

A tool to help lay out Multiple View Visualisations guided by view analysis

Al-Maneea, Hayder Mahdi Abdullah; Roberts, Jonathan C.

Published: 25/05/2020

Publisher's PDF, also known as Version of record

[Cyswllt i'r cyhoeddiad / Link to publication](#)

Dyfyniad o'r fersiwn a gyhoeddwyd / Citation for published version (APA):
Al-Maneea, H. M. A., & Roberts, J. C. (2020). *A tool to help lay out Multiple View Visualisations guided by view analysis*. Poster session presented at Eurographics & Eurovis 2020, Norrköping, Sweden.

Hawliau Cyffredinol / General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A tool to help lay out Multiple View Visualisations guided by view analysis

H. M. Al-manee^{1,2}  and J. C. Roberts¹ 

¹Bangor University, United Kingdom

²University of Basrah, Iraq

Abstract

In this paper, we present ‘Layouts for Multiple View’ (LMV), a tool that helps users build, control and save multiple view visualisations simply and easily using a bespoke grammar. The tool incorporates template multiple view layout strategies as quantified from prior research on view analysis, and the user can build different layouts by defining the grammar, or through the linked visual interface. LMV saves the multiple view layout as a JSON file, including all the details of the layout and attributes of the visualisation, which can be subsequently loaded and adapted or used to create dashboards. LMV guides the user to (i) design and control the multiple view layout, (ii) add data and allocate a specific visualisation technique for each view, and (iii) to adapt specific appearance properties of the layout.

CCS Concepts

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

1. Introduction

Visualization tools, libraries and systems all help users create visualisations. While there are many ways to create a visualisation, controlling the layout of multiple view systems is still difficult. This is not the case for websites, where there are many design tools to help users lay out their websites. Moreover, Java programmers can use methods such as GridBagLayout to easily organise different components, or a BorderLayout layout to place nodes: top, bottom, left, right, and center. Why can we not have the same idea in visualisation? Visualisation developers would likewise benefit from a similar system. Templates to help users follow ‘typical’ layout strategies, and methods to graphically design different layouts. At the start of our research, we asked several questions: How can we develop a system to help visualisation users lay out their views? How can we allow users to quickly lay out their views, and then easily change their design? How do we map data to a view and easily change the appearance of the layout viewer? Our motivation is to give the users the ability to create juxtaposed view layouts in a simple and easy way. We also wanted to allow users to create ‘typical’ layouts, and consequently we drew on recent work by Al-manee and Roberts [AR18, AR19], on quantifying and identifying typical and frequent layout strategies.

We present LMV (Layouts for Multiple views), a tool to help users create and re-configure different multiple view visualisations. We designed the tool by starting with design sketches, and incrementally developed a final prototype in JavaScript and HTML. Users can create different multiple view layouts, associate their choice of visualisation libraries to a view, and change the appearance of the final multiple view layout. They can control this

through either a bespoke shorthand grammar or longhand JSON code, choose default layouts, design their own layout, and go back-and-forth between grammar and visual interface.

2. Background and Related work

Research in multiple-view visualisation is broad [Rob07]. For example, there are papers that focus on the theoretical aspects of multiple views, such as rules and principles for the use of multiple views [WBWK00], investigating juxtaposition, superposition and explicit designs [GAW*11, Gle18], the phraseology of multiple views [RAMB*19] and recently the structure of view layouts [AR18, AR19]. Other researchers have focused on developing coordinated multiple view systems, such as ComVis [MFGH08], Snap-together [NS00], Waltz [Rob98], Improvise [Wea04], Jigsaw [SGL08]. Or investigated techniques such as linked brushing [BC87] and rudiments of coordination [BRR03]. While other researchers have directed their attention to keep multiple views consistent [QH18], or work across large displays [LKD19].

There has been some research on multiple view layout. For instance, Spotfire and IVEE could re-group widgets [AS94] and later Spotfire tools allowed users to drag and drop views. Likewise Snap-Together [NS00] snapped views together to create custom combinations. And Improvise [Wea04] packs many visualisations in a tight space, while Keshif allows users to create visualisations in different parts in the Web page [YEB17]. Layout is also controlled in dashboard visualisations, and there are many ways to create visualisation dashboards, including D3.js [BOH11], highcharts (highcharts.com), and in tools such as Tableau, SAS or Power BI. Tools like datahero.com can be used to create dashboards using a drag-