

DOI: <https://doi.org/10.24297/ijct.v21i.9889>

A Tool Development Framework to Support Design Thinking for Software Engineering

Ayşe Kok Arslan

ay_kaa@protonmail.com

Abstract ;

This research aims to create a framework to guide the development of design thinking support tools - that is, tools that enable people to express themselves creatively and develop as creative thinkers. The main goal is to develop advanced software and social networking sites that empower users to not only be productive, but also have new technologies. Potential users of these interfaces include software with other engineers, various scientists, product and image designers, builders, teachers, students and many more. Improved communication methods can enable effective psychological search, improved interaction between groups, and faster recovery processes. These advanced combinations should also provide strong support for hypothesis formation, rapid testing of alternatives, improved visual perception, and better distribution of results.

Keywords— Tool development, design thinking, software engineering

I. INTRODUCTION

An article in Creative Review's bestselling magazine aptly states that it is "a good time to build." The concept of design thinking has never been so widely regarded as powerful and as culturally influential as it is now.

This research provides an analysis of design principles regarding the development of "design thinking tools" - that is, computer systems and environments that people can use to produce, convert, interact and play with, and / or share logical and / or physical displays. Although it is difficult to study "design thinking" itself, we can study the way people and creative teams implement design thinking, and use their excellent methods in tools that can help others to imitate those processes.

II. RELATED WORK

Design thinking often uses Human Computer Interaction (HCI) methods, which focus on personal experience, or User Centered Design (UCD) methods, which focus on the use and user features of the product. The "building process" includes building methods into a series of actions, steps, or events. More precisely, it refers to the stages or stages of design. No unique process can completely define design thinking. Researchers present a process of creative thinking using different semantics. There is no right or wrong approach to design thinking process. For this reason, the following discussion explores various approaches to design thinking.

Human Centered Design (HCD) was inspired by the need for social capital in developing countries. Presented by IDEO. It is based on the IDEO's original design ideas, but its terms and framework have been simplified to make it easier to use in the context of public naming. In this new context, the HCD abbreviation is translated "hear, create, and present".

- The sensor refers to the need for empathy to address the needs of the user himself.
- Design refers to thinking, assessment and evaluation as learning processes in which solutions can be developed.
- Submission refers to the implementation of the project, and includes addressing remedial issues and overcoming potential challenges during the life span of the project.

One of the well-known design thinking methods is the Hasso Plattner Institute of Design at Stanford (d.school) 5 stage methodology. The "Empathize / Define / Ideate / Prototype" categories can be analyzed as follows:

Step 1 - Empathy

The word "empathy", means the designer's attempt to understand the problem in depth. D.school Bootcamp Bootleg proposes the following methods as a way to empathize with users effectively:



- Embracing the mindset of the beginner, which means letting go of the ignorant thoughts that the designer has built up through his or her experience. This step is important in understanding the needs of users [29].
- Ask questions, and ask directly "what is the situation like?", Which leads to problem building; "How does the user feel about the situation?", Which can be answered by looking at gestures and expressions; "And why does the user have this experience", which leads to understanding the user's motivation
- Step 2 - Explain
- The definition of a problem statement is an important function of the design thinking process. Positive problem statements have the following characteristics:
- Focus on people; a positive statement focuses on people, their needs, their feelings, and their desires rather than specific, solutions, technologies or resources.
- Broad to allow for intelligence; a good statement does not include statements that point to a specific solution, methodology, or technology.
- Mapping user input containing information divided into four categories: what users said, what they did, what they thought, and how they felt.
- Visual problem statement (POV); describing the problem of a particular group of users; this work sets the stage for a solution in the latest stages of the design thinking process.

Step 3 - Prepare

The concept section challenged designers to "think outside the box" [13] by presenting possible solutions to a clearly defined problem statement built into user input. This phase is defined by all the inventions and techniques, which are important in making effective solutions to problems that are difficult to solve [32]. Potential new activities include thinking, challenging thinking, mind mapping, drawing, building story boards, analog drawing, annoying, organizing workshops, and more. SCAMPER is a widely accepted method that allows for innovation in an existing product by looking at different lenses. This approach means replacing, merging, adapting, modifying, replacing, deleting, and reversing [35].

Step 4 - Prototype

Prototyping involves the design of low-resolution versions that reflect performance and other features of the final product. [38]. The purpose of prototypes is to verify ideas and to see how they work and how they are received by users.

There are two different types of prototyping [38]:

- Low reliability: This refers to the basic models of a complete product or fixed solution, used not ultimately but cheap enough to demonstrate the desired performance [38]. The idea of making low fidelity to produce faster and cheaper is a cheaper, discarded version of the final product for testing purposes. Typical methods include storytelling, drawing, and more.
- High reliability performance: This refers to a product version close to the last. While high reliability methods are more productive they offer advantages as users are more likely to engage with a version similar to the end product very closely Both low and low reliability methods have advantages and disadvantages.
- Step 5 - Test

Exploring the final phase of the design thinking process. Includes final product testing or intermediate prototypes either by the launch team or by users. The results of the experimental phase may be reversed in the design thinking process, as, as previously stated, it is a retrospective method [39].

Some of these principles have emerged in collaboration with a large number of students, in the creation of a wide range of design thinking tools. Some of the principles also apply to software development tools, commonly referred to as "User Interface Tools," but specialized design tools highlight new ideas and needs. Potential users of these interfaces include software with other engineers, various scientists, product and image designers, builders, teachers, students and many more. Improved communication methods can enable effective psychological search, improved interaction between groups, and faster recovery processes. These advanced combinations should also provide strong support for hypothesis formation, rapid testing of alternatives, improved visual perception, and better distribution of results.

III. PROPOSED MODEL

The process of creative design is not a standard production process that can be documented, and which tools and presentations people use most affect their studies of thought processes and processes [de la Rocha, 1985] [Zhang, 1997] [Shirouzu et al. 2002].

1. Support Exploration

An important requirement for design thinking is being able to try different alternatives. By definition, creative work means that the final structure is not known at first, so users should be encouraged to explore space [Fischer 1994]. This has many implications for the tools. Test systems have been developed for a long time [Sheil 1983], but many tools still focus on projects where the outcome is known previously. According to Green and Petre [Green 1996], the aim is to develop systems with a "low viscosity" - which make it easy to change all aspects of the design.

First, it should be very easy to try things, and then go back when you are unsuccessful. This means that the tools must be reliable so that users are comfortable with trying things out. For example, excellent 'Undo' power is required for tools. Getting Started Postpartum can be very difficult but [Myers 1996], many research programs are abandoning it. The rich histories needed to support the retreat can also be helpful in building consultation programs [Myers 1992], where users teach the program to perform repetitive tasks by providing examples. Preview methods [Terry 2002] and established works [Terry 2004] have also been proposed and tested to support such processes.

The second requirement is that the tools be "self-explanatory" to make it clear to users what can be done. If flexibility is not detected, it will not be used. This is especially important as users learn the tools. The tools should also be face-to-face and unobtrusive, so that professional users can try other methods very quickly. Ultimately, tools should be fun and enjoyable to use. When people are stressed or too focused on using the tools themselves, then they will have a few logical resources left to be used in finding creative solutions to their tasks.

Spreadsheets are famous for giving people the ability to compare results in if-if situations [Brown 1987], by enabling the user to easily distinguish what will remain consistent (formulas) from what should be different (values to be tested). In the field of user interface, some tools are popular as they allow for the development of prototypes, testing and modification.

Another way to support exploration is to speed up the process of "drawing" alternatives in the early stages of construction. Professional user interface designers often try many ideas by drawing on paper or white board, before starting to write the code for actual use. The goal is to allow a partial effort to achieve a partial result quickly.

Support testing requires performance made available through careful communication design. We view the calculation tool as a set of tools that users interact with to create a "user-friendly" environment [Schoen 1983] [Nakakoji 2000a] Promotional, non-blocking, innovative tools need to be built around understanding what presentations users need to work with [Yamamoto 2005].

The interoperability structure of the tool influences the user's thinking process. Collaborative design refers to the determined representation and operation of the application process [Yamamoto 2005]. Programs to support creative processes need to empower users not only to create unique objects, but also to think about

what they should do with novel artefacts [Nakakoji 2005]. Historically, existing tools and applications have been widely used in the creation of digital art. Examples are software-processing software, software-processing software, or spreadsheet applications. Specified 3D CAD programs have been found to be effective in helping architects to develop solutions but prevent them from their old experiments [Lawson 1994].

2. Low Threshold, High Ceiling, and Wide Walls

Functional tool designs should make it easier for novice starters (low limit) but also possible for professionals to work on progressive projects (high ceilings) [Myers 2000]. The lower limit means that the interface should not be intimidating, and should give users immediate confidence that they can succeed. High ceilings mean powerful tools and can create complex, complete solutions. Often tools that enable creative thinking can be very difficult to learn (they have no bottom line). Instead, they focus on providing more powerful features so that experts can consolidate results faster.

Now, a third objective should be added: broad walls. Design thinking tools should support and elevate the scope of testing. By excluding the predefined widgets, the tool encourages designers to explore many different ways to control communication, instead of using buttons and scrolling bars. When children use MIT's Programmable LEGO Bricks, for example, they can create anything from a robot creature to a "smart" house to a coherent precision to a musical instrument [Resnick 1993] [Resnick 1996]. The aim is to empower users to work on projects that grow to their liking and **preferences** - meaning that technical support tools need to support a wide variety of projects.

When evaluating the use of innovative support tools, variability of outcomes should be considered an indicator of success. If all creations are the same, another feeling might be wrong. If, after completing one project, users feel that they have "finished" with the tool, then the feeling of failure may continue. Design thinking tools should define a test space, not a set of specific tasks. The designer's goal should be to make users feel continuously amazed while exploring the space of opportunity.

The problem with low-end programs is that they are often limited in what they can do, so users are restricted, or even need to find "jobs" to achieve what they want. High-performance tools often require significant training and effort to learn to use. Wide walls mean that there are some common starting areas that users should learn to integrate.

One strategy to achieve all three is to explicitly include materials and materials that can be used in many different ways. The design challenge should be clear enough so that users can quickly understand how to use the features (lower limit), but overall enough so that users can continue to discover new ways to use them (wide walls). The tool should help users learn to use features, for example with mouse-overs, tool tips, and various examples, so users can make the necessary changes to understand several possible uses.

3. Support Many Paths and Many Styles

When MIT researchers explored the classic form of LEGO computer-controlled technology, they explored prototypes in the fourth grade class where students wanted to build an amusement park. One group of students decided to create a fun meeting. They carefully make plans, create routes, and then write a plan to make the ride rotate around the person pressing the touch sensor. Within a few hours, their merry-go-round was working. Another group of students decided to build a Ferris wheel. However, before leaving, they set it aside and began building a food stand next to the Ferris wheel. The developers were concerned: the recreational stand had no car or sensors or systems. They are worried that students will lose some of the powerful ideas associated with the LEGO work. Still, they did not interfere. After completing the refreshment store, the team built a wall around the amusement park, built a parking lot, and added a few LEGO people walking around the park. Then, at last, they went back and finished their Ferris wheel.

These two groups represent two very different styles of play, design, and imagination. Turkle and Papert [Turkle 1990] described these styles as "heavy" (first group) and "soft" (second). Hard and soft ways, they explain,

"are each characterized by a cluster of attributes. Some involve organization of work (the hards prefer abstract thinking and systematic planning; the softs prefer a negotiational approach and concrete forms of reasoning); other attributes concern the kind of relationship that the subject forms with computational objects. Hard mastery is characterized by a distanced stance, soft mastery by a closeness to objects."

In many mathematics and science classes, the hard way is right, considered superior to the soft way. Turkle and Papert opposed "epistemological pluralism" which considers the soft method to be different, not inferior. The same scenario should be taken when it comes to the construction of new design thinking tools, prioritizing the support of students of all styles and styles. Special care should be taken to ensure that technology and services are available and attractive to the sofas. Since mathematics and science have traditionally been reaping hardships, designers may want to work harder to close the gap.

4. Support Collaboration

At the core of this diversity lies the need to provide collaborative support to tools. The focus should be on creating groups that include people with different strengths. It is important that design thinking tools allow each person to donate using his or her talent.

With the advent of the Internet, another "collaborative" approach has become more widespread: getting the best out of others through social media. Design thinking tools should encourage the user community to share their creativity, and the strategies and strategies they have acquired for using the tools.

Social and psychological factors such as trust and budgeting play an important role in supporting collaborative intelligence [Nakakoji 2000b] [Shneiderman 2000].

5. Support Open Interchange

The development process will not usually be supported by a single tool, but will instead require the user to configure various tools each of which supports a part of the task. Design thinking tools should interact freely with other tools. This includes the ability to easily import and export data from common tools such as spreadsheets, word processors and data analytics tools, and other creative support tools. This requires that the data formats in the files be opened and properly defined.

Another form of openness allows for the tendency of the tools themselves. Professional tools are increasingly providing the design of a "plug-in", or "open data model" [Myers 1998] to support expansion. This has long been achieved with state-of-the-creativity tools such as PhotoShop to allow creative people to define their work that works on shared data types. A professional suite of tools from companies such as AutoDesk (<http://www.autodesk.com>) and Adobe (<http://www.adobe.com>), designed primarily to facilitate the capture of results from one tool to another.

The functional integration of the tools can allow for smooth interaction across all windows and better integration of the tools.

6. Make It As Simple As Possible - and Maybe Even Simpler

The designers are eager to return to the pure use of the 1980s Macintosh. When they see the role of stress-that is; using more sophisticated technology, and helping users to perform complex tasks, also make user information easier. Designers are trying to create systems that offer simple ways to make things more sophisticated.

Experience shows that reducing the number of features can improve the user experience. What initially looked like a barrier or limit could encourage new forms of creativity. In the mid-1990's, for example, MIT researchers developed a LEGO Brick Planner that was about the size of a baby juice box. It can control four engines and receive inputs from six sensors. For a sponsor event at the Media Lab, some decoration of the table interaction was required. All of the planned Bricks skills were unnecessary, so a smaller, reduced version was developed quickly, almost the same size as the matchbox car. It can control only two motors with input from only two sensors. This was expected to be a temporary project, because they "knew" that most users would want more engines and more sensors. But with the discovery of a reduced version (called Cricket), people were getting

more and more programs to prepare for it, even though (or maybe it was because of) its limited limits. Although the first unplanned brick was better for some projects, the simplicity of Cricket has been successful.

7. Choose Black Boxes Carefully

When children build robots with MIT's Programmable Bricks, for example, they learn processes and preparation, and they learn about feedback and control. However, they are usually not students of the internal functioning of engines. The car is always a black box. If you wanted to help children learn how motors work, you should design a construction kit with low-grade building blocks, so that children can build their own motors.

Similarly, choosing basic "building blocks" in a programming language determines what children can learn as they use the language. When children combine Logos commands as forward and for commands such as 4 repetitions [**forward 50 right 90**] (square) or 360 repetitions [front 1 right 1] (drawing a circle), they gain a better understanding of the concepts, and it is suggested that they too they learn important mathematical and geometric concepts [Papert 1980]. However, the first command to move **forward** is still in the black box. Each time the tortoise moves, the computer must calculate the new x and y positions from the first positions and y using the trigonometric calculations. These statistics are hidden from the user. If the goal of our building project were to help children learn these types of trigonometric calculations, then the tortoise would be a bad black box. However, by hiding these figures inside the black box, the tortoise lets the user try and explore other mathematical and geometric ideas.

All language designers and tools face the same challenge. This is very much related to the above point about the floor compared to the roof - the higher the level of primitives, the easier they are to use, but there is little they can do. However, at the same level of competence, there are simpler and more complex ways of presenting the same tasks. The goal should be to empower users to achieve a specific outcome, not to teach the basic principles (precise algebra, motion physics, or blue and blue light formulation), so simplification was appropriate.

8. Balance user suggestions, with observation and participatory processes

Some researchers worry that users may ask for features that are impossible or impossible. In some cases, users request only additional changes, unaware of the potential for major changes.

Another concern is that users may request more flexibility than is required or desirable. In general, a structure with well-chosen parameters is more effective than constructed with perfectly adjustable parameters. Designers all like to give control to users - but only when control will make a big difference to their experience. Sometimes users can request multiple "local" or specific items without adequately observing the design requirements of the "earth" design, which has resulted in a "kitchen sink" effect (such as "throw everything except the kitchen sink").

One way to ask users to suggest features is to see users interact with prototypes, and do what they want (and do not want) in their actions. Often, their actions speak louder than their words. It is often seen when users are frustrated, even if they do not express their frustration. When designers look at users they often make the same "mistake" of a certain type, sometimes they are able to update the software to behave the way users expected.

Some design teams have emphasized participatory approaches that involve users who have worked tirelessly for a long time as team members [Muller 2002]. These user representatives may need to invest heavily in learning more about design issues and opportunities. The evidence is strong that projects that engage users in the construction process lead to greater acceptance by the wider user community. Greater acceptance may be due to more accurate data and data from users, as well as ego investment and empathy generated by having user representatives as part of the development team.

9. Iterate, Iterate - Then Iterate Again

Another common goal of user interface design is the importance of iterative design using prototypes. In the design of technical support tools, the designers put a lot of emphasis on "tinkerability" - they want to

encourage users to discuss building materials, experiment with many alternatives, move midway through the process, sort items and create new types.

While developing new technologies, they are constantly criticizing, correcting, correcting and reviewing. The ability to develop fast prototypes is very important in this process. News boards are not enough; there will be active prototypes. Early prototypes don't need to be fully functional, just good enough for designers (and their users) to play with, experiment, and talk about them.

One thing most project planners and designers do not have enough solid knowledge for is the idea of "iterate just enough to do the next test." It is important to be able to speed up

- view users with a given iteration of the program
- Adjust design changes as a result of that response
- apply (and check performance) those changes to the tool

then repeat the process. In a perfect world, one can circle this loop once daily or weekly.

In his book *Serious Play*, Michael Schrage argues that prototypes are especially useful as a starting point for discussion, reviving conversations between designers and potential users [Schrage 1999]. This could not have been more true. The best conversations (and good ideas) happen when designers start playing with new prototypes - and see users playing with prototypes. As soon as the designers started playing (and talking) in one style, they started thinking about the next build. This process requires both the right tools (to support the rapid development of new prototypes) and the right mindset (the willingness to release a particular type after creating it).

10. Design for Designers

By building, one becomes creative. LEGO kits certainly empower users to express themselves creatively, but new computer-based support tools continue, enabling users to create not only static, constructive but also powerful, interactive objects: music, video, animation, interactive. Software-based technical support tools have the added advantage that (and their emerging products) can be widely distributed at low cost.

The analogy with LEGO kits also raises an important contradictory example. While it is possible to use these kits to create a variety of building materials, many children create a model that is suggested in the package, or perhaps a little different, and nothing else. This is like using a paint kit with numbers. These kits clearly promote "handmade work," but they don't work well (compared to traditional LEGO kits) in promoting creative thinking. The aim is to develop technologies that not only involve users in creating new objects, but also encourage them (and support them) to explore ideas under their design.

Another example as a model would be paper and pencil as a tool used by artisans, such as architects. Hand-painted drawings and drawings have been found to be important in the architectural exhibition [Arnheim 1969] [Lawson 1994]. Not only the sketches are drawn, but the drawing process helps designers engage in visual interaction [Schoen 1983]. Tools designed, developed and tested to support such drawing processes across multiple domains, including architecture [Gross 1996], software interface and Web page [Landay 2001], and industrial construction [Hoeben 2005]. The basic features of imaginary search engines have been identified and applied to non-graphic design environments, such as the writing and production of movies, using a two-dimensional landscape as a representative [Yamamoto 2005].

Writing software is a work of art, so another goal should be to develop tools that help people to write software intelligently. This is repeated in some way; because designers are looking for creative support with software writing tools that help them build design thinking tools for other tasks as well. The tools they build for themselves should therefore support all of the guidelines discussed above. Therefore, they should follow good software engineering practices so that the tools themselves can be easily modified.

IV. RESULTS AND DISCUSSION

While predicting styles is not a straightforward science, based on the information collected in this study there are a few trend indicators that can be reduced to date:



- The emergence of Big Data (large and complex data sets produced by multiple sources) is becoming increasingly relevant to many large businesses and governments not only for analytical purposes, but also for the purpose of conveying complex information to the general public in an easy-to-understand way.
- As more designers move to use evidence-based formats and HCD techniques in their work, their need to learn relevant research skills will grow. Universities have begun to introduce design programs that are researched and have reduced design programs that focus exclusively on technical design skills. PhD programs in design will continue to grow as the need for skilled designers in a variety of research methods and able to work on large and complex projects will increase in demand.
- One important problem with the development of design thinking tools is how to test them. How do we know the tool is useful? Human-computer experts are used to measure the efficiency and effectiveness of tools, but how do we measure if it supports intelligence? It remains open to how you can measure the extent to which a tool promotes creative thinking. While the robustness of controlled subjects makes it a traditional method of scientific research, long-term studies with active users for weeks or months seem to be a valid way to gain a deeper understanding of what is helpful (and why) for skilled people [Seo 2005].

While researchers cannot be fooled into believing that testing tools is an easy task, they should also believe that the potential impact of advanced tools can be significant in promoting and reviving innovation.

V. CONCLUSION AND FUTURE SCOPE

The examples highlighted in this study suggest that the world today needs design thinking tools which encourage individuals to be culturally sensitive, inquisitive, and capable of thinking both intellectually (logically) and later (accurate thinking). In addition, these tools should foster individuals to communicate clearly and confidently in visual, oral and written ways. They also need to be able to analyze problems and organize information related to how people interact with information, technology, knowledge, culture, places, objects, and society. Their work should focus on design that is meaningful to specific individuals and situations, rather than producing works of art. They must be curious about the needs of others, not just them.

All of this suggests that the social structure that governs the meaning of the design has changed, and the term "design" now means an evidence-based approach, a way of focusing people on their purpose to help businesses, communities and individuals.

REFERENCES

1. Agre, P. E. (1994). Surveillance and capture: Two models of privacy. *The Information Society*, 10(2), 101–127.
2. Allen, J. (2016). *Topologies of power. Beyond territory and networks*. Routledge.
3. Bratton, B. (2015). *The Stack: On software and sovereignty*. MIT Press.
4. Bucher, T. (2018). *If...then: Algorithmic power and politics*. Oxford University Press.
5. Castañeda, L., & Selwyn, N. (2018). More than tools? Making sense of the ongoing digitizations of higher education. *International Journal of Educational Technology in Higher Education*, 15(1).
6. Decuyper, M. (2019a). Open Education platforms: Theoretical ideas, digital operations and the figure of the open learner. *European Educational Research Journal*, 18(4), 439–460.
7. Decuyper, M. (2019b). Researching educational apps: ecologies, technologies, subjectivities and learning regimes. *Learning, Media and Technology*, 44(4), 414–429.
8. Decuyper, M. (2019c). STS in/as education: where do we stand and what is there (still) to gain? Some outlines for a future research agenda. *Discourse: Studies in the Cultural Politics of Education*, 40(1), 136–145

9. Dieter, M., Gerlitz, C., Helmond, A., Tkacz, N., Vlist, F., Der, V., & Weltevrede, E. (2018). Store, interface, package, connection : Methods and propositions for multi-situated app studies. CRC Media of Cooperation Working Paper Series No 4.
10. Drucker, J. (2020). Visualization and Interpretation: Humanistic Approaches to Display. MIT Press. *Journal of New Approaches in Educational Research*, 10(1)
11. Mathias, Decuypere The Topologies of Data Practices: A Methodological Introduction Fedorova, K. (2020). Tactics of Interfacing. Encoding Affect in Art and Technology. MIT Press. Goriunova, O. (2019). The Digital Subject: People as Data as Persons. *Theory, Culture & Society*, 36(6), 125–145.
12. & Ruppert, E. (2020). Population Geometries of Europe: The Topologies of Data Cubes and Grids. *Science, Technology, & Human Values*, 45(2), 235–261.
13. Gulson, K. N., Lewis, S., Lingard, B., Lubienski, C., Takayama, K., & Webb, P. T. (2017). Policy mobilities and methodology: a proposition for inventive methods in education policy studies. *Critical Studies in Education*, 58(2), 224–241.
14. Gulson, K. N., & Sellar, S. (2019). Emerging data infrastructures and the new topologies of education policy. *Environment and Planning D: Society and Space*, 37, 350–366.
15. Hartong, S. (2020). The power of relation-making: insights into the production and operation of digital school performance platforms in the US. *Critical Studies in Education*, 00(00), 1–16.
16. Hartong, S., & Förschler, A. (2019). Opening the black box of data-based school monitoring: Data infrastructures, flows and practices in state education agencies. *Big Data & Society*, 6(1),
17. Lash, S. (2012). Deforming the Figure: Topology and the Social Imaginary. *Theory, Culture & Society*, 29(4-5), 261–287.
18. Latour, B. (1986). Visualization and cognition: Thinking with eyes and hands. *Knowledge & Society*, 6, 1–40. Retrieved from [http://hci.ucsd.edu/10/readings/Latour\(1986\).pdf](http://hci.ucsd.edu/10/readings/Latour(1986).pdf)
19. Law, J. (2004). *After Method: Mess in Social Science Research*. Psychology Press.
20. Lewis, S. (2020). Providing a platform for “what works”: Platform-based governance and the reshaping of teacher learning through the OECD’s PISA4U. *Comparative Education*, 56(4).
21. Lewis, S., & Hardy, I. (2017). Tracking the Topological: The Effects of Standardised Data Upon Teachers’ Practice. *British Journal of Educational Studies*, 65(2), 219–238.
22. Light, B., Burgess, J., & Duguay, S. (2018). The walkthrough method: An approach to the study of apps. *New Media and Society*, 20(3), 881–900.
23. Lindh, M., & Nolin, J. (2016). Information We Collect: Surveillance and Privacy in the Implementation of Google Apps for Education. *European Educational Research Journal*, 15(6), Lury, C., & Day, S. (2019). Algorithmic Personalization as a Mode of Individuation. *Theory, Culture & Society*, 36(2), 17–37.
24. Mathias, Decuypere The Topologies of Data Practices: A Methodological Introduction Lury, C., Fensham, R., Heller-Nicholas, A., & Lammes, S. (2018). *Routledge Handbook of Interdisciplinary Research Methods*. Routledge.
25. Lury, C., Parisi, L., & Terranova, T. (2012). Introduction: The Becoming Topological of Culture. *Theory, Culture & Society*, 29(4-5), 3–35.
26. Lury, C., Tironi, M., & Bernasconi, R. (2020). The Social Life of Methods as Epistemic Objects: Interview with Celia Lury. *Diseña*, 16, 32–55.
27. Lury, C., & Wakeford, N. (2012). Introduction: A perpetual inventory. *Inventive Methods* (pp. 15–38). Routledge.

28. Martin, L., & Secor, A. J. (2014). Towards a post-mathematical topology. *Progress in Human Geography*, 38(3), 420–438.
29. Piattoeva, N., & Saari, A. (2020). Rubbing against data infrastructure(s): methodological explorations on working with(in) the impossibility of exteriority. *Journal of Education Policy*, 00(00), 1–21.
30. Plantin, J. C., Lagoze, C., Edwards, P. N., & Sandvig, C. (2018). Infrastructure studies meet platform studies in the age of Google and Facebook. *New Media and Society*, 20(1), 293–310.
31. Prince, R. (2017). Local or global policy? Thinking about policy mobility with assemblage and topology. *Area*, 49(3), 335–341.
32. Ratner, H. (2019). Topologies of Organization: Space in Continuous Deformation. *Organization Studies*, 1–18.
33. Ratner, H., & Gad, C. (2019). Data warehousing organization: Infrastructural experimentation with educational governance. *Organization*, 26(4), 537–552.
34. Ratner, H., & Ruppert, E. (2019). Producing and projecting data: Aesthetic practices of government data portals. *Big Data & Society*, 6(2), 1–16.
35. Ruppert, E., Law, J., & Savage, M. (2013). Reassembling Social Science Methods: The Challenge of Digital Devices. *Theory, Culture & Society*, 30(4), 22–46.
36. Suchman, L. (2012). Configuration. In C. Lury & N. Wakeford (Eds.), *Inventive Methods: The Happening of the Social* (pp. 48–60). Taylor and Francis.
37. Thompson, G., & Cook, I. (2015). Becoming-topologies of education: deformations, networks and the database effect. *Discourse: Studies in the Cultural Politics of Education*, 36(5), 732–748.
38. Thompson, G., & Sellar, S. (2018). Datafication, testing events and the outside of thought. *Learning, Media and Technology*, 43(2), 139–151.
39. van de Oudeweetering, K., & Decuyper, M. (2019). Understanding openness through (in)visible platform boundaries: a topological study on MOOCs as multiplexes of spaces and times. *International Journal of Educational Technology in Higher Education*, 16(1).
40. van de Oudeweetering, K., & Decuyper, M. (2020). In between hyperboles: forms and formations in Open Education. *Learning, Media and Technology*, Advance online publication, 1–18.
41. Williamson, B. (2017). Learning in the “platform society”: Disassembling an educational data assemblage. *Research in Education*, 98(1), 59–82.

AUTHORS PROFILE

Ayse received her MSc in Internet Studies in University of Oxford in 2006. She participated in various research projects for UN, Nato and the EU regarding HCI (human-computer interaction). She completed her doctorate degree in user experience design in Oxford while working as an adjunct faculty member at Bogazici University in her home town Istanbul. Ayse has also a degree in Tech Policy from Cambridge University. Currently, Ayse lives in Silicon Valley where she works as a visiting scholar for Google on human-computer interaction design.