A Curriculum for Tools for Thought

Will Crichton
will_crichton@brown.edu
Brown University
Providence, Rhode Island, USA

Abstract

To build the next generation of tools for thought (TfT), the TfT community would benefit from consolidating a foundation of conceptual and technical knowledge that can support this emerging subfield of HCI. However, no existing book or course appears to span the breadth of what I consider the key ideas in TfT. I therefore decided to create a new course at Brown, CSCI 1377 (Tools for Thought), to address this gap. I am currently running this course in Spring 2026. This paper describes my curricular design decisions, along with my preliminary experience teaching the course. The goal of this paper is to spark a broader discussion about what topics are considered foundational for TfT, and to discuss approaches for teaching TfT to undergraduates.

1 Introduction

If you set out to learn about tools for thought, you will be quickly stymied by the lack of a comprehensive text or course on the subject. For example, if I Google for "tools for thought," the top results are Howard Rheingold's book *Tools for Thought* [43], Microsoft's new landing page for work related to the present workshop, and some lovely essays by Maggie Appleton [7] as well as Andy Matuschak and Michael Nielsen [35]. If I pose the question to ChatGPT, it will tell me to start by reading Andy's work, then follow up with *Augmenting Human Intellect* [19], *As We May Think* [14], and finally *Deschooling Society* [24]. These resources, while stimulating, are hardly a comprehensive take on tools for thought.

Another sensible place to learn about TfT would be from resources in the human-computer interaction (HCI) community. General texts such as Introduction to Human-Computer Interaction by Hornbæk et al. [23] provide an admirably broad look across relevant topics such as cognition and interaction design, but cannot address the full range of systems specifically relevant to TfT. At the other extreme, some HCI texts and courses focus on specific slices of TfT, such as data visualization or hypertext systems. A few courses sit in the middle, such as MIT's *Cognitive Augmentation* course [2] and CMU's *Augmenting Intelligence* [1]. However, these courses are closer to graduate-level surveys on recent research than to a comprehensive look at TfT foundations. Moreover, many HCi courses omit topics I personally consider foundational for TfT. For instance, Kenneth Iverson's venerable Turing Award speech "Notation as a Tool for Thought" [25] deserves a spot in any TfT curriculum.

I therefore decided to design my own course, *Tools for Thought*, which I am teaching for the first time at Brown this semester. Being halfway through the first iteration of the course, I can only provide a limited experience report. Instead, I will focus on the preliminary curricular design, both for the goal of getting early feedback from the TfT community, and providing inspiration to others who may be considering designing a similar course of their own. You can view the live curriculum for the course here: https://cel.cs.brown.edu/csci-1377-s26/

2 Curriculum

I believe that a course on TfT should provide the foundations to understand today's most interesting computational TfT: hypertext, data visualization & data analysis, spaced repetition systems, cognitive tutors, end-user programming, computational notebooks, programming languages, to name a few. I think these systems can be usefully understood through three lenses:

(1) *Psychology:* one has to understand thought to build tools for it. The hard part is picking the bits of psychology that are the most relevant to design. I generally focus on cognitive psychology, having a body of results that replicate (cf. social psychology) and are high-level enough to be useful (cf. neuroscience).
(2) *History:* humans have developed analog TfT over thousands of years. A TfT course should help students understand how people of the past used the intellectual materials of their time to develop TfT, and how one could be analogously creative today. The course should take a broad view on what constitutes a TfT— oral tradition, for example, is a mnemonic technology.
(3) *Engineering:* the utility of a TfT for thinking is one dimension in a larger design space. Students should understand the other dimensions: what makes a TfT fun to use, when can one TfT interoperate with another, what makes a TfT have longevity?

One notable difference between a TfT course and a traditional HCI course concerns user research. Many HCI courses will cover some element of iterative design, qualitative analysis, and similar design practices. Such practices are indispensable for user interfaces which people can learn and use at a time scale of minutes, hours, or days. However, tools for thought are most impactful when adopted over much longer timescales. For example, the skill of reading is learned over many years and at great social expense, but we consider literacy worth the investment nonetheless. A TfT course should therefore focus on giving students generative frameworks they can use to evaluate designs before user testing, such as the Cognitive Dimensions of Notation [21].

2.1 Lectures

Table 1 shows the first 19 lectures of the curriculum, spanning 10 weeks. The last four weeks are reserved for in-class seminars on AI+TfT while the students are doing their final project. Most weeks cover a new topic in two parts, with the first part focusing more on history and psychology, and the second part focusing more on systems and design. A few notes on the choice of topics:

- This curriculum is aggressively breadth-oriented. Each week could easily constitute a semester-long course. I opted for breadth because I wanted a course that truly could span TfT and not

Table 1: The curriculum for CSCI 1377, Tools for Thought.

Lecture	Subtitle	Principal Sources
Introduction		"As We May Think" [14]; "Mother of All Demos" [19]; "Dynabook: The Complete Story" [27]; "Knowledge Navigator" [6]; "Computational Public Space" [54]
Mnemonics I	Science and Tradition of Memory	*Cognitive Psychology* Chs 6–7 [4]; *Orality and Literacy* [40]; *Memory in Oral Traditions* [44]
Mnemonics II	Spaced Repetition Systems	"Spaced Repetition for Efficient Learning" [13]; "How can we develop transformative tools for thought?" [35]
Reading I	Technology of the Written Word	*A History of Writing* [20]; *Psychology of Reading* [42]; *Cognitive Psychology* Ch 13 [4]
Reading II	Reading Augmentation Systems	*The Myth of the Paperless Office* [47]; "LiquidText" [48]; "Explorable Explanations" [52]; "The Semantic Reader Project" [33]
Hypertext I	Managing Information Overload	*The Printing Revolution in Early Modern Europe* [18]; *Too Much to Know: Managing Scholarly Information before the Modern Age* [10];
Hypertext II	The Internet and Beyond	"Information Management: A Proposal" [9]; "Prolegomena to Any Future Hypertext System" [17]; "Patterns of Hypertext-Augmented Sensemaking" [61]
Visualization I	Principles of Perception	*Information Visualization* [55]; *Visualization Analysis & Design* [38]
Visualization II	Data Visualization Systems	*The Grammar of Graphics* [57]; "Vega-Lite: A Grammar of Interactive Graphics" [46]
Visualization III	Effective & Expressive Visualization	*The Visual Display of Quantitative Information* [49]; "Automating the design of graphical presentations of relational information" [34]
Multimedia I	Communicating in Space	"Why a Diagram is (Sometimes) Worth Ten Thousand Words" [30]; *Understanding Comics* [37]
Multimedia II	Communicating in Time	*Multimedia Learning* [36]; *The Cognitive Style of PowerPoint* [50]; "Animation: can it facilitate?" [51]
Learning I	Constructing Knowledge	*Mindstorms* [41]; "Learnable Programming" [53]; *The Charisma Machine* [3]
Learning II	Software Tutors	"The 2 Sigma Problem" [11]; "Cognitive Tutors: Lessons Learned" [5]
Notation I	Compacting Concepts	*The Story of Notation* [58]; *A History of Mathematical Notations* [15]; "A Representational Analysis of Numeration Systems" [60]
Notation II	Programming Systems	"Notation as a Tool of Thought" [26]; "Cognitive Dimensions of Notation" [21]; "Mathematical Notation: Past and Future" [59]
Programming I	Spreadsheets	*A Small Matter of Programming* [39]; "Creating, Comprehending and Explaining Spreadsheets" [22]; "Untidy Data" [8]; "Ambsheets" [56]
Programming II	Notebooks	"Exploration and Explanation in Computational Notebooks" [45]; "The Design Space of Computational Notebooks" [31]; "What's Wrong with Computational Notebooks?" [16]
Programming III	Devtools	"Designing the Whyline" [29]; "Mylar" [28]; "Code Bubbles" [12]; "Ply" [32]

just e.g. a data visualization or programming systems course. The consequent challenge for this course is ensuring students leave with something more than a cursory understanding of a hodgepodge of systems and concepts.

- I considered many ways to structure the progression of the course. One way was chronologically: start with the first TfT, and work up to the modern day. Another way was cognitively: start with perception, then memory, then attention, then problem-solving, and work up the ladder of cognitive complexity (e.g. Bloom's taxonomy). Ultimately, I structured the course by centering *systems*: I selected several kinds of TfT systems I felt should be in the course, then worked backwards to build a linear sequence that felt reasonable.

- The readings are mostly for my benefit — I try to cover as much factual information in class as reasonably possible, and reserve students' time out of class for hands-on assignments rather than reading. This is partially Brown-specific, because the other HCI classes here are focused more on reading than building, so I knew the students here wanted a more implementation-focused class.
- I conceptualize TfT as principally about the individual (thoughts happen in one brain after all), so this curriculum is biased away from a focus on teams / collaboration / etc. (computer-supported cooperative work) in favor of the individual.
- One weakness in the curriculum right now is a paucity of frameworks / schemata / etc. that tie together ideas across different units. There are certain repeating themes like "understand how a tool supports task in terms of the user's cognitive capabilities and limitations", but nothing more specific than that. In the future, I'd like to better integrate information across units.
- The treatment of history in the course is necessarily more narrow and technologically-focused than one might hope. For example, in discussing the impact of the printing press on society, we do not have much time to talk about its impact on the Reformation, the Thirty Years' War, and so on.

2.2 Assignments

The goal of the assignments is to have the students engage constructively with the general themes of each unit. Some conceptual material is difficult to fit onto the assignment (e.g., historical connections), so students are primarily incentivized to review it for the midterm.

We ultimately developed six assignments:

(1) For the unit on Mnemonics, we provided students a black-box computational model of a learner. Students were guided in a Juptyer notebook through a series of experiments in understanding this model, and then had to design a new spaced repetition system to find a Pareto-optimal trade off of number of reviews vs. level of recall.
(2) For the unit on Reading, we had students do an "inverse reading" of a text, whereby students are given comprehension questions, they get an LLM to answer the questions, and then selectively read the text in order to verify the quality of the answer. Students then did a design challenge where they designed an LLM interface augmentation which would address a pain point they encountered in the "inverse reading" process.
(3) For the unit on Hypertext, we borrowed a real Notion workspace from a current law students, and had students analyze its hypertextual structure. We then had students implement an Obsidian plugin which integrates with an American legal data service to automatically import hyperlinked court rulings into a user's workspace.
(4) For the unit on Visualization, students learned the grammar of graphics via Altair. Students analyzed the effectiveness and expressiveness of data visualizations, and designed their own visualizations to answer data analysis questions.
(5) For the unit on Multimedia, we had students watch a 3Blue1Brown animated math video, and then both self-test on their understanding of it, and reflect on its pedagogic structure. Students created their own mathematical animation of a spaced repetition concept from earlier in the semester, using the Manim library created by 3Blue1Brown.
(6) For the unit on Notation + Programming, we designed a hybrid spreadsheet/notebook that uses APL for the formula language. Students implement some features for the APL interpreter, and designed the notebook extension for linearly exploring a sequence of formulas in the sheet.

3 Connecting to AI

AI is explicitly not part of the main part of the curriculum, as it has only recently come into vogue as a TfT, so it would not make sense to include in a survey of foundations. But a TfT course should clearly help prepare students to work on AI-augmented TfT. I will ruminate on a few ways to make those connections.

3.1 AI as Automating the Subjective

Many analog tools for thought involve rote/mechanical/objective processes and comparatively subjective processes. The rote processes are the first to get automated with the introduction of the technology. For example, when constructing an index for a book, a computer can easily do full-text search to identify all the page locations where a word is used. However, most indexes are not concordances, so they require subjective judgments such as: which concepts should appear in the index? When does a given sentence or paragraph talk about a concept, and therefore should appear in that index's entry?

These questions are naturally ones that could be addressed through modern generative AI. One could imagine building an index in a "neurosymbolic" manner by combining LLM-based judgements of relevance with algorithmic full-text search. More generally, any instance where development of a TfT has historically required subjective judgment suggests a potential area for application of AI.

3.2 AI as Shifting Task Structures

The process of certain types of knowledge work like reading is rapidly changing with the advent of LLMs. Plato's famous objection that books cannot talk back like a human is no longer true. Today, when people perform tasks like reading a document for the sake of answering a question involving it, they are mediating that process through an LLM. This new question-answering process has shifted the task structure of reading comprehension from linear reading with skimming to something qualitatively different. Certainly programming with agents looks very different from programming in an IDE.

A TfT course is a good place to investigate these new workflows, and ask how old ideas can be applied in new setings. A good example of this is the second assignment for "inverse reading" described in Section 2.2.

3.3 AI as Eroding Productive Friction

Many tools for thought derive their cognitive benefit not from efficiency but from friction. When Luhmann wrote a slip for his Zettelkasten, the act of reformulating an idea in his own words forced a deeper engagement with the material than copying would have. The difficulty is productive in the sense that the effort required to perform the task is what causes learning to occur.

A TfT course rooted in historical foundations should give students a vocabulary for recognizing when friction is productive and when it is merely tedious—a distinction that is easy to collapse when the tool that removes both looks the same. The hope is that students who have, for example, built an index by hand and reflected on what that process demanded of them will be better positioned to make deliberate choices about what to delegate and what to retain.

4 Conclusion

This curriculum is just the beginning, and I will continue iterating on it in the coming years. I welcome any feedback, and I hope you found some ideas in the curriculum's design that inspire you in turn.

References

[1] 2026. Augmenting Intelligence. https://coaugmentation.com/

[2] 2026. MAS.S62: Cognitive Augmentation. https://augmentation.media.mit.edu/schedule/

[3] Morgan G. Ames. 2019. *The Charisma Machine: The Life, Death, and Legacy of One Laptop per Child.* MIT Press. https://mitpress.mit.edu/9780262537445/the-charisma-machine/

[4] John Anderson. 2009. *Cognitive Psychology and Its Implications* (7 ed.). Worth, New York, NY.

[5] John R. Anderson, Albert T. Corbett, Kenneth R. Koedinger, and Ray Pelletier. 1995. Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences* 4, 2 (1995), 167–207. doi:10.1207/s15327809jls0402_2

[6] Apple Computer. 1987. Knowledge Navigator. Video produced for John Sculley's EDUCOM 1987 keynote. https://www.youtube.com/watch?v=umJsITGzXd0

[7] Maggie Appleton. 2022. *Tools for Thought as Cultural Practices, not Computational Objects.* https://maggieappleton.com/tools-for-thought

[8] Lyn Bartram, Michael Correll, and Melanie Tory. 2022. Untidy Data: The Unreasonable Effectiveness of Tables. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2022), 686–696. doi:10.1109/TVCG.2021.3114830

[9] Tim Berners-Lee. 1989. Information Management: A Proposal. Internal proposal, CERN. https://www.w3.org/History/1989/proposal.html

[10] Ann M. Blair. 2010. *Too Much to Know: Managing Scholarly Information before the Modern Age.* Yale University Press. https://yalebooks.yale.edu/book/9780300165395/too-much-to-know/

[11] Benjamin S. Bloom. 1984. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher* 13, 6 (1984), 4–16. doi:10.3102/0013189X013006004

[12] Andrew Bragdon, Steven P. Reiss, Robert Zeleznik, Suman Karumuri, William Cheung, Joshua Kaplan, Christopher Coleman, Ferdi Adeputra, and Jr. LaViola, Joseph J. 2010. Code Bubbles: Rethinking the User Interface Paradigm of Integrated Development Environments. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE '10).* 455–464. doi:10.1145/1753326.1753706

[13] Gwern Branwen. 2009. Spaced Repetition for Efficient Learning. https://gwern.net/spaced-repetition Last updated 2019.

[14] Vannevar Bush. 1945. As We May Think. *The Atlantic Monthly* 176, 1 (1945), 101–108.

[15] Florian Cajori. 1928. *A History of Mathematical Notations.* Open Court. https://monoskop.org/images/2/21/Cajori_Florian_A_History_of_Mathematical_Notations_2_Vols.pdf

[16] Souti Chattopadhyay, Ishita Prasad, Austin Z. Henley, Anita Sarber, and Michael Coblenz. 2020. What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems.* 1–12. doi:10.1145/3313831.3376729

[17] Steven J. DeRose. 2014. What Do We Still Lack? Or: Prolegomena to Any Future Hypertext System. In *Proceedings of Balisage: The Markup Conference,* Vol. 14. http://www.balisage.net/Proceedings/vol14/html/DeRose01/BalisageVol14-DeRose01.html

[18] Elizabeth L. Eisenstein. 2013. *The Printing Revolution in Early Modern Europe* (2nd ed.). Cambridge University Press. https://www.cambridge.org/core/books/printing-revolution-in-early-modern-europe/85AE910ACCCFCE8C068DDA911990AEFC

[19] Douglas C. Engelbart. 1962. *Augmenting Human Intellect: A Conceptual Framework.* Technical Report AFOSR-3223. Stanford Research Institute.

[20] Steven Roger Fischer. 2021. *A history of writing.* Reaktion Books, London, England.

[21] T. R. G. Green. 1989. Cognitive Dimensions of Notations. In *People and Computers V.* Cambridge University Press, 443–460. https://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/papers/Green1989.pdf

[22] David G. Hendry and T. R. G. Green. 1994. Creating, Comprehending and Explaining Spreadsheets: A Cognitive Interpretation of What Discretionary Users Think of the Spreadsheet Model. *International Journal of Human-Computer Studies* 40, 6 (1994), 1033–1065. doi:10.1006/ijhc.1994.1047

[23] Kasper Hornbæk, Per Ola Kristensson, and Antti Oulasvirta. 2025. *Introduction to Human-Computer Interaction.* Oxford University Press. https://introductiontohci.org/

[24] Ivan Illich. 1971. *Deschooling Society.* Harper & Row, New York.

[25] Kenneth E. Iverson. 1980. Notation as a tool of thought. *Commun. ACM* 23, 8 (Aug. 1980), 444–465. doi:10.1145/358896.358899

[26] Kenneth E. Iverson. 1980. Notation as a Tool of Thought. *Commun. ACM* 23, 8 (1980), 444–465. doi:10.1145/1283920.1283935

[27] Alan Kay. 1996. Dynabook: The Complete Story. Lecture, Computer History Museum. https://www.youtube.com/watch?v=IwL3yXdupv0

[28] Mik Kersten and Gail C. Murphy. 2005. Mylar: A Degree-of-Interest Model for IDEs. In *Proceedings of the 4th International Conference on Aspect-Oriented Software Development (AOSD '05).* 159–168. doi:10.1145/1052898.1052912

[29] Andrew J. Ko and Brad A. Myers. 2004. Designing the Whyline: A Debugging Interface for Asking Questions about Program Behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04).* 151–158. doi:10.1145/985692.985712

[30] Jill H. Larkin and Herbert A. Simon. 1987. Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science* 11, 1 (1987), 65–100. doi:10.1111/j.1551-6708.1987.tb00863.x

[31] Sam Lau, Ian Drosos, Julia M. Marber, Nisha Doshi, and Andrew Head. 2020. The Design Space of Computational Notebooks: An Analysis of 60 Systems. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020). doi:10.1109/TVCG.2020.2991588

[32] Sarah Lim, Josh Hibschman, Haoqi Zhang, and Eleanor O'Rourke. 2018. Ply: A Visual Web Inspector for Learning from Professional Webpages. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18).* 991–1002. doi:10.1145/3242587.3242660

[33] Kyle Lo, Joseph Chee Chang, Andrew Head, Jonathan Bragg, Amy X. Zhang, Cassidy Trier, Chloe Anastasiades, Tal August, Russell Authur, Danielle Bragg, Erin Bransom, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Yen-Sung Chen, Evie Yu-Yen Cheng, Yvonne Chou, Doug Downey, Rob Evans, Raymond Fok, Fangzhou Hu, Regan Huff, Dongyeop Kang, Tae Soo Kim, Rodney Kinney, Aniket Kittur, Hyeonsu B. Kang, Egor Klevak, Bailey Kuehl, Michael J. Langan, Matt Latzke, Jaron Lochner, Kelsey MacMillan, Eric Marsh, Tyler Murray, Aakanksha Naik, Ngoc-Uyen Nguyen, Srishti Palani, Soya Park, Caroline Paulic, Napol Rachatasumrit, Smita Rao, Paul Sayre, Zejiang Shen, Pao Siangliulue, Luca Soldaini, Huy Tran, Madeleine van Zuylen, Lucy Lu Wang, Christopher Wilhelm, Caroline Wu, Jiangjiang Yang, Angele Zamarron, Marti A. Hearst, and Daniel S. Weld. 2024. The Semantic Reader Project. *Commun. ACM* 67, 10 (Sept. 2024), 50–61. doi:10.1145/3659096

[34] Jock Mackinlay. 1986. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.* 5, 2 (April 1986), 110–141. doi:10.1145/22949.22950

[35] Andy Matuschak and Michael Nielsen. 2019. *How Can We Develop Transformative Tools for Thought?* https://numinous.productions/ttft/

[36] Richard E. Mayer. 2020. *Multimedia Learning* (3rd ed.). Cambridge University Press.

[37] Scott McCloud. 1993. *Understanding Comics: The Invisible Art.* HarperPerennial.

[38] Tamara Munzner. 2014. *Visualization Analysis and Design.* CRC Press. https://www.cs.ubc.ca/~tmm/vadbook/

[39] Bonnie A. Nardi. 1993. *A Small Matter of Programming: Perspectives on End User Computing.* MIT Press.

[40] Walter J Ong. 2002. *Orality and Literacy* (2 ed.). Routledge, London, England.

[41] Seymour Papert. 1980. *Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books. https://worrydream.com/refs/Papert_1980_-_Mindstorms,_1st_ed.pdf

[42] Keith Rayner, Alexander Pollatsek, Jane Ashby, and Charles Clifton, Jr. 2011. *Psychology of reading.* Psychology Press, London, England.

[43] Howard Rheingold. 2000. *Tools for thought.* MIT Press, London, England.

[44] David C Rubin. 1998. *Memory in oral traditions.* Oxford University Press, New York, NY.

[45] Adam Rule, Aurélien Tabard, and James D. Hollan. 2018. Exploration and Explanation in Computational Notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* 1–12. doi:10.1145/3173574.3173606

[46] 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:10.1109/TVCG.2016.2599030

[47] Abigail J Sellen and Richard H R Harper. 2003. *The myth of the paperless office.* MIT Press, London, England.

[48] Craig S. Tashman and W. Keith Edwards. 2011. LiquidText: a flexible, multitouch environment to support active reading. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11).*

Association for Computing Machinery, New York, NY, USA, 3285–3294. doi:10.1145/1978942.1979430

[49] Edward R. Tufte. 1983. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, CT.

[50] Edward R. Tufte. 2003. *The Cognitive Style of PowerPoint: Pitching Out Corrupts Within.* Graphics Press, Cheshire, CT.

[51] BARBARA TVERSKY, JULIE BAUER MORRISON, and MIREILLE BETRANCOURT. 2002. Animation: can it facilitate? *International Journal of Human-Computer Studies* 57, 4 (2002), 247–262. doi:10.1006/ijhc.2002.1017

[52] Bret Victor. 2011. Explorable Explanations. https://worrydream.com/ExplorableExplanations/

[53] Bret Victor. 2012. Learnable Programming. https://worrydream.com/LearnableProgramming/

[54] Bret Victor. 2024. Computational Public Space. Presentation at the Screenless Cities Conference, Dynamicland. https://dynamicland.org/2024/Computational_Public_Space/

[55] Colin Ware. 2020. *Information Visualization: Perception for Design* (4th ed.). Morgan Kaufmann. https://www.sciencedirect.com/book/monograph/9780128128756/information-visualization

[56] Alex Warth, Geoffrey Litt, and Szymon Bryant. 2025. Ambsheets: Spreadsheets for Exploring Scenarios. Ink & Switch. https://www.inkandswitch.com/ambsheets/notebook/

[57] Leland Wilkinson. 2005. *The Grammar of Graphics* (2 ed.). Springer, New York.

[58] C. F. Abdy Williams. 1903. *The Story of Notation.* Charles Scribner's Sons. https://archive.org/details/storyofnotation00willrich

[59] Stephen Wolfram. 2000. Mathematical Notation: Past and Future. MathML and Math on the Web: MathML International Conference. https://www.stephenwolfram.com/publications/mathematical-notation-past-future/

[60] Jiajie Zhang and Donald A. Norman. 1995. A Representational Analysis of Numeration Systems. *Cognition* 57, 3 (1995), 271–295. doi:10.1016/0010-0277(95)00674-3

[61] Siyi Zhu, Robert Haisfield, Brendan Langen, and Joel Chan. 2024. Patterns of Hypertext-Augmented Sensemaking. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24).* 1–17. doi:10.1145/3654777.3676338