Knowledge Maps for Building Conceptual and Transferable CSS Knowledge

Roxanna G. Wilcox

Department of Computer Science

Northwestern University

Supervisor

Haoqi Zhang Ph.D.

In partial fulfillment of the Honors requirements for the degree of

Bachelor of Arts in Computer Science

April 15, 2022

Acknowledgements

We would like to thank Haoqi Zhang, Gobi Dasu, Fardeem Munir, and the rest of Northwestern University's Design, Technology, and Research program for their guidance and feedback throughout this process. This work was also supported by an Undergraduate Research Grant from Northwestern University's Office of the Provost.

Abstract

Despite the abundance of online resources for learning CSS, novice web developers struggle to develop the expert intuition of choosing the best CSS technique to build a given layout. We present Knowledge Maps (KM), a tool that allows users to build transferable and conceptual knowledge of CSS techniques. By interactively exploring professional websites and by categorizing visual features of those sites and the relevant CSS techniques used to create them, KM users discover the relevant similarities, differences, and use cases of various CSS techniques in the process, developing the knowledge that characterizes experts. In an initial study where 6 users learned from conventional CSS tutorials and 7 users learned through KM, KM users identified the appropriate CSS to build a set of layouts with a 48% increase in accuracy as compared to a 15% increase for non-KM users and showed development of transferable knowledge through KM. In a second study where 6 users learned through KM, 5 out of the 6 users were able to build in-depth knowledge maps that show generalized knowledge of CSS layouts.

Contents

1 Introduction **1**

2 Related Work **4**

3 Initial System Description **7**

3.1 The Explore Phase 7

3.2 The Knowledge Map Phase 9

4 Initial User Study and Evaluation **10**

4.1 Setup 10

5 Initial Study Findings **12**

5.1 Results for RQ1 12

5.2 Results for RQ2 12

6 Initial Discussion **14**

6.1 Limitations 14

6.2 Revision Inspiration 15

7 Revised System Description **17**

7.1 The Explore Phase 17

7.2 The Knowledge Map Phase 20

8 Revised User Study and Evaluation **22**

8.1 Setup 22

9 Revised Study Findings **24**

9.1 Results for RQ1 24

10 Revised Discussion **26**

10.1 Limitations 27

10.2 Future Work 27

References **29**

Chapter 1

Introduction

As the world is increasingly run by software, there is higher demand for professional software developers. While there exists a growing number of novice web developers, there is a shortage of experts. This is because existing resources, such as online tutorials (Codecademy, Stack Overflow, and W3Schools), coding bootcamps, and computer science degrees, do not necessarily bridge the gap between novice and expert developers.

How Learning Works [1] argues that one important way in which experts' and novices' knowledge organizations differ is in the quantity and density of connections among the concepts, facts, and skills they know. In the context of CSS, novices understand explicit CSS syntax and are able to use specific CSS techniques in the context in which they were taught. Experts, along with understanding CSS syntax, have deep conceptual knowledge of the various CSS techniques and their use cases. Experts, like novices, understand the syntax but are also able to draw connections across the different ways this syntax is used to produce various visual outcomes in web development.

As a result of their years of experience and exposure to many use cases and techniques, experts develop an intuition for understanding which CSS techniques to use when and which techniques are appropriate to build out the layout of any given website. Novices are not able to build these complex connections nor develop this intuition from completing tutorials which only teach CSS given in one context. Aside from tutorials, users can utilize Chrome Developer Tools to understand the CSS techniques that are responsible for any given professional website's layout. However, Chrome Developer Tools are much too complex for beginners to utilize and gain knowledge from effectively [2]. Although Chrome Developer Tools provides the capability to examine a large quantity of professional websites to build a user's exposure to the ways CSS can be used to produce different layouts, the lack of scaffolding or guidance makes it difficult for this to be an effective learning tool on its own.

There have been contributions in this space from a previous iteration of this project. In the previous iteration, it was shown that (1) users were able to better understand how specific CSS properties affected the visual features of an example by tinkering with the interactive CSS properties; (2) code highlighting is effective in helping learners identify and form a basis for analyzing visual and code differences in two examples [3]. In this previous iteration, users were able to formulate a rationale for why the author in each example chose their professional CSS technique. Although they were able to identify why a technique was used in a particular use-case, they did not understand the tradeoffs of other CSS techniques that could produce a similar result. Therefore, the users understood the particular examples in context but did not translate this knowledge to a broader mental model for how each CSS technique can be used in general.

We present a new iteration of Knowledge Maps (KM), a platform to help learners understand individual CSS techniques and the various use cases associated with those techniques. The conceptual contribution of this research is that exploring various website examples and then grouping the visual attributes of those websites with the CSS techniques used to produce them creates a more generalized model of CSS knowledge that allows for conceptual understanding of the use cases and applications of each CSS technique.

The mechanisms utilized on KM include, as before, highlighted code properties and interactive CSS code in a new interface with expanded features, along with the addition of a format for creating a general knowledge map using matching and category creation. As stated in How Learning Works [1] and research from the National Research Council [4], how students organize knowledge influences how they learn and apply what they know, highlighting the importance of the easily digestible knowledge map.

During a user test, KM helped users create a general knowledge map for CSS grid and apply that general map to a set of specific website examples in order to identify the appropriate CSS grid techniques used to build each site with three times as much accuracy improvement as compared to users who used alternative methods of learning CSS grid. KM's mechanisms were useful in helping users identify key differentiating features between examples and turn those different features into categories for different visual outcomes. Users then could identify which CSS grid techniques were responsible for producing each visual outcome. They matched the identified visual outcomes with the corresponding CSS techniques producing those outcomes in their knowledge map, forming generalized knowledge that could be applied to examples different from the ones they studied in KM.

Chapter 2

Related Work

A primary goal of web developers in regards to CSS is to be able to build complex layouts utilizing CSS rather than learning any specific CSS technique on its own. One central challenge is to teach CSS grid in a way that the knowledge gained through using KM transfers to the act of building complex web layouts without the help of the tool. This leads to the central problem faced by most research in the learning sciences: the problem of transfer. Psychologist Robert Haskell [5], after conducting a vast review of the literature on transfer in learning, says, “Unfortunately, transfer is also something that, despite more than a century of intense work and research, has largely failed to occur in formal education.” Haskell suggests that transfer tends to be more difficult when our knowledge is limited and that as we delve deeper and gain more experience in a subject area, we are more likely able to apply our knowledge outside the narrow contexts in which it was originally learned. Applying this idea to Knowledge Maps, we propose a solution to overcome the problem of transfer through “directedness”: directly modifying CSS syntax and seeing the results of those modifications rather than passively reading a tutorial, giving users a deeper dive into how CSS techniques work. KM provides various website examples to learn from in order to give exposure to the different use cases of CSS grid, further expanding users’ knowledge of CSS grid.

Many current solutions exist to help novice developers understand CSS techniques. Tutorons [6] helps learners build contextual understanding by providing learners with in-line explanations of CSS properties that they may not be familiar with. Ply [2] identifies the relevance of specific CSS properties for a selected visual feature so users can isolate the relevant code from all of the source code, allowing users to focus on the relevant code without being distracted or overwhelmed by other CSS that is not directly affecting the visual outcome. These existing solutions help learners understand more about particular CSS techniques and their significance on a given visual outcome. However, neither are sufficient to help learners compose their learnings into generalized CSS knowledge that can be applied to building and understanding complex layouts.

Lastly, KM instructs users to build a knowledge map. A knowledge map is a bipartite graph mapping visual outcomes to the corresponding CSS grid techniques used to produce them (Figure 9.1). This is inspired by research from Schwartz and Bransford [7] arguing that by using contrasting cases while learning, learners can identify specific features differentiating each case to generate knowledge structures for conceptual understanding. Example website pairings that show the same CSS technique used in two contrasting layouts or the same layout produced by two different CSS techniques allow users to identify and create visual categories of those layouts in the map and then match the corresponding CSS techniques used to create them. This enables them to build conceptual knowledge about which CSS techniques can be used to produce which kinds of visual outcomes. Additionally, Marcia C. Linn and Michael J. Clancy [8] found that articulating and reflecting on what one has learned is critical to learning from examples. Building a knowledge map requires users to reflect on takeaways from the professional website examples they study with the tool.

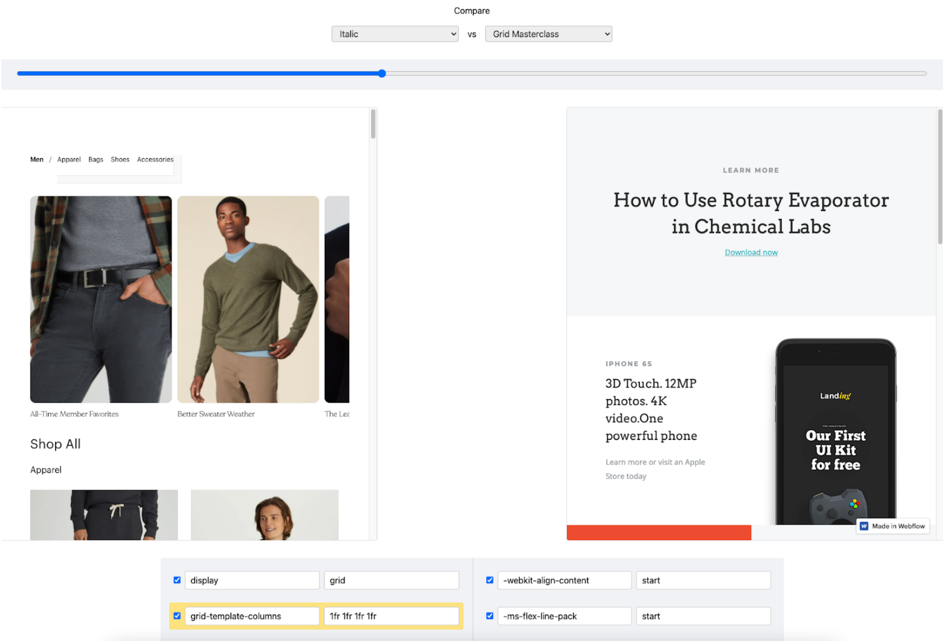
As explained in How Learning Works [1], how students organize knowledge influences how they learn and apply that knowledge. How Learning Works [1] argues that it is most effective to create a knowledge organization that matches the situation the knowledge is needed for. Because the users of KM aim to be able to identify the best technique for a given visual outcome and to build complex web layouts themselves, mapping visual outcomes to the possible CSS techniques used to produce them is a knowledge organization that closely aligns with the given situations in which the knowledge will be used. For example, if a user wants to build a website with a layout that contains static columns, having a knowledge map that contains a visual outcome category of static columns and matches that outcome category to the CSS techniques used to produce static columns is effective, because this knowledge organization directly matches the situation in which the knowledge is needed - using CSS techniques to achieve a particular visual outcome.

Chapter 3

Initial System Description

KM consists of two separate phases: the explore phase and the knowledge map phase.

3.1 The Explore Phase



Compare

Italic vs Grid Masterclass

Men / Apparel Bags Shoes Accessories
All-Time Member Favorites
Better Sweater Weather
The Lea
Shop All
Apparel

LEARN MORE

How to Use Rotary Evaporator in Chemical Labs

Download now

IPHONE 6S

3D Touch. 12MP photos. 4K video.One powerful phone

Learn more or visit an Apple Store today

Landing
Our First UI Kit for free
Made in Webflow

☑ display: grid

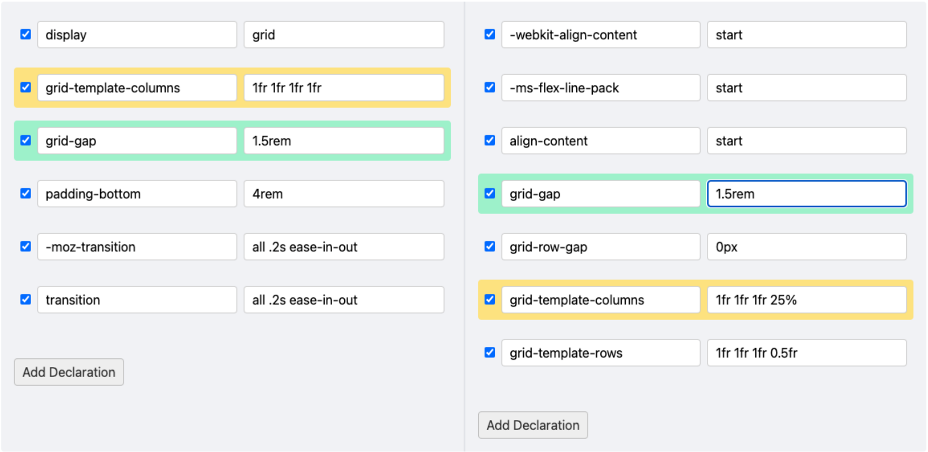
☑ grid-template-columns: 1fr 1fr 1fr 1fr

☑ -webkit-align-content: start

☑ -ms-flex-line-pack: start

Figure 3.1 The split-screen design to compare two examples at once.

To explore the sites, users choose two professional website examples from a set of five. These websites were chosen because they have similar layouts implemented by different CSS techniques or different layouts implemented by the same CSS techniques. Five websites were chosen to ensure enough variety and effective comparison. Users are able to resize the two selected website windows' using the sliding bar in order to examine the way the websites change when their window sizes are changed simultaneously. This allows them to see how the two websites' behavior may differ when performing the same action of window resizing, giving way to noticing different visual outcomes.



☑	display	grid
☑	grid-template-columns	1fr 1fr 1fr 1fr
☑	grid-gap	1.5rem
☑	padding-bottom	4rem
☑	-moz-transition	all .2s ease-in-out
☑	transition	all .2s ease-in-out

☑	-webkit-align-content	start
☑	-ms-flex-line-pack	start
☑	align-content	start
☑	grid-gap	1.5rem
☑	grid-row-gap	0px
☑	grid-template-columns	1fr 1fr 1fr 25%
☑	grid-template-rows	1fr 1fr 1fr 0.5fr

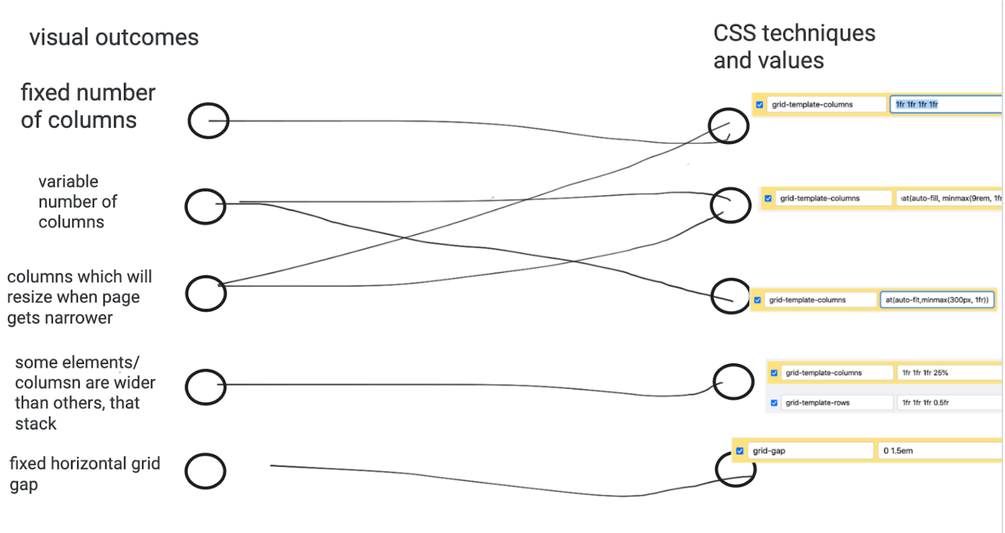
Figure 3.2 The code editor in the explore phase.

Users can inspect the relevant CSS code producing each website's layout in an interactive code editor (Figure 3.2). Here, users can change the values of the CSS techniques and see those changes rendered in the website windows. When edits are made to the editor by the user, the structured data is synthesized into valid CSS. Using the browser postMessage API, the example's output is updated. Javascript is added to the example website in order to receive modified CSS by the user and then apply that modified CSS to the website.

There are affordances for adding new CSS declarations to the parent element as well as editing existing declarations. Declaration blocks are highlighted with yellow if the same CSS property is used in both examples but the value of that property is different; blocks are highlighted with green if the same CSS property and the same value of that property is used in both examples. The highlighted code guides users to focus on the relevant similarities and differences between the two websites' CSS code.

3.2 The Knowledge Map Phase

Users are required to build out a knowledge map in the knowledge map phase. From exploring the sites, users create categories of visual outcomes they notice. Examples of visual outcomes are: (1) the number of columns changes based on the window size of the site; or (2) the number of columns stays the same regardless of window size. Along with creating visual outcome categories, users write the relevant CSS techniques used to produce these visual outcomes and match the visual outcomes to the relevant CSS techniques responsible for producing them by drawing a line connecting one or more visual outcome circles on the left to one or more CSS technique circles on the right. Users go back and forth between exploring the sites and building out their own knowledge map. A completed knowledge map by one of our study participants is included below:


visual outcomes
fixed number of columns
variable number of columns
columns which will resize when page gets narrower
some elements/ columsn are wider than others, that stack
fixed horizontal grid gap
CSS techniques and values

☑ grid-template-columns 1fr 1fr 1fr 1fr

☑ grid-template-columns ›at(auto-fill, minmax(9rem, 1fr

☑ grid-template-columns at(auto-fit,minmax(300px, 1fr))

☑ grid-template-columns 1fr 1fr 1fr 25%

☑ grid-template-rows 1fr 1fr 1fr 0.5fr

☑ grid-gap 0 1.5em

Figure 3.3 A completed knowledge map by one of our participants.

Chapter 4

Initial User Study and Evaluation

The initial user study seeks to answer the following questions:

1. Do users have a better understanding of the tradeoffs and use cases of CSS grid after using the KM tool as compared to alternative ways of learning CSS?

2. Does building out the knowledge map in the KM tool create conceptual understanding that allows users to apply their knowledge to new contexts?

4.1 Setup

We recruited 13 computer science students from Northwestern University who had basic knowledge of CSS. These students had all taken an introductory HTML/CSS/Javascript course, although some had more experience with CSS than others. We held a 32-minute study that consisted of a pre-test, an exploration and learning phase, and a post-test. The participants were randomly divided into two groups. Group 1 used the KM tool during the learning phase, while Group 2 used traditional methods of learning CSS grid.

Each participant in both groups completed the same pre-test (8 minutes): a survey testing their previous knowledge of CSS grid. They were asked to examine three professional websites not included in the tool that used CSS grid as the main element of their layout and describe the visual layout and the CSS techniques they believed to be producing the layout. These websites were chosen because they had similar layouts implemented by different CSS techniques or different layouts implemented by the same CSS techniques. We scored the participants' accuracy on the pre-test by comparing each participant's answers against our rubric. Our rubric outlined the CSS techniques responsible for each website's layout. Each contributing CSS technique or value was given a point value (either 1 or 2). The participant received the corresponding number of points for each CSS technique or value they correctly identified. The participant's score for each site was calculated to be the number of points they achieved over the total number of points possible for that site.

In the learning phase of the study (16 minutes), Group 1 used the KM tool while Group 2 acted as a control group, utilizing traditional methods of learning CSS grid. Participants in Group 1 utilized the explore phase and knowledge map phase described above. Participants in Group 2 used traditional methods of learning CSS grid in the allotted time, such as W3 Schools, Stack Overflow, exploring grid websites, and inspecting them with Chrome Developer Tools. They were told to focus specifically on CSS grid techniques. They did not build out a knowledge map.

In the post-test (8 minutes), participants in Group 1 and Group 2 were presented with the same survey as in the pre-test. They were shown the same three websites, none of which were presented in the KM tool, and asked to identify the CSS techniques responsible for producing each website's layout. Both groups were allowed to use their resources when completing the post-test: the knowledge map for Group 1 and online sources for Group 2. Post-test results were scored using the same rubric as the pre-test. Each participant ended up with 6 scores: 3 scores for the 3 pre-test sites and 3 scores for the same 3 post-test sites. In order to receive more data on the participants' understanding, we asked for verbal feedback after the study as well. To analyze the test results, we summed up each participants' scores for the 3 pre-tests and the 3 post-tests.

Chapter 5

Initial Study Findings

5.1 Results for RQ1

By exploring and tinkering with examples side by side and building out a knowledge map using KM, participants gained a better understanding of the tradeoffs and use cases of CSS grid than when using alternative methods of learning CSS. The average increase in score from pre-test to post-test in the control group was 15%, whereas the average increase in score for the group that used KM was 48%. This validates our hypothesis that KM is more effective at teaching the basics and use cases of CSS techniques, specifically CSS grid, than conventional methods of learning CSS. In a post-study interview, one participant shared about KM that he found it “really interesting how [he] could learn about many different CSS attributes at once. [He] especially liked being able to play around with the CSS values and see what visual change resulted from it.” Another participant shared that through KM, he “found a really clear workflow of finding the unique CSS in the tool and copying what that usage was into the [knowledge map], and reflecting on how that connected to the unique visual characteristics [he] was seeing.”

5.2 Results for RQ2

By guiding participants to build out a knowledge map, participants were able to form transferable knowledge about which visual outcomes

could be produced by which CSS techniques, going beyond making conclusions about a particular example and forming conceptual knowledge. The group that used KM had a 48% increase in score when asked to identify the relevant CSS code producing the layouts of a group of websites. The group of websites that the participants were tested on was different from the group websites they studied in the KM tool, proving that the knowledge they gained from the tool was able to be transferred to a completely different set of examples and is thus transferable. One participant who scored particularly high in the post-test shared that "when [he] started connecting [visual outcomes] to multiple properties, [he] realized that there were different ways of controlling a property, like grid spacing... this helped [him] see patterns and multiple points of similarity across examples." We noticed that students who had more in-depth knowledge maps, like his, performed better on the post-test, pointing to the effectiveness of an in-depth knowledge map in building transferable and conceptual knowledge.

Chapter 6

Initial Discussion

Analyzing the individual results, we noted that the lowest scoring users only mentioned that they would use CSS grid to achieve certain layouts without specifying which particular syntax they would use to achieve a certain outcome. Many learners did not grasp the additional layers of complexity within CSS grid techniques. For example, the grid specification introduces the repeat function, which takes possible values of: auto-fill, auto-fit, and others. The KM platform did not have processes to nudge users to explore these individual techniques more closely. The tool was best utilized by people who had some prior experience with CSS but not grid specifically. Similarly, people who were unable to extract value from the tool had very limited experience with CSS, which indicated a failure in recruiting appropriate users for the study. One significant finding was that there was a strong correlation between users who performed well on the post-test and users who were able to produce more in-depth knowledge maps, leading us to believe that the ability to produce a rich knowledge map goes hand-in-hand with obtaining transferable CSS knowledge and applying that generalized knowledge to new examples. Only one user was able to produce a significant knowledge map.

6.1 Limitations

Although the study results suggest that learners are able to grasp the basics of CSS grid using KM, the analysis does not show that participants were able to learn particular CSS grid techniques deeply. For example, no one in the study fully grasped how the fr fractional units introduced by CSS grid work, which was a relevant contributor to the layout of one of the websites in the tool.

The study had a limited number of users. It is possible that the KM tool's specific interfaces may fit certain types of learners, such as visual learners, more than others. Because there were a limited number of users, it is possible that results were more positively skewed towards learners whom the tool's style favored.

Additionally, the example websites in the KM tool used the same or similar CSS grid techniques to the ones in the pre and post-tests. In this way, the tool guided users to focus on these specific techniques. Users in Group 2, the control group, had less guidance on which CSS grid techniques to focus on and focused on CSS grid as a whole.

6.2 Revision Inspiration

One of the limitations of KM in this iteration is that there is no strong integration between the explore phase and the knowledge map phase. Users extract value from the explore phase without any support or guidance from the knowledge map phase. As such, users focus on the most obvious differences between techniques and not the more subtle differences between them. While users were able to extract value from the explore phase, they had trouble translating their gained knowledge to the knowledge map phase without guidance.

In addition, we hope to expand the KM tool to include more websites so users learn more techniques than just CSS grid. By adding another CSS technique, such as flex, users can explore similar layouts that are produced by different CSS techniques (grid vs. flex) to gain a better understanding of the tradeoffs and use cases of each technique, especially techniques that can implement similar layouts. Additionally, adding more CSS techniques will allow them to gain full knowledge of CSS rather than only knowledge of CSS grid.

Although our initial iteration of KM did provide successful findings that support our hypothesis, these limitations are worth addressing in a new iteration. The limitations we want to address in a revised iteration of KM are: (1) expanding beyond just CSS grid to include other CSS techniques; and (2) further integrating the explore phase and the knowledge map phase in hopes of producing more in-depth knowledge maps. With this in mind, we set out to build a system with a focus on producing stronger knowledge maps in hopes of furthering our research.

Chapter 7

Revised System Description

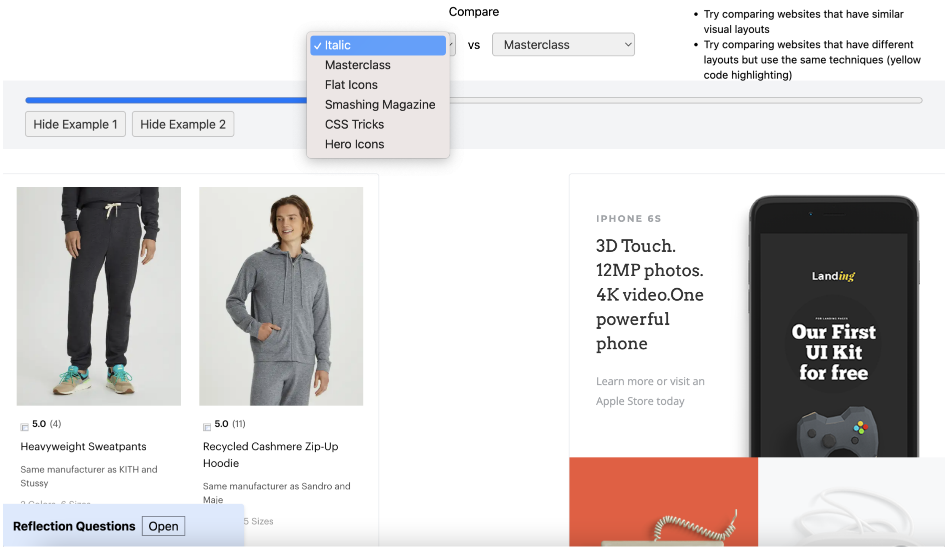
The revised KM consists of the same two phases: the explore phase and the knowledge map phase.

7.1 The Explore Phase

The layout of the explore phase looks largely the same but has a few key differences.

1. We have expanded the pool of websites to include websites with layouts built with CSS flex as well as CSS grid (Figure 7.1). There are three websites built with CSS grid and three websites built with CSS flex. Within the six sites, there is: a pair of sites, one built with CSS grid and one built with CSS flex, that achieve the same visual layout; a pair of sites, both built with CSS grid, that achieve different visual layouts; a pair of sites, both built with CSS flex, that achieve different visual layouts. These example websites were chosen based on research from Schwartz and Bransford [7] that argues for using contrasting cases while learning in order for learners to identify specific features differentiating each case. We hope that the contrasting layouts built with the same CSS technique allow users to see how the same CSS technique can produce different visual outcomes, and we hope that the contrasting techniques (grid vs. flex) producing similar layouts allows users to see how different CSS techniques can produce the same visual outcome.

2. There are hints in the top right corner to guide users when creating pairings (Figure 7.2). Many of the users in the initial study struggled to create visual



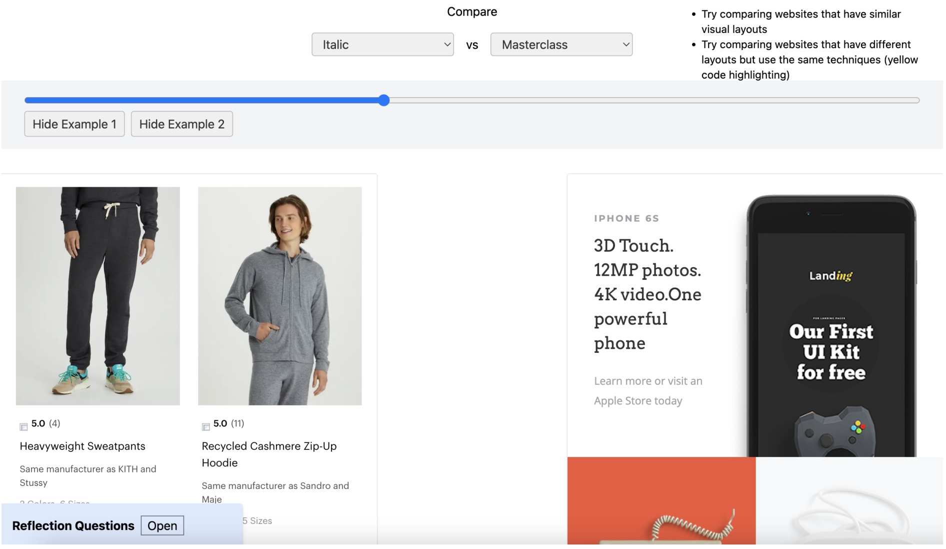
- Try comparing websites that have similar visual layouts
- Try comparing websites that have different layouts but use the same techniques (yellow code highlighting)

Compare
✓ Italic
Masterclass
Flat Icons
Smashing Magazine
CSS Tricks
Hero Icons
vs
Masterclass
Hide Example 1
Hide Example 2
5.0 (4)
Heavyweight Sweatpants
Same manufacturer as KITH and Stussy
5.0 (11)
Recycled Cashmere Zip-Up Hoodie
Same manufacturer as Sandro and Maje
5 Sizes
Reflection Questions
Open
IPHONE 6S
3D Touch. 12MP photos. 4K video.One powerful phone
Learn more or visit an Apple Store today
Landing
Our First UI Kit for free

Figure 7.1 The revised explore phase with a greater number of websites to explore.

outcome categories in their knowledge map. One way to begin coming up with visual outcome categories is to notice a visual outcome that exists on two different websites. To probe users to create this basic category, we provide the first hint, which says: *Try comparing websites that have similar layouts.* We hope that by comparing websites with similar layouts, users will see a common visual outcome in those similar layouts and in turn create a visual outcome category. The second hint says: *Try comparing websites that have different layouts but use the same techniques (yellow code highlighting).* This hint guides users to compare websites with two different layouts, in hopes that they will create two different visual outcome categories from those two layouts. We emphasize the code highlighting to guide users to pay attention to the yellow highlighted code. Yellow highlighting means that the same CSS technique is used in both websites but that technique takes on a different value in each of the two sites. We hope that this guides users to focus on that same CSS technique taking on two different values to produce two visual outcomes.

3. To guide users to begin thinking about their knowledge map while exploring the websites, we implemented reflection questions to answer while exploring the websites (Figure 7.2). The reflection questions are located in the bot-



- Try comparing websites that have similar visual layouts
- Try comparing websites that have different layouts but use the same techniques (yellow code highlighting)

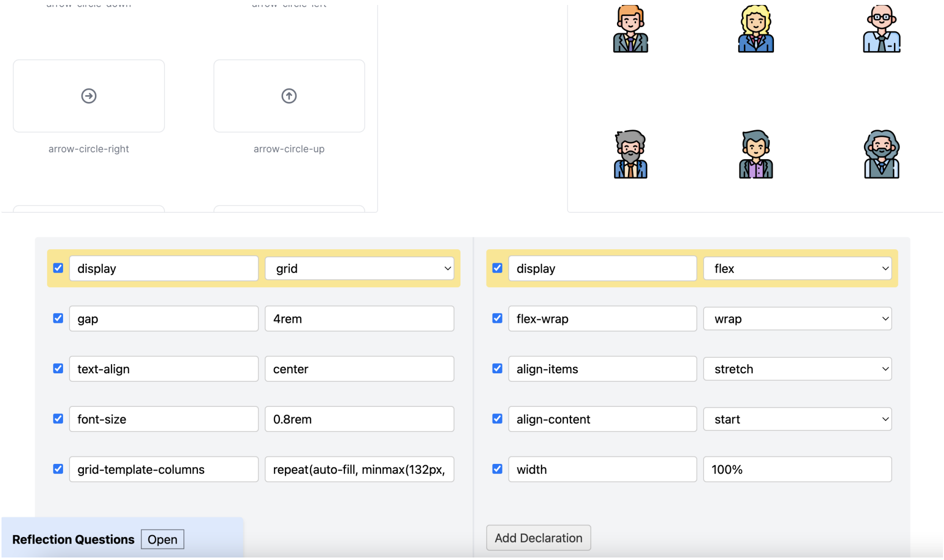
Compare
Italic
vs
Masterclass
Hide Example 1
Hide Example 2
5.0 (4)
Heavyweight Sweatpants
Same manufacturer as KITH and Stussy
5.0 (11)
Recycled Cashmere Zip-Up Hoodie
Same manufacturer as Sandro and Maje
5 Sizes
Reflection Questions
Open
IPHONE 6S
3D Touch. 12MP photos. 4K video.One powerful phone
Learn more or visit an Apple Store today
Landing
Our First UI Kit for free

Figure 7.2 The revised explore phase with hints at the top right and reflection questions at the bottom left.

tom left of the tool and can be opened and closed as desired. The reflection questions are: (1) *Find surprising differences between cases that appear similar, and find surprising similarities between cases that appear different*; (2) *How would you describe site 1's layout? What CSS code is producing this layout?*; and (3) *How would you describe site 2's layout? What CSS code is producing this layout?* Question 1 is based on research from Cognitive Flexibility Theory in the Oxford Handbook of Expertise [9], which argues that students should reflect on the nuances of two cases in order to build expert understanding more rapidly. Questions 2 and 3 aim to probe users to think about the relevant techniques producing each visual outcome in order to produce a visual outcome-CSS technique pairing in their knowledge map. Users are also given a document with the reflection questions pasted in so they have somewhere to write down their thoughts.

4. The final update to the explore phase is the expansion of the CSS code editor. As before, users can modify the values of various CSS techniques. However, many users did not have enough familiarity to know which values were able to be used for a given CSS technique, hindering their ability to tinker with the code. As a result, we added a dropdown menu of suggested values for relevant CSS techniques, so users can see what their options are when changing a value. In addition to providing options, we put the dropdown

menus on the most relevant CSS techniques for users to explore, further probing them in the right direction of which CSS techniques to play with. We also built out the code-disabling feature, which allow users to disable a CSS technique by unchecking its box. This allows the user to see how the layout changes when that CSS technique is no longer used, highlighting the effects that a given CSS technique has on the website's layout. The expansion of the code editor provides guidance for users to make meaningful conclusions about a given CSS technique's effect on a website's visual layout and also guidance to learn the CSS techniques more deeply.



☑ display grid

☑ gap 4rem

☑ text-align center

☑ font-size 0.8rem

☑ grid-template-columns repeat(auto-fill, minmax(132px,

☑ display flex

☑ flex-wrap wrap

☑ align-items stretch

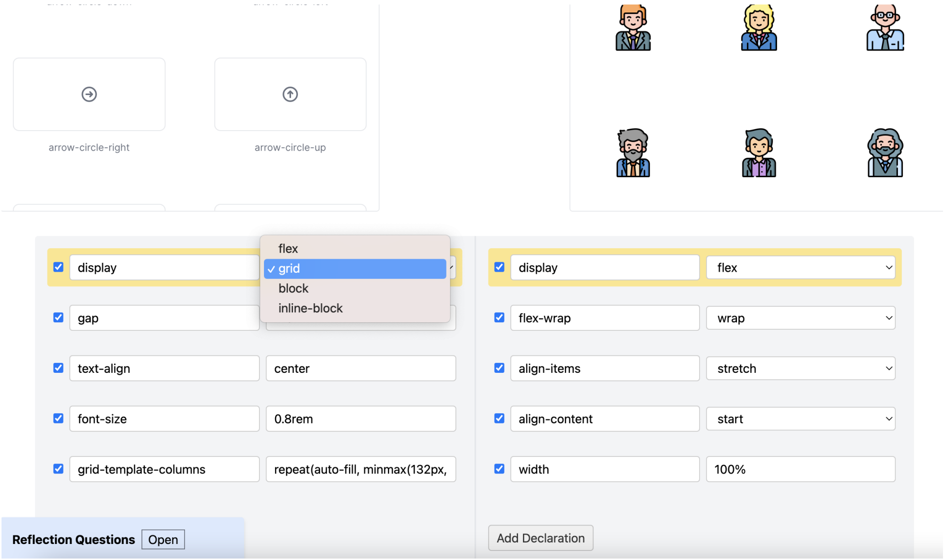
☑ align-content start

☑ width 100%

Figure 7.3 The expanded code editor in the revised explore phase.

7.2 The Knowledge Map Phase

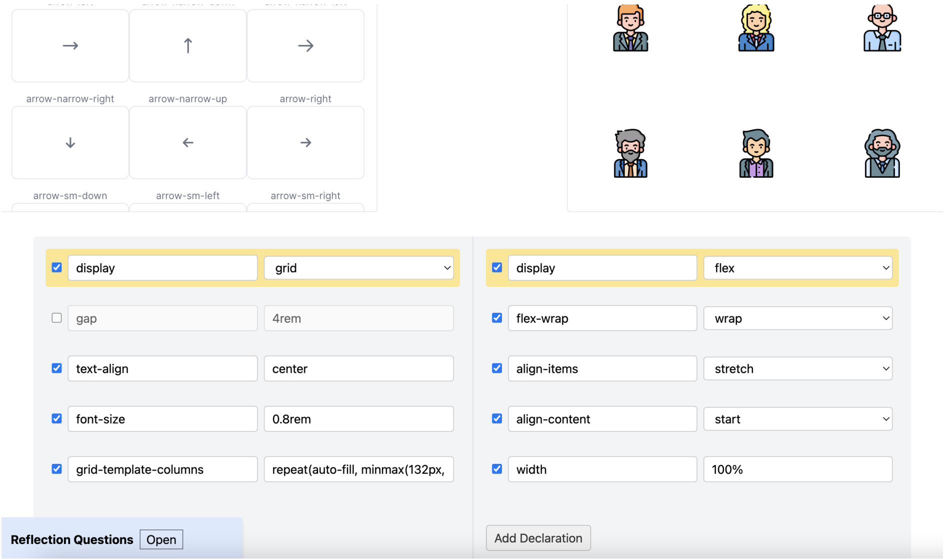
The knowledge map phase remains the same as before.



☑ display | grid
☑ gap |
☑ text-align | center
☑ font-size | 0.8rem
☑ grid-template-columns | repeat(auto-fill, minmax(132px,

☑ display | flex
☑ flex-wrap | wrap
☑ align-items | stretch
☑ align-content | start
☑ width | 100%

Figure 7.4 The dropdown menu of value options in the revised explore phase.



☑ display | grid
☐ gap | 4rem
☑ text-align | center
☑ font-size | 0.8rem
☑ grid-template-columns | repeat(auto-fill, minmax(132px,

☑ display | flex
☑ flex-wrap | wrap
☑ align-items | stretch
☑ align-content | start
☑ width | 100%

Figure 7.5 The disable feature and its effects on the left website in the revised explore phase.

Chapter 8

Revised User Study and Evaluation

The revised user study seeks to answer the follow question:

1. Can we guide users to build out more in-depth knowledge maps that show generalized understanding of CSS techniques and their use cases?

8.1 Setup

We recruited 6 computer science students from Northwestern University who had basic knowledge of CSS. As before, these students had all taken an introductory HTML/CSS/Javascript course, although some had more experience with CSS than others. We held a 53-minute study where users did not complete a pre or post-test and instead only worked with the KM tool. The first ten minutes of the study were spent explaining the tool while the next 43 minutes were spent working with the tool to build out their knowledge map.

Knowledge maps can be subjective to score, so we created a rubric to score them as consistently as possible while still honoring the fact that different people can create different but correct knowledge maps. In our rubric, we award 1 point for identifying a specific visual outcome category, which we define as identifying a singular layout component or layout behavior. An example of a specific visual outcome category is: single row without wrapping - all children of that row taking equal amount of space. We award 1 point for identifying a correct match between a visual outcome and its corresponding CSS technique or technique-value pair. For example, the visual feature of "single row without wrapping" would correspond to display: grid and/or display: flex. If someone points out two properties that could be responsible, we award 2 points. We take away 1 point for each incorrect match between a visual outcome and its corresponding CSS technique or technique-value pair. For the visual feature of "single row without wrapping", display: grid would not be appropriate if coupled with grid-template-rows specifying multiple rows or grid-template-columns, which implicitly specify wrapping should occur. A user's total score is the total number of points they receive.

Chapter 9

Revised Study Findings

9.1 Results for RQ1

By adding features to guide users toward examining the relevant aspects of websites and their CSS code and nudging them towards category creation, KM users were able to build in-depth knowledge maps that showed generalized knowledge of CSS techniques and their use cases. We define an in-depth knowledge map to have a total score of at least 8. Our six users had an average score of 10 with a maximum score of 15 and a minimum score of 4. Five out of the six users were able to build in-depth knowledge maps with a score of at least 8.5. This validates our hypothesis that with more guidance from the explore phase, KM can guide users to build in-depth knowledge maps that show generalized understanding of CSS techniques and their use cases.

As seen below, the user created specific visual outcome categories on the left side of their knowledge map (Figure 9.1). Prompted by window resizing in the explore phase of KM, she noticed visual outcome behavior such as "items squish together when you decrease window size" and "number of columns decreases as you decrease window size," although she did not identify all of the techniques producing this advanced visual outcome behavior. She also noticed trivial visual outcomes such as "items spaced out horizontally." Prompted by tinkering with the dropdown menu of suggested values in the explore phase of KM, the user investigated flex wrap vs. flex nowrap. She concluded that flex nowrap puts the items next to each other in a line while flex wrap puts the items next to each

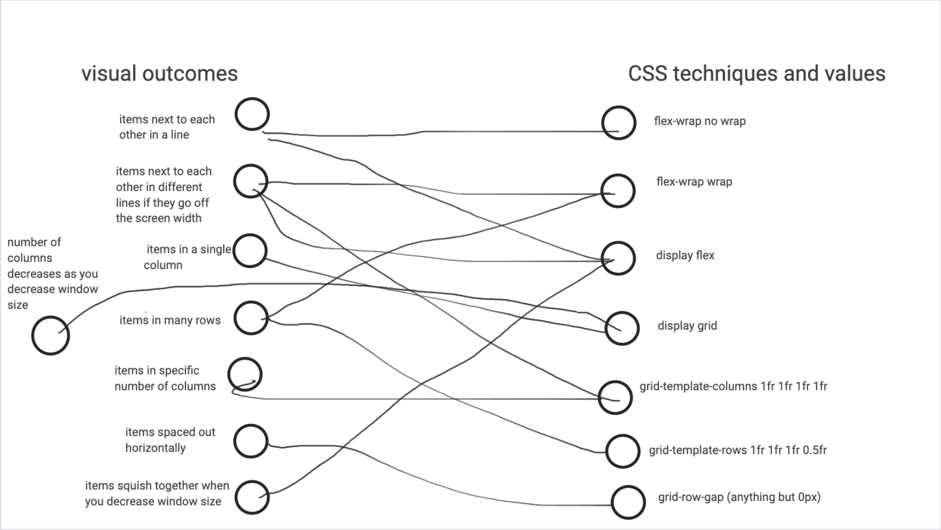

visual outcomes
CSS techniques and values
items next to each other in a line
items next to each other in different lines if they go off the screen width
number of columns decreases as you decrease window size
items in a single column
items in many rows
items in specific number of columns
items spaced out horizontally
items squish together when you decrease window size
flex-wrap no wrap
flex-wrap wrap
display flex
display grid
grid-template-columns 1fr 1fr 1fr 1fr
grid-template-rows 1fr 1fr 1fr 0.5fr
grid-row-gap (anything but 0px)

Figure 9.1 A knowledge map from one user in the revised study.

other with items spilling over onto the next lines if they go off of the screen width, an important and correct distinction that shows a generalized understanding of flex wrap and flex nowrap. After the user test, this user commented, “I liked the knowledge map part. I think the [explore phase] pushed me to figure out the way it worked, or else I wouldn’t have been able to figure out the knowledge map,” validating the necessity of the new interface features in guiding users to build out their knowledge map. Despite some lacking connections, we consider this an in-depth knowledge map due to the quantity and quality of visual outcome categories and the quantity of correct connections between visual outcome categories and the CSS techniques producing them.

Chapter 10

Revised Discussion

Analyzing the knowledge maps created by our users, we noticed that users who had some familiarity with the tool tended to perform better than those who were seeing the tool for the first time. Several of the users in the revised study had used the tool before in a previous iteration (with different websites) so were familiar with how Knowledge Maps works. Users that were familiar with Knowledge Maps built more in-depth knowledge maps than users who were not, pointing to the complexity of the tool and the learning curve of getting used to the tool. This leads us to believe that increased familiarity with the tool will cause an improvement in the tool's efficacy. Users who used the tool for the first time likely did not reach their full potential when exploring the websites and building out their knowledge maps.

The significance of having an in-depth knowledge map is that it reflects the type of knowledge that characterizes experts, as described in How Learning Works [1]. The knowledge map represents connections among the concepts, facts, and skills that users learn from the tool. The facts are the specific CSS syntaxes that produce particular visual outcomes. The concepts are the ability to categorize multiple techniques that produce the same visual outcome or multiple visual outcomes that can be produced by the same technique. The skills gained are the ability to extract these categories and visual outcome-CSS technique pairings from professional websites using the guiding interface features. How Learning Works [1] argues that these connections characterize experts, and these connections are at the heart of building out the knowledge map. Therefore, the act of building out the knowledge map helps users learn expert techniques.

10.1 Limitations

Although our argument for the reflection questions succeeded in that the questions were ultimately helpful in probing users to explore the relevant aspects of the websites, many users were frustrated by the way the reflection questions were incorporated into the tool. Users commented that it was frustrating to go back and forth between exploring the sites and writing in the reflection document. The act of going back and forth between several different mediums added to the complexity of an already complex tool. The complexity of the tool should be addressed in a future iteration to make the process simpler and more fluid.

A scalability limitation is that the Knowledge Maps tool is not generalizable for any website. Each professional site that is included in the tool is slightly modified to reduce clutter and help users focus on the main layout of the website. To achieve this, ads and other distracting sidebars are removed from the website. Additionally, the values in the dropdown menu of suggested values for each CSS technique are put in manually. Because these features have to be manually altered for each website, the tool is not automated to be scalable for a large number of websites.

In an ideal study, the KM tool would be new for all users, so there would not be inconsistencies in users' familiarity with the tool. However, this was not possible given the quantity of novice web developers that were accessible to us.

10.2 Future Work

Future work involves further integrating the multiple aspects of Knowledge Maps into one platform to provide for a smoother learning experience. Currently, the explore phase and the knowledge map phase are on two different platforms: the former is hosted on a website while the latter is hosted in Google Jamboard. Additionally, the reflection questions are pasted in a Google Doc. This causes users to switch between three different platforms when using the tool. Future work would integrate these three phases into one platform to provide a smoother and less complicated experience for the user.

We would also like to generalize the tool so that any website could be easily incorporated into the tool. This would require developing an automated solution for adding websites to the tool. Adding this feature would allow the tool to scale to a large number of websites and would even allow users to pick any website they want to explore.

Through advancing these potential research directions, we hope to help bridge the gap between novice and expert web developers. By expanding the KM tool to include any website, we can expand the learning opportunities available to every site on the Internet, allowing users to discover even more patterns and have more direction over their learning. We hope to reduce the dropoff in computer science, as novice developers often become frustrated by their difficulty in progressing past a novice level of expertise and give up altogether. By bridging the gap between novice and expert web developers and decreasing dropoff in Computer Science, we hope to increase the number of developers able to pursue professional careers in Computer Science. Extending beyond CSS, we hope that KM can provide a learning framework for other domains through exploration of examples and guidance for building out a knowledge map of findings from those examples in order to develop conceptual and transferable knowledge. The learning experience of KM has proven to be both successful and more effective than alternative methods for building expert CSS knowledge rapidly, paving the way for the learnings of KM to be used to develop a professional mindset in other fields.

References

(1) Ambrose, S. A., *How Learning Works : Seven Research-Based Principles for Smart Teaching*; Wiley: Hoboken, 2010.

(2) Lim, S. *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* **2017**.

(3) Zhu, D.; Kariuki, S. W. In Association for Computing Machinery: Honolulu, HI, USA, 2020, pp 1–6.

(4) Council, N. R., *Educating Children with Autism*; Lord, C., McGee, J. P., Eds.; The National Academies Press: Washington, DC, 2001.

(5) Haskell, R. E., *Transfer of learning : cognition, instruction, and reasoning*; Educational psychology series; Academic Press: San Diego, Calif, 2001.

(6) Head, A.; Appachu, C.; Hearst, M. A.; Hartmann, B. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2015, pp 3–12.

(7) Schwartz, D. L.; Bransford, J. D. *Cognition and Instruction* **1998**, *16*, 475–5223.

(8) Linn, M. C.; Clancy, M. J. *Commun. ACM* **1992**, *35*, 121–132.

(9) Ward, P.; Schraagen, J.; Roth, E., *The Oxford Handbook of Expertise*; Oxford Library of Psychology Series; Oxford University Press: 2019.