Demonstration of Masonview: Content-Driven Viewport Management

Bryan Min
University of California San Diego
La Jolla, USA
bdmin@ucsd.edu

Matthew T Beaudouin-Lafon
University of California San Diego
La Jolla, USA
mbeaudouinlafon@ucsd.edu

Sangho Suh
University of California San Diego
La Jolla, USA
sanghosuh@ucsd.edu

Haijun Xia
University of California San Diego
La Jolla, USA
haijunxia@ucsd.edu

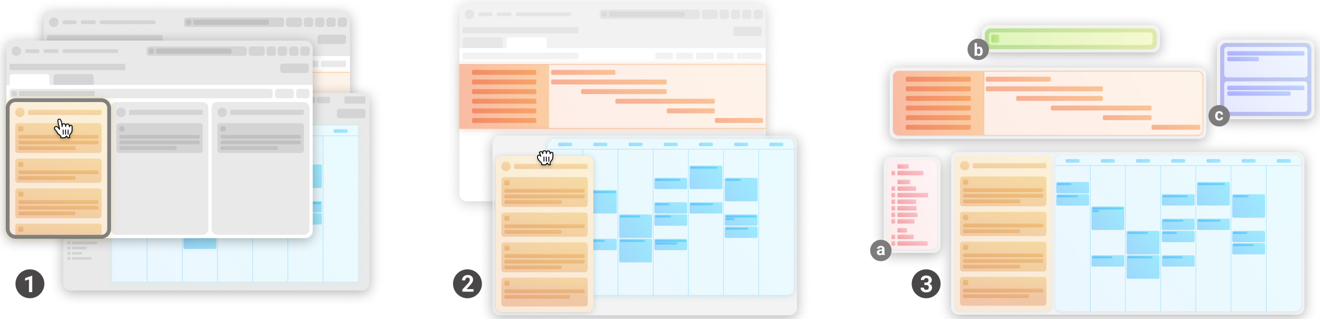


Figure 1: Masonview is a content-driven viewport manager for the desktop that enables users to (1) form *viewports* for web components of webpages and (2) construct mashups of viewports under merged *viewplates*. (3) Masonview opens more space in the screen for useful components such as (a) a company-wide messaging channel list, (b) a documentation website's search bar, and (c) a language translator.

ABSTRACT

The web is now rich with information and functionality, allowing users to only use portions of webpages to complete their tasks. However, current desktop interfaces only allow us to manage entire windows, leading to inefficient use of screen space and suboptimal workflows. We present Masonview, a content-driven viewport management system that provides mechanisms to detach desired elements from their webpages as viewports and compose these views into UI mashups with viewplates. We demonstrate how these mechanisms enable more free-form organization and management of content.

CCS CONCEPTS

• **Human-centered computing** → **User interface management systems**; **Interaction techniques**; *Web-based interaction.*

KEYWORDS

Window Management; UI Mashups; End-user Customization; Web.

ACM Reference Format:
Bryan Min, Matthew T Beaudouin-Lafon, Sangho Suh, and Haijun Xia. 2023. Demonstration of Masonview: Content-Driven Viewport Management. In *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23 Adjunct), October 29–November 01, 2023, San Francisco, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3586182.3615827

1 INTRODUCTION

The web has matured to a state where individual components on webpages feature valuable information and functionality. Consequently, users may only require a subset of the webpage to complete their tasks. However, desktop interfaces only allow users to organize entire windows, leading to inefficient use of screen real estate and suboptimal workflows. We argue that users should directly manage content instead of managing windows. Therefore we envision that enabling users to detach and organize these components independently of their originating webpages can transform how end-users interact with information.

Webstrates [3] has demonstrated that the web affords the composition of content and functionality in the Document Object Model (DOM). This enables users to manipulate content from within the

UIST '23 Adjunct, October 29–November 01, 2023, San Francisco, CA, USA

ACM ISBN 979-8-4007-0096-5/23/10.
https://doi.org/10.1145/3586182.3615827

structure of the code, leading to a novel vision for the web's infrastructure. We instead leverage the richness of *existing* webpages and explore how the end-user can directly manipulate content from the desktop. While prior work has explored creating snippets of windows [5] and composing components into mashups [1, 4, 6], no prior work has brought both to the desktop.

We present Masonview, a content-driven viewport management system that lets users decompose webpages into relevant components and compose them into new layouts. Masonview allows users to create *viewports* by cropping a webpage to specific DOM elements while retaining the webpage's functionality and interactivity. Users can then group these viewports under merged *viewplates*, resizing them to represent the viewports in various organizational forms. We demonstrate the power of Masonview with various information tasks to reveal how it unlocks the untapped potential of the component-rich web.

2 MASONVIEW

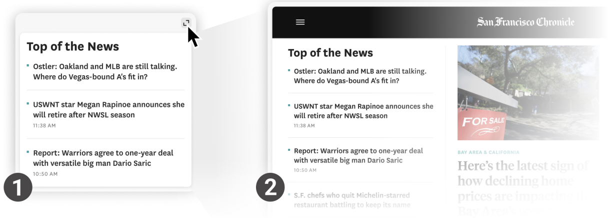
Masonview is a viewport manager that lets users organize and manage components alongside windows. Masonview is built on two concepts: **viewports** and **viewplates**. Viewports can show the full content of the webpage or crop to an individual DOM element. This lets users organize the content they need for their task without dedicating screen space to the content they do not. Viewports are supported by viewplates, substrates that organize and represent viewports in different forms based on the size of the viewplate (Fig. 3). We describe how viewports and viewplates support diverse representations and layouts of content.

2.1 Viewports

Viewports generalize the concept of a window. Windows are limited to showing entire webpages, whereas viewports can display arbitrarily specific components of a webpage. To select a component, the user holds the meta[1] key and points the mouse cursor to a DOM element of interest. The user can then scroll up or down to select a "more parent" or "more child" element. When the user clicks the selection, the viewport crops to the selected element (Fig. 1.1). If the user wishes to retain the original window and create a copy of the element, they can use the meta+shift modifier instead, effectively duplicating the view of that component. The user can also drag the selection to create and then immediately move the viewport to another location.

Cropped viewports can also *unravel* to show their original full views by clicking a button on the header of the viewport. This feature can be especially useful when a viewport hides information that is only occasionally needed. For example, a viewport can contain the top news articles of the day, but the user can unravel the viewport to view the entire front page (Fig. 2). Routing to a new link also unravels the viewport. For instance, making a search in a stock photo search bar component automatically unravels the viewport to reveal the searched result.

Webpages are typically designed to adapt to the size of the windows in which they are rendered. Meanwhile, viewports in Masonview represent a focused element, allowing users to resize the



Top of the News

- Ostler: Oakland and MLB are still talking. Where do Vegas-bound A's fit in?
- USWNT star Megan Rapinoe announces she will retire after NWSL season
  11:38 AM
- Report: Warriors agree to one-year deal with versatile big man Dario Saric
  10:50 AM

Top of the News

- Ostler: Oakland and MLB are still talking. Where do Vegas-bound A's fit in?
- USWNT star Megan Rapinoe announces she will retire after NWSL season
  11:38 AM
- Report: Warriors agree to one-year deal with versatile big man Dario Saric
  10:50 AM
- S.F. chefs who quit Michelin-starred restaurant battling to keep its name

Here's the latest sign of how declining home prices are impacting the Bay Area's

Figure 2: (1) The cropped viewport contains a focused view of the top news articles. (2) Clicking the *unravel* button on the viewport's header reveals the rest of the news webpage.

element itself. As shown in Figures 1.1 and 1.2, the selected task board column is resized to fully display all task items in the column.

2.2 Viewplates

Viewplates are structural substrates for viewports. Viewplates are represented by a rectangle that spans the width and height of all contained viewports. They serve two purposes: 1) consolidating viewports under a unified space and 2) organizing and representing them semantically.

2.2.1 Viewport Mashups with Viewplates. Masonview allows users to group multiple viewports under a single viewplate by dragging one viewport onto another. Users can also drag individual viewports within the viewplate to reorganize the layout or drag them out of the viewplate to detach them. Additionally, dragging the viewplate moves it and the viewports inside.

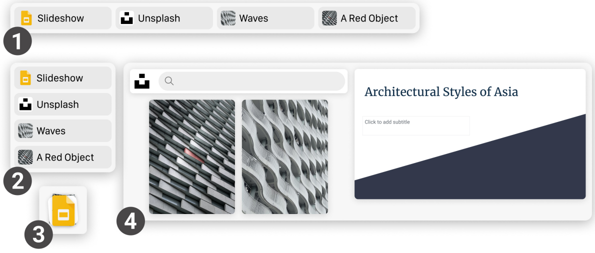


Figure 3: Viewplates dynamically adapt their viewport's representation based on their aspect ratio and size. Like normal windows, users can resize them by dragging the corner of the viewplate. Reducing the viewplate thin landscape-wise or portrait-wise turns its viewports into horizontal (1) or vertical (2) tabs, respectively. Reducing the viewplate to a small square converts them into a pile (3). Expanding the viewplate past its original size automatically lays them out as a grid (4).

2.2.2 Viewport Layout and Representation with Viewplates. Viewplates can further transform the representation of contained viewports with *Semantic Resize*. Semantic Resize leverages semantic zoom capabilities of zoomable user interfaces, where zooming does not scale content but rather adds or removes content to retain the readability of information [2]. To perform Semantic Resize, the user

[1] Command key on macOS and Control key on Microsoft Windows.

resizes the viewplate with direct manipulation by dragging the bottom right corner and releasing it to the desired representation.

Upon resizing the viewplate to certain aspect ratios and sizes, the viewplate changes its representation to improve comprehensibility (Fig. 3). Masonview accomplishes this by directly displaying the most significant information of the viewport’s focused element. For instance, to display tabbed views of images, Masonview uses the image itself as the icon and its alt text as the tab name (Fig. 3.1).

Since all viewplates are the same substrate regardless of their representation state, users can group multiple viewports of different representations under a single viewplate. For instance, dragging a pile into a list of tabs adds the piled viewports to the tab list.

Furthermore, different representations of viewplates offer familiar interactions that echo their intended functions. Double-clicking a pile opens the viewplate to show the original viewplate representation, similar to opening a desktop file, and clicking on a tab in a tabbed viewplate displays the viewport under or beside the viewplate.

3 CONCLUSION AND FUTURE WORK

We presented Masonview, a content-driven viewport management system that enables a more expressive layout of web components. With Masonview, users can create viewports for DOM elements of existing webpages and combine them into UI mashups using viewplates. Together, viewports and viewplates demonstrate unique interaction techniques that can transform how users interact with information on the web.

In future work, we plan to extend Masonview to support long-term archival of users’ information spaces and apply the concept of viewports and viewplates across the rest of the desktop (i.e., files and folders, desktop spaces, menubars). We also plan to formally evaluate Masonview in user studies to understand what behaviors can emerge from more flexible viewport management systems.

REFERENCES

[1] 2022. Arc from The Browser Company. https://arc.net/ Accessed: 2023-8-12.

[2] Benjamin B. Bederson and James D. Hollan. 1994. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology* (Marina del Rey, California, USA) *(UIST ’94)*. Association for Computing Machinery, New York, NY, USA, 17–26. https://doi.org/10.1145/192426.192435

[3] Clemens N. Klokmose, James R. Eagan, Siemen Baader, Wendy Mackay, and Michel Beaudouin-Lafon. 2015. Webstrates: Shareable Dynamic Media. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST ’15)*. Association for Computing Machinery, New York, NY, USA, 280–290. https://doi.org/10.1145/2807442.2807446

[4] Sunjae Lee, Hoyoung Kim, Sijung Kim, Sangwook Lee, Hyosu Kim, Jean Young Song, Steven Y. Ko, Sangeun Oh, and Insik Shin. 2022. A-Mash: Providing Single-App Illusion for Multi-App Use through User-Centric UI Mashup. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking* (Sydney, NSW, Australia) *(MobiCom ’22)*. Association for Computing Machinery, New York, NY, USA, 690–702. https://doi.org/10.1145/3495243.3560522

[5] Desney Tan, Brian Meyers, and Mary Czerwinski. 2004. WinCuts: Manipulating Arbitrary Window Regions for More Effective Use of Screen Space. In *Extended Abstracts of Proceedings of ACM Human Factors in Computing Systems CHI 2004* (extended abstracts of proceedings of acm human factors in computing systems chi 2004 ed.). 1525–1528. https://www.microsoft.com/en-us/research/publication/wincuts-manipulating-arbitrary-window-regions-for-more-effective-use-of-screen-space/

[6] Xiong Zhang and Philip J. Guo. 2018. Fusion: Opportunistic Web Prototyping with UI Mashups. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST ’18)*. Association for Computing Machinery, New York, NY, USA, 951–962. https://doi.org/10.1145/3242587.3242632