Vis, The Next Generation: Teaching Across the Researcher-Practitioner Gap (IEEE VIS Panel)

Marti A. Hearst, Eytan Adar, Robert Kosara, Tamara Munzner, Jon Schwabish, Ben Shneiderman

1 Introduction

Information visualization has escaped the research lab and is now heavily used by practitioners across a wide spectrum of fields. New software tools and programming frameworks appear on a monthly basis. New design paradigms are rapidly gaining acceptance and evolving.

At the same time, methods for teaching in the classroom and beyond are being challenged and influenced by online offerings such as Khan Academy and Massive Open Online Courses (MOOCs), by the adoption of flipped classrooms, and by the adaptation of instructional environments used in other communities. Pedagogy geared towards mastery learning that makes use of active learning and peer learning are being introduced in more contexts, reflecting the results of decades of research showing the benefits of these techniques, as well as their suitability for today's connected students who expect a more interactive learning experience.

As the role of information visualization grows and changes in the world of practice, new methods are needed to teach this dynamic topic. This panel brings together experts with different perspectives to talk about how they are rising to the challenge of teaching information visualization in this new world. Panelists are asked to address three specific themes:

- Visualization in practice versus visualization in research;
- Active versus passive learning; and
- How to address the rapid rise of new tools and frameworks.

Panelists are encouraged to tackle these specific themes — and others that they find pressing — to answer how they incorporate these issues into modern visualization courses.

The invited panelists are instructors who teach across the spectrum from purely research-oriented courses to more applied courses, and with a wide range of styles. Two panelists teach in Computer Science departments, two teach in interdisciplinary Schools of Information, and one is a senior research associate in a non-profit policy center. The moderator is a former academic and practitioner thought-leader who now guides innovation at a leading software developer of visualization tools.

Below are shown the position statements and brief biographies for each of the invited, confirmed panelists. This panel builds on prior surveys of teaching practices [10]. A panel on teaching did appear at VisWeek 2010, but that was one year before the first MOOC appeared [12] and helped launch this disruption in higher education; furthermore, since then practitioners have begun using visualizations much more widely.

- *Professor, UC Berkeley; hearst@berkeley.edu*
- *Associate Professor, U Michigan; eadar@umich.edu*
- *Research Scientist, Tableau Software; rkosara@tableau.com*
- *Professor, U British Columbia; tmm@cs.ubc.ca*
- *Professor, U Maryland; ben@cs.umd.edu*
- *Senior Researcher, The Urban Institute, and author of policyviz.com; jschwabish@gmail.com*

2 The Invited Panelists

Marti Hearst (co-organizer) Inspired by Eric Mazur's writings about how he transformed instruction in physics courses with active and peer learning [5], and by the technology for active learning used in MOOCs, for the past two years I have created my own blend of these ideas and introduced them into my two standard masters-level courses. After reading the pedagogy of active learning and what is sometimes called cooperative learning and other times called structured peer learning [9, 13, 17, 16, 11, 4, 18], my goal was to craft new ideas for introducing these ideas into computer science courses.

Before each class session, students do short readings and a small exercise to prepare for that day's class. This exercise can be a quiz, a programming practice exercise, a peer review, or some other short task. Almost every exercise requires some kind of feedback, so designing these requires care. During class, students are expected to be active as well; most class sessions consists of short lecture segments alternating with activities which have the students think about questions in pairs or do short exercises. Within each class session we refer back that day's preparatory exercise. Anonymous surveys find that students prefer the active and peer learning activities over traditional lecture formats.

The course has four conceptual modules, and each module includes learning a different software tool. We study core principles of graphing and visualization (Highcharts javascript charting tool), narrative infographics (Adobe Illustrator), exploratory data analysis (Tableau), and interaction and animation (d3.js). A final project allows for the integration of all four of these skills; an example of this work appears as a poster in this conference [2].

One example illustrates the idea of integrating active and peer learning in the manner I refer to here. I have taught a course on Information Visualization and Presentation since 1998, but I have never been fully satisfied with students' grasping of the basics of chart design. I think in part this is because students need practice in order to achieve mastery, and in part because they need objective feedback.

Therefore, I developed an exercise I call the *Objectively Evaluated Visualization Assignment*: Inspired in part by work by Dow et al. [7], students first sketch multiple designs individually, then are paired randomly and create one final design together using a software tool. The design must be able to answer a wide range of question types.

Then, people from the target audience use the visualizations to *answer questions* about the underlying data. With the permission of the students, I post their designs on a crowdsourcing platform, along with a set of multiple choice questions, and then we as a class scrutinize what happens.

As a side effect of the learning activity, we have found new design recommendations that are usability tested. Another consequence is the students get real-world feedback about what does and does not work. I also give feedback on those designs that do not do well, but in future I hope to have peers do this.

Bio: Prof. Marti Hearst has taught Information Visualization since 1998 at the School of Information at UC Berkeley. She learned Infoviz while a researcher at Xerox PARC in the early 1990's and her research focus in this field is on text visualization. She published *Search User Interfaces* in 2009 (Cambridge University Press) and received student-initiated Excellence in Teaching Awards in 2014 and 2015.

Eytan Adar (co-organizer) Over the past five years (8 semesters) I have taught our graduate information visualization class. While the course is in the School of Information, it is the only infoviz offering on campus and attracts students from CS, Art and Design, Public Health, and other departments. Teaching to this diverse crowd has led to a number of innovations around engagement.

A key feature of the class is that students are challenged to use what they learn immediately in the context of real problems. The class has evolved to become completely flipped with a video lectures in advance of class meetings. High value quizzes at the start of class ensure that students keep up.

Each 3-hour class sessions is in two parts. The first is an active [5] lecture/group session (students are randomly assigned to a new 5-6 person group every class). The "lecture" sets of 3-5 slides expanding on the video lectures or providing new case studies followed by 5-10 minute group discussion on a 'group question' (generally analysis and critique questions). After each breakout we come back for a full-class discussions. The outcome is that students are much more engaged and consequently appreciate the class more and produce far better designs.

The second half of class involves a longer design exercise (again in groups), based on 'real problems' with published solutions that the students are unaware of. The exercise relates to the class topic of the day. For example, the hierarchical visualization lecture is followed by the design of a tennis vis system and after learning about networks students design a solution exploring papers/people/topics in a scholarly database. The notion of visualization as a wicked design problem [3] is heavily emphasized in class so students learn to pick a subset of tasks to address. At the end of the session the best solution from each group is presented to the class.

After this session students are given the 'published' solution (e.g., TennisViewer [8] and PivotPaths [6]). Their assignment for the *next session* includes making a five-minute presentation comparing the published solution to their own (one student is selected at random to present but feedback is provided to all and we have a discussion after the presentation). This structure forces students to define the domain/abstract questions before experiencing the published solution and forcing them to question both their own approaches and the published version more deeply.

Together with more standard exercises and labs, the format has proven quite popular with the students as the class has evolved we have seen students producing very creative and functional systems.

Bio: Eytan Adar is an Associate Professor of Information and Computer Science at the University of Michigan. He has taught information visualization since 2009. He has published extensively in Infoviz, HCI and information retrieval and data mining and has created the GUESS system [1], a tool for graph visualization which is used extensively in classrooms both at Michigan and outside to teach information visualization, graph theory, and even basic programming.

Robert Kosara (moderator) While I no longer teach, I created and taught a course for several years that I called *Visualization and Visual Communication*. The goal was to bring together students not just from computer science, but also art, design, film, communication, etc. The material presented included an introduction to visualization, plus lectures about photography, design, and some basic art theory. Projects included some drawing exercises, with the final project being the design of a visualization for a given data set.

The results were highly variable, but some were quite amazing. The students came up with some very original ideas for showing data, way beyond the usual visualization techniques.

A recent paper on data sketching by Walny et al. [19], discusses a very similar approach[1]. In addition to describing the basic data sketching experiment, the paper contains an analysis of different approaches that participants took, and the kinds of results they got.

I cannot claim to have a recipe for getting great results, but I believe that encouraging more drawing and sketching, as well as making connections across different disciplines, can help stimulate more creative and novel thinking about data visualization.

Bio: Robert Kosara is a research scientist at Tableau Software. His focus is on the communication of data through visualization and visual storytelling. Robert is also working on furthering our understanding of visual perception and cognition, so we can make data easier to understand and develop tools to communicate it more effectively.

Before joining Tableau in 2012, Robert was Associate Professor of Computer Science at The University of North Carolina at Charlotte. Robert received his M.Sc. and Ph.D. Degrees in Computer Science from Vienna University of Technology (Austria). In his copious spare time, Robert likes to run long distances and writing articles for his website, `https://eagereyes.org`.

Tamara Munzner My foremost goal in my infovis courses is to encourage students to think systematically and critically about the entire possible design space for visualization, with the rationale that learning how to analyze existing systems will help them design new ones. My recent textbook [14] breaks down visualization design according to three questions: what data users need to see, why users need to carry out their tasks, and how the visual representations proposed can be constructed and manipulated.

My own courses fall on the academic end of the spectrum, particularly for the 14-week graduate version at UBC where I combine book chapters with reading original academic papers, in service of teaching students the conventions of paper structure in this field. The class is built around students reading before and submitting questions or comments about the reading beforehand, with most of class time spent in discussion rather than lecture. The students also do a large-scale final project, with a range of possibilities. Design study programming projects are by far the most popular choice, and students often bring their own data and tasks to the table. To make the course accessible to students without a computer science background, analysis projects using existing tools are also possible, as are surveys. Nevertheless, the course is very much designed to bring students into the realm of infovis research. While I am happy with the base structure, I am curious whether and how I can inject more in-class design exercises to add more practice with analysis before the monolithic final project.

The full-day or half-day mini-course version at conferences and workshops a lecture-based presentation of the content covered in the book on the space of possible designs and validation approaches. While there's no direct reading of original academic papers, there are pointers to many of them as further reading. I am curious whether adding hands-on exercises would strengthen the course and make it more useful for practitioners, or weaken it by decreasing the scope of what I cover.

Bio: Tamara Munzner is a professor at the University of British Columbia Department of Computer Science, and holds a PhD from Stanford. She has been active in visualization research since 1991 and has published over sixty papers and book chapters. She co-chaired InfoVis in 2003 and 2004, co-chaired EuroVis in 2009 and 2010, and is chair of the VIS Executive Committee. She has worked on problem-driven visualization in a broad range of application domains, including genomics, evolutionary biology, geometric topology, computational linguistics, large-scale system administration, web log analysis, and journalism. Her technique-driven interests include graph drawing and dimensionality reduction. Her evaluation interests include both controlled experiments in a laboratory setting and qualitative studies in the field.

Ben Shneiderman My ideas about High-Impact Research strategies have been emerging, tested, and refined during two decades of teaching. The strategies drew on earlier education technology ideas in the *Relate-Create-Donate* concept: students working in teams to produce something ambitious for someone to use outside the classroom, which would survive beyond the semester. This concept became my guiding principle in every course I taught, but it became especially effective in my graduate computer science course on Information Visualization with 30 students.

[1] `http://innovis.cpsc.ucalgary.ca/supplemental/Data-Sketching/`

The University of Maryland's 15-week semester gives me enough time to include smaller team projects early on, so that students develop the communication skills necessary for working in teams of 4 or 5 students over a 11-12 week project. There are individual homeworks and midterm plus final exams, so I have good evidence to combine with team project scores to provide a final grade for each individual. I learned that random assignment of students to teams produced greater diversity, and simulated the professional world in which employees usually are assigned to teams.

Computer science students don't automatically seek to work in teams, but I think they are learning valuable professional skills in a relatively safe environment. My best measure of success is that many of the team project reports go on to be published papers in conferences and journals. I used to say that one in ten student projects could lead to a publishable result, but these teamwork methods (external client/mentors, multiple deliverables, open posting, internal reviews, etc.) and the support technologies (class wiki, web access to previous work, discussion board, shared code repositories, etc.) have enabled a much higher rate. My best semester produced 5 published papers out of 7 teams in respected conferences and journals. Students have a strong portfolio item to show future employers and if their work is published they can proudly add it to their resumes. Students may build on their work in a Masters or Doctoral dissertation, or get hired by colleagues and companies. Not every student likes the pressure of working in teams on authentic projects, but Im encouraged by the students who go on to do excellent work and check back with me many years later.

Bio: Ben Shneiderman is a Distinguished University Professor in the Department of Computer Science, Founding Director (1983-2000) of the Human-Computer Interaction Laboratory and a Member of the UM Institute for Advanced Computer Studies (UMIACS) at the University of Maryland. He is a Fellow of the AAAS, ACM, and IEEE, and a Member of the National Academy of Engineering. His contributions include the direct manipulation concept, clickable highlighted web-links, touchscreen keyboards, dynamic query sliders for Spotfire, development of treemaps, novel network visualizations for NodeXL, and temporal event sequence analysis for electronic health records.

Ben is the co-author with Catherine Plaisant of Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th ed., 2010). With Stu Card and Jock Mackinlay, he co-authored Readings in Information Visualization: Using Vision to Think (1999). His book Leonardos Laptop appeared in October 2002 (MIT Press) and won the IEEE book award for Distinguished Literary Contribution. His latest book, with Derek Hansen and Marc Smith, is Analyzing Social Media Networks with NodeXL, 2010.

Jonathan Schwabish The data visualization field combines data analysis, graphic design, storytelling, and statistics in such ways that producers of content can help their audience better understand and gain insight into issues of interest. To create effective visualizations, I argue that people need (at least) three primary skillsets:

1. Statistics: some understanding of how to use and analyze data, grapple with potential biases, shortcomings, and limitations.
2. Design: appreciation and understanding for font, color, layout, and how those factors help draw in an audience, and direct and maintain their attention.
3. Programming: ability to use code to create custom, interactive data visualizations.

The problem, it seems, is that everyone wants to be an expert in all three of these areas. I maintain that people who fit this definition are Unicorns [15] — they don't exist. Instead, people interested in creating effective visualizations need to find the area in which they have interests and strengths, and then develop their skills, appreciation, and respect for these other areas.

In my presentation, I will argue that teaching data visualization requires exposing students to these different skillsets and providing them with some level of instruction in each. It is also important to give students a broad view of the different forms and functions of data visualizations, best practices and strategies, and how to critique visualizations. I will also argue that teaching data visualization requires hands-on creation, critique, and experimentation, and I will offer several example exercises I regularly use.

Most importantly, teaching data visualization requires the instructor to recognize students' skillsets and interests. Teaching data visualization in a computer science department is much different than teaching in a policy school, business school, or in an open, public workshop. Public policy students, for example, will ultimately conduct social and policy research, write research reports and policy recommendations, and to directly communicate with practitioners and policymakers. The skills they have and the skills they will need to succeed in the workplace are very different from those who will specialize in the theory and design of computational systems, for example. By thinking carefully about the needs of the students, the instructor can help students ultimately think about the needs of their audience.

Bio: Jon Schwabish is an economist, writer, teacher, and creator of policy-relevant data visualizations. He is considered a leading voice for clarity and accessibility in how researchers communicate their findings. He has taught countless workshops on data visualization, presentation skills, and data visualization in Excel and Tableau. He has also taught data visualization classes in the McCourt School of Public Policy and the at Georgetown University, the McDonough School of Business at Georgetown University, and the Maryland Institute College of Art. He is a sought-after speaker about data visualization, open data, and data workflow processes. He is currently writing a book with Columbia University Press on presentation design and techniques. He is on Twitter @jschwabish.

REFERENCES

[1] E. Adar. Guess: a language and interface for graph exploration. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 791–800. ACM, 2006.

[2] J. Appleman, A. Gupta, A. Rajagopal, J. Shishido, and M. A. Hearst. Exploring data for fun and profit: Case study of Jeopardy! In *IEEE Conference on Information Visualization [Poster]*, 2015.

[3] R. Buchanan. Wicked problems in design thinking. *Design issues*, pages 5–21, 1992.

[4] J. Chase and E. G. Okie. Combining cooperative learning and peer instruction in introductory computer science. *ACM SIGCSE Bulletin*, 32(1):372–376, 2000.

[5] C. H. Crouch and E. Mazur. Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69(9):970–977, 2001.

[6] M. Dörk, N. H. Riche, G. Ramos, and S. Dumais. Pivotpaths: Strolling through faceted information spaces. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2709–2718, 2012.

[7] S. P. Dow, J. Fortuna, D. Schwartz, B. Altringer, D. L. Schwartz, and S. R. Klemmer. Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In *CHI*. ACM, 2011.

[8] L. Jin and D. C. Banks. Tennisviewer: A browser for competition trees. *Computer Graphics and Applications, IEEE*, 17(4):63–65, 1997.

[9] D. W. Johnson, R. T. Johnson, and K. A. Smith. *Active learning: Cooperation in the college classroom*. Interaction Book Co., Edina, MN, 1991.

[10] A. Kerren, J. T. Stasko, and J. Dykes. Teaching information visualization. *Information Visualization: Human-Centered Issues and Perspectives*, 4950:65, 2008.

[11] T. Lord. Cooperative learning that really works in biology teaching: using constructivist-based activities to challenge student teams. *The American Biology Teacher*, 60(8):580–588, 1998.

[12] J. Markoff. Virtual and artificial, but 58,000 want course. *The New York Times*, 15, 2011.

[13] B. J. Millis and P. G. Cottell. *Cooperative learning for higher education faculty*. Oryx Press (Phoenix, Ariz.), 1998.

[14] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[15] J. Schwabish. Communicating research: build a unicorn, don't look for one. *Urban Wire*, (April 15), 2015. http://www.urban.org/urban-wire/communicating-research-build-unicorn-dont-look-one.

[16] M. K. Smith, W. B. Wood, W. K. Adams, C. Wieman, J. K. Knight, N. Guild, and T. T. Su. Why peer discussion improves student performance on in-class concept questions. *Science*, 323(5910):122–124, 2009.

[17] L. Springer, M. E. Stanne, and S. S. Donovan. Effects of small-group learning on undergraduates in science, mathematics, engineering, and technology: A meta-analysis. *Rev. of Ed. Research*, 69(1):21–51, 1999.

[18] D. A. Trytten. Progressing from small group work to cooperative learning: A case study from computer science. *Journal of Engineering Education*, 90(1):85–91, 2001.

[19] J. Walny, S. Huron, and S. Carpendale. An exploratory study of data sketching for visual representation. *Computer Graphics Forum*, 34(3):231–240, 2015.