GAIA: A Grammar for Animated Infographics

Category: Research

Paper Type: please specify

Abstract—Animated infographics are a potent storytelling tool in various fields, from business to science communication. However, their creation is often laborious, involving meticulous element selection, property adjustments, and timing management. Therefore, we propose GAIA, a declarative grammar designed to streamline animated infographic creation. With GAIA, users can craft animations expressively and intuitively, abstracting away the underlying implementation. Via an abstraction of animated targets, GAIA provides an approach to express, save and share infographic animations from expert designers and allows novices to reuse them, significantly speeding up the creation process. We implemented a componentized GAIA compiler prototype and built a demo system based on it with diverse examples to showcase its usability. In a user study involving non-expert participants, individuals with no prior animation experience successfully replicated real-world animations within just one hour of learning. GAIA has been applied in several practical projects, further advancing the field of animated infographic authoring.

Index Terms—Infographics, animation, declarative language

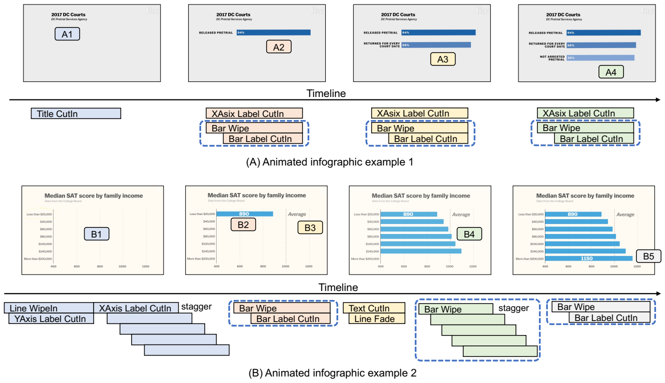
1 INTRODUCTION

Infographics use a combination of graphics, text, images, and data to inform and engage. In animated infographics, some elements are animated to illustrate data or emphasize key points more excitingly and engagingly. The animation of infographics is considered a powerful tool for storytelling. It has been widely used in multiple areas, such as business, publicity, science popularization and so on. Compared to static infographics, it can interpret data or ideas more vividly and grabs the audience's attention quickly, appreciated and shared by people for years [?, ?].

Unfortunately, the creation of animated infographics is tedious and time-consuming, especially for novice users [?, ?, ?]. It can take weeks to authorize graphical animation itself that lasts around one minute [?]. The creation process involves multiple separated steps [?, ?]. Initially, designers need to create static infographics (like the rightmost figure in Figure 1(A)) using feature-rich software such as Adobe Illustrator [?]. Then they can export their infographics as Scalable Vector Graphics (SVGs) or other kinds of files so that downstream tools can be employed for animation creation. Interactive tools (*e.g.,* Adobe After Effects [?]) are usually used for their expressiveness. However, they only provide low-level operations, like keyframes or interpolation, for motion authoring, which requires expertise and is time-consuming [?]. Some users might choose programming tools (*e.g.,* D3 [?]) because they can precisely control property changes and organize timelines through program flow. In recent years, declarative languages are preferred by designers for this task as they favor conciseness over expressiveness [?]. Canis [?] falls into this category, which provides a declarative language for creating animated charts. These techniques simplify the creation process but can only use a predefined library of simple animations usually, which limits their usage. Here, we introduce two examples to better illustrate the challenges of creating animated infographics with existing tools.

Figure 1(A) illustrates four keyframes and the timeline of an animated. The creation process is as follows: To implement the CutIn effect in A1, each element needs to be masked separately. In A1, the y coordinates of title texts should be set at the start/end keyframe to let them move out from the masks. Then following the narration, three similar animation groups are created at the corresponding timestamp (A2-4). In each group, the bar enters using a Wipe effect (the mask is also required) and texts of the axis and the bar enter using CutIn. The relative offsets, durations and easing functions for each animation are carefully set to produce a harmonious result.

A more complex example is shown in Figure 1(B). Given a static SVG shown as the rightmost figure in Figure 1(B), five animations are added (B1-5). First, the chart framework enters at the beginning (B1) when the author explains this chart. Similarly, animations for each element should be handled one by one, using different effects depending on the element. For example, the lines use the WipeIn effect, while the x-label on the left uses the CutIn effect. Increasing offsets are set for the x-axis labels to achieve the stagger effect. Following the narration, the bar and bar label for the category "less than $20,000" enter (B2). A line and a label fade in to illustrate the average score (B3). Then, bars in the middle wipe in with a stagger (B4) to show a positive correlation. Finally, the corresponding bar and label for "More than $200,000" are animated with similar effects (B5).



(A) Animated infographic example 1

(B) Animated infographic example 2

Fig. 1: (A) An animated infographic about the pretrial released defendants of the D.C. court in 2017 (retrieved from https://youtu.be/5mhwDSHKEyM?t=215). (B) An animated infographic about the relation between SAT score and family income (retrieved from https://youtu.be/WjVVwMGJ9S8?t=334). Similar animations are framed in blue.

(A) An animated infographic about the pretrial released defendants of the D.C. court in 2017. (B) An animated infographic about the relation between SAT score and family income. Similar animations are framed in blue.

Two obstacles hinder the designer's creation process. First, **non-intuitive and expert-driven creation approaches**. During the creation, each animation needs to: 1) select one animation target; 2) create a start keyframe at the appropriate timestamp and set initial properties; 3) create an end keyframe at the appropriate timestamp and set final properties; 4) specify an appropriate easing function. Designers need to deal with these cumbersome settings with an understanding of underlying concepts like keyframes, channels, *etc.* [?, ?] This is labor-intensive but unavoidable even for experts [?]. Second, **the repetitive creation of animations**. We observe that animations share similarities, not only between different element groups of the same instance but also across multiple similar instances. For example, animations framed in blue in

Figure 1 are similar and applied on the bars and bar labels, which can be extracted and reused. As another example, entering animations of axes exist in many animated chart videos. As mentioned by Thompson *et al.* [**?**], reusing animations on similar groups is useful, which simplifies and speeds up the creation process. Given these situations, though these existing tools are helpful, designers of animated infographics still take a procedural approach to create them from scratch: draw the elements, select targets and create animations one by one.

In this work, we aim to reduce the burden of creating animated infographics greatly. We propose GAIA, a declarative Grammar for Animated InfogrAphics. GAIA is motivated by three goals.

Expressivity: GAIA is both expressive and intuitive for reading and writing. In GAIA, animations are represented by effects applied to a group of visual elements instead of keyframes created individually for each separate element. Groups in one infographic (as we called, instance) form a hierarchy and each group can apply effects. Besides, grouping is also a way to organize animations [**?**]. Animations for groups can be combined with several alignment strategies (*e.g.,* start before previous), which also form a hierarchical structure simplifying the creation and refinement of timelines. GAIA provides a declarative grammar following these insights, which enables designers to create animations for groups and organize them in a hierarchical timeline. In this way, GAIA can support flexible and intuitive animation creation for infographics and enable users to express and refine complex animation timelines easily.

Reusability: Animations in GAIA can be reused. As mentioned above, reusing similar animations saves time and effort [**?**]. However, due to the diversity of animated objects, it is challenging even for a single simple animation, let alone a well-designed animation combo. In GAIA, animations are declared for a class of infographics, not a concrete instance. To this end, we conduct a target abstraction to describe graphical elements in infographics. The animations are not directly bound to the graphical elements but instead, bound to the role of the elements in a unified representation. For example, GAIA can declare a design for different types of animated elements in axes, such as domain, ticks and labels, without relying on a concrete instance. A wide range of SVGs from the internet or other design tools can be transformed into a format that contains the roles of each animated target and other information (like data). Then users can reuse complex but well-designed animations from themselves or other designers.

Extensibility: Template library in GAIA can be extended. Good design (including the coordination of effects, the design of parameters, *etc.*) is difficult to express, store, and share in multiple creations. Some existing declarative languages provide an animation library, but provided effects are preset and simple, like fade and wipe, designed for generic usage. Meanwhile, these techniques don't provide support for extending the animation library. The only way to do this is through a code-level implementation. Based on the target abstraction, GAIA allows expert designers to define new templates easily through a consistent declarative grammar as animation creation, then register them to the library and all users can use them just as other existing templates. For example, users can combine templates for bars and axes to create a new template for bar charts. Parameter abstraction is also introduced in GAIA so that a template can be more generic and customizable.

We implemented a GAIA compiler and released it as a TS library, which can be embedded in other tools as an animation engine with high-level specs as input. Through a demo system with a variety of examples, we show the expressive of GAIA. We conducted a user study with 8 participants with no animation design experience to illustrate the ease of use of GAIA and gather feedback. All participants can acquire the core features of GAIA within 1h of learning, and successfully replicate the data video designed by professionals. GAIA has been applied in a real-world project, and through interviews with three experts who employed it extensively, we garnered positive feedback and valuable suggestions for future improvements.

In brief, the main contributions of this work, in presentation order, include:

- **A set of high-level grammars** that enables intuitive creation of animations (**??**). Users can build complex animations via a hierarchical view and concatenating operations.
- **Target abstraction** decouples animation declarations from concrete instances, enabling reusability and extendability of GAIA, which is the first attempt for declarative animation grammars (**??**).
- **A compiler prototype**[1] and **workflow** of GAIA in different points of views (**??**).
- **An evaluation** on the usability of GAIA, including **a demo system** with a variety of examples used in the case study, a user study involved novice users, and semi-structured interviews conducted with experts experienced in using GAIA.

[1] The src code and the deployment version of both GAIA compiler and demo system are available in supplementary materials. An online version of the demo system is also available at `https://gaia-ui.azurewebsites.net`.