding lines sharing the same x-axis (time). And from the second scene, users could go back to the overview map if they wheel their mouse on any line. Besides, if the users double click on any position on the map, a line chart comparing the trajectory of all countries in logarithmic scale is displayed. Finally, if the users wheel on any line, it will lead to the next scene displaying the relationship between the governmental stringency level and the total number of cases for each country. These interactions are captured by the interaction graph in Figure 5.

After adding scenes and linking them using interactions, Bob and Wendy can add the presentation notes for each scene. They can then proceed to "View your story" page, where they can conduct a presentation based on the visual data story they just created. When presenting, Bob and Wendy could refer to the insight notes that they saved for each scene. A list of transitions to the next scenes is also displayed to help them select the desired interactions. Bob and Wendy could also export their presentation in the form of a PDF document to their local machine or capture it in a video recording.

For brevity and legibility, we have removed the traceability links in Figure 5. However, the swim lines indicate the state of the system and differentiate which nodes are participating in each system state.

6.2. Example 2: Analysis of different difficulties for having adequate fruit and vegetable for low-income residents

To demonstrate that Gravity++ is able to work in a wide range of scenarios, we propose another example where Amara, a food and nutrition researcher can successfully produce a visual story using Gravity++. The dataset that Amara has access to is the survey response which collects data about different barriers that can prevent low-income dwellers in Ibadan, Nigeria from having the daily recommended amount of fruit and vegetable. A visual summary of this example usage scenario is captured in Figures 6, 7, and 8.

6.2.1. Task 1. Prepare dataset and process data
Upload the dataset. Amara logs in and creates a blank Gravity project. She uploads her dataset by clicking on "Upload new dataset" button in the Data tab.

She then confirms if the dataset is successfully uploaded by clicking on "Current datasets" button where her dataset can be seen from the dropdown. To make it utilizable, she creates a data node (D0 in Figure 6), representing the dataset by clicking on "Add new Node", and fills in name and the data format to ensure Gravity++ can recognize and read the dataset. After the data node is created, she examines its data content by clicking on the node.
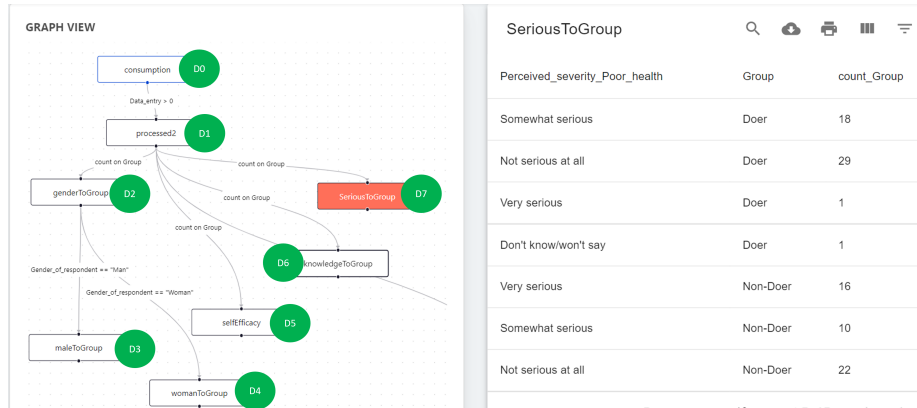

Figure 6: Amara's Data Domain. D0 to D7 are data nodes representing raw and derived datasets.

Clean the dataset. Amara confirms that the dataset is correctly parsed, however she quickly recognizes that some data rows are missing important information. She applies a non-empty filter in one of the columns to remove the unnecessary empty rows. This is achieved by creating a new "transformed" data node, D1 (Figure 6) from the original data node, D0, with the Filter function.

Amara then continues to navigate the dataset using the data exploration functionality. She notices some outstanding parameters in the survey data that can affect whether or not a person consumes enough fruit and vegetable, which she wants to present in her story.

Process datasets. Amara finds that the distribution amongst people with and without enough fruit and vegetation consumption is different between genders. She wishes to investigate this further, hence, she creates a new data node, D2 (Figure 6), from the cleaned data node, D1, with the aggregation function. D2 is a new data node which counts the number of survey respondents categorised by gender and their fruit/vegetable consumption level. By now, Amara wants to advance further by seeing the distribution for male only, she applies a filter function on D2 to create D3, and she applies the filter function on D2 to create D4 which is for female only.

Amara knows that her visual story needs more findings than just the level of fruit and vegetable consumption by gender, therefore, she creates three data nodes D5, D6, D7 which demonstrate the relationship between level of fruit/vegetation

consumption with their self-efficacy to have vegetable and fruit, their measured knowledge of food preparation, and their awareness of risks if they have poor health. It is noted that Amara also creates data nodes concerning other factors, but for the sake of simplicity, those relationships are not mentioned.

6.2.2. Task 2. Create visualizations

To populate her visual data story, Amara first creates two charts (V1 and V2 in Figure 7) to show the proportion of respondents consuming adequate fruit and vegetable against respondents who do not, one for male data and the other for female. To create each, Amara first navigates to the visualization tab and clicks on "Add new Node". As the chart creation wizard shows up, she selects the target data, chart type, encoding attributes such as color and arc length. The outcome of this task is a new visualization node which represents each individual chart. Amara confirms that the chart has been created correctly by clicking on each visualization node to view the chart.
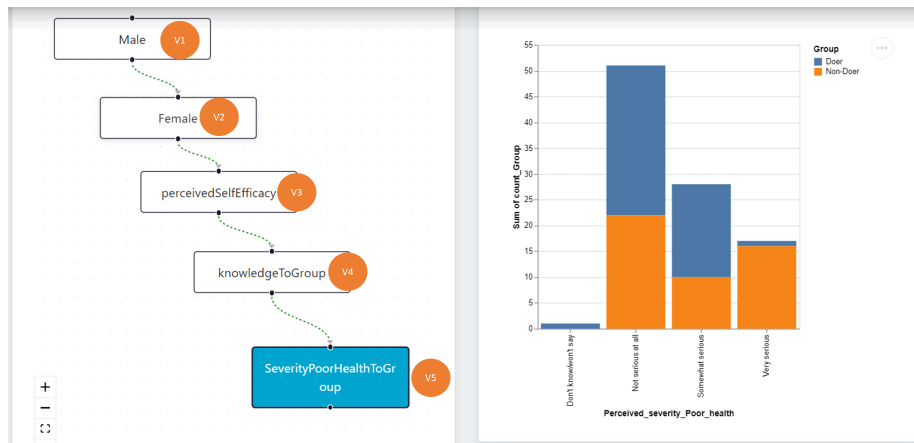

Figure 7: Amara's Visualisation Domain: showing five nodes (V1 to V5) representing visualizations.

Amara repeats the process of creating visualizations by creating three more visualization nodes (V3, V4, and V5 in Figure 7), making it a total of five visualization nodes (Figure 7).

6.2.3. Task 3. Create scenes and add interactions

Finally, Amara links all the visualizations together to produce a functional narrative. Amara creates scene I1, which contains two charts V1 and V2 by following the "Add new Node" wizard in the Interaction view. Similarly, Amara creates I2, I3, I4, each contains a chart from V3, V4 and V5, respectively. After creating all the interaction nodes, Amara creates the transition between I1, I2, I3 and I4 by following the "Add new Edge" wizard. Each transition requires Amara to specify the source and destination scenes and how the transitions are

triggered. Once all the transitions are created, Amara's visual story is ready to be presented at the "Present" tab.
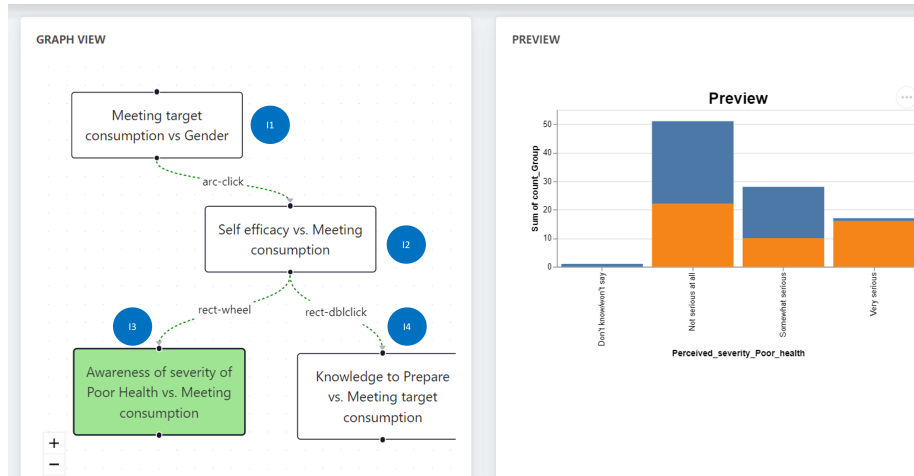

Figure 8: Amara's Interaction Domain: showing the scenes (I1 to I4) and transitions between them using events.

7. Implementation

Gravity++ is implemented using ReactJS on client side and NodeJS on server side. A Neo4J graph database instance is maintained at server side, serving as main storage for the system and performing graph traversals. The client side sends API requests using Axios, which are then rerouted by the server side to perform CRUD operations with the Neo4J instance.

Gravity++ accepts JSON and CSV datasets, and files of both formats are parsed into a JSON-array of data objects. Gravity++ currently supports two data transformation options, i.e. filtering and aggregation, both are supported by Vega transforms. Gravity++ takes advantage of Vega [72] signals to add parameters into the data manipulation, which is helpful when user interaction data is needed to update visualization scenes.

Gravity++ uses Vega-lite [71], a compact visualization grammar to declare and construct the visualization from JSON expressions. Gravity++ also allows multiple visualizations in one scene, supported from Vega concatenation features. Gravity++ supports the generation of optimal visualization sequence recommendations by operationalizing the Graphscape [73] model. Graphscape considers all paths that visit all nodes, each node once, and sums all the encoding distances between two adjacent nodes. Generating all node permutations are computationally expensive, hence, a separate thread at server side is dedicated to running the Graphscape model when requested, to prevent blocking the client side and the execution of other requests on the server side.

22

To support user interaction, a finite state machine is constructed from the interaction graph, determining the next scene, or state based on current scene and generated user interaction data. Gravity++ uses Vega event streams to capture the user interactions, e.g. click, double click and wheel, and when such interaction is detected, an event handler is triggered to update state, where new state and next possible interactions are loaded into the system.

8. Evaluation

We conducted a user study to evaluate the capability of Gravity++ in producing interactive visual narratives. Our evaluation covers all three domains of Gravity++: data, visualization and interaction. *The evaluation instruments including short tutorial videos are available as supplementary materials via this URL:* `https://tinyurl.com/gravitysupplementary`.

8.1. Participants

We invited nine representative participants for the user study consisting of five experts and four non-experts in information visualization. The expert participants create visualizations and visual data stories in their work while the non-experts are familiar with basic data analysis and visualization. Participants were between 19 and 40 years old.

8.2. Procedure

At the beginning of each user evaluation session, an investigator introduced participants to the Gravity++ system, the objective of the user study, and its voluntary nature.

8.2.1. Training stage

To ensure all participants received uniform instruction, each was asked to watch a series of tutorial videos within the first 10 minutes. The content of the videos covers all essential functionalities of Gravity++, which users would use to produce a fully-functioning visual narrative. Participants were encouraged to ask questions about any areas requiring clarification.

8.2.2. Task stage

The task stage lasted on average 70 minutes for participants, with a range of 25 minutes. Participants were asked to follow the task instruction sheet to complete the tasks.

Each participant directly interacted with Gravity++ to produce a full visual narrative of the global Covid-19 pandemic, following a designated task sequence. The task sequence is designed in a way that simulates a realistic process of creating an interactive visual narrative. *We chose the Covid-19 datasets because it is a current topic that participants are familiar with.*

Each participant was asked to import two given datasets covering the Covid-19 pandemic into the data domain for navigation purposes and to follow a

sequence of steps to transform the original datasets using Gravity++'s available features. Once a participant obtained new meaningful derived datasets that were ready to produce visualizations, they moved to the visualization domain and created visualizations. Every participant was also asked to go back to the data domain to generate new datasets. This imitates a typical situation when a data visualization designer wants to look back at the data they have, because new ideas for visualization can come up naturally during creation, which often requires transforming existing datasets in a different way. Once satisfied with the visualizations, participants were asked to link different visualizations together to create scenes, and add insight notes to those scenes via user interaction under the interaction tab. In the end, participants opened the presentation view to present their final visual data story.

8.2.3. Survey stage

In this phase, participants filled in an anonymous survey. The anonymous survey uses a 5-point Likert scale to capture evaluation categories such as ease of use, ease of learning, and satisfaction, as proposed by Amini et al. [76]. Participants were encouraged to provide additional feedback in the comment sections.
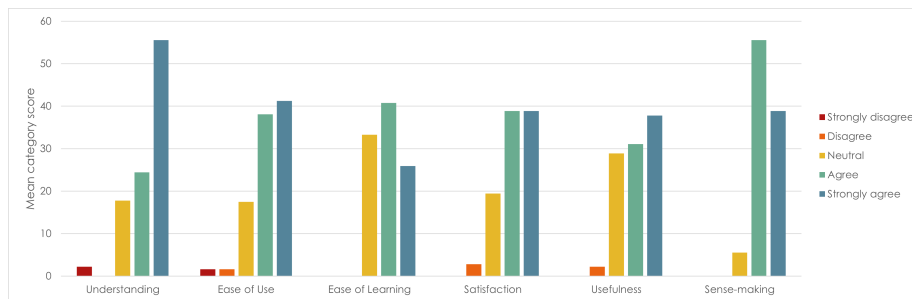

Figure 9: Summary result of the user evaluation study

8.3. Results

8.3.1. Successes

All participants finished the user evaluation study. They successfully created visual data stories with a high level of complexity. They were able to add datasets, create chains of transformations to derive new datasets, create visualizations, scenes and interactions to transition between scenes. Figure 9 summarizes the results of the user study.

Participants agreed that the workflow beginning from data preparation to visualization creation and finally deciding on scenes and transitions very intuitive and easy to work on. They found navigating and transforming a dataset using Gravity++ easy. Participants also agreed that it is easy to create visualizations as well as transitions between scenes, and to evaluate which different visualizations are suitable for a dataset.

The results show that Gravity++ made the relationship between different
₇₃₀ data, visualization and interaction domains easy to be understood and managed.
One participant commented that *"The data modification done via nodes is easy
to understand, it helps me know where the data is coming from after a series
of data processing steps"*. Another participant noted, *"I liked that the flow
of starting from uploading data and observing the columns to visualising and
₇₃₅ connecting them, and finally presenting them"*. All 4 non-expert participants
agreed that Gravity++ gave them new a perspective on perceiving a visual
narrative in the three different layers: data, visualization and interaction, while
the remaining participants also expressed the same sentiment.

In terms of usefulness, participants found it helpful to save their current
₇₄₀ progress into a downloadable *gpp* file or into the database. A participant com-
mented that they like the *"representation of data, visualization and interaction
as a graph with the ability to export this"*, and also appreciated the fact that
Gravity builds on other reusable available tools such as Vega [72], *"...it builds
on-top of tools such as Vega, but provides an interfaces that is much faster
₇₄₅ to learn and use"*. Participants also appreciated the presentation and insight
note-taking features. For example, a participant commented, *"I really liked the
idea of generating presentations from the dataset. It saves the time editing and
re-creating graphs and use them in presentations"*.

Furthermore, sense-making can be understood from two perspectives - the an-
₇₅₀ alyst perspective and the target audience perspective [19]. The results show that
Gravity++ supports sense-making from an analyst perspective, i.e., Gravity++
helped participants (role-playing as analysts in the evaluation) understand the
datasets that they interacted with, the sequencing of visualizations, and the
overall visual data story creation process. Participants agreed that Gravity++
₇₅₅ would help them create better visual narratives in the future. All participants
agreed that Gravity++ aided them with reasoning in designing meaningful visual
data stories. For instance, a participant commented, *"I found the representation
of each stage (data, visualization, interaction) as a graph an intuitive, consistent,
and powerful way to perform analysis"*. Another participant also noted, *"The
₇₆₀ consistent way that data, visualization and interaction were treated in Gravity++
helped me see how each of these stages are connected in the presentation of the
final result"*. We do conjecture that improved sense-making from the analyst
perspective would propagate into sense-making for the target audience. However,
another study is required to (in)validate that conjecture, but it's beyond the
₇₆₅ scope of this paper and is an area for future work.

8.3.2. Current Limitations

Some participants reported that they struggled to apply a change to the graph
structure, for instance, if they choose the wrong type of data transformation,
they had to delete the incorrect data node before creating a new one. This
₇₇₀ difficulty reflects the current limitation of Gravity++ that it does not support
users in modifying the current graph meta-data or graph structure. Future
developments of Gravity++ would include a feature that enables users to modify
the existing graph structure and metadata.

25

Some participants also noted that adding nodes seemed inconvenient and time-consuming, while others commented that node and edge creation should not solely rely on forms. Interaction edges should be created directly by dragging from a source node to a target node in the graph view.

Furthermore, although expert participants appreciated the novelty and usefulness of the tool, they agreed that the prototype would be benefit from more development work in terms of scale. For instance, a participant commented, *"Gravity++ seems to be a promising tool for rapidly analysing/visualising data and allowing users to create complex processing graphs. However, I would need guidelines for integrating it with the rest of my workflow (e.g. version control, integration with custom data analysis scripts/notebooks) before using it on larger multi-developer projects"*. A suggestion from a participant says that, *"Users should be able to create node faster, maybe a node suggestion engine."*. Another participant noted the disadvantage of not knowing in advance how long the visualization sequence recommender would take in processing an arbitrary number of visualization sequences.

8.4. Threats to Validity

While we were mindful in conducting the user evaluation study for Gravity++, there may be some potential threats to the validity of our results. We highlight these threats and the controls applied to them. We summarise these in Table 2.

8.4.1. Internal Validity

Instrumentation. We designed the evaluation protocols to be easy to use by the participants and encouraged participants to seek clarifications in any areas they did not understand. However, it is possible that some participants may have been reluctant to seek clarification and completed the tasks and survey in the way they understood them, and this may have affected their experience of the evaluation. We note that participants raised no concerns regarding the evaluation protocols.

Maturation. Due to the length of the evaluation study (an average of 70 minutes), there is a risk of boredom as participants advanced through the study. However, the investigators made effort to encourage participants with regards to the benefits of of the study in advancing the field of information visualization. Participants were also informed that they were under no obligation to complete the study and they could leave at any time.

8.4.2. External Validity

Population Sample. We aimed to get a representative sample consisting of information visualization experts and non-experts. Our population sample consisted of 5 experts and 4 non-experts. All of our expert participants create information visualizations and visual data stories in their professional engagements. Additionally, the 4 non-expert participants are familiar with data analytics and visualization, with diverse backgrounds in Computer Science and Information Technology, and the Arts and Humanities.

Sample Size. Our sample size consists of a total number of 9 participants for the user evaluation study. We note that our aim was not to compare participants' performance, but instead to gain insights into how our graph-based architecture supports the creation of visual data stories, potential barriers, and areas for improvement. Furthermore, the number of expert participants engaged in our study is within the acceptable range of participants in information visualization and human-computer interaction research [77, 55, 57, 54].

Table 2: Summary of threats to validity.

Threat	Category	Status
Instrumentation	Internal	Addressed
Maturation	Internal	Addressed
Population sample	External	Partially addressed
Sample size	External	Partially addressed

9. Discussion and Future Work

For the design of our Gravity++ framework, we have focused on interaction with visualization and their use in visual data storytelling. The combination of the three graphs generates a supergraph that when parsed by the system results in an interactive visual data story. Note that when a visual data story has been generated, it can be re-imported for future data stories. For example, Wendy's scenario can be reused for a similar analysis. It can also be updated (by plugging new datasets) to support other domains and analyses.

The architecture of the Gravity++ system allows better control over steps of interactive visualization usage. For example, in using our data graph model, it is clear which capabilities are going to be needed at each node and which queries should be prepared. This results in faster preparation of query results. This data graph model can also facilitate other data navigation models, e.g., data cubes. In such an arrangement, nodes of the data cube lattice will be represented by Gravity++'s graph nodes, and their links represent the queries to generate those data aggregation nodes.

It is also important to consider that generating visualizations from a set of data should not necessarily be a one-way procedure. Visualizations can potentially be incorporated to perform tasks and processes on the data they have originated from. This is important, as current techniques mostly focus on how to visualize data and do not focus much on how to utilize the visualizations to alter the data or perform processing tasks on the data [78]. Various research and industrial applications have been working on addressing data mapping solutions in order to make transitioning from one data format to another less expensive and more user-friendly (e.g., [79, 80]). Interactive visualizations can play a significant role in the realization of user-friendly data manipulation and processing, and our Gravity++ framework can accommodate such interaction models.

The examples we described are based on two user scenarios, i.e., an expert user develops the visualization, and a domain analyst consumes them. The domain analyst does not need to know the technical complexity behind the visualizations. There is also the possibility of enabling a third user as the data analyst. The data analyst would alter the data using the interactions on the visualization. This would require more complex many-to-many relationships between the three domains. In this way, a selection of interactions can cause alteration of visualizations, and certain interactions on the visualization would trigger data processing and wrangling. For example, changing the height of a bar in a bar chart, could trigger the data behind that bar to be updated. This operation can be performed as instance-based operation, e.g., for removing outliers in a trend analysis visualization or as demonstrations to generate queries (e.g., to normalize all values represented in a bar chart). This extension would allow for the combination of the capabilities of popular interactive visualization design tools (e.g. Tableau[3]) to that of Extract Transform Load (ETL) tools (like Wrangler [79] and Trifacta[4]).

Another area for future work is the extension and adaptation of techniques and frameworks in tools like Gravity++ to facilitate the consumption of visual data stories on mobile devices, as mobile devices have become more ubiquitous forms of visualization consumption.

10. Conclusion

In this paper, we introduced a novel graph-based framework for designing and generating flexible interactive visualizations to enable alternative data representations, support for new interaction paradigms, and support for the integration of newly acquired data to create visual narratives. We implemented our framework in a stand-alone prototype tool called Gravity++. Based on the results of a user evaluation study with representative participants, Gravity++ can help analysts utilise different interaction capabilities with a variety of complex visualizations and multiple data sources to create coherent visual narratives.

Acknowledgements

---

[3]https://www.tableau.com
[4]https://www.trifacta.com

References

[1] H. O. Obie, C. Chua, I. Avazpour, M. Abdelrazek, J. Grundy, T. Bednarz, The effect of narration on user comprehension and recall of information visualisations, in: 2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2020, pp. 1–4. `doi:10.1109/VL/HCC50065.2020.9127200`.

[2] D. Moody, The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering, IEEE Transactions on Software Engineering 35 (6) (2009) 756–779. `doi:10.1109/TSE.2009.67`.

[3] I. Avazpour, J. Grundy, H. Vu, Generating reusable visual notations using model transformation, International Journal of Software Engineering and Knowledge Engineering 25 (2015) 277–305.

[4] H. O. Obie, C. Chua, I. Avazpour, M. Abdelrazek, J. Grundy, T. Bednarz, A study of the effects of narration on comprehension and memorability of visualisations, Journal of Computer Languages 52 (2019) 113–124. `doi:https://doi.org/10.1016/j.cola.2019.04.006`.

[5] S. Liu, W. Cui, Y. Wu, M. Liu, A survey on information visualization: recent advances and challenges, The Visual Computer 30 (2013) 1373–1393.

[6] H. O. Obie, C. Chua, I. Avazpour, M. Abdelrazek, J. Grundy, Visualising Melbourne pedestrian count, in: 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2017, pp. 343–344. `doi:10.1109/VLHCC.2017.8103501`.

[7] S. K. Card, J. D. Mackinlay, B. Shneiderman, Using Vision to Think, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, p. 579–581.

[8] B. Lee, P. Isenberg, N. Henry Riche, S. Carpendale, Beyond mouse and keyboard: Expanding design considerations for information visualization interactions, Visualization and Computer Graphics, IEEE Transactions on 18 (2012) 2689–2698. `doi:10.1109/TVCG.2012.204`.

[9] J. Heer, M. Bostock, V. Ogievetsky, A tour through the visualization zoo, Commun. ACM 53 (2010) 59–67. `doi:10.1145/1743546.1743567`.

[10] H. O. Obie, Data — driven visualisations that make sense, in: 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2017, pp. 313–314. `doi:10.1109/VLHCC.2017.8103486`.

[11] B. Shneiderman, The eyes have it: A task by data type taxonomy for information visualizations, in: Proceedings of the 1996 IEEE Symposium on Visual Languages, VL '96, IEEE Computer Society, USA, 1996, pp. 336–343.

[12] J. Heer, B. Shneiderman, Interactive dynamics for visual analysis: A taxonomy of tools that support the fluent and flexible use of visualizations, Queue 10 (2) (2012) 30–55. `doi:10.1145/2133416.2146416`.
URL `https://doi.org/10.1145/2133416.2146416`

[13] G. Sun, Y. Wu, R. Liang, S. Liu, A survey of visual analytics techniques and applications: State-of-the-art research and future challenges, Journal of Computer Science and Technology 28 (2013) 852–867.

[14] O. Lock, T. Bednarz, C. Pettit, The visual analytics of big, open public transport data – a framework and pipeline for monitoring system performance in greater sydney, Big Earth Data 5 (1) (2021) 134–159.

[15] C. Weaver, Building highly-coordinated visualizations in improvise, in: IEEE Symposium on Information Visualization, 2004, pp. 159–166. `doi:10.1109/INFVIS.2004.12`.

[16] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, J. Heer, Voyager: Exploratory analysis via faceted browsing of visualization recommendations, IEEE Transactions on Visualization and Computer Graphics 22 (2016) 649–658.

[17] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, J. Heer, Voyager 2: Augmenting visual analysis with partial view specifications, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 2648–2659.

[18] H. O. Obie, C. Chua, I. Avazpour, M. Abdelrazek, J. Grundy, T. Bednarz, A framework for authoring logically ordered visual data stories, in: 2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2019, pp. 257–259. `doi:10.1109/VLHCC.2019.8818925`.

[19] H. O. Obie, C. Chua, I. Avazpour, M. Abdelrazek, J. Grundy, T. Bednarz, Authoring logically sequenced visual data stories with Gravity, Journal of Computer Languages 58 (2020) 100961.

[20] C. Stolte, D. Tang, P. Hanrahan, Multiscale visualization using data cubes, in: IEEE Symposium on Information Visualization, 2002, pp. 7–14. `doi:10.1109/INFVIS.2002.1173141`.

[21] L. Lins, J. T. Klosowski, C. Scheidegger, Nanocubes for real-time exploration of spatiotemporal datasets, IEEE Transactions on Visualization and Computer Graphics 19 (12) (2013) 2456–2465. `doi:10.1109/TVCG.2013.179`.

[22] J. Heer, M. Agrawala, W. Willett, Generalized selection via interactive query relaxation, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08, Association for Computing Machinery, New York, NY, USA, 2008, p. 959–968. `doi:10.1145/1357054.1357203`.
URL `https://doi.org/10.1145/1357054.1357203`

[23] Square Inc., Crossfilter: Fast multidimensional filter for coordinated views, http://square.github.io/crossfilter/, 2012.

[24] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, M. A. Porter, Multilayer networks, Journal of Complex Networks 2 (3) (2014) 203–271. arXiv:https://academic.oup.com/comnet/article-pdf/2/3/203/9130906/cnu016.pdf, doi:10.1093/comnet/cnu016.
URL https://doi.org/10.1093/comnet/cnu016

[25] F. McGee, M. Ghoniem, G. Melançon, B. Otjacques, B. Pinaud, The state of the art in multilayer network visualization, Computer Graphics Forum 38 (6) (2019) 125–149. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13610, doi:https://doi.org/10.1111/cgf.13610.
URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13610

[26] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, A. Vespignani, Epidemic processes in complex networks, Rev. Mod. Phys. 87 (2015) 925–979. doi:10.1103/RevModPhys.87.925.
URL https://link.aps.org/doi/10.1103/RevModPhys.87.925

[27] A. Saumell-Mendiola, M. A. Serrano, M. Boguñá, Epidemic spreading on interconnected networks, Phys. Rev. E 86 (2012) 026106. doi:10.1103/PhysRevE.86.026106.
URL https://link.aps.org/doi/10.1103/PhysRevE.86.026106

[28] D. A. Bright, C. Greenhill, A. Ritter, C. Morselli, Networks within networks: using multiple link types to examine network structure and identify key actors in a drug trafficking operation, Global Crime 16 (3) (2015) 219–237. arXiv:https://doi.org/10.1080/17440572.2015.1039164, doi:10.1080/17440572.2015.1039164.
URL https://doi.org/10.1080/17440572.2015.1039164

[29] M. E. Dickison, M. Magnani, L. Rossi, Multilayer Social Networks, Cambridge University Press, 2016. doi:10.1017/CBO9781139941907.

[30] F. McGee, M. Uring, M. Ghoniem, Towards visual analytics of multilayer graphs for digital cultural heritage, in: 1st Workshop on Visualization for the Digital Humanities, 2016, pp. 1 – 5.

[31] B. Sluban, M. Grčar, I. Mozetič, Temporal Multi-layer Network Construction from Major News Events, Springer International Publishing, Cham, 2016, pp. 29–41. doi:10.1007/978-3-319-30569-1_3.
URL https://doi.org/10.1007/978-3-319-30569-1_3

[32] C. Collins, S. Carpendale, Vislink: Revealing relationships amongst visualizations, IEEE Transactions on Visualization and Computer Graphics 13 (6) (2007) 1192–1199. doi:10.1109/TVCG.2007.70521.

[33] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, J. Fekete, Graphdice: A system for exploring multivariate social networks, Computer Graphics Forum 29 (3) (2010) 863–872. `arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01687.x`, `doi:https://doi.org/10.1111/j.1467-8659.2009.01687.x`.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01687.x`

[34] J. Xia, E. E. Gill, R. E. W. Hancock, Networkanalyst for statistical, visual and network-based meta-analysis of gene expression data, Nature Protocols 10 (2015) 823–844.

[35] R. Bourqui, D. Ienco, A. Sallaberry, P. Poncelet, Multilayer graph edge bundling, in: 2016 IEEE Pacific Visualization Symposium (PacificVis), 2016, pp. 184–188. `doi:10.1109/PACIFICVIS.2016.7465267`.

[36] Y. Liu, C. Wang, P. Ye, K. Zhang, Hybridvis: An adaptive hybrid-scale visualization of multivariate graphs, J. Vis. Lang. Comput. 41 (2017) 100–110. `doi:10.1016/j.jvlc.2017.03.008`.
URL `https://doi.org/10.1016/j.jvlc.2017.03.008`

[37] K. Xu, A. Ottley, C. Walchshofer, M. Streit, R. Chang, J. Wenskovitch, Survey on the analysis of user interactions and visualization provenance, Computer Graphics Forum 39 (3) (2020) 757–783. `doi:https://doi.org/10.1111/cgf.14035`.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14035`

[38] J. Heer, J. Mackinlay, C. Stolte, M. Agrawala, Graphical histories for visualization: Supporting analysis, communication, and evaluation, IEEE Transactions on Visualization and Computer Graphics 14 (6) (2008) 1189–1196. `doi:10.1109/TVCG.2008.137`.

[39] C. Dunne, N. Henry Riche, B. Lee, R. Metoyer, G. Robertson, Graphtrail: Analyzing large multivariate, heterogeneous networks while supporting exploration history, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12, Association for Computing Machinery, New York, NY, USA, 2012, p. 1663–1672. `doi:10.1145/2207676.2208293`.
URL `https://doi.org/10.1145/2207676.2208293`

[40] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, M. Streit, From visual exploration to storytelling and back again, Computer Graphics Forum 35 (3) (2016) 491–500. `arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12925`, `doi:https://doi.org/10.1111/cgf.12925`.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12925`

[41] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, H. T. Vo, Vistrails: Visualization meets data management, in: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, SIGMOD '06, Association for Computing Machinery, New York, NY, USA,

2006, p. 745–747. `doi:10.1145/1142473.1142574`.
URL `https://doi.org/10.1145/1142473.1142574`

[42] C. Bors, T. Gschwandtner, S. Miksch, Capturing and visualizing provenance from data wrangling, IEEE Computer Graphics and Applications 39 (6) (2019) 61–75. `doi:10.1109/MCG.2019.2941856`.

[43] S. Xu, C. Bryan, J. K. Li, J. Zhao, K.-L. Ma, Chart constellations: Effective chart summarization for collaborative and multi-user analyses, Computer Graphics Forum 37 (3) (2018) 75–86. `arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13402`, `doi:https://doi.org/10.1111/cgf.13402`.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13402`

[44] A. Mathisen, T. Horak, C. N. Klokmose, K. Grønbæk, N. Elmqvist, Insideinsights: Integrating data-driven reporting in collaborative visual analytics, Computer Graphics Forum 38 (3) (2019) 649–661. `arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13717`, `doi:https://doi.org/10.1111/cgf.13717`.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13717`

[45] K. Dextras-Romagnino, T. Munzner, Segmentifier: Interactive refinement of clickstream data, Computer Graphics Forum 38 (3) (2019) 623–634. `arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13715`, `doi:https://doi.org/10.1111/cgf.13715`.
URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13715`

[46] F. Dabek, J. J. Caban, A grammar-based approach for modeling user interactions and generating suggestions during the data exploration process, IEEE Transactions on Visualization and Computer Graphics 23 (1) (2017) 41–50. `doi:10.1109/TVCG.2016.2598471`.

[47] P. H. Nguyen, K. Xu, A. Bardill, B. Salman, K. Herd, B. W. Wong, Sensemap: Supporting browser-based online sensemaking through analytic provenance, in: 2016 IEEE Conference on Visual Analytics Science and Technology (VAST), 2016, pp. 91–100. `doi:10.1109/VAST.2016.7883515`.

[48] P. H. Nguyen, K. Xu, A. Wheat, B. W. Wong, S. Attfield, B. Fields, Sensepath: Understanding the sensemaking process through analytic provenance, IEEE Transactions on Visualization and Computer Graphics 22 (1) (2016) 41–50. `doi:10.1109/TVCG.2015.2467611`.

[49] Y. B. Shrinivasan, J. J. van Wijk, Supporting the analytical reasoning process in information visualization, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08, Association for Computing Machinery, New York, NY, USA, 2008, p. 1237–1246. `doi:10.1145/1357054.1357247`.
URL `https://doi.org/10.1145/1357054.1357247`

[50] Y. B. Shrinivasan, J. van Wijk, Supporting exploration awareness in information visualization, IEEE Computer Graphics and Applications 29 (5) (2009) 34–43. `doi:10.1109/MCG.2009.87`.

[51] Y. Shrinivasan, D. Gotzy, J. Lu, Connecting the dots in visual analysis, in: Proceedings of the 2009 IEEE Symposium on Visual Analytics Science and Technology, 2009, pp. 123 – 130. `doi:10.1109/VAST.2009.5333023`.

[52] E. Segel, J. Heer, Narrative visualization: Telling stories with data, IEEE Transactions on Visualization and Computer Graphics 16 (6) (2010) 1139–1148. `doi:10.1109/TVCG.2010.179`.

[53] R. Eccles, T. Kapler, R. Harper, W. Wright, Stories in geotime, in: 2007 IEEE Symposium on Visual Analytics Science and Technology, 2007, pp. 19–26. `doi:10.1109/VAST.2007.4388992`.

[54] A. Satyanarayan, J. Heer, Authoring narrative visualizations with Ellipsis, Comput. Graph. Forum 33 (3) (2014) 361–370. `doi:10.1111/cgf.12392`. URL `http://dx.doi.org/10.1111/cgf.12392`

[55] N. W. Kim, N. Riche, B. Bach, G. A. Xu, M. Brehmer, K. Hinckley, M. Pahud, H. Xia, M. McGuffin, H. Pfister, Datatoon: Drawing dynamic network comics with pen + touch interaction, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, ACM, 2019.

[56] Z. Zhao, R. Marr, N. Elmqvist, Data Comics: Sequential art for data-driven storytelling, Tech. rep., Human Computer Interaction Lab, University of Maryland (2015).

[57] F. Amini, N. H. Riche, B. Lee, A. Monroy-Hernandez, P. Irani, Authoring data-driven videos with dataclips, IEEE Transactions on Visualization and Computer Graphics 23 (1) (2017) 501–510. `doi:10.1109/TVCG.2016.2598647`.

[58] M. Conlen, M. Vo, A. Tan, J. Heer, Idyll Studio: A Structured Editor for Authoring Interactive & Data-Driven Articles, Association for Computing Machinery, New York, NY, USA, 2021, p. 1–12.

[59] B. Lee, N. H. Riche, P. Isenberg, S. Carpendale, More than telling a story: Transforming data into visually shared stories, IEEE Computer Graphics and Applications 35 (5) (2015) 84–90. `doi:10.1109/MCG.2015.99`.

[60] M. Brehmer, T. Munzner, A multi-level typology of abstract visualization tasks, IEEE Transactions on Visualization and Computer Graphics 19 (12) (2013) 2376–2385. `doi:10.1109/TVCG.2013.124`.

[61] M. Tory, T. Moller, Rethinking visualization: A high-level taxonomy, in: IEEE Symposium on Information Visualization, 2004, pp. 151–158. `doi:10.1109/INFVIS.2004.59`.

[62] H. Schulz, T. Nocke, M. Heitzler, H. Schumann, A design space of visualization tasks, IEEE Transactions on Visualization and Computer Graphics 19 (12) (2013) 2366–2375. doi:10.1109/TVCG.2013.120.
URL http://dx.doi.org/10.1109/TVCG.2013.120

[63] R. Roth, An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization, IEEE Transactions on Visualization and Computer Graphics 19 (12) (2013) 2356–2365. doi:10.1109/TVCG.2013.130.

[64] B. Lee, C. Plaisant, C. S. Parr, J. Fekete, N. Henry, Task taxonomy for graph visualization, in: Proceedings of the 2006 AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization, ACM, New York, 2006, pp. 1–5. doi:10.1145/1168149.1168168.
URL http://doi.acm.org/10.1145/1168149.1168168

[65] N. Andrienko, G. Andrienko, Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach, Springer-Verlag, Berlin, Heidelberg, 2005.

[66] J. Ahn, C. Plaisant, B. Shneiderman, A task taxonomy for network evolution analysis, IEEE Transactions on Visualization and Computer Graphics 20 (3) (2014) 365–376. doi:10.1109/TVCG.2013.238.
URL http://dx.doi.org/10.1109/TVCG.2013.238

[67] P. Murray, F. Mcgee, A. Forbes, A taxonomy of visualization tasks for the analysis of biological pathway data, BMC Bioinformatics 18 (21) (2017) 1–13. doi:10.1186/s12859-016-1443-5.

[68] R. Kosara, J. Mackinlay, Storytelling: The next step for visualization, Computer 46 (5) (2013) 44–50. doi:10.1109/MC.2013.36.

[69] F. Chevalier, M. Brehmer, G. Bolduan, C. Elmer, B. Wiederkehr, Data-Driven Storytelling, 1st Edition, A K Peters/CRC Press, Boca Raton, FL, 2018, Ch. From Analysis to Communication: Supporting the Lifecycle of a Story.

[70] H. O. Obie, C. Chua, I. Avazpour, M. Abdelrazek, J. Grundy, T. Bednarz, Pedaviz: Visualising hour-level pedestrian activity, in: Proceedings of the 11th International Symposium on Visual Information Communication and Interaction, VINCI '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 9–16. doi:10.1145/3231622.3231626.
URL https://doi.org/10.1145/3231622.3231626

[71] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, J. Heer, Vega-lite: A grammar of interactive graphics, IEEE Transactions on Visualization and Computer Graphics 23 (1) (2017) 341–350.

[72] A. Satyanarayan, R. Russell, J. Hoffswell, J. Heer, Reactive Vega: A streaming dataflow architecture for declarative interactive visualization,

IEEE Transactions on Visualization and Computer Graphics 22 (1) (2016) 659–668. `doi:10.1109/TVCG.2015.2467091`.

[73] Y. Kim, K. Wongsuphasawat, J. Hullman, J. Heer, Graphscape: A model for automated reasoning about visualization similarity and sequencing, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 2628–2638.

[74] J. Hullman, R. Kosara, H. Lam, Finding a clear path: Structuring strategies for visualization sequences, Computer Graphics Forum 36.

[75] J. Hullman, S. Drucker, N. Riche, B. Lee, D. Fisher, E. Adar, A deeper understanding of sequence in narrative visualization, IEEE Transactions on Visualization and Computer Graphics 19 (2013) 2406–2415.

[76] F. Amini, M. Brehmer, G. Bolduan, C. Elmer, B. Wiederkehr, Evaluating Data-Driven Stories & Storytelling Tools, A K Peters/CRC Press, 2018, Ch. 11.
URL `https://www.microsoft.com/en-us/research/publication/evaluating-data-driven-stories-storytelling-tools/`

[77] R. J. Sarmah, Y. Ding, D. Wang, C. Y. P. Lee, T. J.-J. Li, X. A. Chen, Geno: A developer tool for authoring multimodal interaction on existing web applications, in: Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology, UIST '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 1169–1181. `doi:10.1145/3379337.3415848`.
URL `https://doi.org/10.1145/3379337.3415848`

[78] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. H. Riche, C. Weaver, B. Lee, D. Brodbeck, P. Buono, Research directions in data wrangling: Visuatizations and transformations for usable and credible data, Information Visualization 10 (4) (2011) 271–288. `doi:10.1177/1473871611415994`.
URL `https://doi.org/10.1177/1473871611415994`

[79] S. Kandel, A. Paepcke, J. Hellerstein, J. Heer, Wrangler: Interactive visual specification of data transformation scripts, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, Association for Computing Machinery, New York, NY, USA, 2011, p. 3363–3372. `doi:10.1145/1978942.1979444`.
URL `https://doi.org/10.1145/1978942.1979444`

[80] I. Avazpour, J. Grundy, L. Grunske, Specifying model transformations by direct manipulation using concrete visual notations and interactive recommendations, J. Vis. Lang. Comput. 28 (C) (2015) 195–211. `doi:10.1016/j.jvlc.2015.02.005`.
URL `https://doi.org/10.1016/j.jvlc.2015.02.005`