Towards a Unified Visualization Platform for Ubiquitous Analytics

Sriram Karthik Badam
University of Maryland
College Park, MD, USA
sbadam@umd.edu

Roman Rädle
Aarhus University
Aarhus, Denmark
roman.raedle@cc.au.dk

Clemens N. Klokmose
Aarhus University
Aarhus, Denmark
clemens@klokmose.net

Niklas Elmqvist
University of Maryland
College Park, MD, USA
elm@umd.edu

MobileVis '18 Workshop at CHI 2018, April 21, 2018, Montreal, QC, Canada.
https://mobilevis.github.io/

Abstract
We propose the vision of a unified visualization platform that supports the full range of the classic "anytime" and "anywhere" motto for data analysis: from mobile and on-the-go usage, through office settings, to collaborative conference rooms and smart environments. Furthermore, such a platform, in order to be truly unified and universal, would have to scaffold a complete set of use-cases for visualization and analytics, including exploration, sensemaking, development, and, eventually presentation and dissemination. We discuss the design space of this unified platform and each of the required components in detail. Finally, we present several existing technologies and research efforts that may be leveraged to realize this vision.

Author Keywords
Ubiquitous analytics, ubilytics, cross-device visualization, mobile visualization, immersive analytics, situated analytics.

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction
Data analysis using visualization is performed through the use of purpose-made applications—just like most computer-based knowledge work. The choice of a specific applica-

tion for a specific task is based on a combination of personal preferences, the data and type of task, and the user's skills [21]. For visualization and data analytics, the choice stands between shelf configuration or template-based [11] applications (e.g., Tableau, Spotfire, and Microsoft PowerBI), high-level visualization specification grammars (e.g., Vega/-Lite [31, 32] and Atom [22]), or low-level visualization frameworks that require programming in a text editor (e.g., D3 [6] and Raphaël). For example, programming-savvy analysts may prefer using a low-level visualization framework, while a less technically-skilled analyst may prefer an application based on direct manipulation. When users with different skills or preferences need to collaborate, they, however, have to make a "collective compromise" and negotiate a common software denominator [21], as well as develop strategies for cross-application collaboration. These strategies require significant "mental gymnastics" and additional effort when copy&pasting content from one application to another or exporting and importing data back and forth.

Collaboration for visualization [13] is just one of several cross-cutting concerns that unify all realistic uses of visualization. Another is the *activities* that users perform using visualization and visual analytics: programming, tinkering and crafting, exploring, analyzing, and presenting visualizations. Yet, most modern visualization software support—at most—collaboration only in one of these activities. Finally, the choice of tool or application also dictates the *devices* applicable for a given task: e.g., desktop, mobile, pen-based, tangible, and pen-based ones. Analytics applications developed for desktop computers are impossible to use on mobile platforms, limiting migration between mobile and stationary devices—what Jokela et al. [15] call *sequential use*. Again, the parallel use of devices [15] involving multiple heterogeneous devices (e.g., personal devices and large displays) as well as potentially multiple users is generally poorly supported by contemporary software.

In an ideal world all these three cross-cutting aspects of data analysis using visualization—(i) collaboration, (ii) diverse activities, and (iii) heterogeneous devices—would be supported by the same visualization and visual analytics platform, and transitions between them would be both transparent and effortless. Embodied by Mark Weiser's classic "anytime" and "anywhere" moniker for ubiquitous computing [35], we think of this as an *ubiquitous* paradigm for analytics, or *ubilytics* [4, 9]. To achieve this paradigm, the implication is that the visual analysis environment needs to accommodate both single and multiple users, a diverse and heterogeneous set of input and output devices, and the full spectrum of activities, from exploration and analysis to dissemination and presentation [33]. This includes both the use of off-the-shelf visual and analytics components, which is most practical in a mobile setting where interaction is limited, as well as the ability to modify or even develop new visualizations from scratch within the same system.

We propose the notion of a *unified visualization platform* that supports both a complete range of physical usage settings as well as a complete set of practical use cases where people may want to perform cooperative data analysis without sacrificing their skills due to a collective compromise. This platform is based on a design space that incorporates factors such as collaboration, literate computing, input and output devices, and component models, as well as activities such as design, development, analysis, and presentation. The intention of this unified visualization platform is to provide an exemplar of what an ideal platform would look like, essentially outlining a vision for future research in this area. We take the first steps on this path by identifying a host of current tools and technologies that we believe could contribute to realizing this grand vision.

Motivating Scenario

Ellis is a data scientist working in the real estate investment branch of a large North American coffeehouse chain. His team, the store location team, decides where to buy property and open new stores. This involves analyzing data on competition, demography, safety, real estate prices, etc. of a given city or neighborhood. Ellis and his colleagues develop interactive visualizations where the interdisciplinary store location team can experiment with “what if” scenarios. The team comprises data scientists as well as financial analysts and economists. Team members collaborate synchronously in meetings and asynchronously in-between them. Some team members are located in different time zones across the country, which is why—among co-located meetings—the team regularly collaborates remotely. The co-located team will interact with visualization components developed by the data scientists across both personal devices and large shared displays. A typical scenario of use will revolve around a large display with a map with various visualizations of potentials for sales and growth—visualizations that can be configured live from the personal devices of the participants. At any time a visualization component can be reconfigured or even reprogrammed if the need arises. At their desks, analysts can combine datasets and visualization components in a direct manipulation fashion. The visualization components Ellis and his colleagues have developed are accessible on mobile devices as well. This enables field scouting, where an analyst can quickly get an overview of the suitability of a neighborhood or particular address, and explore “what if” scenarios directly on the phone in the field.

Background

Supporting data analytics anywhere, anytime [9] can help realize workflows described in the motivating scenario. However, this is not exactly straightforward. In fact, going beyond the traditional analytical activities on a personal computer into large display environments that support collaboration as well as mobile visualization is ongoing. Among recent work, VisPorter [7] enables visual exploration between large displays and smartphones in a co-located space. BodyLenses [17] and Proxemic lenses [1] explore full-body interaction models for users working in front of a wall-sized display. GraSp [16] enhances such interactions with a handheld device allowing graph analysts explore the data with more flexible workflows in the large display environment. Finally, spatial analytic interfaces [10] introduce the notion of accessing visualizations in any target environment with the help of augmented reality. These advances pave the way for our Unified Visualization Platform as it needs to bring together the interaction models and visualization tasks tackled in these background works.

Early collaborative visualization platforms (e.g., Sense.us [12] and ManyEyes [34]) exemplified asynchronous distributed collaboration in visual analysis. Munin [4] presents a peer-to-peer platform for interaction management and user interface distribution across devices in a co-located environment. Going beyond that, PolyChrome [2] supports cross-device visualizations over the web to utilize devices of different modalities together or transform visual interfaces to work on any device. Panelrama [36] supports distribution of webpages across devices, thus, enabling ubiquity for general web applications. However, these frameworks still showcase early steps towards reaching our vision as they rely on the end-user application developers to implement data models, create visualization techniques, make them responsive to devices, and even support collaboration.

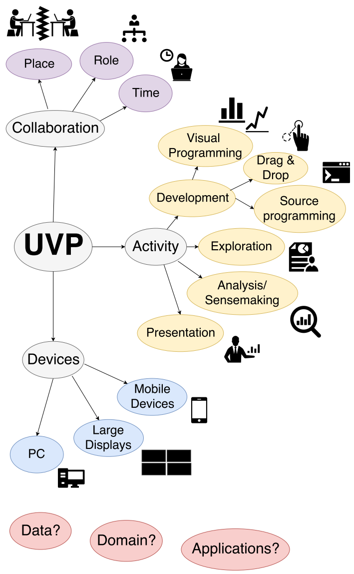

Figure 1: The conceptual model for a unified visualization platform. We currently focus on three core aspects—activity, collaboration, and devices/platforms—to outline the design space. There are also some secondary aspects such as data, application, and domain that can influence the envisioned visualization platform.

A Unified Visualization Platform

In the motivating scenario, we see analysts performing data analysis while collaborating with others to develop their insights and working across devices. We, therefore, identify three core aspects of a unified visualization platform (UVP): *collaboration*, *activities*, and *devices*. First, a UVP should support the different types of *collaboration* and seamless transitions between them: remote vs. co-located and asynchronous vs. synchronous collaboration [8]. Second, a UVP should support the various *activities* involved in visualization work and the seamless transitions between them: development, exploration, analysis and sensemaking, and presentation [25]. And finally, a UVP should allow users to span and migrate their work across heterogeneous devices.

Activities

The activities represent the use context, and thus convey the support expected from the platform.

- *Exploration:* investigative data analysis where the hypotheses are not initially known.
- *Sensemaking:* collecting actionable insights.
- *Development:* creating and modifying visualizations and analytical components; ranges from drag-and-drop or visual programming to actual programming.
- *Presentation and dissemination:* communicating the results from an analysis [33].

Activities exist on a spectrum. On one end, development and exploration involve creation and interaction with visualizations to identify patterns, trends, and outliers. On the other, sensemaking supports much more including hypotheses testing and management of insights developed, while presentation brings them to the target audience. On this scale, other activities also exist such as learning and education. Ideally, new visualizations that emerge from development during learning and education seamlessly transition into the tool portfolio of an analyst such that their tools become a tangible manifestation of their expertise. These custom tools also allow analysts to develop their own analyses strategies without having them dictated by a system with a pre-defined toolset. Therefore, different users can have their own custom tools and strategies for data analyses.

Collaboration

The activities are not performed by just a single user. Different types of collaboration require specific features.

- *Time:* whether the collaboration happens at the same (synchronous) or different (asynchronous) times.
- *Space:* whether the collaboration happens in the same (co-located) or different (distributed) space.
- *Role:* the unique capabilities, expertise, and permissions of participating collaborators.

When collaboration happens asynchronously in distributed settings—the most common form of collaboration in visual analysis [12, 13]—annotations and data coverage representations [5, 28] help the analytical activity. In contrast, synchronous collaborations require real-time communication and deixis [14] to coordinate multiple users.

Devices and Platform

The target platforms and their input and output modalities play an important role in the analytical activity.

- *Mobile devices:* small, handheld devices such as smartwatches, mobile phones, and tablets.
- *Personal computers:* laptop or desktop computers, equipped with a mouse and keyboard, or touch input.
- *Large displays:* large screens such as wall-mounted, wall-size, or tabletop displays.
- *Input sensing:* depth cameras, full-body sensing, motion capture, etc.

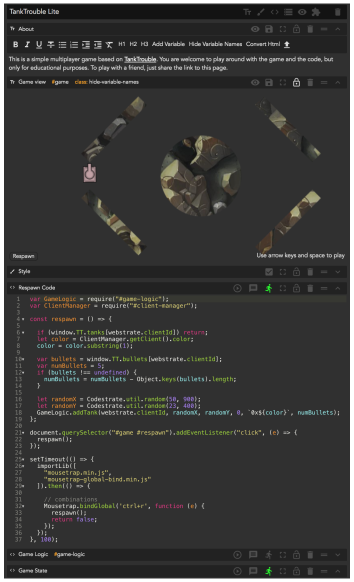

Figure 2: An example codestrate [26] for development of a web game, where the code is organized in blocks enriched with rich text and media (literate programming). The Codestrates system [26] supports real-time collaboration in creating web applications, enhances reprogrammability, and bridges the gap between development and use of web applications.

The differences with output modalities, display sizes of small handheld devices vs. large displays, calls for responsiveness in the user interface elements. Furthermore, the UI should also adapt to the input modalities of the target devices to seamlessly support analytical activities across devices as seen in our motivating scenario.

Inspirational Technologies

Frameworks such as D3 [6], Vega [32], and Vega-Lite [31] have gained popularity for developing visualizations. They offer flexible ways of expressing interactive visualizations for a programming-savvy user that can be deployed as a webpage to make it accessible on all devices that have a web browser. Shelf configuration [11] applications such as Tableau[1] or Polestar[2] allow for drag-and-drop based creation of visualizations without the need for programming. Lyra [30] builds on D3 and allows users to create custom visualizations using drag-and-drop. For storytelling, Ellipsis [29] takes D3 code and unravels parameters to allow users create scenes and tell a visual story.

Interactive notebooks such as Jupyter Notebook[3] [24], Observable Notebook[4], or Codestrates [26] (Figure 2) bridge the gap between development and use of software. These applications leverage the literate computing paradigm for reprogrammable interactive narratives [19]. In these notebooks, text and media such as images, audio, and video can be interleaved with executable code in a modular fashion, to support re-usability in data analytics. Observable is specifically designed to be used with visualization frameworks such as D3, and allows users to tinker with code to visualize data, and share the visualizations with others.

[1] https://www.tableau.com/
[2] http://vega.github.io/polestar/
[3] http://jupyter.org
[4] http://observablehq.com

Real-time and concurrent text editing has been a vital aspect for synchronous writing, and it has been widely adopted by a large user base (e.g., Google Docs). Only recently, this type of collaboration found its way into other types of “making together” such as programming (e.g., Codestrates [26]), data analysis (e.g., Colaboratory[5]), or user interface prototyping (e.g., Figma[6]). A different aspect of collaboration is “exploring together” where users work on the same artifact in a WYSIWIS[7] style. Visualization tools like Sense.us [12] or ManyEyes [34] showcase ”exploring together” through view sharing, annotation, and social navigation [27].

Cross-device interaction has grown to be a popular research area in human-computer interaction. VisPorter [7] is a visual analytics system supporting co-located group activity through large shared displays and handheld personal devices, and cross-device interactions for sharing data across them. VisTiles [20] (Figure 3) is a conceptual framework that proposes multi-device interactions to coordinate visualization views across small-screen devices (e.g., smartphones and tablets) for visual data exploration. Panelrama [36] demonstrates a component based approach to distributing user interface elements across multiple heterogeneous devices using integer programming. AdaM [23] solves the problem of allocating components to devices with combinatorial optimization and accounts for device capabilities, user roles, preferences, and access rights. Visfer [3] connects devices merely using QR-codes captured by a handheld device from a large display.

Finally, Munin [4] and PolyChrome [2] (Fig. 4) provide shared event and device management for ubilytics. They support application development for many collaboration styles [8].

[5] http://colab.research.google.com
[6] https://www.figma.com/
[7] What You See Is What I See

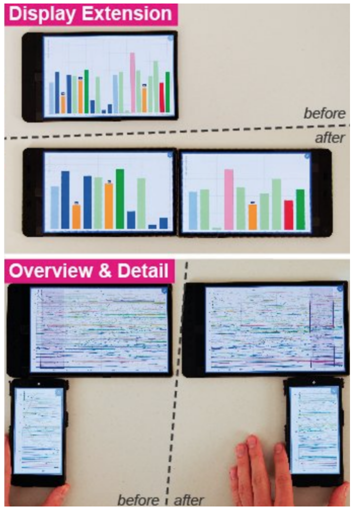

Figure 3: Two example configurations for combining small-screen devices for visual exploration from VisTiles [20].

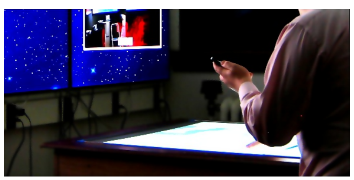

Figure 4: Current Ubilytics frameworks such as Munin [4] and PolyChrome [2] provide the infrastructure for managing input events and display space across devices in a visual analytics environment. However, as middleware they rely on end-user application developers to actually create the visualization interface and define workflows across the devices for multiple users.

Agenda for Ubilytics Research

A truly universal platform for ubiquitous analytics cannot be built by a single vendor, team, or researcher. In fact, the definition of a universal platform is that it should support integration across any usage, user, or setting. The scope of the existing research and industry projects we draw upon to realize the motivating scenarios in the beginning of the paper should be evidence enough that multiple efforts are required across both academics and practitioners.

For this reason, we believe that an important endeavor in the quest for a universal visualization platform should be the development of a *component model for visualization*. In software engineering, a *component* is a module, service, or resource that encapsulates a set of related functions and state. Only by formalizing the separation of concerns between different parts of the visualization platform using a well-defined software interface would it be possible to realize massive software engineering undertakings such as the universal visualization platform proposed in this paper.

We believe that the literate computing paradigm used in Jupyter Notebook, Observable Notebook and Codestrates has a yet untapped potential for supporting our agenda. Codestrates has demonstrated how literate computing can scale from data exploration to application development, and through the underlying Webstrates platform support collaboration both in development and use of software and applications. The web platform used by Codestrates and Observable, allows for unprecedented combination and remixing of existing tools and frameworks including the most popular visualization frameworks. Combining a component based model with a literate computing paradigm would allow novice users to remix components, and expert users to develop or reprogram components in the same environment and collaboratively. We imagine bite-sized components that can easily be distributed to mobile devices, or shown in combination on a large display, e.g. automated in a fashion as proposed with AdAM [23].

Conclusion and Future Work

We have proposed a *unified visualization platform* (UVP) to support the full range of activities, devices and settings, and modes of collaboration necessary for seamless interactive data analysis. The goal is to define a grand vision that can serve as an exemplar for future work. Furthermore, we have also identified inspirational technologies that partially support this vision, as well as a future research agenda.

The MobileVis workshop will allow us to present the vision of a unified visualization platform to researchers and practitioners, discuss challenges and benefits, and brainstorm solutions for realizing it.

In our future work, we will begin to actually make inroads towards realizing this vision. While we have so far in this work identified possible technologies and research efforts that can be leveraged to do this, our goal is to actually start implementing some of these ideas in the near future. Our plan is to primarily use the Webstrates [18] and Codestrates [26] platforms as starting points, working towards a system that we are tentatively calling “Vistrates.”

Acknowledgments

This work was partially supported by the Aarhus University Research Foundation and the U.S. National Science Foundation award IIS-1539534. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. We also thank the Noun Project for the icons used in Figure 1.

REFERENCES

1. Sriram Karthik Badam, Fereshteh Amini, Niklas Elmqvist, and Pourang Irani. 2016. Supporting Visual Exploration for Multiple Users in Large Display Environments. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*. IEEE, 1–10. DOI: http://dx.doi.org/10.1109/vast.2016.7883506

2. Sriram Karthik Badam and Niklas Elmqvist. 2014. PolyChrome: A Cross-Device Framework for Collaborative Web Visualization. In *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. 109–118. DOI: http://dx.doi.org/0.1145/2669485.2669518

3. Sriram Karthik Badam and Niklas Elmqvist. 2017. Visfer: Camera-based visual data transfer for cross-device visualization. *Information Visualization* (2017). DOI: http://dx.doi.org/10.1177/1473871617725907

4. Sriram Karthik Badam, Eli Fisher, and Niklas Elmqvist. 2015. Munin: A Peer-to-Peer Middleware for Ubiquitous Analytics and Visualization Spaces. *IEEE Transactions on Visualization and Computer Graphics* 21, 2 (Feb 2015), 215–228. DOI: http://dx.doi.org/10.1109/TVCG.2014.2337337

5. Sriram Karthik Badam, Zehua Zeng, Emily Wall, Alex Endert, and Niklas Elmqvist. 2017. Supporting Team-First Visual Analytics Through Group Activity Representations. In *Proceedings of the Graphics Interface Conference*. Canadian Human-Computer Communications Society, 208–213. DOI: http://dx.doi.org/10.20380/GI2017.26

6. Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2301–2309. DOI: http://dx.doi.org/10.1109/TVCG.2011.185

7. Haeyong Chung, Chris North, Jessica Zeitz Self, Sharon Lynn Chu, and Francis K. H. Quek. 2014. VisPorter: facilitating information sharing for collaborative sensemaking on multiple displays. *Personal and Ubiquitous Computing* 18, 5 (2014), 1169–1186. DOI: http://dx.doi.org/10.1007/s00779-013-0727-2

8. Clarence A. Ellis, Simon J. Gibbs, and Gail Rein. 1991. Groupware: some issues and experiences. *Commun. ACM* 34, 1 (1991), 39–58. DOI: http://dx.doi.org/10.1145/99977.99987

9. Niklas Elmqvist and Pourang Irani. 2013. Ubiquitous Analytics: Interacting with Big Data Anywhere, Anytime. *IEEE Computer* 46, 4 (2013), 86–89. DOI: http://dx.doi.org/10.1109/mc.2013.147

10. Barrett Ens and Pourang Irani. 2017. Spatial Analytic Interfaces: Spatial User Interfaces for In Situ Visual Analytics. *IEEE Computer Graphics & Applications* 37, 2 (2017), 66–79. DOI: http://dx.doi.org/10.1109/MCG.2016.38

11. Lars Grammel, Chris Bennett, Melanie Tory, and Margaret-Anne Storey. 2013. A Survey of Visualization Construction User Interfaces. In *Proceedings of EuroVis - Short Papers*. The Eurographics Association. DOI:http://dx.doi.org/10.2312/PE.EuroVisShort.EuroVisShort2013.019-023

12. Jeffrey Heer. 2009. The Design of Sense.us. In *Beautiful Data: The Stories Behind Elegant Data Solutions*, Toby Segaran and Jeff Hammerbacher (Eds.). O'Reilly Media, Inc., 183–204.

13. Petra Isenberg, Niklas Elmqvist, Jean Scholtz, Daniel Cernea, Kwan-Liu Ma, and Hans Hagen. 2011. Collaborative visualization: Definition, challenges, and research agenda. *Information Visualization* 10, 4 (2011), 310–326. DOI: http://dx.doi.org/10.1177/1473871611412817

14. Petra Isenberg and Danyel Fisher. 2009. Collaborative Brushing and Linking for Co-located Visual Analytics of Document Collections. *Computer Graphics Forum* 28, 3 (2009), 1031–1038. DOI: http://dx.doi.org/10.1111/j.1467-8659.2009.01444.x

15. Tero Jokela, Jarno Ojala, and Thomas Olsson. 2015. A Diary Study on Combining Multiple Information Devices in Everyday Activities and Tasks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 3903–3912. DOI: http://dx.doi.org/10.1145/2702123.2702211

16. Ulrike Kister, Konstantin Klamka, Christian Tominski, and Raimund Dachselt. 2017. GraSp: Combining Spatially-aware Mobile Devices and a Display Wall for Graph Visualization and Interaction. *Computer Graphics Forum* 36, 3 (2017), 503–514. DOI: http://dx.doi.org/10.1111/cgf.13206

17. Ulrike Kister, Patrick Reipschläger, Fabrice Matulic, and Raimund Dachselt. 2015. BodyLenses: Embodied Magic Lenses and Personal Territories for Wall Displays. In *Proceedings of the ACM Conference on Interactive Tabletops & Surfaces*. ACM, 117–126. DOI: http://dx.doi.org/10.1145/2817721.2817726

18. Clemens N. Klokmose, James R. Eagan, Siemen Baader, Wendy Mackay, and Michel Beaudouin-Lafon. 2015. Webstrates: Shareable Dynamic Media. In *Proceedings of the ACM Symposium on User Interface Software & Technology*. ACM, 280–290. DOI: http://dx.doi.org/10.1145/2807442.2807446

19. D. E. Knuth. 1984. Literate Programming. *Comput. J.* 27, 2 (1984), 97–111. DOI: http://dx.doi.org/10.1093/comjnl/27.2.97

20. Ricardo Langner, Tom Horak, and Raimund Dachselt. 2018. VisTiles: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 626–636. DOI: http://dx.doi.org/10.1109/tvcg.2017.2744019

21. Midas Nouwens and Clemens N. Klokmose. 2018. The Application and Its Consequences for Non-Standard Knowledge Work. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/3173574.3173973 To appear.

22. Deok Gun Park, Steven M. Drucker, Roland Fernandez, and Niklas Elmqvist. 2018a. ATOM: A Grammar for Unit Visualization. *IEEE Transactions on Visualization & Computer Graphics* (2018). DOI: http://dx.doi.org/10.1109/TVCG.2017.2785807 To appear.

23. Seonwook Park, Christoph Gebhardt, Roman Rädle, Anna Maria Feit, Hana Vrzakova, Niraj Ramesh Dayama, Hui-Shyong Yeo, Clemens Nylandsted Klokmose, Aaron Quigley, Antti Oulasvirta, and Otmar Hilliges. 2018b. AdaM: Adapting Multi-User Interfaces for Collaborative Environments in Real-Time. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM. DOI: http://dx.doi.org/10.1145/3173574.3173758 To appear.

24. Fernando Pérez and Brian E Granger. 2007. IPython: a System for Interactive Scientific Computing. *Computing in Science & Engineering* 9, 3 (2007), 21–29. DOI: http://dx.doi.org/10.1109/MCSE.2007.53

25. Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of International Conference on Intelligence Analysis*, Vol. 5. 2–4.

26. Roman Rädle, Midas Nouwens, Kristian Antonsen, James R. Eagan, and Clemens N. Klokmose. 2017. Codestrates: Literate Computing with Webstrates. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM, 715–725. DOI: http://dx.doi.org/10.1145/3126594.3126642

27. Mark O. Riedl. 2001. A Computational Model and Classification Framework for Social Navigation. In *Proceedings of the ACM Conference on Intelligent User Interfaces*. 137–144. DOI: http://dx.doi.org/10.1145/359784.360320

28. Ali Sarvghad and Melanie Tory. 2015. Exploiting analysis history to support collaborative data analysis. In *Proceedings of the Graphics Interface Conference*. 123–130. DOI: http://dx.doi.org/10.20380/GI2015.16

29. Arvind Satyanarayan and Jeffrey Heer. 2014a. Authoring narrative visualizations with Ellipsis. 33, 3 (2014), 361–370. DOI: http://dx.doi.org/10.1111/cgf.12392

30. Arvind Satyanarayan and Jeffrey Heer. 2014b. Lyra: An interactive visualization design environment. 33, 3 (2014), 351–360. DOI: http://dx.doi.org/10.1111/cgf.12391

31. Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 341–350. DOI: http://dx.doi.org/10.1109/TVCG.2016.2599030

32. Arvind Satyanarayan, Kanit Wongsuphasawat, and Jeffrey Heer. 2014. Declarative interaction design for data visualization. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM, 669–678. DOI: http://dx.doi.org/10.1145/2642918.2647360

33. James J. Thomas and Kristin A. Cook. 2005. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press.

34. Fernanda B Viégas, Martin Wattenberg, Frank Van Ham, Jesse Kriss, and Matt McKeon. 2007. ManyEyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1121–1128. DOI: http://dx.doi.org/10.1109/TVCG.2007.70577

35. Mark Weiser. 1991. The computer for the 21st Century. *Scientific American* 265, 3 (1991), 94–104. DOI: http://dx.doi.org/10.1145/329124.329126

36. Jishuo Yang and Daniel Wigdor. 2014. Panelrama: enabling easy specification of cross-device web applications. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, 2783–2792. DOI: http://dx.doi.org/10.1145/2556288.2557199