

Interactive Data Comics

Zezhong Wang, Hugo Romat, Fanny Chevalier, Nathalie Henry Riche, Dave Murray-Rust and Benjamin Bach

Abstract—This paper investigates how to make data comics interactive. Data comics are an effective and versatile means for visual communication, leveraging the power of sequential narration and combined textual and visual content, while providing an overview of the storyline through panels assembled in expressive layouts. While a powerful static storytelling medium that works well on paper support, adding interactivity to data comics can enable non-linear storytelling, personalization, levels of details, explanations, and potentially enriched user experiences. This paper introduces a set of *operations* tailored to support data comics *narrative goals* that go beyond the traditional linear, immutable storyline curated by a story author. The goals and operations include adding and removing panels into pre-defined layouts to support branching, change of perspective, or access to detail-on-demand, as well as providing and modifying data, and interacting with data representation, to support personalization and reader-defined data focus. We propose a lightweight specification language, COMICSCRIPT, for designers to add such interactivity to static comics. To assess the viability of our authoring process, we recruited six professional illustrators, designers and data comics enthusiasts and asked them to craft an interactive comic, allowing us to understand authoring workflow and potential of our approach. We present examples of interactive comics in a gallery. This initial step towards understanding the design space of interactive comics can inform the design of creation tools and experiences for interactive storytelling.

Index Terms—Data comics, Non-linear narrative, interactive storytelling

1 INTRODUCTION

Data comics leverage the storytelling and engaging power of sequential art [21,44] to communicate compelling data-driven stories [8,74]. Built upon the visual language of comics, this medium uses multiple panels integrating visualization and text organized in a meaningful sequence to convey a rich narrative. So far, data comics encountered in the wild [11,15] and research [7,67] are mostly static; the panel layout itself encodes the flow of information, guiding the reader through a predefined sequence of panels [44]. A major advantage of static comics compared to other storytelling mediums is the absence of programming or animation skills required for authoring them, making them more approachable to a wide audience of creators.

Conversely, interactivity is a core element in understanding data through visualization and enables viewers to consume a story in different ways, giving more agency around how the narrative unfolds, and eventually may improve comprehension and recall in storytelling [38,45,50]. Interactivity in storytelling enables the viewer to e.g., drill into parts of a story to access additional information, context, or explanations [12,18,64], or to curate their own version of the story given their own approach to explore the data [25]. The comic artist community has already experimented with interactivity in creative ways such as letting the reader navigate an infinite canvas [43], or linking between nested panels to create multiple narrative paths for a personalized story [29]. These compelling explorations hint at a larger design space for adding interaction to aid storytelling with data comics.

In this paper, we contribute the formalization of a core set of operations for interactive data comics, which can help inform the development of future graphical user interfaces for creation tools. We propose a lightweight scripting approach enabling to add these operations to a set of existing comic panels. Triggered by a user interaction, an operation transforms one or more of the comic components: content, structure, or visual appearance. These operations dynamically append and replace

panels, load and switch comic layouts, highlight and filter elements, as well as enable data input and manipulation.

To inform the set and design of interactive operations, we establish six goals of interactivity for data-driven storytelling, and report on a systematic review of traditional (non-data) comics to understand which interactions the community of comics authors is currently providing to support some of these goals (Sect. 3). When attempting to design and create interactive comics building on these operations, we found no appropriate tool to support rapid prototyping for story authors (Sect. 4). Creating interactive and potentially non-linear data comics in prototyping tools such as Adobe XD or Figma leads to exponentially complex views and a substantial performance challenge. The only existing alternatives are full-fledged programming approaches (e.g., HTML, CSS, and JavaScript), which are less approachable to many designers through their steep learning curve, but also require developers significant development time.

To mitigate this problem and allow story authors to more quickly explore interactivity, we propose a lightweight declarative scripting approach: COMICSCRIPT (Sect. 5). COMICSCRIPT is grounded in our formalization and operationalization of the core building blocks for interactive data comics. COMICSCRIPT is inspired by other declarative languages such as Vega-Lite [52] and DataTheater [39]. In our approach, designers first create panels in traditional graphics editing tools or digitize hand-drawn sketches. Then, they define comic layouts and *script* interaction in a web editor. Thus, rather than specifying graphical elements, COMICSCRIPT allows to augment existing graphics with interactivity using a core set of operations.

To explore the design space of interactive operations and iterate on COMICSCRIPT, we engaged in a two-week-long design process with six illustrators, designers, and postgraduate computer science students (Sect. 6). Feedback suggests that our scripting approach is understandable by non-programmers and fosters exploration of the potential of interaction for data comics. Our study shows potential for interactive storytelling, while providing an extensible basis for exploring future authoring tools and types of interaction. It suggests possibilities for a range of scenarios such as exploration, personalization, learning data literacy as well as engaging viewers with messages. A gallery of interactive data comics can be found online, alongside the editor, tutorials, and detailed documentation: <https://interactivedatacomics.github.io>.

2 BACKGROUND

2.1 Interaction in Storytelling

Interaction plays a crucial role in information visualization, enabling analysts and domain scientists to explore multiple facets of their data

• Z. Wang and B. Bach are with University of Edinburgh. E-mail: {zezhong.wang, bbach}@ed.ac.uk.
• H. Romat is with ETH Zurich. E-mail: hugo.romat@gmail.com.
• F. Chevalier is with University of Toronto. E-mail: fanny@cs.toronto.edu.
• N. Henry Riche is with Microsoft Research. E-mail: nath@microsoft.com.
• D. Murray-Rust is with TU Delft. E-mail: D.S.Murray-Rust@tudelft.nl.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

and discover patterns [17, 31]. Interactive visualization is also valuable to a more general audience and a variety of tasks and contexts [27] such as searching histories of names [70], browsing through a corpus of books [59], or learning experiences in museums [33]. Interaction is prevalent in data-driven storytelling [49], whether it allows the viewer to enhance story structure and enable reader-driven navigation, link different story elements [55], or provide controlled explorations such as dynamic queries [53, 54]. In fact, most data-driven stories encountered in the wild from news media outlets, designers and practitioners provide some degree of interaction (see, e.g., [34] for a comprehensive list).

The value of interaction in storytelling is manyfold: it is used to engage viewers more deeply with the data to enhance their comprehension and memorability, to prompting self-reflection, foster critical thinking, or personalize reading, to name a few [34, 45]. Depending on the type of message evidenced by the data, viewers may learn something new [32, 46], change their perspective on a topic [56, 57], or modify their behavior [58]. A few research studies demonstrated several of the direct benefits of adding interaction to data-driven stories. Hullman et al. [38] demonstrated that interactions eliciting the viewer to externalize their prior knowledge before contrasting it with the data enabled a deeper comprehension and recall of the information. Romat et al. [50] found that providing interactions to personalize visual elements was likely to impact how viewers related to data and encouraged deeper reflection. Zhi et al. [75] demonstrated that interactions may also impact levels of user engagement.

As pointed by Amanda Cox in her keynote at VisWeek 2011 [23], one of the challenges of interactive storytelling is how viewers discover the interactive elements and how to use them. Boy et al. [14] provide solutions to enhance affordances and make these interactions discoverable. Other research shows that different interaction mechanisms may appeal to different people, making it difficult to recommend specific types of interactions [45].

To summarize, while there are challenges to creating appropriate, compelling interactions for storytelling media and making them discoverable, several studies demonstrate direct benefits to readers. Interactive experiences may connect readers more deeply to the data, enhance comprehension and recall of data-driven stories and open up new kinds of engagement. This present work seeks to explore what interactions can be added to the data comics medium, and how it could be done.

2.2 Data Comics

Data comics have attracted the attention of the visualization and storytelling community in recent years [8, 74] (see a collection on the data comics website [11]) and have been defined as “*juxtaposing multiple visualizations into comic strip layouts consisting of a sequence of panels, each appropriately annotated and decorated with both visual and textual elements, and arranged into a sequence that progressively develops the overarching story told in the comic*” [74].

Compared to more open-ended infographics, data comics support understanding of complex information by breaking down the complexity of stories into smaller chunks (i.e. panels), laid out in a sequential order [69]. They have been used in multiple domains such as telling stories about temporal networks [7], presenting statistical analysis [67] and teaching people to decode novel data visualizations [66].

Data comics can be graphically very rich, with styles ranging from clean vector graphics and visualizations created with charting software to free-hand illustrations [66]. This graphic expressiveness can be key to reaching specific audiences and attracting readers, with artistic and engaging presentations often giving richer context around more formal data visualizations. As with infographics or comics for science communication [28], data comics can include more literal references to data through characters, numbers, and metaphors [11]. Data comics can use the unfolding narrative structure to provide a rich space for information about people and places, the origin and context of data, as well as information about data transformation and analysis [67].

Data comics are typically created as ‘static’ pieces, which makes the creation and sharing easy compared to dynamic work such as animations and interactive media. Comics can be read at an individual pace: a reader can pause, reflect, glance back to previous panels, or com-

pare panels. Large and information-rich panels can invite a reader into exploration and discovery. Small multiples, as used in exploratory visualization [6, 40], can provide the reader with an *at-a-glance* overview of actors, data, visualization, insights, and events in a data story, facilitating individual access. While authors such as Chris Ware have been experimenting with non-linear storytelling in static comics, adding dynamic interaction to paper versions allows more possibilities [5].

In this paper, we look at interactions that work with a digital comic on a screen, making static data-comics dynamic by e.g., changing the layout, the visibility of panels, or referencing particular elements. This is designed to complement hand-drawn and computer-generated comic styles equally, by opening spaces for exploration. We hope to increase engagement with the story and data, especially by developing the way that data and narrative presentations relate to each other.

2.3 Authoring Comics, Visualization, and Interaction

Creating comics requires many skills and steps such as storyboarding, illustration, layout planning, and more. Adding interactivity demands an additional set of abilities around defining and keeping track of the state of panels and objects, and configuring dynamic operations. These are currently supported through a range of diverse tools, some targeted directly at comics, and some more general purpose. Besides the mainstream graphic design tools such as Adobe Photoshop, Adobe Illustrator, or design tools like Figma, an array of tools particularly to create traditional comics exist. For example, Clip Studio Paint [2] provides a set of toolkits and features for comic drawing, including layout templates, environments, and characters. Comics can even be created without drawing, through tools that support creation by drag and drop of assets from a library [3, 30, 62], automatic layout solution [16], automatic generation of data story in comic strips [73], presentation of exploratory analyses in a computational notebook as a comic [35], and creating comics from other media formats [19, 26, 60, 63].

Several digital comic creation systems have been developed for authoring and presenting *interactive* comics, including branching comics with multiple storylines [61] or non-branching comics which trace the consequences of audience actions [24]. Making comics tangible, Andrews et al. [5] built a physical and digital environment allowing an audience to interact with predefined objects in comics drawn on paper above a multi-touch table. Other ad-hoc physical solutions involve creative cutting and folding of paper to support branching stories [E28, E29] (see Section 3). However, these tools lack support for the kind of interactions that data practitioners require, such as filtering, leaving a considerable workload for the creator.

For visualization and visual storytelling, researchers have focused on interfaces for non-programmers, supporting direct manipulation and clever data binding methods. Tools have been created for creating visualization from templates [42] as well as creating expressive visualization designs [37, 41, 48]. Tools for data-driven storytelling can create infographics [72], static graph comics [36], personalized glyphs [71] or animated video clips [4]. Only few tools provide interactivity through, e.g., navigating a slideshow [51], showing annotations [47], or scroll-telling [1, 55]. Libraries for creating interactive explorables and articles, such as Tangle [65] and Idyll [22] provide for scripting interactions such as sliders and update visual representations based on these values. Tableau also allows the export of interactive visualizations into a specific layout, resembling comics or dashboards. Visual elements in visualizations are highlighted on mouseover and show tool-tips.

To support more flexible workflows, Bigelow et al. [13] provide a bridge between Illustrator and D3. Our approach is similar to Bigelow et al. and Idyll in spirit, as we also separate the visual design of comics from the specification of interactivity.

3 INTERACTION FOR DATA COMICS

To streamline our exploration and creation of interactive data comics, we define six goals for interaction in data comics. These goals either address shortcomings in static data comics, or open up new opportunities for storytelling, exploration, personalization, and engagement with this medium. We then report on a systematic analysis of interactions in

traditional (non-data) comics to collect specific low-level interactions (Figure. 1) that inform interactions for data comics.

3.1 Goals of Interaction

Our goals are informed by extensive discussion among the authors, our own exploration of creating interactive data comics, our analysis of traditional static and interactive comics and existing static data comics, as well as our own review of the body of work on interaction in storytelling (Sect. 2.1). In defining these goals, we aim to exclude interactions that merely support general engagement without a specific link to data-driven storytelling. For example, general interactions may trigger entertaining animations by moving or shaking elements or panels without any semantics associated with it; objects can be made interactive for the reader to play with it, like a music instrument, or a toy present in a panel scene. As the design space of such playful interactions is virtually limitless, and few studies indicate most effective ones to retain and focus the reader, we opted to scope them out in the present work.

[G1] NAVIGATE—Navigation moves the readers between different parts of the story and provides agency to decide what parts to focus on. Static data comics already enable different levels of navigation through skimming, following the proposed sequence, pausing, focusing only on the text or graphics, or skipping entire panels and pages. Interactions with comics can support or reinforce these different reading experiences, or provide entirely for new modes of navigation through, e.g., hyperlinks, drilling-down into specific panels, flashbacks, or selecting panels of interest. *Interactive navigation in data comics can provide more flexibility to experience a story, follow one’s interests, as well as revisit the information in an already read story.*

[G2] DETAILS ON DEMAND—Well known in information visualization, details on demand provide additional information to the interested reader. Static comics and stories are inherently author-driven with the author deciding on the type of information and level of detail in each panel. Moreover, the space for each panel is naturally limited to show multiple panels side by side, which limits the amount and complexity of content that each panel can afford. Interaction can help provide details about, e.g., visual encodings and visualization techniques [68], data provenance and transformation [67], context, side-stories or other background information. *Designing with DETAILS ON DEMAND in mind, allows an author to streamline the storyline while remaining inclusive to audiences with different interests, backgrounds, and data and visualization literacy skills.*

[G3] CHANGE PERSPECTIVE—Author-driven stories usually present a single perspective of the author. This often requires balancing the amount of information with how these information are presented, the author’s interests with those of the audience, objective aspects with perspectives of subjectivity, as well as between different stories centered around a particular data element. For example, a story focusing on geographic data might favor a ‘global’ holistic perspective over a ‘local’ more personal one. Interaction can enable readers to select and change main characters, seeing different facets of the data. Switching the main character may filter out panels irrelevant to the new character, add panels directly relevant, and/or update existing visuals to put the spotlight on this new character. *CHANGE PERSPECTIVE allows authors to integrate different perspectives in their story and use complementary, contrasting, or contradictory perspectives as narrative devices [9].*

[G4] BRANCH—Branching introduces forking paths into the storyline, where each path results in a slightly, or sometimes drastically different story. This type of non-linear storytelling provides for a very personal reading experience defined by a set of choices that the reader makes. Branching in static comics leads to structures akin to flow charts. Such comics may prove quite challenging to layout, use a lot of space, and make any single path in the story tree difficult and tedious to reconstruct when looking at the overview (see [E30]). Interaction can support branching while offering a linear reading experience for each path in the tree at any given moment, or displaying two or more paths at once for comparison. *In data comics, branching can be used, e.g., to show different versions and perspectives, contrast data and alternatives, as well as explain possible future scenarios (i.e. “what*

if?”), or as reactions to the reader’s answer to a knowledge question.

[G5] PAUSE & REVEAL—Gradually revealing information can be used as a device to build tension and force the reader to pause to think, or make a choice. Many web comics have played with gradual reveal through continuous scrolling, or upon clicking on a visual element or panel. Interaction can build artificial barriers for the reader to advance in the story. These range from a simple proof of attention (e.g. mouse over a panel), to an interaction demonstrating the knowledge of the reader or understanding of the prior panels (e.g. multiple choice question), to prompting the reader to make guess or a decision [38].

[G6] INPUT DATA—Data comics and data stories usually contain data provided by the author. However, interaction allows the reader to input or alter data used in the story, decide on manipulations to apply to the existing data, or on parameters for data simulation. For example, providing input mechanisms for readers to enter their calorie consumption or favorite movie can update the panels content for a more personalized, and potentially compelling message. Inputting (personal) data could make data stories more engaging and informative through personalization and explore the effect of different parameters onto the data through dynamic queries.

3.2 Analysis of Interactive (non-data) Comics

We collected 33 traditional interactive comics (found online) to analyze which interactions existing interactive comics already provide, and to inform the authoring process of interactive data comics. Two of the authors examined the collection and converged on a code book. One of the authors then proceeded to the coding of the corpus, categorizing interactions into higher level operations (e.g. *add/remove element, reorder panel*, Fig. 1) and relating them to our goals G1-G6.

Interaction Analysis—Trigger and Effects We describe an interaction as having a *trigger* mechanism, that is, how the interaction is activated, and the resulting *effects* of the interaction, that is, what changes occur in the comics. We also found interactions to either impact the *content* within panels, or the *structure* of the comic and its panels. References to example comics are indicated as E1, E2. The full list of example comics is found in the Appendix.

Interaction trigger At the top of the table, we list the elements in the comic that acts as the trigger for an interaction. Interactive triggers include (i) a particular *visual element* in a panel, such as an object, or a data point that can be hovered over, clicked, or dragged; (ii) a *widget*, such as a button, a slider, or a text field for the reader to input data or specify a choice, which may be integrated within a panel, or placed on the side; (iii) an entire *panel*, that can be clicked to indicate a choice, or reveal what comes next; or (iv) the whole *canvas*, that the reader can navigate through scrolling [E1], panning or zooming, or even explore in augmented reality by moving a device around [E19, E32].

Panel Content lists all of the changes that occur within the panels of the comics. These include (i) *change viewport* e.g. when the viewer focuses the story on a particular geographical area, a panel showing geographical automatically recenters on that area [E4]; (ii) *add/remove element* in a panel, for instance, when the reader makes a choice in a given panel, this choice is reflected in the story (e.g. selecting an apple or an orange in [E20] results in the character holding the corresponding fruit in their hand); (iii) *change properties* of existing elements, as for instance updating their color, size or location as the result of a story choice [E4], or putting a halo effect to put emphasis on the visual element; (iv) *input data*, where the content of the panel is updated based on the underlying data that changes, e.g. a line chart [E4].

Panels Layout lists all of the changes that impact the comics structure, i.e., the number and/or layout of panels. These include (i) *reorder panel*, e.g. when panels can be swapped to result in a different story [E12], or moved around the canvas to reveal panels placed underneath [E11]; (ii) *add/remove panels*, e.g., when a choice results in a different ending, panels that are no longer relevant are removed, and new panels are added [E20], or when panels are revealed progressively, either to build up tension and create a surprise effect [E13], or as the result of particular choices [E22]; (iii) *change panel properties*, such as its aspect ratio, or orientation [E12].

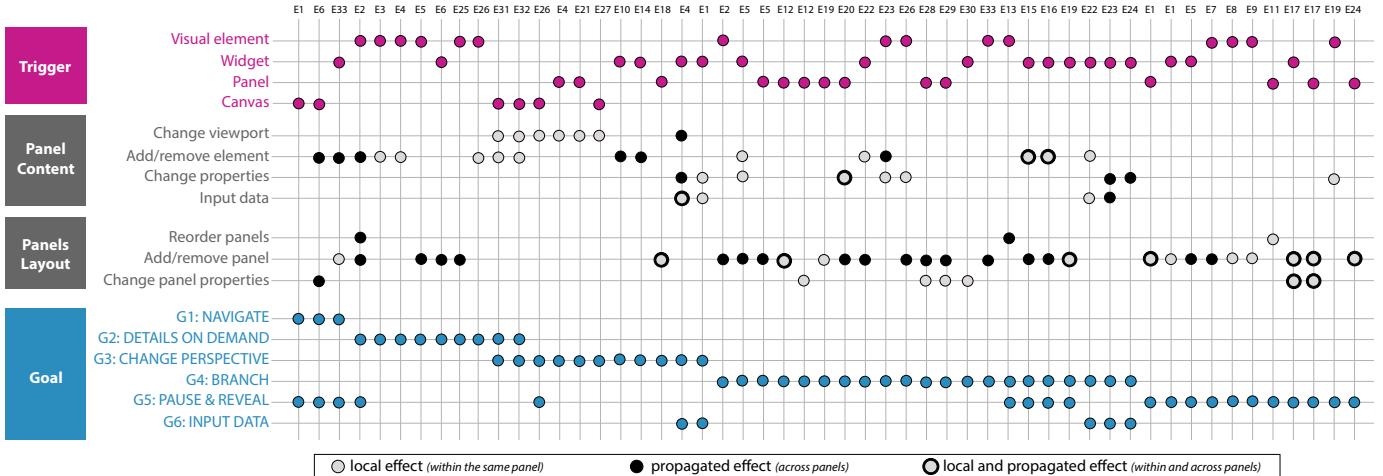


Fig. 1. Results of the manual coding of interactive comics from our corpus (see list at the end of the paper). Interactions are defined by a *trigger*, results in visual effects that can affect *panel content*, and/or *panel layout*, to achieve a particular communication *goal*.

Findings The above triggers and effects can be used to enrich the storytelling experience in otherwise static comics. Below, we discuss trends and examples relating to our goals.

Interactions vary in the *scope of their effect*: changes are sometimes contained within the panel where interaction was triggered, or have an effect that propagates throughout the story. Most commonly, we found a choice or object involved in an early panel to be made consistent in the rest of the story [E10,E14]. These are more profound actions that change the course of the story, and therefore the set of panels presented to the reader. We note also instances where data is inputted in a panel, and where this data is reflected throughout the story [E4,E23].

NAVIGATE is typically supported by scrolling, or hyperlinks to jump to a particular place in the story, when it is not an enforced slideshow presentation that resembles flipping pages through. We found interesting instances of aids to navigation in the form of providing a hint of what choices are possible [E20], or what will come next as the result of a choice [E2], when hovering over different interactive panels or visual elements. Other supporting effects include highlighting the panel where the reader is currently at [E13, E30], or even embodying the reader as a character progressing through panels [E15,E16,E12].

DETAILS ON DEMAND is supported mostly by two types of interaction: showing pop-ups or adding and filtering out elements [E4], as commonly found in visualizations; as well as providing side stories, i.e. branching on a set of secondary panels, to access more details, context, or other perspectives on a point [E6] by adding and removing panels.

CHANGE PERSPECTIVE can be supported through instances where the same character changes their personal perspective, e.g., by changing the viewport through drag-and-drop, or moving and tilting a device in the context of augmented reality [E26,E27]. An interesting instance of a perspective change is by moving a phone device to a particular location on a poster, which then triggers the story of a particular set of characters at the location of focus [E32]. This metaphor is also found in the famous project HERE [E18], where multiple spatio-temporal viewports are materialized as panels around a larger big room, each of which providing a “window” in time of a location in the room.

BRANCH is a popular goal that interaction enables in digital comics. We found many instances where the reader is prompted to make a choice that has repercussion on the rest of the story, by e.g. clicking on a visual element or widget [E22], panel [E20], or indicating the direction of progression [E13]. Other interesting branching instances are the result of personalized input, e.g. the reader writes what a character says [E22], or indicates physical characteristics of a character, which is then used to calculate the outcome of the story [E23].

PAUSE & REVEAL in our corpus uses add/remove elements and panels, for purposes including hiding the result of choices away from the reader to not influence their decision [E13, E28], creating an effect of surprise [E2, E7], prompting the reader to guess a result, or answer a

question [E1], or simply because of screen real estate [E5, E17, E26]. Other examples of reveal are achieved through flipping panels [E9], or through changing the perspective with parallax, revealing potentially important/relevant details only visible from a different angle [E26].

We found only a few instances of INPUT DATA, through a range of traditional widgets, that can be hand-drawn and fully integrated within the comic (like the form and slider in [E22], or buttons in [E4]), or computer-generated, like that found in traditional explorable explanations [E1]. An interesting instance of data input we found is randomization [E10]: random generation of content of balloons (in emoticons), resulting in a new story left to the interpretation to the reader, every single time, and even order and content of panels [E14].

Summary There is a rich set of interactions found in traditional comics and many of these operations can directly be used to support storytelling with data comics; hovering elements to emphasize perspectives, adding panels to show details on demand, or posing the reader a question to decide on which panels to show next. The design space in Fig.1 serves as a blueprint to apply these interactions to data comics.

4 EXAMPLES OF INTERACTIVE DATA COMICS

Over the course of several months, we engaged in creating several interactive data comics, available on our website: <https://interactivedatacomics.github.io/examples.html>. This section describes two interactive comics to show how interaction has been used for interactive data-driven storytelling. We then report and reflect on our creation process and the lessons learned that motivate our scripting approach discussed in Sect. 5.

4.1 Example 1: Interactions for Detail-on-Demand, Navigation, and Changing Perspectives

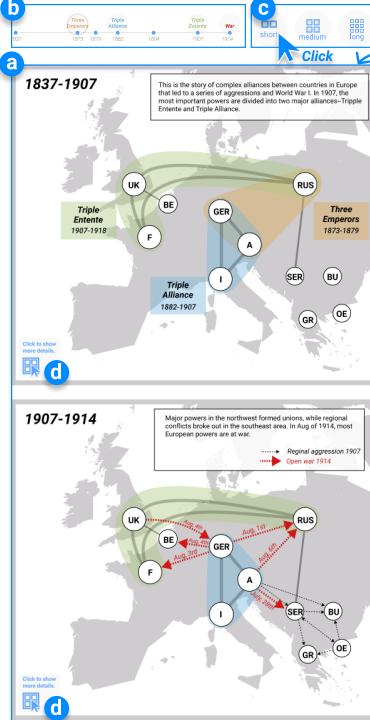
This story presents a dynamic network of **historic alliances** between countries in Europe (Fig. 2), based on a static data comic [7]. With the goal of informing and educating, this interactive comic offers the option to read the story at different predefined information granularities.

The comic is initialized with two panels (a), a timeline highlighting major events (b), and three version buttons able to change the detail of the comic (c). Clicking on a version button shows the corresponding version of the story: (1) short, (2) medium, and (3) long version (NAVIGATE), adding more events and descriptions at a finer level of granularity. Some panels show small blue *detail* buttons (d). Upon click, these buttons drill down into the respective panel, again providing more information about the respective event shown in that panel (DETAILS ON DEMAND). Hovering over a country label or one of the events in the timeline highlights all occurrences of this element throughout the comic, supporting navigation by indicating what panels are relevant to

Alternative versions at different levels of granularity

A reader can click one of the three buttons to switch between a short, medium or long version of the story.

European Alliances before World War I



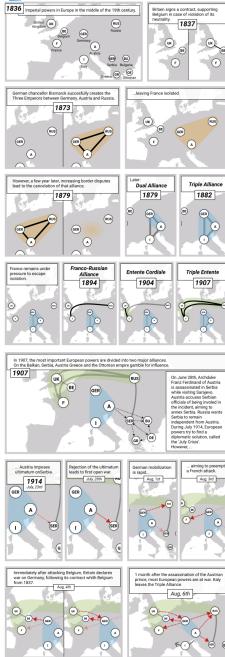
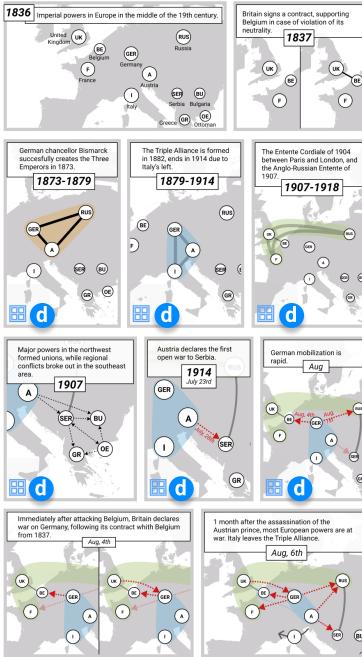
c
d
e
f

Click

short

medium

long



Applied operation

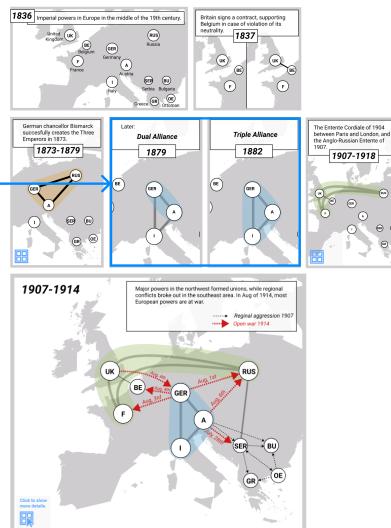
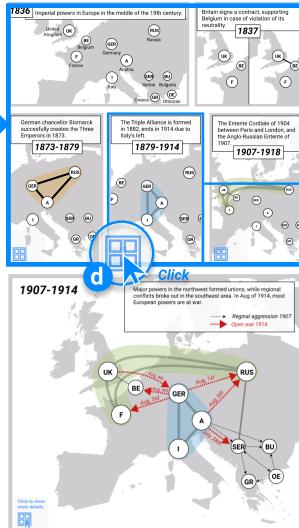
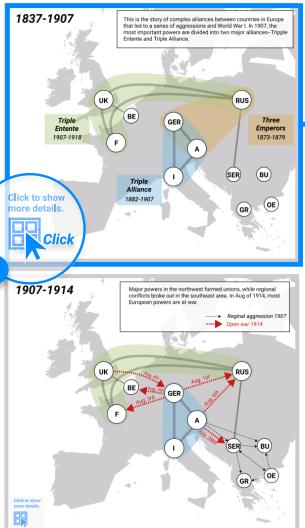
Load layout



These interactions are created by the "Load Layout" operations, which load the panels by three arrays of their panel ids (Section 6.5).

Drill down to reveal more details

Clicking the small event buttons at the bottom corner in a panels can replace that panel with other panels, showing more detail about the events described inside that panel



Applied operation

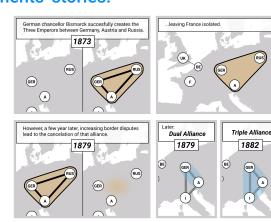
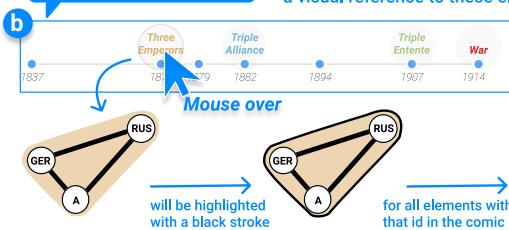
Replace



These interactions are supported by the "replace" operation (Section 6.5).

Highlighting elements

Mouseover highlights occurrences of individual elements across the comics, providing a visual reference to these elements' stories.



Or highlighting the same country by filling a color across all panels.

Applied operation

Highlight



These interactions are supported by the "highlight" operation (Section 6.4).

Fig. 2. The interactive comic of *European Alliances before World War I* presents a dynamic network with a set of countries in Europe, from forming alliances to regressions. With a goal of informing and educating, this interactive comic presents the story of different information granularity (NAVIGATE), provides drill down to acquire information on demand (DETAILS ON DEMAND), and highlights elements as visual navigation (NAVIGATE).

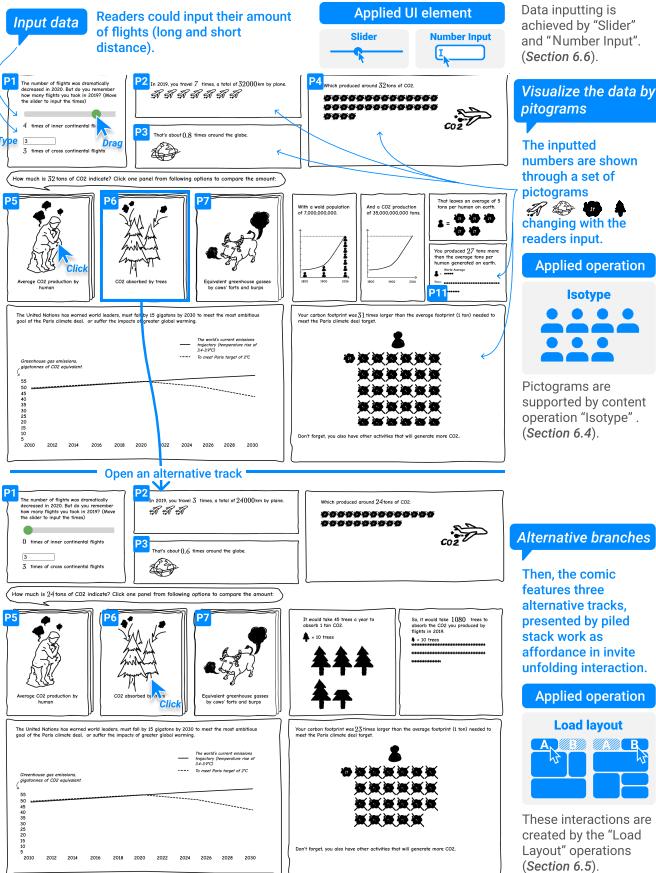


Fig. 3. The comic of *CO₂ emissions* presents viewers with personalized contents by allowing data input, and provides different equivalent comparisons (BRANCH).

a particular event or country (NAVIGATE), as well as allowing to follow a single element through the story (CHANGE PERSPECTIVE).

4.2 Example 2: Interaction for Data Input and Branching

The example in Fig. 3 displays the amount of CO₂ emissions resulting from personal air-traffic and puts these numbers into context using concrete scales [20]. The goal is to engage an audience with their personal data (INPUT DATA) and to promote sensitivity for an important issue. The comic allows a reader to input their amount of flights using a widget (P1). Internally, this number is converted into (roughly) the equivalent CO₂ emissions, and travel distance. Both numbers are shown through a set of pictograms (clouds and globes, respectively) and change with the readers' input (P2-P4). Then, the comic features three alternative tracks (BRANCH), relating the amount of CO₂ emitted to the average CO₂ produced by a human (P5), CO₂ absorbed by trees (P6) and equivalent green house gasses produced by cows (P7). Panels P5-P7, showing these options, are presented as piled stack inspired by [E11]. Clicking one of these three units adds a set of panels showing the values of the respective comparison. The last row of this comic explains the goal of the Paris climate deal in a static panel and finishes by illustrating the exceed of CO₂ of the reader, based on their input.

4.3 Authoring Challenges in Using Design Tools

Authoring interactive data comics like those described above requires a wide range of skills and techniques to function, which makes many existing tools not quite the right fit. As traditional comics, they require strong visual design, with fonts, appealing color schemes, layout, and potentially characters and illustrations. As interactive data storytelling media, they require identifying data elements, placing triggers for interactions, and specifying their effects within and across panels.

In our design process, we first tried common design tools such as Figma, and Adobe XD. These interfaces come with sophisticated

Fig. 4. Workspace in Figma for the European Alliances comic (Fig. 2); each small white square is a full replication of the comic (state) while blue links represent interactive state transitions.

vector graphics drawing capabilities, as well as features for prototyping interactivity, and live previews. Visualizations and content can be created both inside the tool as vector graphics, or imported as hand-drawn pictures in PNG. Then, a designer can set regions in elements that trigger specific operations which are then prototyped inside the tool, by showing or hiding different frames in elements. While being able to support simple interactions, these no-coding prototyping tools suffer from serious limitations; an abundance of manual specification and an increasing amount of duplicating panels was required for even simple comics and interaction and made it a week-long process to create the first versions of the comics in Sect. 6.

The resulting workspace in Figma for the example in Sect. 4.1 is presented in Fig. 4. Maintaining complete states of the comics at all points of the interactions required many panel duplication. These number of elements made the workspace challenging to navigate, the general interaction design hard to preview and possibly fixed. In addition, the number of visual assets required affected the performance of the software itself, making it more tedious to author and preview.

Separating the specification of interactions from the visual assets appeared a necessity, to simplify the authoring process but also to enable the creation of rich sets of interactions. In addition, enabling interactions tailored to data comics and triggered when the reader inputs data or select a specific value are missing from generic design tools. For these reasons, we started developing COMICSCRIPT.

5 AUTHORING INTERACTIVE DATA COMICS

To address the issues raised by our initial exploration in Sect. 4 and to further explore the possibilities for interactive data comics, we developed COMICSCRIPT, a lightweight, declarative language for specifying comics and interactions over them, with particular support for data comics. COMICSCRIPT is our approach to balance and combine aspects from graphical user interfaces and the power of coding interaction. COMICSCRIPT is based on JSON and can be edited and interpreted by an online editor, and the resulting interactive comics can be exported and linked into any website. COMICSCRIPT captures most of the interactions set out in our design space (Sect. 3) through a set of *operations* that change content, layout, structure and appearance:

- **structure operations** manipulate structure and layout of a comic, adding or replacing panels or **specifying panel layout** to make wholesale changes to the layout of all or part of the comic. This includes navigation such as jumping the reader to a specific panel.
 - **content operations** use **classes of objects** to **highlight** and group visual elements across the comic, or apply Isotype techniques to indicate the value of variables.
 - Techniques for **data input** allow users to input numbers through text or sliders and compute **variables and formulas** that can be used in the content or to create **conditions** on whether other operations can be triggered.

In the remainder of this section, we first explain the steps involved in creating interactive comics and then explain how COMICSCRIPT supports scripting interactive comics.

5.1 Creating Comics with COMICSCRIPT

Our process of creating comics has two stages. In the **preparation phase**, a designer specifies each individual panel and its content. This phase includes thinking through the comic layout, as the designer needs to decide on a specific layout of panels, i.e., the size and their order to achieve a meaningful reading order. Here, they can use their usual workflows: a designer could create visualization contents and panels inside vector graphics tools, import hand-drawn images, or rely on existing visualizations in both pixel and vector format. The only restriction is that a single SVG or PNG file should be produced for each panel in the comic, and they should be provided at a public URL so that the web editor can load them.

A COMICSCRIPT comic starts with definitions for each panel, containing a unique ID and a URL for the image (Sect. 5.2). An initial layout is also specified, using the panel IDs in a nested structure to describe rows and columns (Sect. 5.2). At this point, the comic can be displayed as a static webcomic, while interaction is added next.

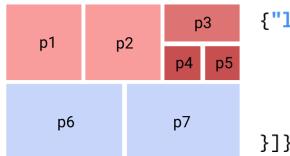
In the **scripting phase**, the designer defines the list of operations that can be applied to the comic. Operations are attached to specific elements through their unique IDs and parameterized in the script. Operations such as adding panels or setting a new layout refer to the panels through their IDs. The set of **structure** operations (Sect. 5.5) works equally well with PNG or SVG files, and allows a large degree of flexibility with panel manipulation. However, if the graphics have been created as vector-based SVG files, a wider range of interaction is possible, as the individual elements in the graphic can be addressed. Vector elements can be used as triggers for operations, e.g., functioning as buttons, or as the targets for effects, such as being added or removed, or highlighted. Objects with the same ID will be applied the same effects, allowing for cross panel interaction, such as brushing to highlight related data. Vector elements with IDs also allow the integration of text input, sliders, isotype pictograms and other types of interaction (Sect. 5.4). The remainder of this section explains the individual components of COMICSCRIPT, and how they come together to create interactive comics. Our set of operations represents a careful balance of low-level operations as they are described in the design space in Sect. 3.2 (e.g., append a panel) and high-level operations that subsume a set of low-level operations (e.g., load layout). More information and a detailed tutorial are included on our website.

5.2 Panels and Layout

COMICSCRIPT loads panels from a URL and assigns them a string ID, which is used throughout the specification to refer to this panel.

```
{"panels": [{"id": "myPanel", "url": "panel1.svg"}]}
```

Since the existence and order of panels can vary over the course of the interaction, COMICSCRIPT provides a flexible way to define layouts. With the width of each panel given by the SVG or PNG image, a layout in COMICSCRIPT can be specified as a nested structure (e.g. panels organized in rows, organized in pages). Alternative layouts can be specified and loaded upon interaction, resulting in rearranging, adding or removing panels. Layouts are thus expressed as nested arrays,



```
{"layouts": [{"name": "init", "panels": [[{"id": "p1", "x": 0, "y": 0}, {"id": "p2", "x": 1, "y": 0}, {"id": "p3", "x": 2, "y": 0, "width": 2}, {"id": "p4", "x": 0, "y": 1, "width": 1}, {"id": "p5", "x": 1, "y": 1, "width": 1}, {"id": "p6", "x": 0, "y": 2, "width": 1}, {"id": "p7", "x": 1, "y": 2, "width": 1}]]}}
```

the first corresponding to a row in a comic, the second to a column in this row and so on (see example above). Panels within a row or column are laid out side-by-side, separated by a gutter which size is specified by the gutter parameter.

5.3 Data and Variables

COMICSCRIPT allows operations to refer to constructs and data: classes and variables. **Classes** group visual elements in the comic into groups, which can be used to highlight elements and refer to groups of elements. Classes are only possible with SVG panels since they require

the referred elements have unique IDs in the file. IDs can be the same across panels, e.g., the first bar can always be called `bar1` across several panels with a bar chart.

```
{"classes": {"name": "countries", "elements": ["france", "germany", "uk"]}}
```

Variables store numerical values which in turn can be shown inside a text element through a place holder that has the same ID as the variable, or used to render a number of ISOTYPE like symbols. Variables can be obtained from data, or through user input, e.g., using a slider.

```
{"variables": [{"name": "movement", "value": 0}, {"name": "totalkm", "value": 0, "what": ["movement", "*1000"]}]}
```

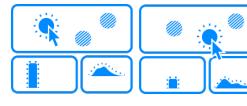
The optional `value` field specifies the default value if the variable is not set. Calculated variables are created by functions, expressed through a `what` attribute. Variables can come directly from data or input via input elements (Sect. 5.6).

5.4 Content Operations

According to our design space in Sect. 3.2, content operations define which elements are shown inside panels and how they are shown. COMICSCRIPT currently provides three content operations.



Highlight changes the appearance of all visual elements with the same ID across all panels (e.g., panels or SVG elements) on mouseover. Highlighting can be used as a visual reference when the author wants the audience to look back/forward at a certain panel or show that an element is the same across the panel (CHANGE PERSPECTIVE, NAVIGATE). An example of highlight is shown in Fig. 2-bottom). The appearance of a highlighted element is specified through its CSS attributes `scale`, `stroke`, `color`.



Layers enable showing and hiding SVG layers and their elements. This operations could be used for presenting DETAILS ON DEMAND and linking texts and its visualisations NAVIGATE. Fig. 5 shows an example how layers can be used to show temporal data as the users clicks dates.



Fig. 5. Using layers, hovering over different time on the timeline triggers different layers.



Isotype aims to reduce the difficulty of understanding abstract content and to support INPUT DATA through the user. The operation creates one or more ISOTYPE like pictograms inside a specified element (i.e., `TreesPlaceholder` in Fig 3) and binds the number of icons to the value of the variable. The size of each icon could be adjusted in `widthIcon`. The icon is loaded from the SVG specified by `url`.

5.5 Structure Operations

Structure operation change the set of shown panels and potentially their order. COMICSCRIPT provides the following structure operations: `append panel`, `load layout`, `replace panel`, and `pan and zoom`.

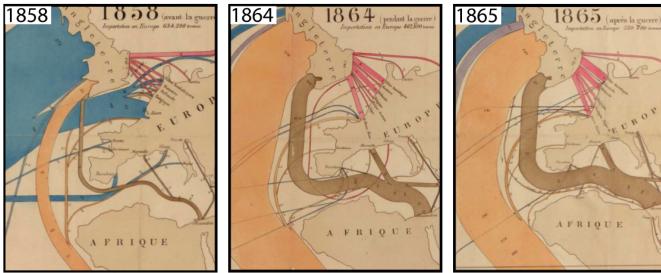
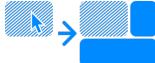


Fig. 6. Example for a pan and zoom operation: panning and zooming in one of the panels with a map synchronize the other panels so that the same detail of the map is shown in all three years.



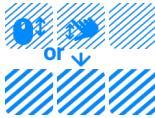
Append adds one or more panels after a specified panel. This operation could be used for different narrative purposes, e.g., **BRANCH**; providing **DETAILS ON DEMAND** by drilling down from this panel (Fig. 2-center); revealing the answer for a question raised in the panel; or **PAUSE & REVEAL**. This operation does not remove or replace any panels. If the creator wants an operation to remove any panel and load new panels, ‘replace’ or ‘Load layout’ should be applied instead. If multiple panels are inserted, the same notion as in layouts is used: e.g., `[“p3”, “p4”, “p5”]`.



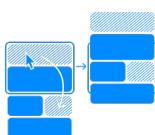
Replace removes a panel to replace it with one or more (new) panels. This operation can be used for replacing panels presenting overview with details or replacing the following narratives by selecting different data set as inputs (**CHANGE PERSPECTIVE**). Replacing an overview with details (**DETAILS ON DEMAND**) (e.g., clicking the button (d) in Fig. 2).



Load layout enables to display a fresh set of panels in a specific layout, removing any panel specified by the `after` attribute. This high-level operation can be used similar to a menu on a website (**NAVIGATE**), which shows alternative versions, e.g., in length, visual style or narration, (Fig. 2). Similar to append, loadlayout could also be used for presenting **DETAILS ON DEMAND**, **CHANGE PERSPECTIVE** and **BRANCH**.



Pan and Zoom creates a simple pan and zoom behavior for a single panel, and can synchronize pan and zoom across multiple panels (e.g., in Fig. 6). Pan and zoom help to achieve the goals **CHANGE PERSPECTIVE** and **DETAILS ON DEMAND**.



Jump brings the user to another panel, similar to anchors in HTML. This operation is designed for non-linear reading by **NAVIGATE**, and to provide visual navigation in a web comic when a pointed panel is not visible due to limited screen real estate.



Conditions can be attached to operations to express constraints when an operation is triggered. This operation allows for personalization, e.g., through **INPUT DATA**. For example, depending on the user’s input in P11, Fig. 3 the comic tells *tons more than*, *tons less than* or *equal to* when comparing that user’s data to other values.

5.6 UI Elements for Data Input

UI elements can be added into SVG panels, replacing an element with a specific ID. Values from these UI elements are then mapped to variables and used to, trigger conditions or show ISOTYPE symbols. COMICSCRIPT currently supports a slider and number input, both demonstrated in Fig. 3.

6 CREATING INTERACTIVE DATA COMICS WITH DESIGNERS

We asked designers, illustrators and visualization practitioners to create interactive data comics with our tool. We wanted to gain insights on many aspects of the creation process as well as improve COMICSCRIPT. We gathered insights on the workflow people follow to craft interactive data comics, their biggest challenges, the role of COMICSCRIPT in the

creating process, the purpose of interaction and the corresponding selected interactive operations as well as identifying missing interactions and operations to iterate on COMICSCRIPT. Respective tutorials and examples can be found on our website.

6.1 Procedure

We conducted a creation workshop consisting of a series of live sessions to help participants create their own interactive comics using COMICSCRIPT. First, we ran a tutorial session, introducing the concept of data comics and interactive data comics. We showed examples of data comics, interactive data and non-data comics. Then we used a simple example to explain the workflow we followed (Sect. 5) including how to create and export individual comic panels with a design tool (Figma) and how to add interactions to the comics using our scripting approach. Participants attended the tutorial live or could consult the recorded version. We offered 2 hours of daily face-to-face online drop-in sessions for answering questions about designing and using COMICSCRIPT. Participants volunteered for an individual 15-minute semi-structured interview after the completion of their comics. During the interviews, we asked five questions followed by an open discussion. Questions asked for 1) the desired intentions of interaction for data comics, 2) inspiration and ideation of interactions, 3) creation process and challenges with COMICSCRIPT, 4) possible future interactions, and 5) the potential of interactive storytelling.

6.2 Participants

Six participants (four from Europe, one from Canada and one from Asia) completed their interactive comics and attended the interview. The participants were: a professional illustrator for 15 years (*Illu*), a postgraduate student in digital media with 6-year experience of creating illustration (*IlluDigit*), a computer science postgraduate student with 15-year experience working as an illustrator (*IlluDigitComp*), a post-graduate student in design and data science (*DesignData*), a doctoral student in creative art and visualization (*ArtViz*), and a team of data storytellers from the company Gramener [30] (*DataStr*) with experience in creating data comics and interactive data comics by programming.

6.3 Qualitative Findings

We gathered feedback during the creation process, and the interview. We organize insights around three themes: creation process, COMICSCRIPT, and participants’ perspective on the value of interaction.

Creation process. Participants worked at their own pace and contacted the instructor occasionally, mostly towards the end of their creation process. Once selecting a topic and gathering data for their story, participants devoted the largest portion of their time to crafting the visuals (e.g. elements and panels) of their comics (seven to ten days). For this stage, they used graphics editing tools such as Adobe Illustrator, Photoshop, Figma, INKSCAPE (<https://inkscape.org/>) and Microsoft PowerPoint. Participants reported thinking of the interactions as they designed the comics and crafted the visuals. They integrated the interactions using COMICSCRIPT over one to three days.

The overall comments suggest that all participants appreciated the final product. This was particularly salient for creators with no coding experience: *“I have all this inspiration imagination and creative ideas but I’ve never been able to transfer them into code because it’s a completely different realm, so what this does is it allows you (a creator) to take your creative imaginative processes ideas and concepts put them into an already constructed editor and then feel proud of yourself.”* (*ArtViz*). Several participants gravitated towards “*different perspective*” (*Illu*) (**CHANGE PERSPECTIVE**) and non-linear comics (**BRANCH**) “*like Chris Ware [in which] you can almost just choose a panel on a page and then see where it leads*” (*IlluComp*).

Participants reported that spending the **majority of their time in graphics editing software** to create the story and design the comic panels. To create different layers or branches, they needed to make more images and variations than it presented. Also, participants **switched between tools**, e.g. taking raster images to vector editors to add IDs if there were buttons or placeholders in the panels, and making sure the size of the image was not changed across tools (*IlluDigit*, *DataStr*).

After the images were exported, participants started editing from one of the examples on our website, since “*the editor already had like a pretext in it and because I could see what your functionality did and with the what your process was I could just change parts of that code and I didn’t have to know too much as a user.*” (ArtViz).

COMICSCRIPT usage. Participants completed their comics **with little to no help** from the facilitator. Overall, they used nine out of eleven operations in the comics they created. Most participants used operations dealing with the panel layout (*Load layout*, *Replace* and *Append*), as well as content operations across panel (*Highlight*, *Layers*). Half of the participants also used operations enabling readers to modify the data (*Slider*, *Condition Number input*, and *Isotype*). COMICSCRIPT definitely involved a **learning curve**: “*Once you understand it then it’s totally fine*” (IlluDigit). After a few explorations using the code samples provided, participants were able to implement their ideas: “*I always felt like I knew what I wanted to do and I knew how to do it*” (IlluComp). Two participants expressed their enthusiasm for a user interface layer, enabling them to drag-and-drop the interaction of their choice directly on top of the comics elements. The JSON language was commented as “*a very refined code for a specific task*” (ArtViz), and “*really structured*” (IlluDigit). The most challenging aspect reported by participants was the **low-tolerance to common syntax error** of COMICSCRIPT. Four participants suggested **extending current operations**, for example, to enable panel replacement when using a slider. Two participants wanted a **finer control** over operations such as resizing or translating elements, which is implemented as low-level operations in COMICSCRIPT but was not surfaced to them at that point.

Goal of interaction. With regards to the **potential of interaction** in data comics, participants indicated that providing navigation (NAVIGATE) and branching (BRANCH) enabled conveying information from different perspectives (CHANGE PERSPECTIVE), e.g., “[*Readers*] can just dive right into it click on something see the connection and it’s way easier to get” (IlluDigit). Participants commented that interactions made data more engaging, “*get the curiosity of people*” (DataStr), “*with interactivity you can kind of be[ing] inside of it*” (IlluDigit), and potentially being used “*as weekly reports replacing dashboard*” (DataStr) or in news article to enable audience to “*think beyond just the passive presentation of imagery*” (IlluComp). A participant suggested that crafting interactive comics with COMICSCRIPT could both **foster creativity and [enable] learning programming**, since this activity “*introduces people to code but also within the creative process, creativity and data can be pushed apart so it’s kind of merging them together in this very nice safe space*” (ArtViz).

7 DISCUSSION AND FUTURE WORK

Crafting interactive data comics involves challenges at multiple levels of the creation process. For several of our workshop participants, the experience of using COMICSCRIPT was their first attempt to create dynamic, computationally mediated material. Challenges occur at multiple levels: from the design and conception of the interactions and non-linear narratives, to the creation and organization of the collection of assets composing the comics, to the implementation of the interactive medium. While the effort is substantial, our workshop revealed that creators see tremendous opportunities in incorporating interactions in data comics, making it worthwhile.

Participants’ feedback also suggests that integrating interaction adds many design considerations to the design of comics—what comics structures can benefit most from interaction, how to make interactivity discoverable—and presents a set of new challenges—designing multiple combinations of content and panel layout and keeping track of the state of the comic at a given point. To provide guidance and potential solutions for these design considerations, further research is needed to evaluate use and engagement with interaction in data comics.

COMICSCRIPT: a lightweight language to create interactive data comics. COMICSCRIPT provides a scripting mechanism to add interactivity to static comics. It serves as a conceptual middle ground between traditional interactive prototyping tools, and full-fledged programming languages. Compared to interactive prototyping tools, COMICSCRIPT facilitates experimenting with interactions and scaling

in number and complexity of the comics (e.g., exploring non-linear data comics). Compared to classic web programming combining HTML, CSS, and Javascript, COMICSCRIPT lowers the complexity of the code and thus lowers the learning curve and saves time to professional developers. COMICSCRIPT provides a core set of interactions that designers, illustrators and creators with minimal or no coding skills can copy and paste and apply to sets of elements and panels crafted in graphics design tools. Our goal with COMICSCRIPT is to strike a balance between approachability of the language and interaction capabilities added to static comics.

Extending COMICSCRIPT while preserving its simplicity. Participants suggested many extensions to COMICSCRIPT: from low-level fine control of interactions (e.g. resizing or rotating elements in panels) to higher-level more advanced layouts (e.g. nesting or overlaying panels). Other desirable features include more sophisticated animated transitions and the creation of visualizations, annotations, data comics design patterns [10] as well as enabling incorporating different affordances to convey interactivity. Animated transitions can help readers understand changes to the comic following an interaction. Creating visualizations, on the other hand, could be done by adding specific directives to the language, e.g., by calling on D3 or Vega-Lite [52]. Creating other content, such as text, characters or annotations, could happen through similar approaches, as partially explored elsewhere [30]. Adding different affordances to suggest interactivity is a research direction in itself [14]. The difficulty with these extensions is to balance the power of the language *vs.* its simplicity. While adding lower-level operations (e.g. hide or resize elements) is trivial, adding more complex functionality may necessitate writing more complex specifications even for simpler examples. To avoid re-implementing functionality and avoid frequent switching between design tools, COMICSCRIPT could be integrated with other visualization and comic design tools.

Towards easier authoring of interactive data comics. COMICSCRIPT is a first step towards understanding what interactions matter most to authors and constitutes a probe into the process they are following when creating them. Still, COMICSCRIPT requires designers to learn a new syntax, although it is arguably simpler than learning full-fledged programming languages. To further lower the effort to create interactive data comics, future work should explore graphical user interfaces that can generate COMICSCRIPT code from directly manipulating the graphical elements in comics and their interactive operations. Perhaps one of the most pressing issues in streamlining the authoring process is to facilitate the creation, reuse and organization of the many graphical and interactive assets necessary for interactive data comics. Observations and comments from our study participants confirm that interactive data comics require a much larger set of visual assets than traditional comics. Managing these assets and generating the variations involved in each interactive state is arguably the main bottleneck for the creation of this new medium and which should be aided through future user interfaces.

Eventually, since our approach is agnostic of panels’ content, it can be extended to other genres of data-driven storytelling such as infographics, flow diagrams or data-driven articles; sets of panels can simply be defined as individual graphical components of these artifacts.

Beyond tools to craft interactive comics, we need to support a change in practice. Authors and designers need to adapt both their mental model and their creation workflow to design and execute interactive data comics. The learning curve is not only about mastering tools (e.g. COMICSCRIPT) to craft them, it largely lies in adjusting the mental model to design them: understanding what is possible, and then what is desirable. A key enabler is certainly to provide inspiration and examples to draw from—participants relied on examples created with the system and existing interactive comics to develop a sense of what is possible. Additional material to support and inspire the design of interactive comics is certainly critical for the future.

ACKNOWLEDGMENTS

We thank our workshop participants for their time and valuable insights. The CO2 emission comic was inspired by earlier exploration of interactive data comics involving Jacob Ritchie and Dandi Huang.

REFERENCES

- [1] Flourish. <https://flourish.studio/>.
- [2] Clip studio paint. <https://www.celsys.co.jp/en/>, 2013.
- [3] Comic life. <http://plasq.com/>, 2021.
- [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*, 23(1):501–510, 2016.
- [5] D. Andrews, C. Barber, S. Efremov, and M. Komarov. Creating and using interactive narratives: Reading and writing branching comics. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, p. 1703–1712. New York, NY, USA, 2012. doi: 10.1145/2207676.2208298
- [6] B. Bach, N. Henry-Riche, T. Dwyer, T. Madhyastha, J.-D. Fekete, and T. Grabowski. Small multiples: Piling time to explore temporal patterns in dynamic networks. In *Computer Graphics Forum*, vol. 34, pp. 31–40, 2015.
- [7] B. Bach, N. Kerracher, K. W. Hall, S. Carpendale, J. Kennedy, and N. Henry Riche. Telling stories about dynamic networks with graph comics. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 3670–3682, 2016.
- [8] B. Bach, N. H. Riche, S. Carpendale, and H. Pfister. The emerging genre of data comics. *IEEE Computer Graphics and Applications*, 37(3):6–13, 2017.
- [9] B. Bach, D. Stefaner, J. Boy, S. Drucker, L. Bartram, J. Wood, P. Ciucarelli, Y. Engelhardt, U. Koeppen, and B. Tversky. Narrative design patterns for data-driven storytelling. In *Data-Driven Storytelling*, pp. 107–133. CRC Press, 2018.
- [10] B. Bach, Z. Wang, M. Farinella, D. Murray-Rust, and N. Henry Riche. Design patterns for data comics. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1–12, 2018.
- [11] B. Bach, Z. Wang, N. H. Riche, and S. Carpendale. Data comics collection. <https://datacomics.github.io>.
- [12] S. K. Badam, Z. Liu, and N. Elmquist. Elastic documents: Coupling text and tables through contextual visualizations for enhanced document reading. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):661–671, 2018.
- [13] A. Bigelow, S. Drucker, D. Fisher, and M. Meyer. Iterating between tools to create and edit visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):481–490, 2016.
- [14] J. Boy, L. Eveillard, F. Detienne, and J.-D. Fekete. Suggested interactivity: Seeking perceived affordances for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):639–648, 2015.
- [15] S. Cagle. Humans have made 8.3bn tons of plastic since 1950. <https://www.theguardian.com/us-news/2019/jun/23/all-the-plastic-ever-made-study-comic>, 2019.
- [16] Y. Cao, A. B. Chan, and R. W. Lau. Automatic stylistic manga layout. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.
- [17] M. Card. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.
- [18] B.-W. Chang, J. Mackinlay, and P. T. Zellweger. Fluidly revealing information in fluid documents. In *Proc. Smart Graphics AAAI Spring Symposium*, 2000.
- [19] B. Chen, R. Jablonsky, J. B. Margines, R. Gupta, and S. Thakkar. Comic circuit: an online community for the creation and consumption of news comics. In *CHI’13 Extended Abstracts on Human Factors in Computing Systems*, pp. 2561–2566. 2013.
- [20] F. Chevalier, R. Vuillemot, and G. Gali. Using concrete scales: A practical framework for effective visual depiction of complex measures. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2426–2435, 2013.
- [21] N. Cohn. *The Visual Language of Comics: Introduction to the Structure and Cognition of Sequential Images*. A&C Black, 2013.
- [22] M. Conlen and J. Heer. Idyll: A markup language for authoring and publishing interactive articles on the web. In *Proc. ACM Symposium on User Interface Software and Technology (UIST)*, pp. 977–989, 2018.
- [23] A. Cox. How editing and design changes news graphics. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. xiii–xiii, 2011.
- [24] E. S. De Lima, B. Feijó, A. L. Furtado, S. D. J. Barbosa, C. T. Pozzer, and A. E. Ciarlini. Non-branching interactive comics. In *Proc. Conference on Advances in Computer Entertainment Technology*, pp. 230–245. Springer, 2013.
- [25] P. Dragicevic, Y. Jansen, A. Sarma, M. Kay, and F. Chevalier. Increasing the transparency of research papers with explorable multiverse analyses. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1–15, 2019.
- [26] A. Durrant, D. Rowland, D. S. Kirk, S. Benford, J. E. Fischer, and D. McAuley. Automics: souvenir generating photoware for theme parks. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1767–1776, 2011.
- [27] N. Elmquist, A. V. Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, 10(4):327–340, 2011.
- [28] M. Farinella. The potential of comics in science communication. *Journal of science communication*, 17(1):Y01, 2018.
- [29] D. M. Goodbrey. From comic to hypercomic. In *cultural excavation and formal expression in the graphic novel*, pp. 291–302. Brill, 2013.
- [30] Gramener. Comicgen. <https://gramener.com/comicgen/#>, 2019.
- [31] J. Heer and B. Shneiderman. Interactive dynamics for visual analysis. *Communications of the ACM*, 55(4):45–54, 2012.
- [32] J. Hilton, N. Cammarata, S. Carter, G. Goh, and C. Olah. Understanding rl vision. *Distill*, 5(11):e29, 2020.
- [33] U. Hinrichs, H. Schmidt, and S. Carpendale. EMDialog: Bringing information visualization into the museum. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1181–1188, 2008.
- [34] F. Hohman, M. Conlen, J. Heer, and D. H. P. Chau. Communicating with interactive articles. *Distill*, 5(9), 2020.
- [35] D. Kang, T. Ho, N. Marquardt, B. Mutlu, and A. Bianchi. Toonnote: Improving communication in computational notebooks using interactive data comics. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1–14, 2021.
- [36] N. W. Kim, N. Henry Riche, B. Bach, G. Xu, M. Brehmer, K. Hinckley, M. Pahud, H. Xia, M. J. McGuffin, and H. Pfister. Dataatoon: Drawing dynamic network comics with pen+ touch interaction. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1–12, 2019.
- [37] 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*, 23(1):491–500, 2016.
- [38] Y.-S. Kim, K. Reinecke, and J. Hullman. Explaining the gap: Visualizing one’s predictions improves recall and comprehension of data. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1375–1386, 2017.
- [39] S. Lau and P. J. Guo. Data Theater: A live programming environment for prototyping data-driven explorable explanations. In *Workshop on Live Programming (LIVE)*, 2020.
- [40] F. Lekschas, X. Zhou, W. Chen, N. Gehlenborg, B. Bach, and H. Pfister. A generic framework and library for exploration of small multiples through interactive piling. *IEEE Transactions on Visualization and Computer Graphics*, 2020.
- [41] 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 *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1–13, 2018.
- [42] M. Mauri, T. Elli, G. Caviglia, G. Ubaldi, and M. Azzi. Rawgraphs: a visualisation platform to create open outputs. In *Conference on Italian SIGCHI Chapter*, pp. 1–5, 2017.
- [43] S. McCloud. *Reinventing comics: How imagination and technology are revolutionizing an art form*. Harper Collins, 2000.
- [44] S. McCloud and A. Manning. Understanding comics: The invisible art. *IEEE Transactions on Professional Communications*, 41(1):66–69, 1998.
- [45] S. McKenna, N. Henry Riche, B. Lee, J. Boy, and M. Meyer. Visual narrative flow: Exploring factors shaping data visualization story reading experiences. In *Computer Graphics Forum*, vol. 36, pp. 377–387, 2017.
- [46] R2D3. A visual introduction to machine learning. <http://www.r2d3.us/visual-intro-to-machine-learning-part-1/>, 2015.
- [47] D. Ren, M. Brehmer, B. Lee, T. Höllerer, and E. K. Choe. Chartaccent: Annotation for data-driven storytelling. In *Proc. IEEE Pacific Visualization Symposium (PacificVis)*, pp. 230–239, 2017.
- [48] D. Ren, B. Lee, and M. Brehmer. Charticulator: Interactive construction of bespoke chart layouts. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):789–799, 2018.
- [49] N. H. Riche, C. Hurter, N. Diakopoulos, and S. Carpendale. *Data-driven*

- storytelling*. CRC Press, 2018.
- [50] H. Romat, N. Henry Riche, C. Hurter, S. Drucker, F. Amini, and K. Hinckley. Dear pictograph: Investigating the role of personalization and immersion for consuming and enjoying visualizations. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1–13, 2020.
- [51] A. Satyanarayan and J. Heer. Authoring narrative visualizations with ellipsis. In *Computer Graphics Forum*, vol. 33, pp. 361–370, 2014.
- [52] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2016.
- [53] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, 2010.
- [54] C. Stolper, B. Lee, N. Henry Riche, J. Stasko, C. Hurter, N. Diakopoulos, and S. Carpendale. Data-driven storytelling techniques: Analysis of a curated collection of visual stories. *Data-Driven Storytelling*, pp. 85–105, 2018.
- [55] N. Sultanum, F. Chevalier, Z. Bylinkskii, and Z. Liu. Leveraging text-chart links to support authoring of data-driven articles with vizflow. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, p. In press., 2021.
- [56] The New York Times. You draw it: How family income predicts children’s college chances. <https://www.nytimes.com/interactive/2015/05/28/upshot/you-draw-it-how-family-income-affects-childrens-college-chances.html>, 2015.
- [57] The New York Times. You draw it: Just how bad is the drug overdose epidemic. <https://www.nytimes.com/interactive/2017/04/14/upshot/drug-overdose-epidemic-you-draw-it.html>, 2017.
- [58] The Washington Post. Why outbreaks like coronavirus spread exponentially, and how to ‘flatten the curve’. <https://www.washingtonpost.com/graphics/2020/world/corona-simulator/>, 2020.
- [59] A. Thudt, U. Hinrichs, and S. Carpendale. The bohemian bookshelf supporting serendipitous discoveries through visualization. Technical report, University of Calgary, 2011.
- [60] H. Tobita. Digestmanga: interactive movie summarizing through comic visualization. In *CHI’10 Extended Abstracts on Human Factors in Computing Systems*, pp. 3751–3756. 2010.
- [61] H. Tobita. Comic-crowd: interactive comic creation that supports multiple storylines, visualizations, and platforms. In *Proc. Conference on Mobile and Ubiquitous Multimedia*, pp. 163–172, 2015.
- [62] H. Tobita and K. Shibasaki. Enformanga: Interactive comic creation using drag-and-drop and deformation. In *Proc. IEEE International Symposium on Multimedia*, pp. 269–274, 2009.
- [63] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky. Video manga: generating semantically meaningful video summaries. In *Proc. ACM Conference on Multimedia*, pp. 383–392, 1999.
- [64] B. Victor. Explorable explanations. <http://worrydream.com/ExplorableExplanations/>, 2011.
- [65] B. Victor. Tangle: explorable explanations made easy. <http://worrydream.com/Tangle>, 2013.
- [66] Z. Wang, H. Dingwall, and B. Bach. Teaching data visualization and storytelling with data comic workshops. In *CHI’19 Extended Abstracts on Human Factors in Computing Systems*, pp. 1–9, 2019.
- [67] Z. Wang, J. Ritchie, J. Zhou, F. Chevalier, and B. Bach. Data comics for reporting controlled user studies in human-computer interaction. *IEEE Transactions on Visualization and Computer Graphics*, 2020.
- [68] Z. Wang, L. Sundin, D. Murray-Rust, and B. Bach. Cheat sheets for data visualization techniques. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1–13, 2020.
- [69] Z. Wang, S. Wang, M. Farinella, D. Murray-Rust, N. Henry Riche, and B. Bach. Comparing effectiveness and engagement of data comics and infographics. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, p. 1–12, 2019.
- [70] M. Wattenberg. Baby names, visualization, and social data analysis. In *Proceedings of the IEEE Symposium on Information Visualization (Infovis’05)*, pp. 1–7, 2005.
- [71] H. Xia, N. Henry Riche, F. Chevalier, B. De Araujo, and D. Wigdor. DataInk: Direct and creative data-oriented drawing. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1–13, 2018.
- [72] J. E. Zhang, N. Sultanum, A. Bezerianos, and F. Chevalier. DataQuilt: Extracting visual elements from images to craft pictorial visualizations. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1–13, 2020.
- [73] J. Zhao, S. Xu, S. Chandrasegaran, C. Bryan, F. Du, A. Mishra, X. Qian, Y. Li, and K.-L. Ma. Chartstory: Automated partitioning, layout, and captioning of charts into comic-style narratives. *arXiv preprint arXiv:2103.03996*, 2021.
- [74] Z. Zhao, R. Marr, and N. Elmqvist. Data comics : Sequential art for data-driven storytelling. Technical report, University of Maryland, 2015.
- [75] Q. Zhi, A. Ottley, and R. Metoyer. Linking and layout: Exploring the integration of text and visualization in storytelling. In *Computer Graphics Forum*, vol. 38, pp. 675–685, 2019.

CATALOG OF INTERACTIVE COMICS

- [E1] N. Case. How to Remember Everything Forever-ish. 2018. <https://ncase.me/remember/>
- [E2] R. Chang. What the Blob. Undated. <https://rosalind-chang.myporfolio.com/what-the-blob>
- [E3] M. Cisneros. Is That True?. 2017. <https://public.tableau.com/profile/mikevizneros#/vizhome/IsThatRight/IsThatTrue>
- [E4] Gramener. Yikes! 6ft please. Undated. <https://gramener.com/nyc311/>
- [E5] Goodboy Digital. Tell me your secrets. <https://www.goodboyydigital.com/case-study/tell-me-your-secrets>
- [E6] M. Huynh. The boat. Undated. <https://www.sbs.com.au/theboat/>
- [E7] M. Kühni. CLAMS II. Undated. <https://abwaesser.net/comic/clams-ii/#drop>
- [E8] M. Kühni. Coming of age. Undated. <https://abwaesser.net/comic/coming-of-age/>
- [E9] M. Kühni. Heidi. Undated. <https://abwaesser.net/comic/heidi/>
- [E10] M. Kühni. New speak. Undated. <https://abwaesser.net/comic/new-speak/#>
- [E11] M. Kühni. Visiting the aquarium. Undated. <https://abwaesser.net/comic/visiting-the-aquarium/>
- [E12] Loveshack Entertainment. FRAMED Collection. 2018. <http://framed-game.com/>
- [E13] D. Merlin Goodbrey. A Duck Has An Adventure. 2012. <https://www.youtube.com/watch?v=YBRhf2jMHVm>
- [E14] D. Merlin Goodbrey. Hidden factions. 2000. <http://e-merl.com/hyper/hapf/hf1.htm>
- [E15] D. Merlin Goodbrey. Icarus Need. 2012. <https://www.youtube.com/watch?v=tuzakLowWTA>
- [E16] D. Merlin Goodbrey. The Empty Kingdom. 2014. <https://www.youtube.com/watch?v=9vKuLap55CM>
- [E17] S. McCloud. The Right Number. 2003. <http://www.scottmccloud.com/1-webcomics/trn-intro/index.html>
- [E18] R. McGuire. HERE. Undated. <https://screendiver.com/directory/here-by-richard-mcguire/>
- [E19] Monde Binaire. Hello World !!. 2012. <https://transmii.com/project/monde-binaire/>
- [E20] C. Muckenhoupt. Interactive comic prototype. <https://muckenhoupt.itch.io/interactive-comics-prototype>
- [E21] R. Munroe. xkcd: Click and Drag. <https://xkcd.com/1110/>
- [E22] R. Munroe. xkcd: Lorentz <https://xkcd.com/1350/>
- [E23] R. Munroe. xkcd: Throw calculator. <https://xkcd.com/2198/>
- [E24] R. Munroe. xkcd: xkcloud <https://xkcd.com/1506/>
- [E25] R. Munroe. xkcd: Zach Weiner (SMBC) <https://xkcd.com/826/>
- [E26] Netwars. The Butterfly Attack. 2014. <http://www.netwars-project.com/novels/graphic-novel>
- [E27] PLASTIEK. RRR2. 2018. <http://plastiek.com/RRR-2/RRR-2.html>
- [E28] J. Shiga. How To Make Your Own Interactive Comic. 2010. <https://www.youtube.com/watch?v=A9Er7kmCPcI>
- [E29] J. Shiga. Numberphile. Paper Caluculator. 2016. <https://www.youtube.com/watch?v=siawhQBRC8I>
- [E30] J. Shiga. Meanwhile. <https://www.zarfhome.com/meanwhile/>
- [E31] Sutu et al. Modern Polaxis. 2014. <https://screendiver.com/directory/modern-polaxis/>
- [E32] Transmii Studio. Memory Catcher. 2014. <https://screendiver.com/directory/memory-catcher/>
- [E33] Turbomedial. Coup de Pompe. Undated. <https://turbointeractive.fr/coup-de-pompe/>