DataToon: Drawing Data Comics About Dynamic Networks with Pen + Touch Interaction

Nam Wook Kim[1,2]   Nathalie Henry Riche[2]   Benjamin Bach[3]   Guanpeng Xu[4]   Matthew Brehmer[2]
Ken Hinckley[2]   Michel Pahud[2]   Haijun Xia[5]   Michael J. McGuffin[6]   Hanspeter Pfister[1]

[1] Harvard University   [2] Microsoft Research
[3] University of Edinburgh   [4] Philips Academy   [5] University of Toronto   [6] École de technologie supérieure
{namwkim, pfister}@g.harvard.edu   {nath, mabrehme, kenh, mpahud}@microsoft.com
bbach@inf.ed.ac.uk   gxu19@andover.edu   haijunxiadgp.toronto.edu   michael.mcguffin@etsmtl.ca
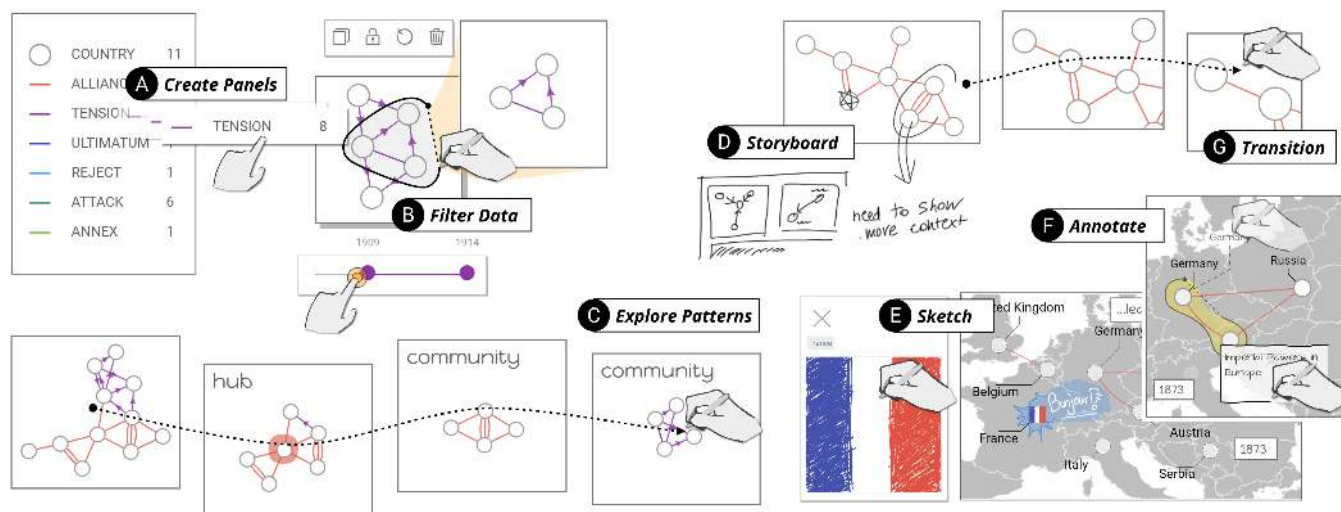
Figure 1: DataToon is a pen & touch environment for producing data comics. A storyteller can rapidly isolate aspects of their data via filtering and pattern detection, as well as assemble a rich narrative via annotation and automatic panel transitions.

ABSTRACT

Comics are an entertaining and familiar medium for presenting compelling stories about data. However, existing visualization authoring tools do not leverage this expressive medium. In this paper, we seek to incorporate elements of comics into the construction of data-driven stories about dynamic networks. We contribute DataToon, a flexible data comic storyboarding tool that blends analysis and presentation with pen and touch interactions. A storyteller can use DataToon to rapidly generate visualization panels, annotate them, and position them within a canvas to produce a visual narrative. In a user study, participants quickly learned to use DataToon for producing data comics.

CCS CONCEPTS

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

KEYWORDS

Data comics, pen + touch interfaces, direct manipulation, storytelling, dynamic networks, data visualization.

1 INTRODUCTION

Visualization is a pivotal component in data-driven storytelling, providing an audience with the means to understand patterns in data without requiring advanced statistical literacy [40]. One genre of data-driven storytelling [43] is the *data comic* [4], in which a narrative grounded in data is conveyed by leveraging the well-established visual language of comics [30]. Data comics integrate captions and annotations with visualization, suppressing the complexity of data by incrementally revealing aspects of the data across multiple panels, arranged thoughtfully on one or more pages [6, 47].

A recently curated collection of manually-created data comics [7] demonstrates the richness of this genre and its applicability to telling stories about datasets of different natures. From a storytelling standpoint, one of the most challenging forms of data is a dynamic network. Dynamic networks appear in many contexts, from analyzing social networks to modeling neural connections in the brain. In addition to evolving in time, such networks may contain multiple types of nodes and links exhibiting different connectivity patterns. Due to this complexity, it is notoriously difficult to communicate insights about dynamic networks to a general audience with a single large visual representation. Since conventional comics often illustrate the dynamic nature of characters and the interactions that occur between them over time by identifying and sequencing salient moments, dynamic networks are ideally suited for a comic treatment [3].

However, producing a data comic is a difficult and laborious process, one that involves switching between visualization and graphic design tools [9], the former being ideal for generating accurate data representations, and the latter being ideal for stylizing visual elements and arrange panels in space. While several recent tools support the construction of visual data stories [25, 37, 42, 51], they do not take advantage of the comic form as a storytelling medium. Thus authors have to resort to illustration and design tools such as Adobe Illustrator and Photoshop.

We contribute DataToon as a storytelling tool for producing data comics with a focus on dynamic networks. DataToon offers fluid storyboarding by blending analysis and presentation in a unified environment supported by pen and touch interactions. A storyteller can use DataToon to rapidly explore their data and generate visualization panels via interactive filtering and from recommendations of interesting data patterns, resulting in a visual story with custom annotations, automatic panel transitions, and layout templates.

The direct manipulation of panels and their data contents further facilitates the storytelling process. Natural touch interaction supports the iteration of story ideas by experimenting with different ways to compose panels and lay them out on a page. The use of a digital pen also allows storytellers to annotate panels with drawings and handwriting, or to draw custom glyphs for data entities. DataToon leverages the underlying data to eliminate the tedious duplication of actions necessary in conventional illustration software, such as propagating visual designs to other panels.

To demonstrate the expressivity of DataToon, we created a set of comics showing different rendering styles, panel layouts, visualization types, and narrative structures. Results from a reproduction study suggest that novice participants can successfully learn to use DataToon with minimal guidance to produce comics about dynamic networks. Insights about usability that led to improvements of DataToon included difficulties in discovering features, the inconsistency of interactions, and the complexity of visualization contents.

2 RELATED WORK

DataToon draws from and extends previous research and tool development relating to communicative visualization, data-driven storytelling, and pen and touch interaction.

Communicative Visualization

Although the visualization research community has been primarily devoted to the study of visualization in support of data analysis tasks, visualization has throughout history been used to communicate insight to an audience. Recent research has examined the aspects of memorability [10, 11], visual embellishment [8], and annotation [36] in the context of communicative visualization, which has in turn informed the design of increasingly expressive interactive visualization authoring tools. For example, tools like Lyra [42] or iVisDesigner [37] both provide a palette of graphical styling options that can be applied to a visualization. More recently, Data-Driven Guides [25], Data Illustrator [29], DataInk [51], and Charticulator [38] allow further expressivity in terms of custom visual marks and custom layouts, while tools like ChartAccent [36] or DataWrapper [17] provide rich annotation options. It is important to note that most of these tools are devoted to visualizing tabular data; Graph Coiffure [44] is an exception in that it provides an interface for visualizing, styling, and laying out static node-link diagrams. However, to our knowledge, there exists no interactive authoring tool for producing communicative visualization about dynamic networks, this being the purpose of DataToon.

Data-Driven Storytelling

Most communicative visualization tools allow for the production of one visualization at a time. While these tools may be sufficient for conveying simple messages about the data, they cannot support the design of a fuller narrative and thus their ability to produce a comprehensive story is limited.

Recent research has examined the integration of communicative visualization within a linear narrative sequence [22].

This research is reflected in another category of tools that focus on sequence and narration. These include commercial tools including Tableau's Story Points [45] and Bookmarks for Microsoft's Power BI [32], which provide interfaces for composing a sequence of story points with embedded visualizations. Meanwhile, tools emerging from the research community aim for greater expressivity. These include: Ellipsis [41] and Timeline Storyteller [12], which augment a sequence of visualizations with annotations and state-based scene transitions; DataClips [2], which focuses on sequencing data-driven video clips; and Vistories [18], which leverages the interaction history produced during data exploration to automatically generate a sequence that can be curated and annotated into a presentable story. In each of these tools, a set of annotated visualizations are arranged in a linear narrative sequence, revealed one at a time via stepping or scrolling interactions [31].

Unlike linear slideshows and scroll-based stories, the layout and juxtaposition of panels in a comic allows for non-linear narrative structures, in which a reader can consume narrative points in various orders in a glanceable format that affords both skimming and revisitation. Unfortunately, no single existing data-driven tool can produce such narrative structures. The sole existing data comics editor by Zhao and Elmqvist [53] allows for the composition of linear slideshow comics and the embellishment of visualization with speech bubbles and a narrator character. However, like illustration software, this tool requires the importation of preexisting visualization generated by other tools. In contrast, we provide the first all-in-one visualization and narrative design tool where multiple panels can be arranged freely on a page.

Another assumption inherent to many existing data-driven storytelling tools is that the storyteller already has a preconceived story, perhaps developed in the course of data analysis performed with other tools, in consultation with data analysts and subject matter experts, or in some combination thereof. However, this separation of analysis and storytelling hampers rapid experimentation of alternative narrative structures and the process of refining a story. In other words, dedicated analysis tools often do not have flexible storytelling features while dedicated storytelling tools lack data exploration capabilities such as ways to collect and organize insights. One of the benefits of an interface that allows for the flexible arrangement of comic panels is that storytellers can rapidly iterate with alternative narrative structures. Furthermore, they can quickly generate and discard panels the process of data exploration without disrupting completed parts of the comic. Finally, DataToon integrates automatic suggestions and transitions between panels as a way to scaffold a story, eliminating the tedium of alternating between a dedicated visualization tool and a dedicated storytelling tool.

Pen + Touch Interaction for Creativity Support

Historically, comics have been drawn by hand, and thus we gravitated to interfaces that could leverage expressive pen-based input for drawing and styling comic elements. Such interfaces have become increasingly popular in recent years, and along with them we have seen an emerging body of research that focuses on the combination of pen and touch interaction for content creation and manipulation. By combining these forms of interaction, users report feeling more directly engaged as compared to manipulating elements via a WIMP interface [51]. Hinckley et al. introduced a rich palette of compelling interaction techniques for manipulating content, all following the principle that the pen writes and touch manipulates [21, 34]. Other research has sought to identify and evaluate pen and touch gestures for common operations on interactive surfaces, including selection, deletion, and copy / paste [33], and these gestures have applied such gestures in various applications including diagram editing [16], digital drawing [48, 49], early-stage ideation [50], and active reading [20]. Visualization researchers are also beginning to take advantage of touch and pen interaction in various contexts [27], including visualization authoring [51], storytelling [28], and data exploration [23, 52], though until DataToon, they have yet to apply such interaction to the creation of data comics.

3 DESIGN GOALS

We conceive DataToon to accomplish three main goals.

D1. Support the creation of data *comics*. Data comics have unique characteristics and components [4, 6]. They expose the story via a juxtaposed and sequenced panels, each containing one (or a few) insight(s). Getting the reader to focus on each insight requires that the author carefully crafts the view of each panel, such as zooming in on an important part of a network. However, identifying different characters of interest (e.g., nodes) and tracking them across panels requires that the author gives each a custom visual style to maintain consistency [35]. DataToon aims to support the crafting of panels and the expressive design of characters by direct manipulation using pen and touch, while ensuring consistency with data-driven propagation.

D2. Enable data-driven design. Authoring tasks such as propagating visual designs across panels or creating transitions between panels are tedious and time-consuming. DataToon leverages the underlying data to automatically propagate visual designs of graphical elements and to generate textual labels and captions from data. DataToon also enables an author to generate transitional panels. For example, given two panels containing data at different times, DataToon can automatically create a series of panels representing the data at interim time points.
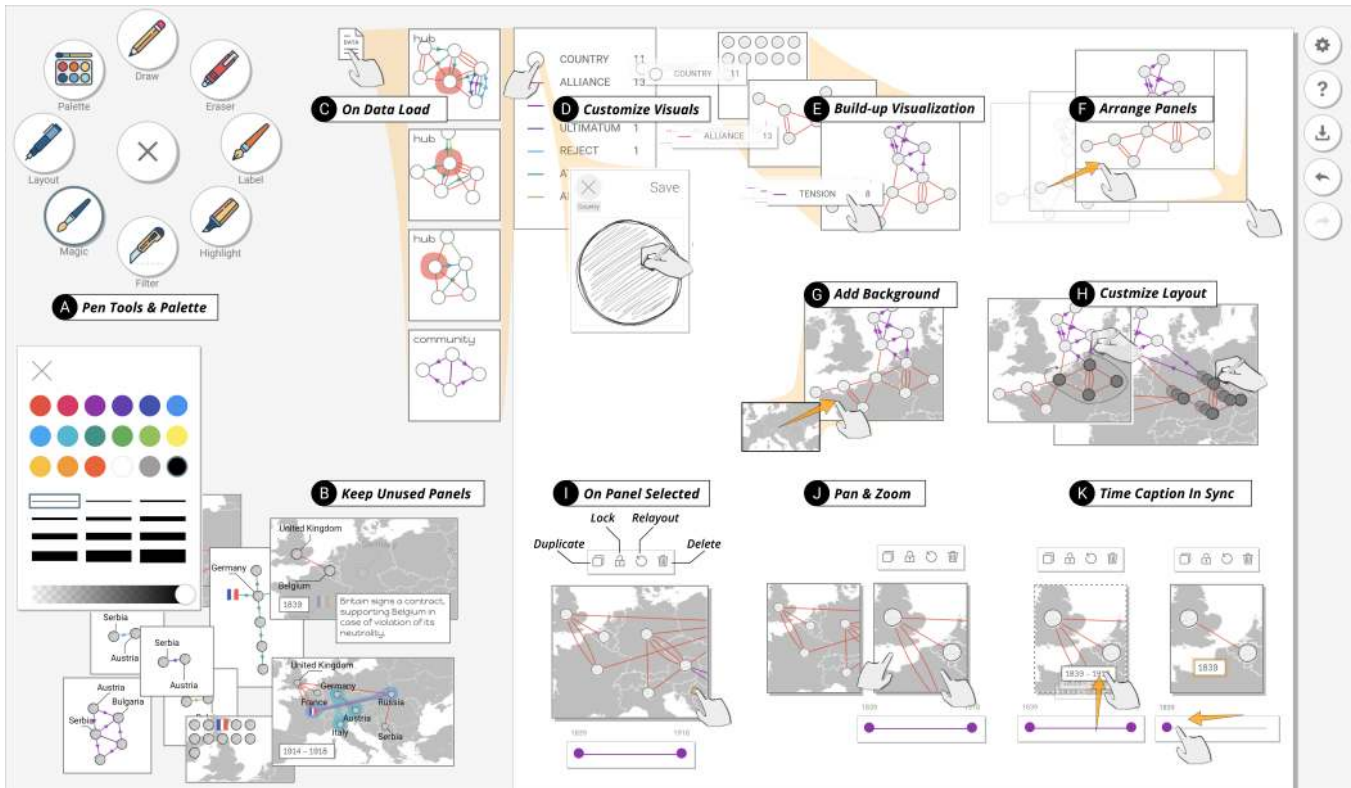
Figure 2: DataToon's interface: the pen can acquire different functions, such as labelling or filtering. The canvas area provides an infinite space for ideation and exploration, as well as a dedicated page area for presentation.

D3. Support exploration and authoring activities. Like any storytelling medium, the production of a comic is a creative and iterative process, often requiring the author to smoothly transition from reviewing the data and its patterns to styling them to craft a compelling story. DataToon facilitates the data exploration and review process by providing recommendations of interesting patterns in the data, such as a large cluster in the network, while enabling the rapid creation of visualizations using direct pen and touch manipulation. DataToon enables flexible workflows by providing a unique platform in which authors can review salient data aspects, rapidly generate and filter data visualizations, craft expressive visual design for data elements, compose a story by leveraging existing data comic templates, create a storyboard, or directly sequence and reorder panels.

4 USAGE SCENARIO

To illustrate how DataToon accomplishes our goals and to describe key components of its interface, we describe the process taken by a hypothetical comic author named Aidan to create the comic in Figure 4, adapted from Bach et al. [3].

Aidan opens DataToon in his web browser on a pen + touch-enabled device. The pen-tools menu (Figure 2A) represents the set of instruments his digital pen can acquire. Aidan loads a dataset he previously created: a dynamic geographic and multi-faceted network containing countries and their alliances and oppositions over time. Dragging the file into the application instantiates both the legend panel and automatically generates a set of panels depicting notable structural patterns in the data (Figure 2C).

Aidan recalls that the evolution of alliances was interesting and he decides to explore these. He creates a new content panel by dragging the *Country* node type from the legend panel onto the canvas. Using the time slider for these panels (Figure 1B), he filters ✎ the data back and forth to explore how the alliances changed over time. He duplicates panels as he finds interesting times, jotting down notes on them using the pencil ✎ (Figure 1D).

Continuing this process of exploration, Aidan now has multiple panels with annotated insights. He proceeds to craft a story for his comic by organizing them on the canvas. He wants to start with an overview of the nations involved, so he drops an image file containing a map of Europe (Figure 2G) into a panel, adjusting the location of each node with the

Layout pen ✏ (Figure 2H). He extracts country names from the data to place them on the map with the *Label* pen ✏.

After establishing the context of the story, Aidan wants to show the evolution of alliances in Europe. He reuses panels that he created earlier, transforming his rapid handwritten annotations into visual clusters around nodes, captions and labels. While he pans and zooms his first panel to emphasize two nodes (Figure 2J), he realizes that the difference between this panel and the first one may be too great and that his audience may fail to see the connection. He acquires the *Magic* pen ✏ to interpolate between these panels and generate intermediary ones (Figure 1G).

He generates a time caption for the last panel by dragging the time label from the slider (Figure 2K, left). Duplicating this panel (Figure 2I) and adjusting the time automatically updates the caption (Figure 2K, right). Uncertain about what to show next, Aidan uses the *Magic* pen ✏ to trigger suggested panels with interesting patterns (Figure 1C).

As his comic nears completion, he customizes the node and link representations. For instance, he draws a custom sketchy representation for all nodes (Figure 2D). To emphasize France among all countries, he paints its flag (Figure 1E). This custom node representation is automatically propagated to all panels. Satisfied with his comic, he exports the comic as an image that he can share with his students.

5 DATATOON

We designed DataToon for a broad range of people who wish to craft data comics that communicate insights about their data. This may include graphic designers without programming expertise, data analysts without design expertise tasked with communicating their findings, or educators seeking new ways to engage their students.

Data Abstraction

As mentioned in the introduction, we chose to focus on dynamic network data since it is poorly supported by existing communicative visualization and storytelling tools, and because of its inherent parallel to the dynamic interactions between characters appearing in comics. In particular, DataToon supports multivariate dynamic network data, which consists of nodes and links and their attributes. A node can have a label and a type, as well start and end dates. A link must have source and target nodes and may have a direction and a weight along with same set of attributes describing nodes. Given this criteria, DataToon also supports static networks, where neither nodes nor links have associated start or end times. Note that DataToon is primarily a storytelling tool, one suitable for communicating different aspects of dynamic network data; we do not address scalability issues and analysis tasks related to very large networks in this paper.

Story Abstraction

Like a comic book, a data comic consists of pages (Figure 4), in which each page can contain multiple panels. A panel is the essential building block of a narrative structure, which can in turn contain visualization, images, text captions, and annotation. The spatial arrangement of panels having varying size and content generates a unique narrative flow, enabling a nonlinear reading experience unlike linear sequences produced by other storytelling tools.

Interaction Design

DataToon is comprised of a canvas for storyboarding; content and legend panels for presenting visual representations of data; a set of pen tools for content creation and manipulation; and a contextual canvas for sketching custom visuals.

Figure 3 summarizes our pen + touch interactions. In general, our interaction design choices reflect the mantra: *the pen writes, touch manipulates* [21]. However, since individual nodes and links within panels are often too small to manipulate with a finger, the pen is also used to manipulate visual elements in some circumstances, as described below. Throughout the interface, we chose to visually expose interactive controls rather than rely on implicit multi-touch gestures that are difficult to discover and remember. Note that we describe our final system, which improves upon the version used in our reproduction study described below.

Interacting with the canvas. DataToon provides an infinite canvas to support flexible authoring and rapid ideation, transitioning from data exploration to authoring activities (D3). The author can navigate the canvas via pan and zoom gestures. Meanwhile, using the pen, the author can draw or write anywhere on the canvas, either to annotate content or to add storyboarding notes and ideas. The author can create an empty panel by simply drawing a rectangle, to be filled later with content, or create panels from data. Panels can be freely arranged and resized with touch interactions, leading to different layouts at any point in the authoring process.

Interacting with the legend panel. A legend panel is created when the author drags a dataset file onto the canvas. This panel provides an overview of the dynamic network, displaying a list of node types and link types along with the cardinality of each type. This legend also serves as an interface for creating content panels. Dragging a node or link type from the legend onto the canvas creates a new content panel displaying a filtered visualization of the data. Dragging a node type automatically creates a unit chart of all nodes in the data matching this type. Dragging a link type automatically creates a force-directed node-link diagram of nodes connected by this link type. Note that links can convey both
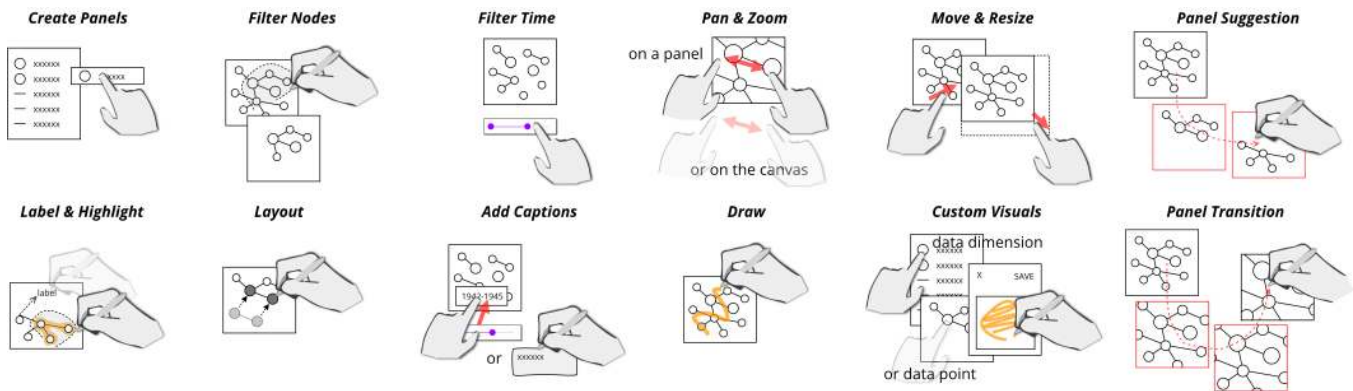
Figure 3: An overview of the pen + touch interactions supported by DataToon.

weight and direction via line thickness and arrows, should these optional attributes be provided.

Node and link types can also be dragged from the legend panel to an existing content panel, whereupon its contents are updated to reflect the additional type (Figure 2E) and its layout recomputed. For instance, dragging a link type to a content panel containing a unit chart will convert the chart into a node-link diagram. Similarly, adding a link type to a panel containing a node-link diagram will generate multiple link types (see Figure 2).

Interacting with content panels. A content panel may contain visualizations, text, annotations, a background image, or some combination thereof (D1). Panels can be nested: drawing a rectangle inside a panel creates a child panel, which is useful for text captions or inset visualizations. It is possible to duplicate an existing panel, copying all of the content of the source panel to a new panel (Figure 2I). This interaction is particularly useful for progressively building a narrative using the previous panel as a starting point.

Tapping on a panel containing a visualization selects it and enables panning and zooming within the panel. This also reveals a time slider for the panel, which applies additional temporal filtering to nodes and links displayed within the panel. Dragging this slider onto the panel creates a nested time caption panel (D2), which remains updates as the user changes the time slider (Figure 2K).

Acquiring different pen modes. Content editing occurs through the use of the following pen modes:

✏️ **Pencil** mode is the default pen mode and allows for freeform inking on the canvas. If the author draws atop a content panel, the ink belongs to the panel and moves along with it when that content panel is manipulated.

🖌️ **Label** mode generates a text label for nodes and links and allows for adjusting label positions (Figure 1F). Label and leader lines move when the element is moved.

🖊️ **Highlight** mode allows the author to lasso a set of nodes to highlight them in a colored group (Figure 1F). As with labels, highlights also adjust when elements are moved.

🖊️ **Filter** mode allows the author to lasso a set of nodes to filter them from a panel or to create a new panel with these nodes (Figure 1B). Filtering can de-clutter panels and help focus on different subsets of the network.

🖊️ **Layout** mode leverages the high-precision of the digtial pen to adjust the positions of nodes and labels (Figure 2H).

🖌️ **Magic** mode offers a way to automatically generate content panels (Figure 1C, D), described in more detail below.
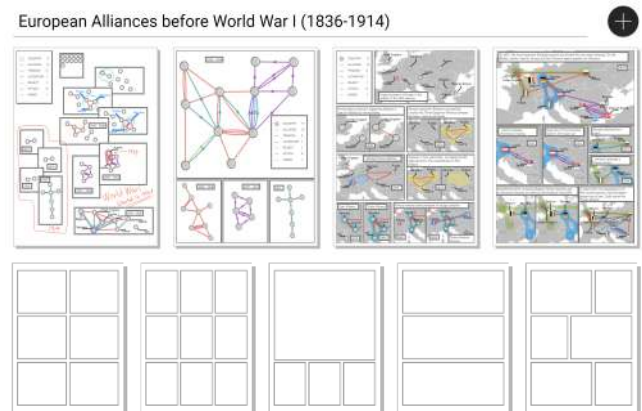

Figure 4: DataToon uses a comic book metaphor to allow authors to create multiple pages of data comics using more than one dataset. Each page can be created with a a predefined layout such as the ones shown in the second row.
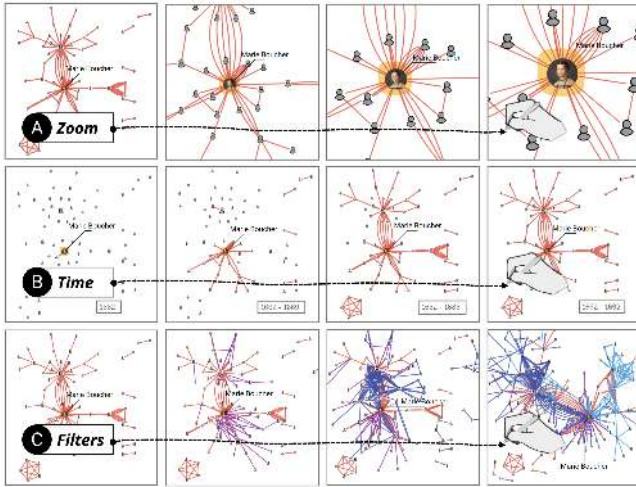
Figure 5: Automatic transitions between panels involve interpolating differences between panels, incorporating zoom levels, time ranges, filters, and combinations thereof.

✏️ **Eraser** mode deletes any item on the canvas including panels, ink, annotations, and highlights. This mode can also be invoked via the pen's eraser button, should it have one.

🎨 **Palette** mode allows the author to choose a different color, line thickness, and fill opacity, which will be applied to subsequent pencil strokes, annotations, and highlights.

Drawing custom node and link representations. A secondary canvas is invoked when the author taps on the iconic representation of a node or link type in the legend (Figure 2D). Within this canvas, the author can draw a new iconic representation for the selected node or link type, and this custom representation is immediately propagated across the comic for consistency (D2). The author can also invoke this canvas to change the representation of a single node or link, by tapping it with the pen while in pencil mode (✏️).

Facilitating Ideation

DataToon provides several ways to scaffold and accelerate the data comics creation process (D1):

1. Multiple pages and layout templates facilitate the creation of multiple iterations of comics (Figure 4). Pages can load different datasets or contain different notes. Templates are a set of panel layouts commonly used in comics [6] such as grid, overview+detail, parallel, and staggered. When selected, the new page is auto-populated with empty content panels specified by the template. Authors can simply drag data on top of them to fill them.

2. Automatic transitions. DataToon creates intermediary panels by interpolating the difference between the two panels, taking into account their respective panel sizes, data

filters, and zoom states (Figure 5). These transitions may also incorporate a temporal progression between two time filter states and the addition or removal of nodes and links [6]. It does not, however, attempt to interpolate annotations between two panels.

As an example, if a source panel and a target panel have zoom factors of 1.0 and 1.8 respectively, intermediary panels will have interpolated zoom factors between 1.0 and 1.8. In addition, if the source and target panels have different sizes and display data at different time periods, DataToon will interpolate over these dimensions as well, producing intermediary panels that would gradually increase in size and depict a progression over time intervals.

DataToon places intermediary panels along the path drawn by the author. Greater distances between panels results in more intermediary panels along the path. It discretizes the path and conducts a linear search for panel positions, while ensuring no overlap between intermediary panels.

3. Panel suggestions for new panels are shown in two contexts: either when loading a new dataset or when drawing a line out of a panel using the Magic pen. In the first case, it attempts to detect interesting subnetwork patterns in the whole dataset and places suggested panels at the left side of the canvas. This helps authors become familiar with the dataset and serves to kickstart the story design process. In the second case, it takes into account the current state of the panel and detects patterns within the panel to generate suggestions. It thereby enables authors to discover potential compelling directions for their story. Similar to transitions, these suggested panels are placed along the path the author draws, while preventing overlap between them.

To detect patterns in the network data, DataToon relies on a pattern detection engine. The engine accepts any network data including highlighted nodes as input and returns a list of detected patterns. It currently looks for four structural patterns including articulation points (or bridges), cliques, hubs, and communities (Figure 6). Once patterns are found, it heuristically prunes the results such as by removing overlapping patterns (e.g., a bridge and a hub can often show a completely identical structure), as well as cliques with less than four nodes. Finally, it ranks the patterns based on how much each pattern overlaps with the source data. This is to promote closure between the source panel and the suggested panel. Finally, the rankings give a high priority to patterns that contain highlighted nodes (Figure 6 C).

Implementation Details

DataToon is a browser-based application written in Javascript. It uses React.js for building user interface components and Redux.js for application state management. It uses Web-CoLa [14] to generate the layout of the node-link diagram
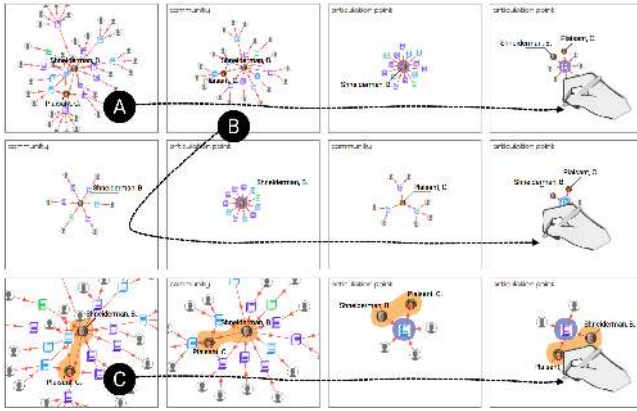
Figure 6: Automatic panel suggestions depicts structural patterns: communities, hubs, articulation points, and cliques. The author can trigger suggestions from existing panels (A) and suggested panels (B). Patterns are ranked based on network coverage and inclusion of highlighted nodes (C).

and a Javascript implementation [26] of Bubble Sets [13] to highlight a group of nodes as a cluster. DataToon consists of a front-end interface without a back-end server, though one can be attached if needed; currently, DataToon makes use of localStorage and indexedDB in HTML5 to persist the application state. The panel recommendation engine is implemented in Python and uses NetworkX to detect patterns [19].

6 EVALUATION

Our evaluation methodology is representative of other recent evaluations of visualization authoring systems [39], in that we demonstrate the *expressiveness* of DataToon with an example gallery (Figure 7) and also evaluate its *learnability* and *usability* via a reproduction study.

Example Gallery

Figure 7 shows example comics varying in their comic style, including diverse rendering styles (abstract, sketchy, realistic), panel layouts (inset, overlapping, staggered) [6], visualizations (unit charts or pictographs, maps, set visualizations, node-link diagram), and narrative structures (overview+detail, nonlinear-temporal, cut-out, build-up [6]). The gallery also exemplifies a diversity of datasets, including multivariate and temporal social networks and co-authorship networks.

Reproduction Study

To evaluate whether people can learn how to use DataToon to create comics from data, we conducted a qualitative reproduction study, in which participants are asked to reproduce them completed examples with DataToon. This type of study is particularly common in the evaluation of visualization authoring tools [39], having been used to evaluate Lyra [42],

ChartAccent [36], DataInk [51], Data-Driven Guides [25], Data Illustrator [29], and Charticulator [38].

Participants. We recruited eight participants from a large software company in the United States. Half of the participants were graphic designers with limited data literacy (P1-P4: 3M, 1F; ages 30−50, avg: 44), while the other half were data analysts with minimal experience in design tools (P5-P8: 2M, 2F; ages 31−42, avg: 37).

Apparatus. Participants used a earlier version of DataToon on a Microsoft Surface Studio with a 28-inch screen at 4500 × 3000px (192 PPI), a device that enables simultaneous pen and multi-touch input.

Procedure and tasks. Beginning with a demographic survey, each study session lasted ~90 minutes, with two participants finishing in ~60 minutes, and one taking ~120 minutes. We asked them to reproduce two comics: 1) the first with guidance from us using the comic about *World War I alliances* (Figure 2) and 2) the second without any assistance using the comic inspired by Fathom's *Scaled in Miles* project [15] about the evolving instrumentation on Miles Davis' records (Figure 7-left). The first task served as a training session and lasted 30 to 40 minutes, which included a 15-minute tutorial, while the second task lasted between 30 and 50 minutes. The study ended with three Likert-style questions about ease of use & learning, and enjoyment, along with a semi-structured interview about their experience.

Observations

All participants successfully completed both comics, while we discovered several usability insights into the usability of DataToon. We describe our observations below.

Learning to interact with both pen and touch. All of the participants appeared to grasp DataToon's interaction design by the end of the study, except for P2, who had no prior experience with pen + touch devices. It took a long time (approx. 10 min) for P2 to complete the first task and the effort spent to learn the interactions are reflected in their low ease of learning (3/7) and use (4/7) scores. P4, P5, P6, and P7 also repeatedly appeared to be frustrated when attempting to use pen and finger interchangeably, with P7 stating *"I kept using my hand instead of the pen"*.

Participants bimanual pen and touch interaction to be engaging, with P8 remarking on the simplicity of the interactions: *"I love the power of just dragging* [shows fingers] *and creating* [shows the pen]*"*. P4 spoke about the empowering experience of bimanual pen and touch input for content creation, making DataToon *"unique"* and *"fun"* compared to other tools: *"I feel like a surgeon because I got precise and used both of my hands, not something I do ever. It's pretty cool!"*.

A focused tool set for design exploration. The graphic design participants all expressed that one notable strength of DataToon was a *"focused tool set"* (P1), its interface *"streamlining the set of tools"* (P4) compared to existing illustration tools. We observed that our interface enabled alternative workflows to achieve the same result, reflecting what Ren et al. refer to as the *flexibility* of a visualization authoring tool [39]. For example, several participants began with multiple panels, adjusted the content of each panel before customizing each of them in turn. Others would create and modify one panel until it was polished, only then duplicating to instantiate the next panel.

Participants' difficulties often related to feature discoverability, as not every pen mode was visually shown in the interface. For example, in the version of DataToon used in the study, the pen button was used to activate the highlight pen. P8 commented that *"Minimalism is in, it looks just like a simple drawing app, but then it can be intimidating because how do you achieve all of this?"* [pointing to print out of the data comics] *I was nervous"*. Similarly, P1 commented that the principle of dragging and dropping elements into panels was violated in the case of time captions, which required a double tap, making it challenging to discover.

Closing the gap between analysis and storytelling. Participants appreciated the ability to discover patterns during the story authoring process, suggestive of a possible advantage over visualization authoring tools that are disjoint from data analysis tools. We observed that data analysts tended to explore the data before constructing on their comics. For example, P5 started by creating many panels (one per node type) and by commenting on the structural patterns in the data. P8 used a different strategy, adding each node type to a single panel in succession, where each node type was a different instruments featured on Miles Davis records; upon doing so, P8 stated that *"now I am beginning to see the relationships between instruments [...] I am going to move things around so I can understand my data"*. Finally, some participants noted the necessity of additional data abstraction. For example, P3, looking at a particularly dense node-link diagram, said *"I wish there were a way to untangle that because that is a super full graph"*, suggestive of a need for capabilities that aggregate nodes and links.

Lessons Learned from the Reproduction Study

The results of the study illuminated a set of usability insights regarding the difficulty of discovering features, the inconsistency of pen and touch interactions, and the complexity of visualization contents. These insights led to several improvements to the design of DataToon.

To address the feature discoverability issue, we ensured that all pen modes are visible (Figure 2A) without cluttering the interface and degrading the authoring experience. In addition, after observing participants repeatedly attempting to use fingers where we enforced use of the pen, which initially included panning and zooming panel content, we opted to accommodate more touch interaction, reflecting *the pen writes, touch manipulates* mantra [21]. We also replaced the double-tap gesture for creating time captions with a consistent drag gesture. Finally, to handle the visual complexity issue, we added the ability to filter nodes of interest from a panel, as well as the panel suggestion functionality for assisting with exploring complex data.

7 DISCUSSION AND FUTURE WORK

We now reflect on broader issues and opportunities that arose during the development and evaluation of DataToon.

Generalization to other data and visualization types. The design of DataToon (See Figure 3) is mostly data-type-agnostic and generalizable beyond network data, such as gestures, panel manipulation, annotations, time sliders, highlighting and removing data elements, etc. Accommodating other data types (e.g., tabular data), visualizations (e.g., bar charts), and specific components (e.g., axes and scales) is an interesting research direction. The overarching research question in this space would be how to enable fast and fluid creation and manipulation of such panels, as visualizations can involve complex grouping and filtering operations on data. However, core "comic" features such as automatic propagation of style, automatic generation of transitions and panel recommendation would require minimum redesign.

Pen + touch interaction for data-driven storytelling. Pen + touch interaction was seen as engaging and flexible by our study participants, advantages that may prove beneficial beyond data comics to other data-driven storytelling experiences. While novices require time to acclimate to pen + touch interfaces, we observed that after an initial learning phase, this input modality stimulates creativity and encourages experimentation. However, to promote learning and discoverability, we must design appropriate visual cues and affordances that remind users of their capabilities. In our study, we observed that people initially want to use pen and touch interchangeably to accomplish a single action. This observation warrants further investigation and a revisitation of the *pen writes, touch manipulates* mantra [21].

Beyond traditional comics. DataToon exports pages as static images, like traditional comic books. While being respectful of this tradition, creating dynamic and interactive data comics is an interesting research direction. For example, a "presentation mode" might allow for presenters or viewers to touch parts of the comic and reveal content on demand, or
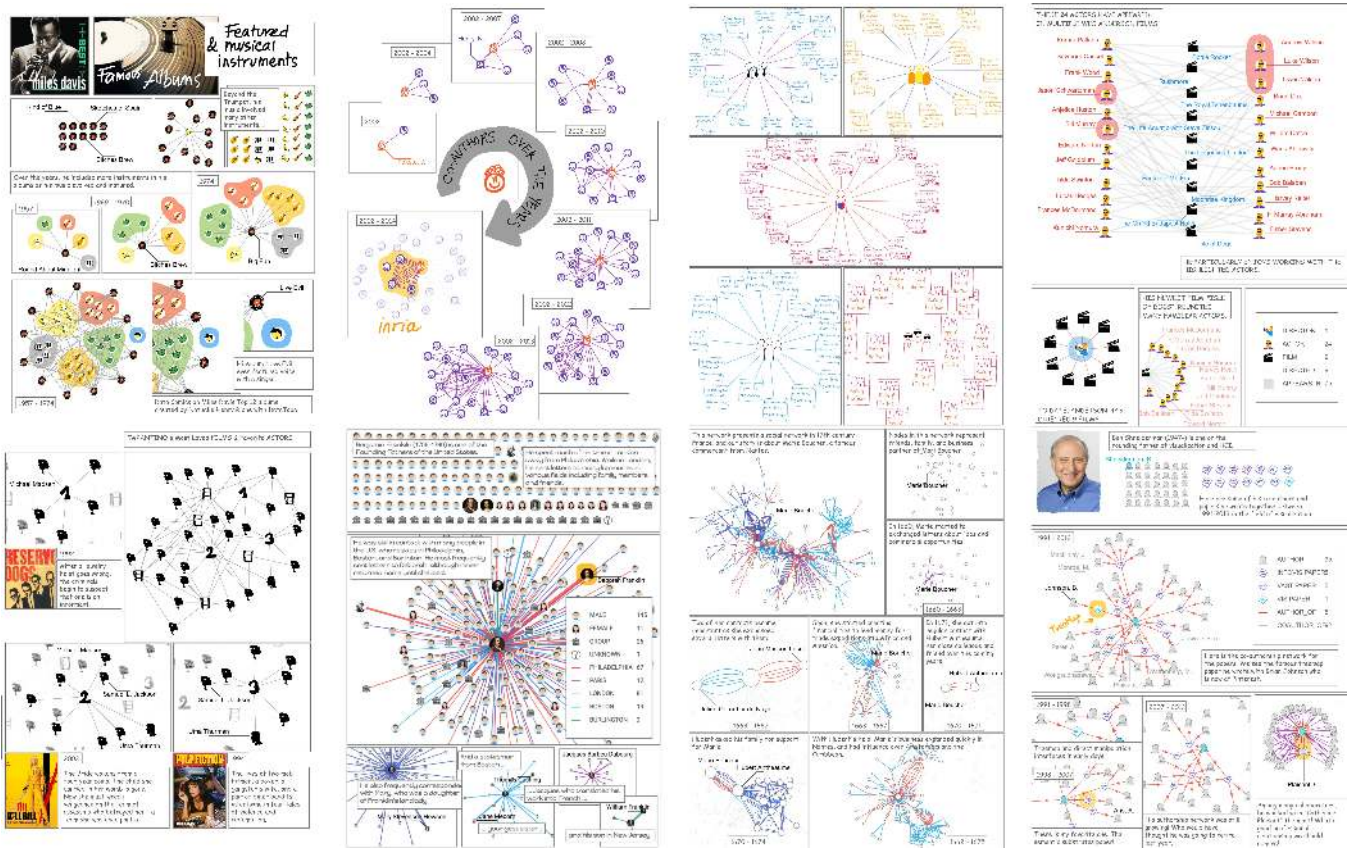
Figure 7: A gallery of eight data comics created with DataToon using different comic styles and datasets.

add annotations as part of an active reading process [46]. Alternatively, DataToon could export comics as websites that invite viewer interaction, potentially by integrating techniques such as brushing and linking across panels.

Toward higher-level narrative design support. Our design considerations emphasized use of the structural elements of comics such as panels and captions to convey a data story. However, producing an engaging story still depends on the contents of the data and the creativity of the author. DataToon does not explicitly incorporate higher-level narrative design patterns [5, 6] into its interface.

The automatic transition, suggestion, and layout template features are a step toward narrative design guidance, but there are further opportunities to improve . For example, we might auto-populate an entire template with visualization content as a way to seed a story, though we must take care to not absolve authors of their creative agency. Similarly, generating panel transitions that precisely match an author's narrative intent is challenging, but such transitions can be used as a way discover new narrative directions. Also, being

able to quickly evaluate the overall narrative structure will greatly aid the iterative process of crating a story [24].

8 CONCLUSION

We contributed DataToon, an interactive system for producing comics about dynamic networks. It leverages the form of comics to construct a narrative structure and offers a flexible pen + touch authoring interface for content creation and manipulation. DataToon provides automatic transitions and panel recommendations for narrative ideation and accelerated storyboarding. We plan to extend our evaluation to study the authoring process in a more longitudinal *free-form study* [39], focusing on comprehensive evaluation metrics for visualization authoring tools [1].

9 ACKNOWLEDGMENTS

REFERENCES

[1] Fereshteh Amini, Matthew Brehmer, Gordon Bolduan, Christina Elmer, and Benjamin Wiederkehr. 2018. Evaluating data-driven stories & storytelling tools. In *Data-Driven Storytelling*, Nathalie Henry Riche, Christophe Hurter, Nicholas Diakopoulos, and Sheelagh Carpendale (Eds.). A K Peters/CRC Press. https://aka.ms/dds_book.

[2] Fereshteh Amini, Nathalie Henry Riche, Bongshin Lee, Andres Monroy-Hernandez, and Pourang Irani. 2017. Authoring data-driven videos with DataClips. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 23, 1 (2017), 501–510. http://doi.org/10.1109/TVCG.2016.2598647.

[3] Benjamin Bach, Natalie Kerracher, Kyle W. M. Hall, Sheelagh Carpendale, Jessie Kennedy, and Nathalie Henry Riche. 2016. Telling stories about dynamic networks with graph comics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. 3670–3682. http://doi.org/10.1145/2858036.2858387.

[4] Benjamin Bach, Nathalie Henry Riche, Sheelagh Carpendale, and Hanspeter Pfister. 2017. The emerging genre of data comics. *IEEE Computer Graphics and Applications (CG&A)* 37, 3 (2017), 6–13. http://doi.org/10.1109/MCG.2017.33.

[5] Benjamin Bach, Moritz Stefaner, Jeremy Boy, Steven Drucker, Lyn Bartram, Jo Wood, Paolo Ciuccarelli, Yuri Engelhardt, Ulrike Köppen, and Barbara Tversky. 2018. Narrative design patterns for data-driven storytelling. In *Data-Driven Storytelling*, Nathalie Henry Riche, Christophe Hurter, Nicholas Diakopoulos, and Sheelagh Carpendale (Eds.). A K Peters/CRC Press. https://aka.ms/dds_book.

[6] Benjamin Bach, Zezhong Wang, Matteo Farinella, Dave Murray-Rust, and Nathalie Henry Riche. 2018. Design patterns for data comics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. https://doi.org/10.1145/3173574.3173612.

[7] Benjamin Bach, Zezhong Wang, Nathalie Henry Riche, Matteo Farinella, Dave Murray-Rust, Sheelagh Carpendale, and Hanspeter Pfister. 2018. Data Comics. http://datacomics.net.

[8] Scott Bateman, Regan L Mandryk, Carl Gutwin, Aaron Genest, David McDine, and Christopher Brooks. 2010. Useful junk?: The effects of visual embellishment on comprehension and memorability of charts. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. 2573–2582. https://doi.org/10.1145/1753326.1753716.

[9] Alex Bigelow, Steven Drucker, Danyel Fisher, and Miriah Meyer. 2017. Iterating between tools to create and edit visualizations. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 23, 1 (2017), 481–490. http://doi.org/10.1109/TVCG.2016.2598609.

[10] Michelle A Borkin, Zoya Bylinskii, Nam Wook Kim, Constance May Bainbridge, Chelsea S. Yeh, Daniel Borkin, Hanspeter Pfister, and Aude Oliva. 2016. Beyond memorability: Visualization recognition and recall. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 22, 1 (2016), 519–528. https://doi.org/10.1109/TVCG.2015.2467732.

[11] Michelle A Borkin, Azalea A Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. 2013. What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 19, 12 (2013), 2306–2315. https://doi.org/10.1109/TVCG.2013.234.

[12] Matthew Brehmer, Bongshin Lee, Nathalie Henry Riche, David Tittsworth, Kate Lytvynets, Darren Edge, and Christopher White. 2019. Timeline Storyteller: The Design & Deployment of an Interactive Authoring Tool for Expressive Timeline Narratives. In *To appear in proceedings of the the Computation + Journalism Symposium*. https://aka.ms/TSCJ19.

[13] Chistopher Collins, Gerald Penn, and Sheelagh Carpendale. 2009. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1009–1016. http://doi.org/10.1109/TVCG.2009.122.

[14] Tim Dwyer. 2018. WebCoLa. http://ialab.it.monash.edu/webcola/. [Online; accessed 31-03-2018].

[15] Fathom. 2015. Scaled in Miles. https://fathom.info/miles/. [Online; accessed 11-Sep-2018].

[16] Mathias Frisch, Jens Heydekorn, and Raimund Dachselt. 2009. Investigating multi-touch and pen gestures for diagram editing on interactive surfaces. In *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces (ITS)*. 149–156. http://doi.org/10.1145/1731903.1731933.

[17] Datawrapper GmbH. 2018. Datawrapper GmbH. https://www.datawrapper.de. [Online; accessed: 2018-08-04].

[18] Samuel Gratzl, Alex Lex, Nils Gehlenborg, Nicola Cosgrove, and Marc Streit. 2016. From visual exploration to storytelling and back again. In *Computer Graphics Forum (Proceedings of EuroVis)*. 491–500. http://doi.org/10.1111/cgf.12925.

[19] Aric Hagberg, Dan Schult, and Pieter Swart. 2018. NetworkX. https://networkx.github.io/. [Online; accessed 31-03-2018].

[20] Ken Hinckley, Xiaojun Bi, Michel Pahud, and Bill Buxton. 2012. Informal information gathering techniques for active reading. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. 1893–1896. http://doi.org/10.1145/2207676.2208327.

[21] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. 2010. Pen + touch = new tools. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. 27–36. http://doi.org/10.1145/1866029.1866036.

[22] Jessica Hullman, Steven Drucker, Nathalie Henry Riche, Bongshin Lee, Danyel Fisher, and Eytan Adar. 2013. A deeper understanding of sequence in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2406–2415. https://doi.org/10.1109/TVCG.2013.119.

[23] Jaemin Jo, Sehi L'Yi, Bongshin Lee, and Jinwook Seo. 2017. TouchPivot: Blending WIMP & post-WIMP interfaces for data exploration on tablet devices. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. 2660–2671. http://doi.org/10.1145/3025453.3025752.

[24] Nam Wook Kim, Benjamin Bach, Hyejin Im, Sasha Schriber, Markus Gross, and Hanspeter Pfister. 2018. Visualizing nonlinear narratives with story curves. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 24, 1 (2018), 595–604. http://doi.org/10.1109/TVCG.2017.2744118.

[25] Nam Wook Kim, Eston Schweickart, Zhicheng Liu, Mira Dontcheva, Wilmot Li, Jovan Popovic, and Hanspeter Pfister. 2017. Data-driven guides: Supporting expressive design for information graphics. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 23, 1 (2017), 491–500. http://doi.org/10.1109/TVCG.2016.2598620.

[26] Josua Krause. 2018. Bubble Sets. https://github.com/JosuaKrause/bubblesets-js. [Online; accessed 31-03-2018].

[27] Bongshin Lee, Petra Isenberg, Nathalie Henry Riche, and Sheelagh Carpendale. 2012. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 18, 12 (2012), 2689–2698. https://doi.org/10.1109/TVCG.2012.204.

[28] Bongshin Lee, Rubaiat Habib Kazi, and Greg Smith. 2013. SketchStory: Telling more engaging stories with data through freeform sketching. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 19, 12 (2013), 2416–2425. http://doi.org/10.1109/TVCG.2013.191.

[29] Zhicheng Liu, John Thompson, Alan Wilson, Mira Dontcheva, James Delorey, Sam Grigg, Bernard Kerr, and John Stasko. 2018. Data Illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. https://doi.org/10.1145/3173574.3173697.

[30] Scott McCloud. 1993. *Understanding Comics: The Invisible Art.* Harper Collins.

[31] Sean McKenna, Nathalie Henry Riche, Bongshin Lee, Jeremy Boy, and Miriah Meyer. 2017. Visual narrative flow: Exploring factors shaping data visualization story reading experiences. *Computer Graphics Forum (Proceedings of EuroVis)* 36, 3 (2017), 377–387. http://doi.org/10.1111/cgf.13195.

[32] Microsoft. 2018. Use bookmarks to share insights and build stories in Power BI. https://docs.microsoft.com/en-us/power-bi/desktop-bookmarks. [Online; accessed 13-09-2018].

[33] Meredith Ringel Morris, Jacob O. Wobbrock, and Andrew D. Wilson. 2010. Understanding users' preferences for surface gestures. In *Proceedings of Graphics Interface (GI)*. 261–268. http://dl.acm.org/citation.cfm?id=1839214.1839260.

[34] Ken Pfeuffer, Ken Hinckley, Michel Pahud, and Bill Buxton. 2017. Thumb + pen interaction on tablets. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. 3254–3266. http://doi.org/10.1145/3025453.3025567.

[35] Zening Qu and Jessica Hullman. 2018. Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 24, 1 (2018), 468–477. http://doi.org/10.1109/TVCG.2017.2744198.

[36] Donghao Ren, Matthew Brehmer, Bongshin Lee, Tobias Höllerer, and Eun Kyoung Choe. 2017. ChartAccent: Annotation for data-driven storytelling. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*. 230–239. http://doi.org/10.1109/PACIFICVIS.2017.8031599.

[37] Donghao Ren, Tobias Höllerer, and Xiaoru Yuan. 2014. iVisDesigner: Expressive interactive design of information visualizations. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 20, 12 (2014), 2092–2101. http://doi.org/10.1109/TVCG.2014.2346291.

[38] Donghao Ren, Bongshin Lee, and Matthew Brehmer. 2018. Charticulator: Interactive construction of bespoke chart layouts. *IEEE Transactions on Visualization and Computer Graphics* (2018), 1–1. https://doi.org/10.1109/TVCG.2018.2865158

[39] Donghao Ren, Bongshin Lee, Matthew Brehmer, and Nathalie Henry Riche. 2018. Reflecting on the evaluation of visualization authoring systems. In *Workshop Proceedings of Evaluation and Beyond - Methodological Approaches for Visualization (BELIV)*. https://aka.ms/renbeliv18.

[40] Nathalie Henry Riche, Christophe Hurter, Sheelagh Carpendale, and Nicholas Diakopoulos (Eds.). 2018. *Data-Driven Storytelling*. AK Peters. CRC Press. https://aka.ms/dds_book.

[41] Arvind Satyanarayan and Jeffrey Heer. 2014. Authoring narrative bisualizations with Ellipsis. *Computer Graphics Forum (Proceedings of EuroVis)* 33, 3 (2014), 361–370. http://doi.org/10.1111/cgf.12392.

[42] Arvind Satyanarayan and Jeffrey Heer. 2014. Lyra: An Interactive Visualization Design Environment. *Computer Graphics Forum (Proceedings of EuroVis)* 33, 3 (2014), 351–360. http://doi.org/10.1111/cgf.12391.

[43] Edward Segel and Jeffrey Heer. 2010. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 16, 6 (2010), 1139–1148. https://doi.org/10.1109/TVCG.2010.179.

[44] Andre Suslik Spritzer, Jeremy Boy, Pierre Dragicevic, Jean-Daniel Fekete, and Carla Maria Dal Sasso Freitas. 2015. Towards a smooth design process for static communicative node-link diagrams. *Computer Graphics Forum (Proceedings of EuroVis)* 34, 3 (2015), 461–470.

http://doi.org/10.1111/cgf.12658.

[45] Tableau. 2018. Story Points. https://tabsoft.co/2jxghOC. [Online; accessed 13-09-2018].

[46] Jagoda Walny, Samuel Huron, Charles Perin, Tiffany Wun, Richard Pusch, and Sheelagh Carpendale. 2018. Active reading of visualizations. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 24, 1 (2018), 770–780. http://doi.org/10.1109/TVCG.2017.2745958.

[47] Zezong Wang, Shunming Wang, Matteo Farinella, Dave Murray-Rust, Nathalie Henry Riche, and Benjamin Bach. 2019. Comparing Effectiveness and Engagement of DataComics and Infographics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*.

[48] Haijun Xia, Bruno Araujo, Tovi Grossman, and Daniel Wigdor. 2016. Object-oriented drawing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. 4610–4621. http://doi.org/10.1145/2858036.2858075.

[49] Haijun Xia, Bruno Araujo, and Daniel Wigdor. 2017. Collection objects: Enabling fluid formation and manipulation of aggregate selections. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. 5592–5604. http://doi.org/10.1145/3025453.3025554.

[50] Haijun Xia, Ken Hinckley, Michel Pahud, Xiao Tu, and Bill Buxton. 2017. WritLarge: Ink unleashed by unified scope, action, and zoom. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. 3227–3240. http://doi.org/10.1145/3025453.3025664.

[51] Haijun Xia, Nathalie Henry Riche, Fanny Chevalier, Bruno Araujo, and Daniel Wigdor. 2018. DataInk: Direct and creative data-oriented drawing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. https://doi.org/10.1145/3173574.3173797.

[52] Emanuel Zgraggen, Robert Zeleznik, and Steven M. Drucker. 2014. PanoramicData: Data analysis through pen & touch. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)* 20, 12 (2014), 2112–2121. http://doi.org/10.1109/TVCG.2014.2346293.

[53] Zhenpeng Zhao, Rachael Marr, and Niklas Elmqvist. 2015. *Data Comics: Sequential Art for Data-Driven Storytelling*. Technical Report. Human Computer Interaction Lab, University of Maryland. http://www.cs.umd.edu/hcil/trs/2015-15/2015-15.pdf.