DataToon: Authoring Data Comics for Dynamic Networks

Category: Research

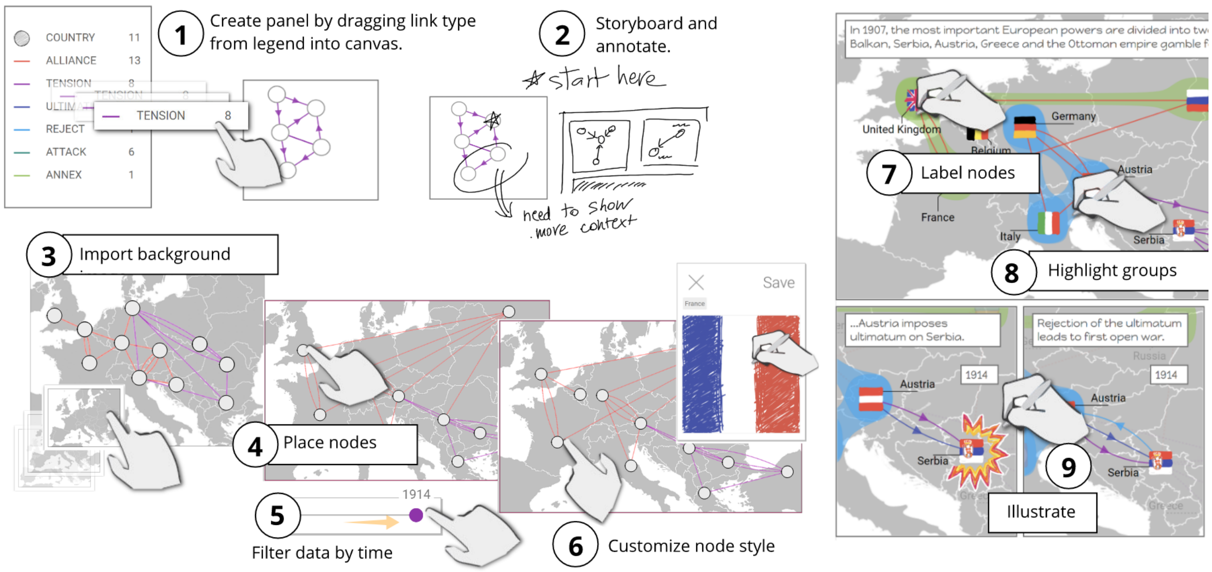


Fig. 1. DataToon provides a flexible authoring environment for exploring data and creating a data comic through direct pen and touch input. A user can rapidly generate visualization panels, customize visuals, add rich annotations, and establish a narrative flow with automated transitions.

Abstract— This paper presents DataToon, a browser-based tool for creating highly expressive data comics about dynamic networks via pen and touch interactions. Comics are an entertaining and familiar medium, one that can be leveraged to tell compelling stories about data by arranging visualization, annotation, and text across a series of panels. The need to craft expressive visuals from data, maintain consistency accross panels and compose them in space to create a compelling narrative are unique challenges of data comics that are not well supported by current authoring tools. DataToon provides a direct manipulation pen + touch interface along with expressive annotation options, allowing storytellers to sketch with data and experiment with different designs to produce dynamic network comics. DataToon smoothly integrates data binding, visual encoding, story authoring, and annotation, activities which would have previously required use of multiple tools with little or no connection to the underlying data. We conducted a qualitative study to evaluate the usability and demonstrate the expressiveness of our tool through examples.

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

1 Introduction

Data comics are emerging as a promising genre of data-driven storytelling [6]. They leverage the well-established visual language of comics [30], integrating captions and annotations with visual representations of data to convey a narrative. In this format, it is possible to suppress the complexity of data by incrementally revealing aspects of the data across multiple views or "panels", arranged thoughtfully on one or more pages [7]. Previous work manually generated comics on personal data, environmental data, or global health[1], demonstrating the richness of this genre. However, producing such data comics is a difficult and laborious process requiring switching between two kinds of tools, mastering both visualization authoring environments to generate accurate data representations, and graphic design applications to stylize visuals and arrange panels in space.

While several recently proposed tools from the research community have aspired to unify this process for infographics [9, 25, 28, 44], data comics pose unique design challenges. In a data comic, we visualize aspects of the data across multiple panels, and thus the design challenges we face include a need to maintain visual consistency between them [35] and to provide appropriate transitions from one panel to the next in an effort to convey a coherent narrative. They also require crafting expressive visual designs that align well with the semantics of the data and experimenting with different ways to break down the story into panels and arrange them in space. Addressing these challenges with existing tools is extremely tedious and time-consuming.

This paper introduces DataToon, an editor for authoring data-driven data comics via direct pen + touch interactions. DataToon enables authors to quickly generate panels from data, relying on data binding to ensure consistency of the visuals. It also provides direct and fluid interactions to explore different expressive visual designs by sketching, and to experiment with different ways to break down panels and lay them out in space by touch interactions. DataToon also features mechanisms to assist in data comic creation, such as the automatic generation of transitions between two panels. We opted to target DataToon to the creation of comics from dynamic multivariate networks. This scope allowed us to build upon previous work [5] which demonstrated the power of the genre for this type of data.

We start with a reflection on design challenges for creating data comics (Section 2). With these in mind, we review current authoring tools (Section 3) and introduce the key concepts and interaction paradigms for DataToon (Section 4). A gallery of examples available in supplemental material demonstrates the expressiveness of the tool (Section 5). Findings from a study with eight designers and data analysts demonstrate its usability (Section 6).

[1] http://datacomics.net

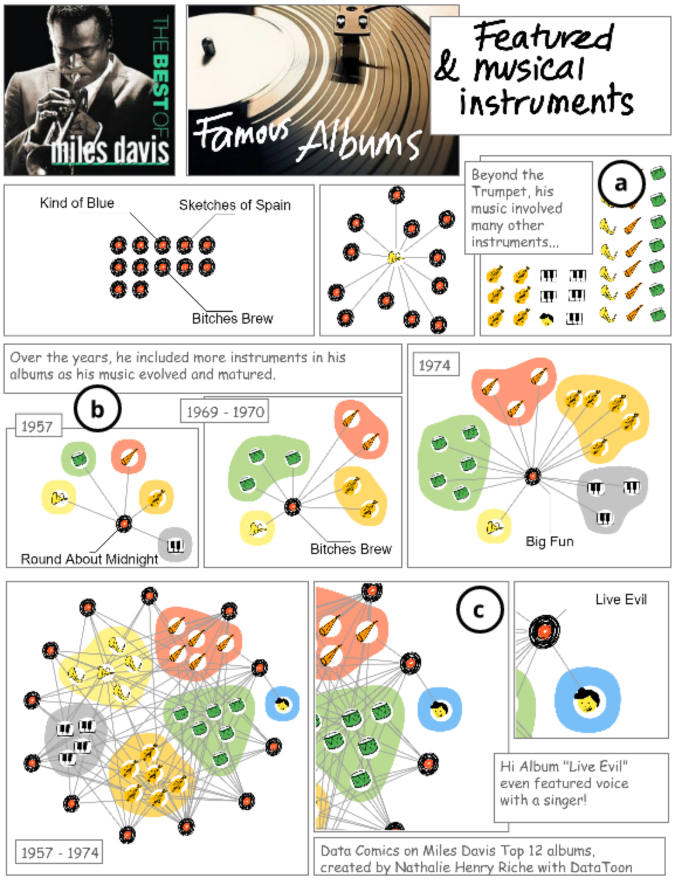


Fig. 2. Data Comics created with DataToon featuring (a) expressive visuals for representing instruments (C1), (b) breakdown of pane to convey temporal evolution (C2), (c) transition between panels at different scales (C3), laid out in 2D space to convey a linear story (C4).

2 Challenges in Creating Data Comics

Comics is a well-established storytelling medium [17, 30], used in many contexts such as storyboarding [21, 32], science education [20, 42], or information communication [13]. McCloud, a comics theorist, describe comics as *juxtaposed pictorial and other images in deliberate sequence, intended to convey information to the viewer* [30] or, to put it simply, *sequential art*. *Data comics* employ sequences of annotated visualizations to communicate insights about data to a viewer. Crafting data comics raise unique challenges, which motivated the design of DataToon. This section describes four challenges based on our own experience in crafting data comics and the research we conducted to help others create them [5, 7].

C1 - Crafting Expressive Data Visualizations: Most characters depicted in comics exhibit some human traits, if caricatured, and generally use words in speech balloons and captions to tell their story. By analogy, data comics seek to visually portray some aspects of the data and may use text to reveal underlying values or dimensions to convey higher-level insights. Crafting expressive, unique and memorable visuals in line with the semantic of the data and tailoring the level of details of the representation to the message to convey are not well supported by data visualization creation tools and thus require graphics design software support akin to infographics [8]. Creating accurate visual representations of the data may then be tedious and error-prone to achieve. As data comics also closely integrate textual annotations and captions to reveal underlying data values, dimensions and filters also introduces additional efforts and sources of potential inconsistencies.

C2 - Breaking Down Information into Panels: A panel is a basic communication unit in data comics, ideally conveying a simple single insight about the data. Reading panels in sequence enables the reader to gain multiple insights and comprehend a higher-level and more complex story about the data. Thus, deciding on the content of each panel, their sequence and arrangement in space is the basis of data comics design. This process requires the creation of multiple views of the data that each accurately represent different aspects or subsets of the data (e.g. filtering data), possibly adjusting the level of details (e.g. aggregating data) to best convey a point.

In addition to the sheer effort required to generated these multiple views, authors also need to provide consistent visual representations across panels to avoid confusing the reader and leading to incorrect decoding of the information [35]. Maintaining this consistency (e.g. same shapes for each data point or same color for a specific data dimension) across all panels is particularly costly if changes are made late in the authoring process as it requires updating all relevant panels. Assisting with these post-hoc changes however may prove particularly important for data comics, as existing comics may often need to be updated to include more recent data for example.

C3 - Creating Transitions between Panels: Panels in comics are spatially juxtaposed to convey a story as a collection of moments (space and time). Readers use their imagination to fill the gap between these moments [], a process of reading called *closure* [15, 30]. Facilitating closure is critical for data comics as leaving the reader to infer the missing parts can lead to incorrect understanding missing out of information. Introducing intermediary panels to convey the relationship between two panels more explicitly can mitigate this issue.

For example Figure 2(c) depicts the transition between an overview panel and a detailed panel. To accompany the reader and convey the zooming operation, the author may create transitional panels. Depending on the type of transition required [6], this technique can prove tedious as it may require producing series of panels with similar content but subtle progressive alterations.

C4 - Conveying a Linear Narrative in a Spatial Layout: Data comics have the power to guide readers through a narrative by providing a sequential reading experience while also offering them to gain an overview of the story and control their reading pace by laying out story pieces in 2D space. However, achieving a compelling narrative in an optimal layout requires a delicate balance between the number of panels, their size and ratio as well as their arrangement in space, which in turn might require adjusting their content and order.

In addition, working with data induces a set of constraints as the accurate depiction of data points (C1) impacts panel sizes or ratios for example, and the selection of data aspects and insights to represent (C2) and their transitions (C3) has a direct influence on the number of panels. Juggling with all of these parameters and experimenting with different sets of panels and arrangements can be a daunting, time-consuming and laborious task, especially for an audience aiming at communicating about their data and unlikely versed in comics design.

3 Related Work

In recent years, many tools emerged for authoring data-driven stories. We review them below, outlining challenges they fail to address.

3.1 Tools for Data-Driven Storytelling and Comics

The large majority of these tools focus on generating a single data visualization (e.g. infographics) via a graphical user interface. For example, tools like Lyra [39] or iVisDesigner [37] provide graphical styling mechanisms after loading and visualizing a data set. Tools such as Data-Driven Guides [25], DataIllustrator [28] or DataInk [48] further allow for expressive customization of visual marks and their layout, and even provides labels, annotations, and emphasis on an existing svg-file of a chart [36]. The only data-driven approach to communicative network representations we are aware of is Graph Coiffure [40], an interface that allows for styling and laying out static node-link diagrams. However, to create comics using these systems, many steps need to be repeated (i.e. for each panel) and assuring consistency (C2), creating manual transitions (C3), and elaborate the final layout (C4) require to be done manually via additional software.

Another category of tools focus on sequence and narration for data-driven storytelling. For example, Ellipsis [38] offers a graphical interface for augmenting an existing visualization with annotations and interactions and sequencing a narrative through state-based scene transitions. Tableau Story Points [41] provides a drag-and-drop interface for composing a sequence of story points that can embed visualizations generated through a separate exploration interface. Similarly, Timeline Storyteller [11, 12] provides an integrated tool for creating and annotating timeline visualizations and capturing snapshots of visualization states that can be played in sequence, while DataClips [4] provides a similar sequencing interface but focuses on creating data-driven video clips. On the other hand, Vistories [19] leverage interaction histories to implicitly generate visualization pieces that can be later annotated to compose a story. In these tools, multiple and sequenced visualizations serve as story pieces and are manually arranged to construct a narrative structure. The visualizations are usually annotated and highlighted to guide readers through the narrative. To support transitions between visualizations, or story pieces, animation and discrete steppers are most commonly used [31]. Although these tools attempt to address C2 and C3 to some extent, they are limited in the expressiveness of the data representations they enable people to create (C1) and, most importantly, do not support experimenting with different spatial juxtaposition of panels (C4) to compose data comics.

Tools for creating conventional comics are mostly illustration software featuring several comic-specific features such as comic-styled brushes and 3D character modeling [1]. Comic editors offer convenient features for ease-of-use, including pre-drawn characters, text effects, and page templates [2, 3] but are not tailored to data visualization. Using these tool requires importing images of data visualizations created from other software. Previous work by Zhao and Elmqvist presented an editor for data comics [50], allowing to create sequences, linear layouts, and embellish them with comic elements such as speech bubbles and a narrator character. However, this tool is in the same vein of conventional comics authoring ones, based on importing images created from other data visualization tools. Editing and iterating over the visualization design to create labels or deemphasize parts of the visualization (C1) requires much back and forth with other data visualization tools. In contrast to these tools, DataToon aims at providing data binding to enable authors to experiment with visual designs by minimizing laborious effort required to maintain consistency across panels (C2).

3.2 Pen & Touch Interaction for Creative Exploration

Crafting data comics is a creative and iterative process. To providing a flexible environment, facilitating storyboarding, experimentations with visual designs and rapid iteration, we propose to leverage direct manipulation using pen and touch. By leveraging natural human sketching and manipulation skills, pen and touch interaction brings enhanced the feeling of direct engagement compared to manipulating configurations and parameters through WIMP UIs.

A number of recent visualization tools take advantage of multi-touch and pen gestures in various contexts. DataInk [48] enables the expressive design of personal data visualizations by integrating data binding with freeform sketching. SketchStory [27] also leverages freeform sketching to create charts to communicate data on a whiteboard. Other tools focus on supporting interactive data manipulation and exploration [24, 49]. Our work builds on these prior work and enables the creation of data comics with direct pen and touch input.

We also draw inspirations from existing tools that employ pen and touch for content creation. Hinckley et al introduced a slew of compelling interaction techniques for metaphorical manipulation of content, following the principle that pen writes and touch manipulates [23, 34]. Previous studies illustrated a set of pen and touch gestures for common operations on interactive surfaces, including selection, deletion, and copy & paste [33], as well as diagram editing [18]. A number of direct pen and touch systems have been proposed in various application domains, such digital drawing [45, 46], early-stage ideation [47], and active reading [22]. We employ a set of gestures used in these tools to inform the interaction design of our tool.

4 DATATOON

To allow people to craft expressive data representations (C1) as well as experiment with panel layout composition (C4), we opted to design DataToon as a pen+touch application. We reasoned that sketching afforded by the pen would stimulate creative and expressive designs, while direct touch interactions would encourage authors to experiment with panel arrangements. The overarching principle of DataToon is to offer a focused set of instruments (materialized by different modes of use of the pen) to enable the creation, editing and styling of multiple panels representing different aspects of the data and experiment with their organization in space (materialized by a set of manipulable views on a zoomable canvas).

A key aspect of DataToon's design is to support data binding across panels, enabling functionality that other applications do not support, such as automatically propagating changes of visual representations to other panels to maintain consistency (C2) and automatically generating transitions between two panels (C3). By combining a focused tool set with data binding, DataToon allows for rapid iterative design of data comics that could only be achieved via laborious manual efforts today.

4.1 Walkthrough

To introduce DataToon, we describe the process of creating a data comic based on a scenario adapted from Bach et al. [5]. In this scenario, Aiden is a high-school teacher who wants to educate his students about the dynamics of European alliances preceding World War I. To engage his students, he wants to use a data comic to communicate how a series of intricate alliances and conflicts evolved over time. He created a dataset by collecting information from various online resources. The dataset represents a dynamic geographic and multi-faceted network where nodes are nations with changing alliances and oppositions.

Aiden opens DataToon using a web browser on a pen-and-touch based tablet. He first sees a list of existing comic pages previously created. He creates a new page and loads the network dataset by dragging it onto the page's canvas. A legend panel is created (Fig. 4a). He starts exploring the dataset by dragging and dropping a data dimension (e.g., a node type of *Country*) to create a visualization. Using the time slider, he further examines how alliances changed over time.

Once he is assured of the data's integrity, he begins creating a comic strip. To show the countries in a geographic context, he loads an image showing the map of Europe (Fig. 4b). He wants to start by showing all the countries on the map. He uses the image as a background of the panel and positions each country node in its geographic location using the *Control* pen ( ). He also creates labels to reveal the names of the countries using the *Annotation* pen ( ). Finally, he creates a caption inside the panel to set the initial tone of the story (Fig. 4c).

After establishing the context through the first panel, he now wants to advance the narrative in a stepwise fashion. He duplicates the first panel and adds the link type of *Alliance* (Fig. 4d). He adjusts the time slider to 1839, to begin with, showing the neutrality contract between the United Kingdom and Belgium. He double-taps the slider to create a caption using the current time as its text (Fig. 4e). He lassos the two nodes to highlight, which make the rest of the countries muted into the background (Fig. 4f,). Switching to the *Control* pen ( ), he pans and zooms to set the focus of the panel to the two nodes.

Now he wants to explain the Three Emperors alliance between Germany, Russia and Austria, which began in 1873. He again starts by duplicating the previous panel, which copies everything including the caption as well as underlying data bindings. Once he set the year to 1873 in the new panel, the time caption is automatically updated (Fig. 4g left). He now zooms into the three countries and highlights them into the foreground. To show the significance of the alliance, he adds a colored background to the three nodes, using the *Annotation* pen but now with the pen button pressed (Fig. 4h).

In the next panel, he highlights the isolation of France due to the alliances of the surrounding countries. This time, he uses freeform sketching to emphasize the isolation (Fig. 4g right,). He likes the freedom of creating his own unique style. He draws a custom, comic-like, sketchy appearance for each node by invoking a contextualized

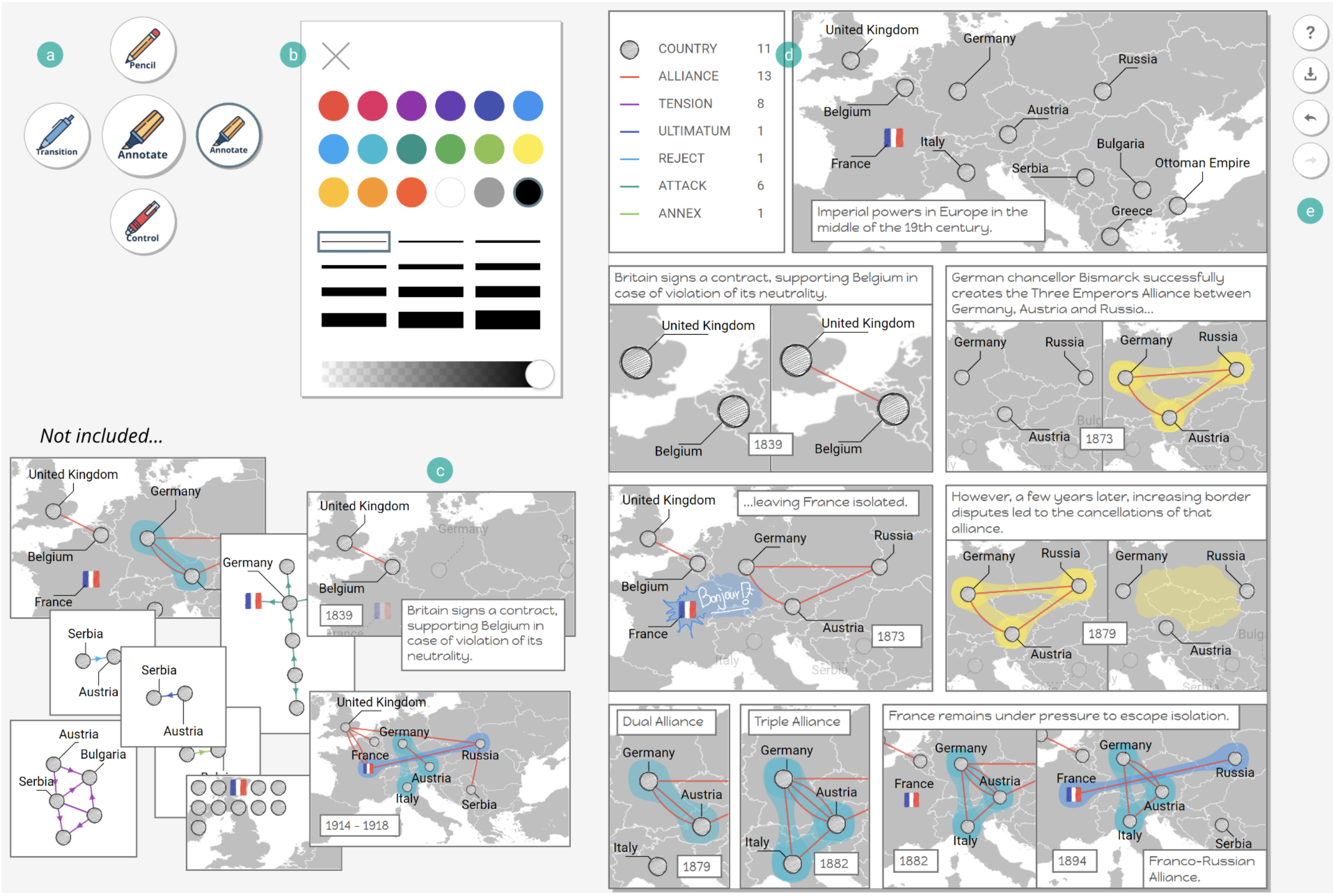


Fig. 3. The overall interface for editing a data comic page, initially revealing (a) the pen tool menu, (b) a color palette, (c, d) the canvas area, and (e) a side menu. A user can switch between different pens to apply different functions. The canvas area provides both (c) an infinite space for ideation and exploration, as well as (d) a dedicated page area for presentation.

canvas from the legend panel (Fig. 4i). He similarly draws a country flag for France by lassoing its node using *Pencil* (✎) with the pen button pressed. These custom visuals are automatically propagated to all other panels. Before finishing, he saves the current comic strip as an image to see if it looks good in print.

4.2 User Interface

DataToon has a few main user interface components (Fig. 3). The page canvas (Fig. 3d) provides an authoring environment for managing visualization panels and drawing freeform annotations. The legend panel (Fig. 3d, upper left) not only shows the overview of visual mappings but also serves as an interface for creating content. The contextual canvas (Fig. 4i,j) is invoked on demand from the legend for customizing visual mappings, e.g., for sketching custom icons for nodes. The pen tool menu (Fig. 3a) allows the author to switch between functions from sketching to annotation and transition.

Our design rationale for interaction design follows the mantra: *the pen writes, touch manipulates* [23]. In DataToon, touch always manipulates panels (e.g., moving and resizing). By default, the pen writes (✎) but acquires different tools depending on the selected pen type (✎,✎,✎). We made this design decision because a visualization usually consists of a multitude of data points, making it cumbersome to manipulate using fat fingers.

4.2.1 Comic Book Metaphor

DataToon uses a comic book metaphor, allowing the author to create multi-page comics (Fig. 5). Each page can embed a separate dataset, if necessary, to show a different facet of the story. The author can export each page into an image for printing and sharing.

4.2.2 Storyboarding and Freeform Sketching

DataToon also leverages a canvas metaphor to provide a flexible storyboarding environment for creating data comics (Fig. 3). The author can pan and zoom through the canvas using a non-preferred hand while using the pen to draw illustrative marks and handwritten text for ideation and annotation. The author can create an empty panel by simply drawing a rectangle, which can serve as a note area and be filled with content later on such as a node-link visualization. All the visual elements on the canvas can be directly manipulated through pen and multi-touch interaction. The author can freely arrange panels and resize them to create an engaging layout of the page. Our design decision for providing a flexible authoring environment is to allow for multiple workflows and fluidity in the iterative design process, while sketching can further amplify the expressive capability of such an environment due to its inherent informality [48].

4.2.3 Panel Types

Legend Panel: The author can import a dataset by dragging a .JSON data file onto the canvas (Fig. 4a). This causes a legend panel to appear, providing an overview of the dataset, displaying data dimensions including a list of node types and link types as well as the number of elements within each type. The legend panel plays an important role in helping readers make sense of the underlying data and understand visual mappings of the data.

The legend panel also serves as an interface for creating visualization panels and customizing visual encodings (**C1**). The author can simply drag a data dimension (e.g., node type) from the legend onto the canvas to create a visualization panel or an existing panel to add additional data to it (Fig. 4d). The latter action is akin to adding a data filter to the

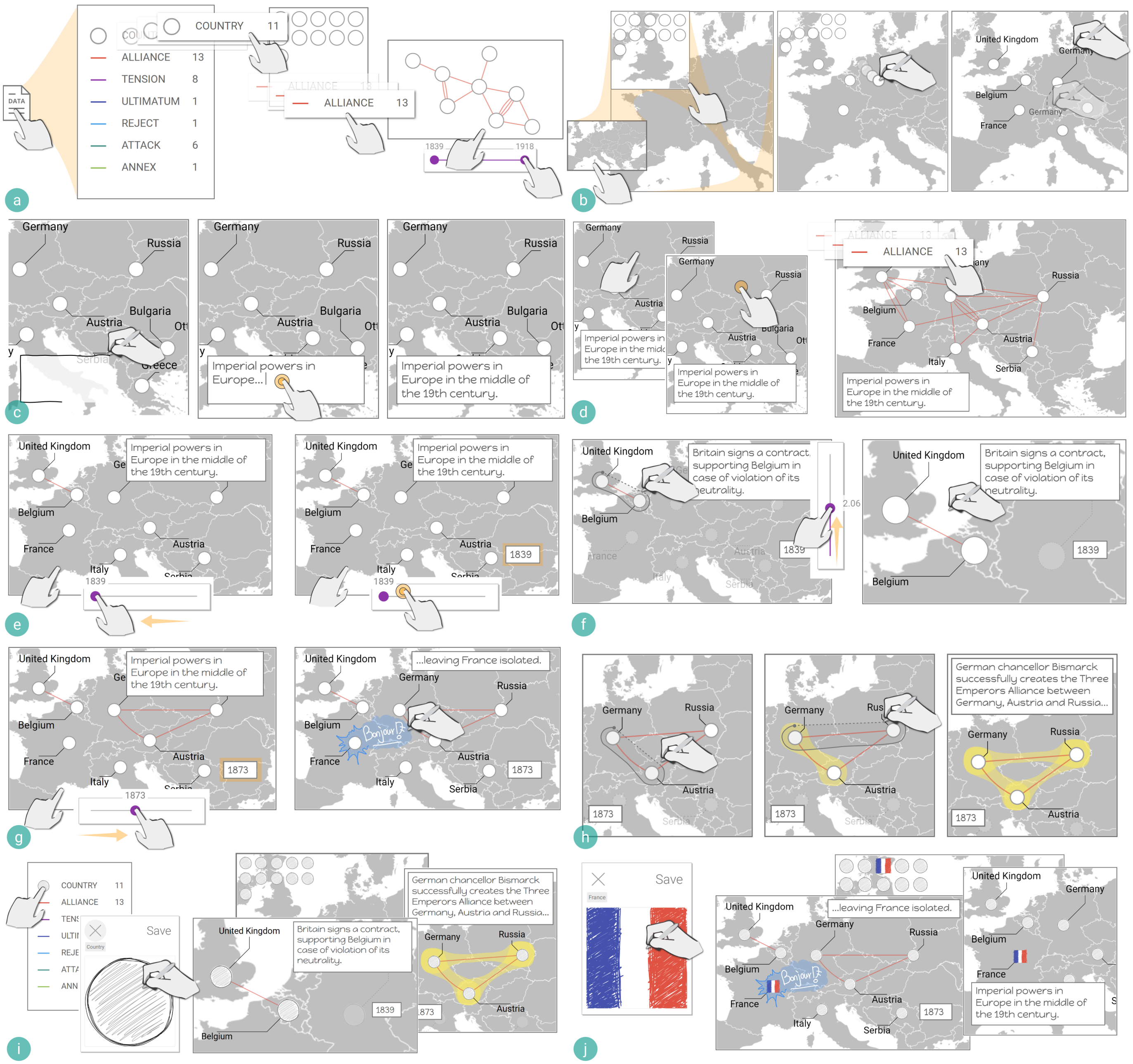


Fig. 4. The workflow of creating the data comic described in Sect. 4.1 and shown in Fig. 3. (a) importing a dataset to create the legend panel and fluidly exploring the data through simple drag and drop and the time slider. (b) importing a background image, laying out nodes, and creating labels. (c) creating a caption by drawing a rectangle and double-tapping it to edit its text. (d) duplicating the panel through hold and tap, and incrementally adding additional data to the panel. (e) filtering data using the time slider and creating a time caption by double-tapping it. (f) lassoing nodes to show in the foreground and panning and zooming through the combination of pen and touch. (g) updating the time caption automatically and adding a freeform annotation through sketching. (h) Creating a highlighted group. (i) the contextual canvas allows the author to sketch new visual mappings for all country nodes and for (j) France specifically.

panel. In addition, the author can update the visual encoding of the data dimension by tapping its current encoding in the legend panel (Fig. 4i).

Content Panel: A regular panel can contain any type of context including a visualization, caption text, annotation, and background image. The core content of the panel is a visualization created from the legend panel. The number of data dimensions added to the panel determines the type and complexity of the visualization within. For example, multiple node types create a stacked unit chart (Fig. 8b), while having at least one link type generates a node-link diagram. Each node and link can use its shape size and line thickness respectively to encode a weight, while the link can additionally encode its direction.

Additional temporal filtering can be applied using the time slider that appears on demand when the author is touching the panel (Fig. 4e). Double-tapping the slider creates a caption panel within the panel, which is always in sync with the slider as it is updated (Fig. 4g). The time caption belongs to the parent panel. Similarly, panels can be nested. For example, drawing a rectangle inside another panel create a child panel, to be used as a caption or an inset.

The author can duplicate an existing panel, which transfers all the content of the source panel to the new panel (Fig. 4d). The duplicate action is particularly useful for progressively building a narrative using the previous panel as a basis (e.g., overview → detail).

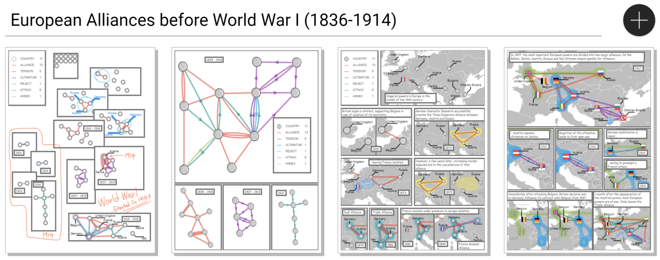


Fig. 5. DataToon uses a comic book metaphor to allow authors to create multiple pages of data comics using more than one dataset.

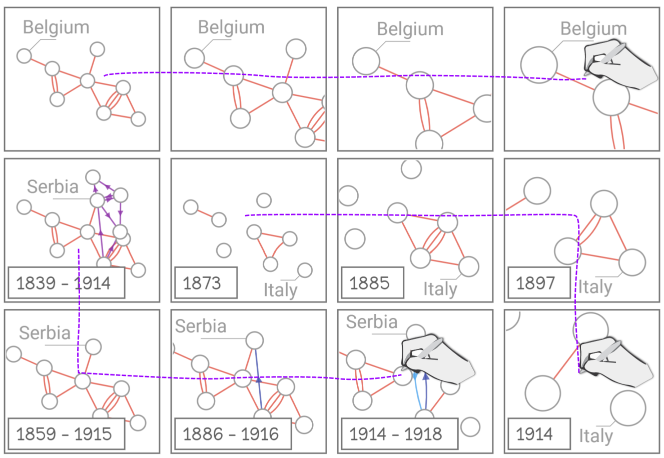


Fig. 6. Automatic transitions between two panels, detecting and interpolating the difference between the panels such as zoom transforms, time ranges, data filters or combinations of all.

4.2.4 Pen Types

Pencil : This is the default pen type that allows the author to perform freeform sketching on the canvas. The author can choose a different color, thickness, and opacity of the pencil stroke. If the author draws on top of a panel, the sketch belongs to the panel and moves along with it when manipulated.

Annotation : DataToon provides rich annotation and highlighting options by combining both freeform sketching and data-aware annotation (**C1**). Using the *Annotation* pen, the author can lasso a set of nodes to show them in the foreground while muting the rest (Fig. 4f) or generate a colored bubble surrounding them with the pen button pressed (Fig. 4h). The author can also create a node label by drawing a line out of each node (Fig. 4b). These types of annotations are all data-driven, making it easier to update iteratively. For instance, if the author updates the position of a node, the corresponding label and bubble are automatically updated. Just like the regular pencil, the author can choose a different style, and the selected style will then be applied to the style of the annotation.

Control : To control which part of the content to show in the panel (e.g., cut-out or lens [7]), DataToon supports simple panning and zooming through the *Control* pen (Fig. 4f). The author can also control the layout of a visualization (i.e., unit chart or node link diagram) that is initially auto-generated (Fig. 4b), as well as the positions of labels.

Transition : DataToon currently provides three automatic transitions, namely a temporal sequence, build-up of data dimensions, and zoom transform [7] (Fig. 6). To create the transitions, the author needs to draw a path from one panel to another using the *Transition* pen. DataToon then tries to interpolate the two panels by detecting the difference between them. A typical continuous value interpolation works for the temporal sequencing and zoom transition (e.g., 2014 $\rightarrow$ {2015,2016,2017} $\rightarrow$ 2018). For the build-up transition, we progressively add or remove filters (i.e., node type or link type). The greater the distance between the panels is, the more transition panels are generated along the path.

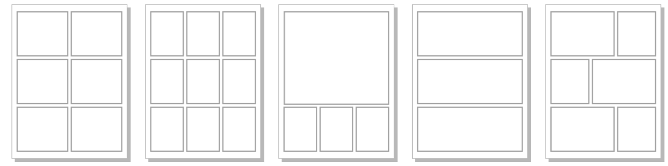


Fig. 7. Layout templates for basic narrative structures: canonical grid (2×3, 3×3), overview and detail, parallel, and staggered layouts

4.2.5 Style Propagation

The author can customize the visual encoding of nodes and links. The custom style is then immediately propagated across the comic for consistency. There are mainly two ways to modify the visual encoding, at the data point level and dimension level. When the author taps the visual encoding of a data dimension (node or link type) in the legend panel, it invokes the contextual canvas in which the author can draw a custom icon. On the other hand, the use can change the visual encoding of a specific data point (node or link) through lassoing using *Pencil* with the pen button pressed.

4.2.6 Layout Templates

DataToon provides a set of panel layouts to begin with, including canonical grid, overview and detail, staggered, and parallel layouts that are commonly used in existing comics [7] (Fig. 7). When a page is created with a layout template, the page is prefilled with panels based on the layout. The panels are not filled with content since the purpose of the layout template is to provide a narrative structure rather than narrative content. We leave automatically filling the content of the layout as future work.

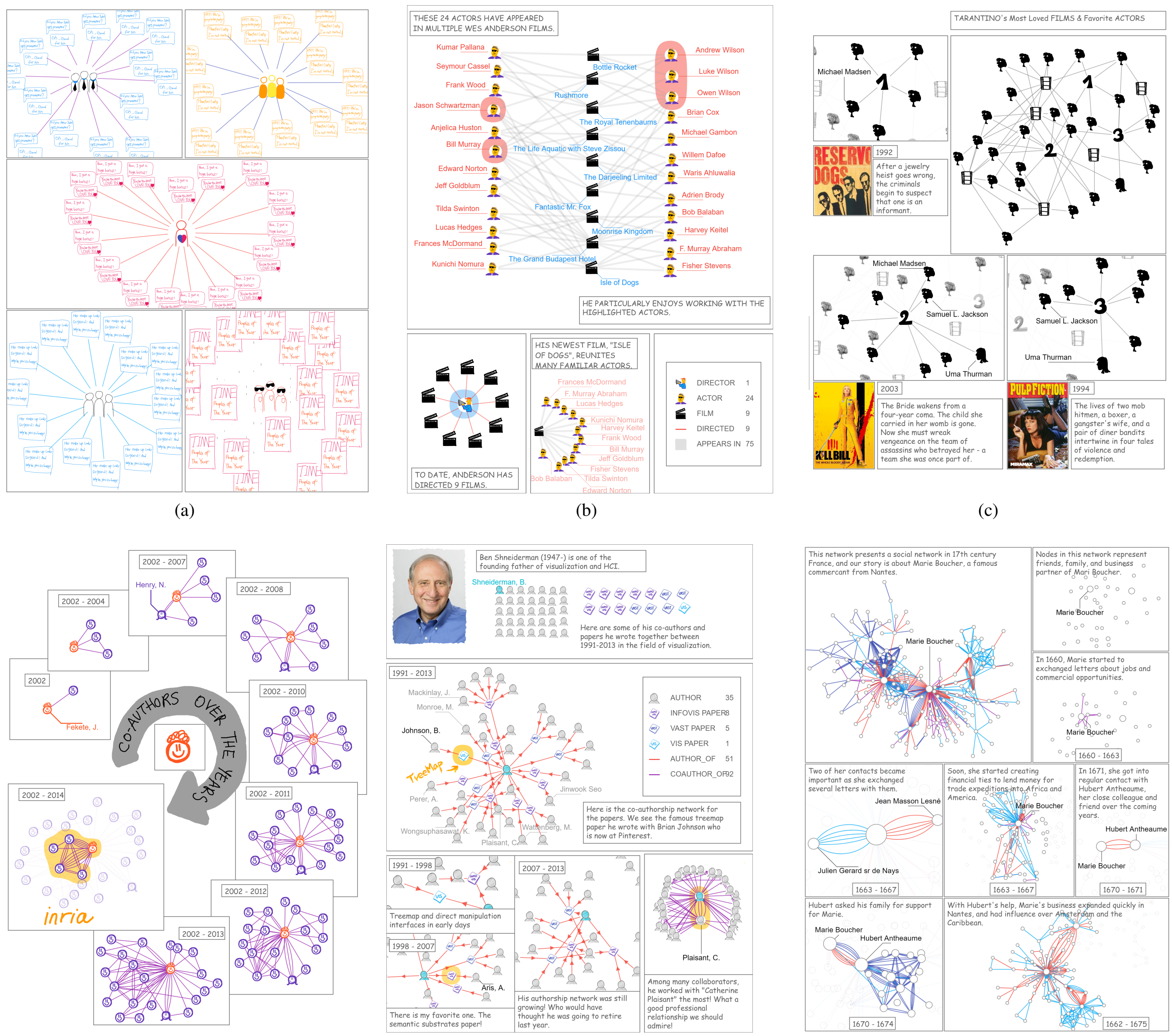
4.3 Implementation

DataToon is a web-based application written in Javascript. It uses React.js for building user interface components and Redux.js for application state management. The layout of the node-link diagram is generated through WebCoLa [16] and highlighting a group of nodes as a cluster uses a Javascript implementation [26] of Bubble Sets [14]. We also make use of utility functions in D3 [10] such as pan and zoom, and react-annotation for labelling [29]. DataToon only consists of a front-end interface without a back-end server, which can be easily attached if needed, and makes use of localStorage and indexedDB in HTML5 to persist the application state. DataToon will be freely available at `http://datatoon.datacomics.net` upon acceptance of this paper.

5 Expressiveness

To demonstrate the expressiveness of comics that can be created with DataToon, we assembled a gallery available in supplemental material. Figure 8 presents six comics showing multivariate and temporal social networks and co-authorship networks, as well as networks depicting relationships around movies, actors, and directors. We refer to the specific comics in Figure 8 through letters (a-f) in this section.

DataToon supports different styles ranging from sketchy (a,d), to clean vector-graphics (e,f), including pictures (c), as well as small pictograms for nodes (b,c,d,e). Free placement of panels allows for a variety of layouts such as described in [7]; tiles implying a clear sequential narrative (e.g., a,b), larger panels for an overview and smaller for details (b,c). Comic (d) still employs a sequential layout but aligned in a clock-wise manner to break the convention of the zig-zag layouts. Some of the comics use sequence (panels) to show temporal evolution (d,e,f), others imply different facets (a) and details of the data (b,c).



(a) (b) (c)

(d) (e) (f)

Fig. 8. Six examples of data comics created with DataToon using different datasets. Higher resolution versions are available in supplementary material as well as on `https://datatoon.datacomics.net`.

6 Comic Creation Study

To understand if people can understand DataToon and use it to create comics from data, we conducted a qualitative study with eight participants. Half of the participants were graphic designers, while the other half were data analysts. From the graphic designers we sought to gather feedback on the interface paradigms we selected and implemented in DataToon, to assess if key interactions or concepts were missing to produce data comics, and to gain an understanding of how it compares with tools that are more oriented to graphic design. From the data analysis, we aimed to assess whether they can learn DataToon and use it to create comics from data without guidance.

We recruited our eight participants from a large software company in the United States. The graphic designers (P1-P4: 3 males, 1 female, aged 30 to 50, average 44) had professional experience in graphic design (e.g. over 20 years of experience using Adobe Illustator). Three of them had relatively low expertise in data analysis but one reported creating charts from data on a monthly basis. The data analysts (P5-P8: 2 males, 2 females, aged 31 to 42, average 37) perform data analysis and create charts and visualizations as part of their job. They had low or no experience with graphic design tools. All participants received \$20 in lunch coupons for their time. Three of our participants were left-handed.

We ran DataToon as a web application on Google Chrome using a Microsoft Surface Studio with a 28 inch screen at 4500×3000px (192 PPI) enabling simultaneous pen and multi-touch input.

6.1 Procedure

The experimenter conducted one individual session with each participant. Sessions lasted 90 minutes on average, with two participants finishing in ≈60 minutes, and one taking ≈120 minutes. In the beginning, the experimenter asked each participant to complete a demographics questionnaire, then asked them to create two data comics (#1,#2) with DataToon and concluded with three 7-point Likert scale questions (ease of learning, ease of use and enjoyment) as well as a short interview.

Comic #1: Demonstration and exploration: With the first comic, participants had to replicate a data comic on *World War I alliances* depicted in Fig. 3 and provided as a print out. This phase served as a tutorial and training for participants to learn and explore what was possible with DataToon. The experimenter presented the main principles of the interface and demonstrated the interactions required to create comics. At the end of this tutorial, participants started from a blank page and proceeded to recreate the comic, asking questions as needed. Participants were encouraged to try their own experiments with the tool until they felt comfortable to proceed without guidance. This first task lasted 30 to 40 min (including the 15 min tutorial).

Comic #2: Replication without guidance: In the second task, participants had to replicate the data comic on *Miles Davis' famous albums* depicted in Fig. 2, provided as a print out. This time, the experimenter did not offer any guidance. Participants were asked to think aloud as they worked, and the experimenter proceeded to observe and note issues with recalling or using functions to create the comics, and to capture quotes of interest. This task lasted between 30 and 50min.

6.2 Results

All participants successfully completed both comics. Seven participants used DataToon without any guidance from the experimenter to replicate the second comic. One participant required a few hints to recall how to complete certain interactions. On average, participants rated DataToon at 5.6 out of 7 for ease of learning and 5.9 out of 7 for ease of use.

Learning Bi-Manual Interactions: All participants mastered the interactions and made few if no errors by the end of the session. Only one participant (P2) had no previous experience with pen and touch devices. Their additional learning effort was reflected in the time to complete the first task (10 minutes longer than other participants) and the final ratings, 2 to 3 points lower than other participants for ease of learning (3/7) and ease of use (4/7). During the first task, the experimenter also observed four other participants (P4, P5, P6, P7) repeatedly having issues mastering bi-manual interaction. The main issue faced by all participants was the desire to use pen and finger interchangeably. However, only P2 and P6 reported it as a significant usability issue in the interview: *"I kept using my hand instead of the pen"*.

From the other six participants, we received many comments on the benefits and the engaging nature of bi-manual pen and touch interaction, once mastered. This is also reflected in the final ratings as the average enjoyment of using our tool is over 6 out of 7. P8 commented on the simplicity of the interactions: *"I love the power of just dragging* [shows fingers] *and creating* [shows the pen]*"*. P4 commented on the empowering experience it provided, making the app *"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: Graphic designers all expressed that the strength of DataToon was to propose a *"focused tool set"* (P1), *"streamlining the set of tools"* (P4) compared to existing advanced graphics illustration tools. Three of our graphic designers also commented that the look & feel of the interface, *"the soul of the app, this handwritten feel"* (P1), and sketching on a canvas were features that encouraged them to explore, ideate, and generally experiment with different data comic designs.

We observed that this interface paradigm of relying on a small set of instruments enabled flexibility in workflow. For example, all but one participant replicated the first comic using a different order of interactions than demonstrated by the experimenter. We also observed a large variation in working styles between participants. Several participants would start with multiple panels, adjusting the content of each panel first and then styling them all in a second wave. Others would polish one panel first, modify its content and style, and then duplicate it to work on the next panel. Design explorations also varied from participants: some created many panels, altering and erasing them and recreating new ones to iterate while other participants worked mostly in a single panel, extensively using the undo/redo feature.

One particular challenge for those using these types of authoring interfaces (having a focused set of instruments with different functions, as opposed to an exhaustive set of menus and buttons) is that it may prove intimidating upon first use, since many functions are hidden. This was particularly true for our data analysts. *"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"* (P8). However, it is worth noting that all participants mastered the interface after the first task.

We did observe that small inconsistencies in interaction principles can rapidly degrade the experience and make it hard for participants to remember how to perform specific operations. For example, P3 noted that an inconsistency where holding the pen button selected items when in pencil mode but it inked when used as a highlighter, which caused him to repeatedly make errors in the tutorial. P1 also commented that the general principle of dragging and dropping elements into panels was violated in the case of time captions, which required a double tap, which made it particularly challenging to remember.

From Data Visualization to Information Design: All participants noted that the data binding was a key advantage over graphic design and sketching tools, and we observed data analysts explore the data before working on their comics. For example, P5 started by creating many panels (one per node type) and commented on structural patterns found in the data. P8 used a different strategy, adding each type of node, one after the other, into the same panel and commenting *"so now I am beginning to see the relationships between instruments [...] I am going to move things around so I can understand my data"*.

Data analysts (P5-P8) particularly enjoyed using DataToon (rated enjoyment as 6.5/7 on average), many commenting that they were not artists or did not know how to draw but yet made aesthetically pleasing visuals: *"the appearance of the cartoons was much nicer than I could have created"* (P7). P8 also noted that she liked *"being shielded from all the data complexity and just focus on presentation"*.

As participants created comics, we gathered a number of comments on the possibility to transform visual representations of data into more "designed" illustrations to make insights more salient. For example, P3, looking at a particularly densely connected node-link diagram, commented *"I wish there was a way to untangle that because that is a super full graph"* and asked if we had features (such as bundling or link aggregation) to simplify the visualization in places to illustrate that two of its clusters were highly connected. P3 then reflected that *"there is an interesting balance between what I want to provide you with a sketch to give you a general idea of the data but not show ALL the data"*. We believe that supporting this subtle shift from data visualization towards what we call *information design* is an interesting direction for future work.

7 Discussion

Our study demonstrated that DataToon was learnable, and that data analysts without expertise in graphic design could create data comics. In this section, we discuss broader issues that arose during the development and evaluation of DataToon.

Expanding to Other Types of Data: Interacting directly with data provides opportunities to explore and iterate, or to easily apply and propagate custom styles and labels (C1). Though DataToon was designed primarily for graphs, the general concept and interaction design of DataToon can be adapted to other types of data (e.g. time-series, charts). Supporting different types of datasets such as tables as well as other common visualization types such as bar charts and line charts would further increase DataToon's expressivity and usefulness.

Most of our design choices can generalize to other data and visualization types, though some adjustments to our interactions may be required to accommodate them (e.g., free-form lassoing is unlikely to be effective in the context of matrix diagrams). Though perhaps the most obvious next step in extending DataToon involves adding automatic geo-encoding so that the author no longer needs to manually position nodes on images of maps, as in Fig. 4. Our transition feature is likely to be particularly useful for any data that changes over time, as a first step toward automatically generating panels, helping authors to lay out their story.

Pen and Touch Interaction: Our choice of pen + touch input was engaging for our study participants and the flexibility offered by this type of interaction may prove beneficial beyond data comics, to data-driven storytelling authoring tools in general. While it might take time for novices to acclimate to bi-manual pen + touch interactions, we believe that after an initial learning phase, this input modality stimulate their creativity, encourage experimentation and proves satisfying. However, to promote learning and potentially discoverability, we need to envision visual cues and better affordances to draw attention and remind users of the available interactions. From our study findings, we also discovered that many users initially wanted to use pen and their finger interchangeably. This may warrant further investigation, revisiting prior work from Hinckley et al. [23] and attempting to provide better affordances or different visual feedback.

Additional issues arise when polishing the comic to create a final version for presentation. Although the flexibility of storyboarding in DataToon enabled rapid exploration of design alternatives, it lacks some precise control in aligning panels for example. Thus, in the future we plan on providing an intelligent snapping of panels while taking into account the gutters between them. Additional features related to polishing a data comics include possibilities to turn handwriting to text, and beautified versions of sketched elements.

Expressiveness and High-Level Narration Support: With DataToon, we tried to find the middle ground between an unconstrained yet unspecific graphic authoring tool and a visualization-specific authoring environment. Our solution thereby allows for considerable expressiveness in terms of panel layout and node styling. These features borrow from the features available in many graphics editing tools such as Adobe Illustrator. Our examples in Fig. 8 show that DataToon allows for the realization of most of the design patterns for data comics described by Bach et al. [7], including “time-sequence” and “the-larger-picture”. Eventually, we are interested in investigating higher-level narrative structures that help authors to produce long-form multi-page stories with a beginning, middle, and end. Alternatively, DataToon could export comics as interactive websites that invite viewer exploration.

Beyond Traditional Comics: In this work, we have only scratched the surface in terms of the design space of authoring tools for data comics. We see several opportunities to further leverage the underlying data in the design of future authoring tools.

DataToon currently focuses on creating structural elements of data comics (visualizations, panels, etc), as opposed to semantic elements such as narrative design patterns. Our current support for transitions and layout templates touches upon this aspect, and we need to further study these components to determine their potential. Generating transitions that precisely match an author’s expectation is challenging, but it can be used as a way to explore different narrative flows.

DataToon’s layout templates are currently initialized with empty panels. It would be an interesting challenge to partially or entirely fill these panels automatically as a way to seed a story. As with transitions, we imagine that it is unreasonable to automate the generation of narrative structure in way that precisely mirrors the author’s intent. Thus, a mixed initiative approach could be a viable solution. For example, while an author can draw a path between two panels to generate transitions between then, we could similarly attempt to predict future panels an existing panel or an existing transition, thereby recommending panels to authors.

The benefit of having access to the underlying data may also be useful in a live presentation context, or in a context that allows for limited exploration, such as in an interactive museum or gallery exhibit. We originally developed DataToon to be used specifically for authoring data comics to communicate known insights. However, we could consider a “presentation mode” that allows for presenters and viewers to touch parts of the comic, thereby revealing content on demand, or allowing them add new annotations or comments to a comic as part of an active reading process [43].

8 Conclusion

We presented DataToon, a bi-manual pen- and touch-based tool for producing graph comics. DataToon was designed with four challenges in mind: a need to craft expressive visualization content, dividing this content across panels, producing transitions them, and ultimately conveying a linear narrative. DataToon provides various features to manipulate visualization elements such as styling and highlighting, as well as features for laying out panels and transitioning between them. We conducted a study in which graphic designers and data analysts used DataToon to produce comics, and they found it to be both usable and easy to learn.

References

[1] Clip Studio Paint. https://www.clipstudio.net/en, 2018. [Online; accessed 31-Mar-2018].

[2] Comic Creator Studio. http://summitsoft.com/products/comic-creator/, 2018. [Online; accessed 31-Mar-2018].

[3] Comipo. http://www.comipo.com/en/, 2018. [Online; accessed 31-Mar-2018].

[4] F. Amini, N. H. Riche, B. Lee, A. Monroy-Hernandez, and P. Irani. Authoring data-driven videos with DataClips. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 23(1):501–510, 2017. http://doi.org/10.1109/TVCG.2016.2598647.

[5] B. Bach, N. Kerracher, K. W. Hall, S. Carpendale, J. Kennedy, and N. Henry Riche. Telling stories about dynamic networks with graph comics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 3670–3682, 2016. http://doi.org/10.1145/2858036.2858387.

[6] B. Bach, N. H. Riche, S. Carpendale, and H. Pfister. The emerging genre of data comics. *IEEE Computer Graphics and Applications (CG&A)*, 37(3):6–13, 2017. http://doi.org/10.1109/MCG.2017.33.

[7] B. Bach, Z. Wang, M. Farinella, D. Murray-Rust, and N. H. Riche. Design patterns for data comics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 2018.

[8] A. Bigelow, S. Drucker, D. Fisher, and M. Meyer. Reflections on how designers design with data. In *Proceedings of the ACM Conference on Advanced Visual Interfaces (AVI)*, pp. 17–24, 2014. http://doi.org/10.1145/2598153.2598175.

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

[10] M. Bostock. D3. https://d3js.org/, 2018. [Online; accessed 31-Mar-2018].

[11] M. Brehmer, B. Lee, B. Bach, N. H. Riche, and T. Munzner. Timelines revisited: A design space and considerations for expressive storytelling. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 23(9):2151–2164, 2017. http://10.1109/TVCG.2016.2614803.

[12] M. Brehmer, B. Lee, N. H. Riche, D. Edge, C. White, K. Lytvynets, and D. Tittsworth. Microsoft Timeline Storyteller, 2017. https://timelinestoryteller.com.

[13] J. Caldwell. Information comics: An overview. In *2012 IEEE International Professional Communication Conference*, pp. 1–7, 2012. http://doi.org/10.1109/IPCC.2012.6408645.

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

[15] R. Duncan, M. J. Smith, and P. Levitz. *The Power of Comics: History, Form, and Culture*. Bloomsbury Publishing, 2015.

[16] T. Dwyer. WebCoLa. http://ialab.it.monash.edu/webcola/, 2018. [Online; accessed 31-Mar-2018].

[17] W. Eisner. *Comics and Sequential Art: Principles and Practices from the Legendary Cartoonist*. WW Norton & Company, 2008.

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

[19] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit. From visual exploration to storytelling and back again. In *Computer Graphics Forum*

(Proceedings of EuroVis), pp. 491–500, 2016. http://doi.org/10.1111/cgf.12925.

[20] M. J. Green and K. R. Myers. Graphic medicine: use of comics in medical education and patient care. *BMJ: British Medical Journal (Online)*, 340, 2010.

[21] M. Haesen, J. Meskens, K. Luyten, and K. Coninx. Draw me a storyboard: Incorporating principles & techniques of comics... In *Proceedings of the BCS Interaction Specialist Group Conference*, pp. 133–142, 2010. http://dl.acm.org/citation.cfm?id=2146303.2146323.

[22] K. Hinckley, X. Bi, M. Pahud, and B. Buxton. Informal information gathering techniques for active reading. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 1893–1896, 2012. http://doi.org/10.1145/2207676.2208327.

[23] K. Hinckley, K. Yatani, M. Pahud, N. Coddington, J. Rodenhouse, A. Wilson, H. Benko, and B. Buxton. Pen + touch = new tools. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pp. 27–36, 2010. http://doi.org/10.1145/1866029.1866036.

[24] J. Jo, S. L'Yi, B. Lee, and J. Seo. 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)*, pp. 2660–2671, 2017. http://doi.org/10.1145/3025453.3025752.

[25] N. W. Kim, E. Schweickart, Z. Liu, M. Dontcheva, W. Li, J. Popovic, and H. Pfister. Data-driven guides: Supporting expressive design for information graphics. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 23(1):491–500, 2017. http://doi.org/10.1109/TVCG.2016.2598620.

[26] J. Krause. Bubble sets. https://github.com/JosuaKrause/bubblesets-js, 2018. [Online; accessed 31-Mar-2018].

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

[28] Z. Liu, J. Thompson, A. Wilson, M. Dontcheva, J. Delorey, S. Grigg, B. Kerr, and J. Stasko. 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)*, 2018.

[29] S. Lu. React Annotation. http://react-annotation.susielu.com/, 2018. [Online; accessed 31-Mar-2018].

[30] S. McCloud. Understanding Comics: The Invisible Art, 1993.

[31] S. McKenna, N. H. Riche, B. Lee, J. Boy, and M. Meyer. Visual narrative flow: Exploring factors shaping data visualization story reading experiences. *Computer Graphics Forum (Proceedings of EuroVis)*, 36(3):377–387. http://doi.org/10.1111/cgf.13195.

[32] N. Moraveji, J. Li, J. Ding, P. O'Kelley, and S. Woolf. Comicboarding: Using comics as proxies for participatory design with children. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 1371–1374, 2007. http://doi.org/10.1145/1240624.1240832.

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

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

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

[36] D. Ren, M. Brehmer, B. Lee, T. Höllerer, and E. K. Choe. ChartAccent: Annotation for data-driven storytelling. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*, pp. 230–239, 2017. http://doi.org/10.1109/PACIFICVIS.2017.8031599.

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

[38] A. Satyanarayan and J. Heer. Authoring narrative visualizations with Ellipsis. *Computer Graphics Forum (Proceedings of EuroVis)*, 33(3):361–370. http://doi.org/10.1111/cgf.12392.

[39] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. *Computer Graphics Forum (Proceedings of EuroVis)*, 33(3):351–360. http://doi.org/10.1111/cgf.12391.

[40] A. S. Spritzer, J. Boy, P. Dragicevic, J. Fekete, and C. M. D. S. Freitas. Towards a smooth design process for static communicative nodelink diagrams. *Computer Graphics Forum (Proceedings of EuroVis)*, 34(3):461–470. http://doi.org/10.1111/cgf.12658.

[41] Tableau. Story Points, 2018. https://tabsoft.co/2jxghOC.

[42] M. Tatalovic. Science comics as tools for science education and communication: a brief, exploratory study. *Journal of Science Communication*, 8(4), 2009.

[43] J. Walny, S. Huron, C. Perin, T. Wun, R. Pusch, and S. Carpendale. Active reading of visualizations. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 24(1):770–780, 2018. http://doi.org/10.1109/TVCG.2017.2745958.

[44] Y. Wang, H. Zhang, H. Huang, X. Chen, Q. Yin, Z. Hou, D. Zhang, Q. Luo, and H. Qu. InfoNice: Easy creation of information graphics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 2018.

[45] H. Xia, B. Araujo, T. Grossman, and D. Wigdor. Object-oriented drawing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 4610–4621, 2016. http://doi.org/10.1145/2858036.2858075.

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

[47] H. Xia, K. Hinckley, M. Pahud, X. Tu, and B. Buxton. WritLarge: Ink unleashed by unified scope, action, and zoom. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 3227–3240, 2017. http://doi.org/10.1145/3025453.3025664.

[48] H. Xia, N. H. Riche, F. Chevalier, B. de Araujo, and D. Wigdor. DataInk: Direct and creative data-oriented drawing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 2018.

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

[50] Z. Zhao, R. Marr, and N. Elmqvist. Data comics: Sequential art for data-driven storytelling. Technical report, Human Computer Interaction Lab, University of Maryland, 2015. http://www.cs.umd.edu/hcil/trs/2015-15/2015-15.pdf.