Causeway: Scaling Situated Learning with Micro-Role Hierarchies

David T. Lee[1], **Emily S. Hamedian**[1], **Greg Wolff**[1], **Amy Liu**[2]
[1]UC Santa Cruz, [2]LinkedIn
{davidtlee,esh,gwolff}@ucsc.edu,ayliu@linkedin.com

ABSTRACT

While educational technologies such as MOOCs have helped scale content-based learning, scaling situated learning is still challenging. The time it takes to define a real-world project and to mentor learners is often prohibitive, especially given the limited contributions that novices are able to make. This paper introduces micro-role hierarchies, a form of coordination that integrates workflows and hierarchies to help short-term novices predictably contribute to complex projects. Individuals contribute through micro-roles, small experiential assignments taking roughly 2 hours. These micro-roles support execution of the desired work process, but also sequence into learning pathways, resulting in a learning dynamic similar to moving up an organizational hierarchy. We demonstrate micro-role hierarchies through Causeway, a platform for learning web development while building websites for nonprofits. We carry out a proof-of-concept study in which learners built static websites for refugee resettlement agencies in 2 hour long roles.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **Collaborative and social computing**;

KEYWORDS

Learning-at-scale, situated learning, micro-role hierarchies

ACM Reference Format:

David T. Lee, Emily S. Hamedian, Greg Wolff, Amy Liu. 2019. Causeway: Scaling Situated Learning with Micro-Role Hierarchies. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4-9, 2019, Glasgow, Scotland UK.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3290605.3300304

CHI 2019, May 4–9,2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300304

1 INTRODUCTION

MOOCs and other educational technologies have motivated a vision of continuous life-long learning widely accessible to all [45]. In a labor market subject to rapid technological change, with as many as 60 million jobs turning over in the next two decades, this vision is more important than ever [31], but still faces significant obstacles. For one, MOOCs have largely focused on content-based learning. But expert work consists of tacit knowledge best learned through apprenticeships situated in a real-world context, a.k.a. situated learning. Second, even in content-based learning, MOOC completion rates have remained low, especially for the less affluent or educated [13, 19, 22]. If workers are to keep up in a world of constant change, we need "new ways to learn - a kind of situated learning-in-action" where anyone can access learning embedded in real-world contexts [3].

In this paper, we seek to introduce a model that: 1) makes it possible for learners to participate in small experiential roles that sequence into pathways for mastering professionally relevant skills, and 2) makes it possible to carry out complex projects using these small roles. Both are important. Providing situated learning in *small units* enables participation from those with limited time, and gives learners the motivation needed to engage in deliberate practice. Making it possible to use these small roles to successfully *carry out complex projects* prevents the preparation and mentorship time required from being a prohibitive barrier.

Our model rests on two ideas. *Doing-by-learning*, the act of achieving large real-world goals through novices engaged in learning, is the conceptual and aspirational goal motivating our approach. *Micro-role hierarchies*, a form of coordination integrating algorithmic logic with organizational hierarchies, represent one way to advance this goal. In a micro-role hierarchy, the algorithmic logic for carrying out a goal is structured not as a plain workflow, but as a series of workflows nested in a hierarchical structure (**Figure 1**), and designed to generalize to a particular type of goal, e.g. web development. Each workflow corresponds to a role in the hierarchy and captures the step-by-step process and delegation pattern for that role. These roles are scoped to small roughly 2-hour long commitments, possibly spread out over time. Roles are the basic units of participation, which can be sequenced into learning pathways similar to that of moving

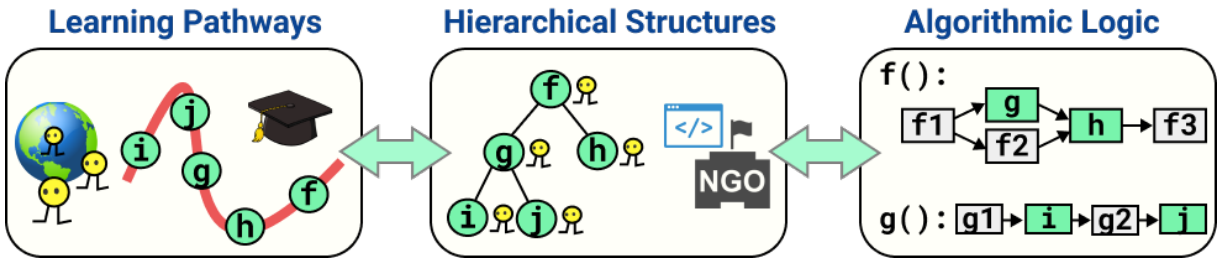

Figure 1: Circles represent roles and rectangles represent steps in a workflow. Individuals work their way through micro-roles (left) corresponding to nodes in project hierarchies (middle). These roles are supported by and linked together with algorithmic workflows (right). Workflows for higher-level roles have delegated steps corresponding to child roles.

up an in organizational hierarchy. As an individual moves through this pathway (typically occurring across different project instances of the same goal type), they learn skills while simultaneously contributing to real-world projects.

To demonstrate micro-role hierarchies, we developed Causeway, a platform where users learn web development while building websites for non-profits. We describe a field study in which we organized 22 students, most of whom had little to no knowledge of HTML and CSS, to code three static webpages based on hi-fidelity mocks defined with refugee resettlement agencies. We found participants were able to successfully develop websites far beyond their initial capabilities, only a small amount of time was required per role, and work could be robustly passed between roles.

In summary, this paper makes the following contributions: 1) *micro-role hierarchies*: a model for integrating situated learning with algorithmic, hierarchical coordination, 2) *doing-by-learning*: a conceptual mechanism and aspirational goal for situated learning, 3) *Causeway*: a system using micro-role hierarchies to teach web development while building websites for nonprofits, and 4) a field study demonstrating successful coordination of novices to build static websites for nonprofits, and an analysis of time, quality, and motivation.

2 BACKGROUND AND FORMATIVE WORK

Experiential learning and learning-at-scale

Experiential learning has its roots in the Pragmatist philosophy of thought as rooted in the "the exchange between an organism and its environment" [30], an idea that was carried into education by John Dewey [9, 10], a renowned Pragmatist. Expert work is complex. It is embedded in rich contexts full of tacit knowledge that cannot be easily transferred through raw content. Acquiring such expertise requires deliberate practice [12], often gained through experiential and apprenticeship learning, whether in physical or cognitive domains [4]. Advocates of situated learning emphasize this take place as closely as possible to a real-world work context. Knowledge cannot be separated from the context activity takes place in, whether the functional context of a complex task or the social context of a work environment [7, 39]. Finding such opportunities, however, is non-trivial and requires overcoming resource constraints, motivation constraints, and effort constraints [12]. Resource constraints are perhaps the most obvious: "available time and energy... and access to teachers, training material, and training facilities". Motivation constraints are also important since practice itself is not inherently rewarding. Individuals need to be "motivated to improve performance before they begin deliberate practice." Finally, effort constraints are also a factor since deliberate practice needs to sustain focus over "extended periods without leading to exhaustion".

One thread of research studies online communities of practice such as open source software [29, 32] and online creative communities [28], and is one of the most successful examples of large-scale situated learning in complex real-world work environments. Processes like legitimate peripheral participation help newcomers gradually move into a community by carrying out simple peripheral tasks [24, 48], often with the help of a mentor. Maintainers label "low-hanging fruit" [18] such as fixing bugs or writing documentation to help novices get started. However, since there are no clear pathways to go beyond peripheral tasks to core tasks, newcomers face numerous participation barriers [17, 38, 44]. Most core development is done by a few long-term contributors [27].

Learning, crowdsourcing, and complex crowd work

Many threads of work have also considered crowdsourcing as a site for situated learning. These include online labor markets providing opportunities for on-the-job learning [5, 11, 23] and micro-mentorship [16]; Citizen Science [8, 37] for service learning; and computational ecosystems like Agile Research Studios [51] and Crowd Research [40] supporting novel forms of communities of practice. Some use learning to support crowdsourcing [11, 41], or even to achieve crowdsourcing objectives through clever coordination of learner activity [14, 43, 47, 49], an idea called learnersourcing [20].

Micro-role hierarchies build on this thread of work. Specifically, they seek to integrate situated learning and crowd

algorithms, and are the first to introduce a method for coupling a sequence of small situated learning experiences to an algorithmic work process. This makes it possible to provide small consumable amounts of situated learning, while still supporting predictable progress towards mastery of complex tasks. Prior work has focused on supporting capstone-like programs requiring large (e.g. quarter-long) time commitments, or focus on micro-tasks in isolation that do not have a pathway towards complex tasks. We want to combine both.

From the crowdsourcing perspective, micro-role hierarchies build on a rich literature in complex crowd work. Early efforts utilized micro-task workflows to knit tiny contributions together to achieve complex goals [2, 6, 26]. To push the boundaries of complex crowdsourcing to more complex, interdependent goals, researchers extended micro-task workflows to macro-tasks [15, 34], utilizing expert workers hired from freelancer platforms such as Upwork [1]. In typical workflows, each step in the workflow is assigned without regard to prior involvement. This means that context can be lost from worker to worker, making highly interdependent tasks hard to carry out. To address this, hierarchies or leadership structures [21, 25] assign longer-term or expert workers to roles that provide continuity across a set of novice tasks. A recent promising direction is team-based crowdsourcing [34–36, 42, 52]. In contrast to workflows, team-based crowdsourcing almost always uses experts working on large macro-tasks. This makes it possible to support adaptivity resulting from "lack of structure and constraints" and "concurrent work and mutual adjustment" [33], but is less suitable for novices.

Micro-role hierarchies integrate algorithmic workflows and hierarchies to obtain (some) critical properties of each. Workflows are great at coordinating short-term novices since steps are small and well-defined. However, they can be fragile when project complexity increases since context is easily lost from step to step. Hierarchies help to preserve context [46], but require roles that are large in scope and better suited for long-term experts. In micro-role hierarchies, individuals are assigned to roles, which helps to preserve context. However, each role uses a workflow to codify the leader-to-novice interaction pattern between its actions and those of the roles it oversees, enabling a more granular decomposition. Unlike team-based crowdsourcing, our emphasis is not on adaptivity, but on greater scaffolding of roles by integrating hierarchical work with algorithmic coordination.

The journey to micro-role hierarchies

This project was birthed out of interviews with refugee resettlement agencies and non-profits. Our initial goal was not around situated learning or even around complex crowdsourcing. We wanted to understand the services and operations of non-profits and find opportunities to support these activities with micro-tasks. However, as we brainstormed, there were few opportunities in "Services" *or* "Operations". Instead, needs kept getting lumped into an "Other (Special Projects)" bucket. For *existing* services or operations, the abstract benefit of giving staff more time for other work was not compelling. They were more interested in new initiatives (requiring digital platforms) they believed could improve their services. For example, a small resettlement agency did not have the capacity to provide high-quality in-person cultural orientation, and wanted to create an online app to support this process. *This led us to focus on complex crowd work.*

For our first pilot study in July 2016, we created specifications and low-fidelity mocks for a cultural orientation app and designed a macro-task workflow to build a website. We hypothesized it would be possible if volunteers had prior expertise, committed for 2-hour long tasks, and were provided tutorials. For this pilot, the tutorials were written at a high level. We recruited 18 student participants and carried out the study over 3 weeks. Results were mixed. There was strong validation in engagement: volunteers went way beyond their time commitment, spending full days and asking for more tasks. However, the process was unsuccessful even when still working on the static visual layout. We heavily relied on the most experienced participants, and there was tremendous fragility when handing off tasks because different people had experience in different frameworks or approaches to writing code. We learned that, in some sense, *everyone* is a novice in complex volunteer crowdsourcing due to the alignment necessary across approaches. *This led us design highly opinionated workflows and tutorials.* It also revealed that *students were hungry for situated learning*, leading us to emphasize learning as a critical aspect of our work.

Our second pilot study in December 2016 doubled down on workflows, but focused on creating just static views. We made this portion of the workflow more much more detailed and opinionated. To stress test this, we recruited 6 participants with little to no experience. The results were mixed again. Volunteers learned a lot and wanted to participate more. However, the level of detail ended up being too much. Conditional branching statements were hard to follow since they required participants to absorb and evaluate the current state of the page. It was hard for participants to build on style files of past participants. Workflows are great at decomposing processes into small steps, but may not be successful when short-term participants need to absorb large amounts of existing context. *This led to our idea of using micro-role hierarchies to decompose work while still preserving context.*

3 MICRO-ROLE HIERARCHIES

Coordination of Work in Micro-Role Hierarchies

Micro-role hierarchies are defined by: 1) the roles making up the nodes of the hierarchy, 2) the workflows defined for each

role that link roles together, and 3) a pathway describing the learning sequence through the set of roles (**Figure 1**).

Algorithmic logic is represented through roles and workflows, linked by delegated steps. The desired objective (e.g. developing a website) is represented by the root node of the hierarchy. This node has an associated workflow representing steps to complete, which can be one of three types: *concept*, *action*, or *delegated*. Concept and action steps are short, and are accompanied by guides that either convey a concept or describe an action to carry out. Delegated steps are similar to action steps, but are typically large in scope. In an ideal role, everything modular is carved out and delegated, so that action steps correspond to the context-dependent "glue" hard to separate out. If a step is delegated, it is represented in the hierarchy as a child node and given its own workflow. These new nodes can also have delegated steps, and so on. Repeated delegation results in a granular hierarchy with each node representing a single step within its parent's associated workflow and each decomposed into its own workflow. Individuals are assigned to roles, and are responsible for carrying out all undelegated steps and for overseeing all delegated steps of their workflow.

From a computational perspective, the root node can be thought of as a function, with delegated nodes representing subroutines called by that function. In this framing, the difference between micro-role hierarchies and traditional workflows is that individuals are also assigned to represent subroutines rather than only computational "primitives".

Integrated roles, workflows and tutorials create opinionated organizations. One way to think about micro-role hierarchies is as opinionated organizations analogous to the opinionated programming paradigm championed by Ruby-on-Rails. Rails programmers face a higher learning curve to understand "the Rails way". However, once this is learned, relative novices are able to contribute to platforms much more complex than they could build otherwise. In micro-role hierarchies, role-specific workflows help to codify an interaction pattern between an "expert" and novices, making it possible to keep action steps bounded in time. This enables the creation of granular roles with minimal amounts of actual work time (though the clock time from start to finish can be long due to waiting for child roles). Every step is accompanied by a tutorial providing the instructions and conceptual content needed to carry out that specific step. This creates a learning experience similar to existing online coding platforms except where tutorials are centered around real roles embedded in a work process.

Situated Learning in Micro-Role Hierarchies

Micro-role hierarchies can also be described from a learning point of view (as opposed to a coordination point of view).

The hierarchy naturally produces a learning pathway. The hierarchy induces a situated learning pathway similar to that of working oneself up an organizational hierarchy. Since goals for delegated micro-roles are a subset of their parent roles' goals, they naturally decompose knowledge in addition to decomposing work. The hierarchy can be thought of as a prerequisite graph of knowledge. For example, one could progress through micro-roles according to a depth-first search such that parent nodes come after children, and later steps come after earlier steps so that new knowledge or context is minimized. Of course, the optimal pattern of traversal may not be so simplistic, and decomposition will likely also need to be done with learning in mind.

Resource, motivation, and effort constraints in micro-role hierarchies. One of the biggest resource constraints to situated learning is the mentorship required. Collins, Brown, and Newmann [4] describe three stages of apprenticeship: modeling, where an apprentice observes the master executing a task; coaching, where the apprentice attempts to execute the task with help; and fading, where the master reduces his participation to limited feedback. Finding a master able to go through these stages is a serious bottleneck. In micro-role hierarchies, workflows, steps, and tutorials do their best to approximate the role of a master by synthesizing best practice processes and by giving examples within the tutorials. Modeling, coaching, and fading are blended so that a learner jumps into a task as quickly as possible, with just enough "modeling" provided for the learner to carry out the next action within a micro-role, and with minimal "master" support if possible. In most communities of practice, the lack of a clear pathway means individuals take a long time to find their way from peripheral tasks like bug-fixing and documentation to core tasks. Legitimate peripheral participation is like peeling an onion, with one spending time in outside layers while working slowly but steadily in [24, 50]. Micro-role hierarchies can be described as helping learners to take a deep (though possibly thin) slice into the core of a complex task to help them quickly get situated in the broader context.

In addition to raw resources, micro-role hierarchies also support motivational and effort constraints in two ways. First, they enable learners to acquire and demonstrate a very clear set of skills in a bounded amount of time, as opposed to only seeing fruit after long periods of investment. Second, they enable learners to align the time they spend learning with supporting a nonprofit cause they believe in.

Doing-by-learning. The central concept in experiential learning, that we learn best by doing, is captured by the phrase learning-by-doing. We introduce doing-by-learning, the act of accomplishing large real-world goals through novices engaged in learning. Doing-by-learning is a form of learnersourcing, the act of organizing learner activity

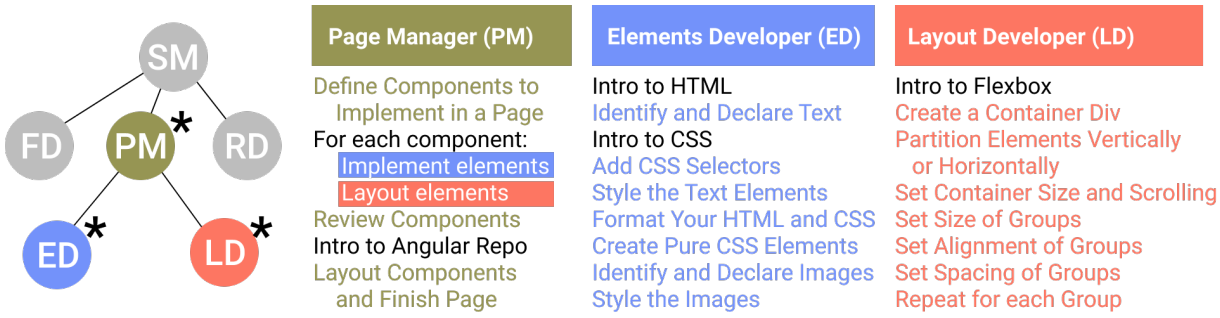

Figure 2: Our micro-role hierarchy for static websites has six types of micro-roles: a site-manager (SM), fonts developer (FD), page manager (PM), elements developer (ED), layout developer (LD), and routing developer (RD). We focused on the main challenge for novices, which is creating the visual layout of each page. We defined workflows and guides for these three roles.

to crowdsource desired goals. It is not identical, however, because doing-by-learning specifically refers to 1) learners engaged in learning-by-doing, and 2) aggregated outcomes that meet complex, real-world needs. Learnersourcing is focused on the crowdsourcing objective (achieved through learners), whereas doing-by-learning is focused on the educational objective (and how it is tied to doing real-world work). Doing-by-learning can be viewed from multiple perspectives: as a mechanism to make mentorship incentive-compatible, as a set of associated platforms or technologies, or as an aspirational standard for evaluating new technologies.

We believe scaling opportunities for learning-by-doing requires finding ways to enable doing-by-learning. If it is possible for learners to make contributions to real-world needs, then people will be incentivized to create opportunities for them to engage. If learners are unable to, then the time it takes to prepare authentic experiences and to mentor learners will continue to be an obstacle. Micro-role hierarchies are just one example of how one might enable doing-by-learning. In the whitewater world of automation, many have said that work and learning need to be deeply integrated. Doing-by-learning is a helpful conceptual tool for evaluating our progress towards this goal. Can our learning environments and resources enable learners to contribute to real-world objectives? If not, there is more work to do.

4 CAUSEWAY AND A PROOF-OF-CONCEPT

To substantiate these ideas, we defined a micro-role hierarchy for static websites (**Figure 2**) and embedded this hierarchy within Causeway (**Figure 3**), a platform for learning web development. The micro-role hierarchy takes a hi-fidelity mock of a website as input. It coordinates learners to create a website coded in HTML and SCSS matching the mock.

A Micro-Role Hierarchy for Web Development

The root micro-role is a `site manager`, who delegates out the task of importing site-wide fonts to a `font developer` and then delegates out each individual page to a `page manager`. Each page manager identifies components within their page to delegate out. The implementation of each component starts with an `elements developer`, who implements and styles each element independently. This code is then passed to a `layout developer` who positions the implemented elements to achieve the visual design. After all components have been successfully created, the page manager completes the page by positioning the completed components with any remaining elements in the page. Finally, a `routing developer` adds links that wire up the pages to each other.

For the scope of this paper, we limited our workflow to static websites because it was the smallest scope still far beyond the ability of people new to web development. In formative studies, we found that even those who self-reported being experienced struggled with creating the visual layout, especially when they needed to pass on their work to others. Because of this, and because each micro-role required extension tutorial writing, we also only wrote workflows and guides for the main challenge of creating the visual layout of each individual page: the `page manager`, `elements developer`, and `layout developer` (**Figure 2**). Tutorials are written independent of specific project instances to make them reusable. We return to the extensive tutorial writing needed in the discussion section on scalability.

The learning pathway. The learning pathway starts with the `elements developer`, then the `layout developer`, and finally the `page manager`. Participants can come in with zero knowledge of web development. They learn basics of HTML and SCSS by implementing individual text, image, and other elements in the `elements developer` micro-role. After mastering basics, they can move onto the `layout developer` micro-role to learn how to structure HTML tags hierarchically and how SCSS properties can be used to layout elements according to this hierarchy. When participants have mastered these two roles, their experience will enable them to identify appropriately sized components to delegate as needed in the `page manager` micro-role. They will also be

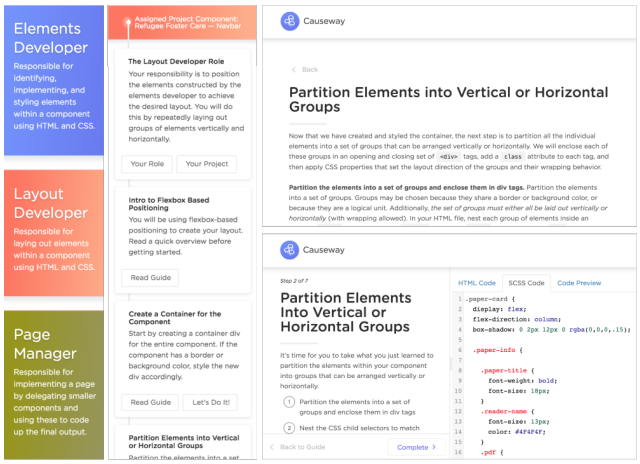

Figure 3: Causeway's interfaces (condensed). The first view a user sees is a list of roles (left) that are unlocked one at a time. Navigating into a specific role gives them an assigned instance and a list of steps to carry out (middle). Action steps have a tutorial (top-right) followed by an exercise (bottom-right). Concept steps just have the tutorial and delegated steps have a management view.

able to layout components to create the full page. In a complete implementation, assignment of roles could be done in many ways: automatically by the platform, initiated by a user who wants to contribute to a specific cause, or specified by a volunteer coordinator. The platform would ensure that prerequisite roles have been completed.

How Learners Use Causeway

In many respects, Causeway is reminiscent of other online platforms for learning to code. Users progress through learning modules, each containing a series of tutorials followed by coding exercises, and carried out in embedded coding environments. Modules and the sections within them are only unlocked when earlier modules/sections have been completed. The key difference is that "modules" in Causeway are explicit roles with defined responsibilities. When a participant clicks into a module, they are given an unassigned instance (or a duplicate if none is available). "Sections" in a typical learning module correspond to the concept, action, and delegated steps of the associated workflow.

Most of these steps are action steps with exercises to complete. Unlike common coding platforms, these exercises are not only toy problems, but also actual contributions to building a website for a nonprofit. These exercises are preceded by step-specific tutorials containing code examples users can edit and play with. Syntax cheatsheets are provided in the exercise view. Concept steps consist of tutorials without exercises, and focus on providing broader conceptual framing. For example, the Intro to HTML section introduces HTML tags and attributes, the Intro to CSS section introduces selectors and properties, the Intro to Flexbox based positioning section introduces the document hierarchy in a webpage, and the Intro to Angular Repo section describes the repository structure. Delegated steps, like that of the `page manager`, take the form of a managerial view, where the individual approves or rejects completed roles.

5 EVALUATION

Can micro-role hierarchies successfully coordinate novices to accomplish complex objectives as they learn? Can this be done in small units of time? To test this existence claim, we carried out a field study of Causeway using hi-fidelity mockups of a single page in three different apps defined with refugee resettlement agencies. To validate that participants were novices, we also integrated a within-subjects study of their ability to implement single components. We collected survey data to evaluate motivation and engagement of participants and logged the time it took to complete each step.

Method

Recruitment. We used campus mailing lists to recruit students to participate in building websites for refugee resettlement agencies while learning web development. No participants were compensated, so all participants were motivated to participate for reasons such as learning or supporting refugee resettlement agencies. The study occurred over the course of two weeks while school was in session, so that volunteering was competing with a typical student courseload.

Pre-study survey. Participants signed up by filling out a survey on their background in HTML and CSS. To understand their motivation, we asked whether they would only participate if we were building a new website for a non-profit or if they would still participate if it were just building an existing website for the purpose of learning.

Page manager bootstrapping. The micro-role hierarchy we implemented starts with a page manager who decomposes the page into smaller components to delegate. Since lower-level roles are prerequisites for the page manager, we had one participant first work their way through the elements and layout developer on a predefined component.

Component implementation with and without Causeway. To understand people's abilities prior to Causeway, we performed a within-subjects study to evaluate single component implementation with and without Causeway. After completing their survey, participants carried out a `control` treatment in which they implemented a component without the help of Causeway. This component was assigned uniformly at random from the components defined by the page manager. They were asked to implement the component to the best of their ability in one hour and could make use of web search

and online resources. After completing the control, participants continued onto the causeway treatment by signing into Causeway and carrying out a single elements and layout developer micro-role for randomly chosen components different than the one assigned them in the control. We further divided participants into two groups at random for this treatment. Participants in the same condition worked on the same component, building on their elements developer code when conducting the layout developer micro-role. Participants in the swap condition received the elements developer code from a different participant by swapping code whenever two participants completed the elements developer micro-roles. Adding the swap condition allowed us to evaluate whether tasks could be successfully handed-off between participants.

We note our study was *not* counter-balanced. All participants did the *control* before the *causeway* condition. In principle, this means they could have applied what they learned from the first condition to the second. We do not believe this one-hour control resulted in noticeable ordering effects. In the status quo, it easily takes novices several days before they can do real work. Our main purpose was to measure the "pre-study" state for all participants.

Completing all components. Completed components were given to the page manager who could accept them or ask for them to be redone. Since we did not implement chat for the scope of this paper, any components not accepted were assigned to new participants and reimplemented from scratch. These "redos" and unassigned components were assigned to the next available participants willing to do additional rounds of work. Recording the time taken during this second round also gave us some data to evaluate the speed up after the learning-intensive first round.

Finishing the page. Finally, the completed components were used to create the full page layout. Instead of having the original page manager do all of these, we split these among all participants who wanted to do the page manager role.

Post-Study Survey. All participants were asked to complete a post-study survey in which they gave qualitative and quantitative feedback on their experience and stated whether they would like to participate again and how likely they would be to recommend Causeway to a friend.

Data and Analysis. To evaluate whether micro-role hierarchies enabled learners to successfully contribute to complex work, we show (i) participants were able to successfully build webpages, (ii) this was far beyond their initial ability, and (iii) they only needed to spend small amounts of time. Quality was determined by evaluating and labeling completed components by error types, ranging from smaller style errors to serious structural errors. We used sign-up and exit surveys to evaluate learning experience and motivation.

6 RESULTS

22 people completed the control treatment and continue onto the causeway treatment. 2 dropped out after only the elements developer, leaving 20 who completed both the elements and layout developer. 10 participants continued beyond the within-subjects study, completing an additional 14 components either unassigned in the within-subjects study (8), rejected by the page manager (2), or were arbitrary duplicates (6). 8 participants, plus the original page manager, carried out the final layout step, three per project. In this section, we describe the three websites and results.

Webpage Descriptions

Each of the three projects came out of conversations with refugee resettlement agencies. Two of them were dynamic apps, but as described earlier, our micro-role hierarchy only implemented the view for these apps. The `Information Page` (**Figure 4, top**) is a redesign of the information page for a refugee foster care program. It consists of program information, testimonials, and a sitemap. It was decomposed by the page manager into 10 components of varying complexity. The `Welcome Page` (**Figure 4, middle**) is a dashboard within an app for coordinating volunteers to create welcome packages for refugees. It is the most complicated page, with a mix of information and widgets. It was decomposed by the page manager into 7 components. The `Orientation Page` (**Figure 4, bottom**) is the homepage within a cultural orientation app for helping refugees navigate the communities they enter. The homepage shows basic stats, upcoming lessons, and the user's progress so far. The page manager decomposed this page into 13 (very simple) components.

Successful Development of Static Websites

Participants successfully created all three pages (**Figure 4, right**) in 394 total lines of HTML and 1525 lines of SCSS.

Quality of single components in control vs. causeway. Our within-subjects study comparing single components in the control versus causeway treatments showed that implementing a static webpage was far beyond the starting ability of our participants (**Figure 5**). All components were labeled with four potential errors. There were two types of significant structural errors: `missing-layout`, when there was no layout structure at all, and `missing-element`, when an element was left out. There were two smaller types of errors: `missing-style`, when an element was not styled correctly, and `missing-resource`, when an image or font was not added. For the 20 participants who completed both treatments, only 6 did it perfectly or close to it in the control compared to 14 in the causeway treatment. There were no components with significant structural errors in the causeway treatment. The distribution of errors was roughly the same

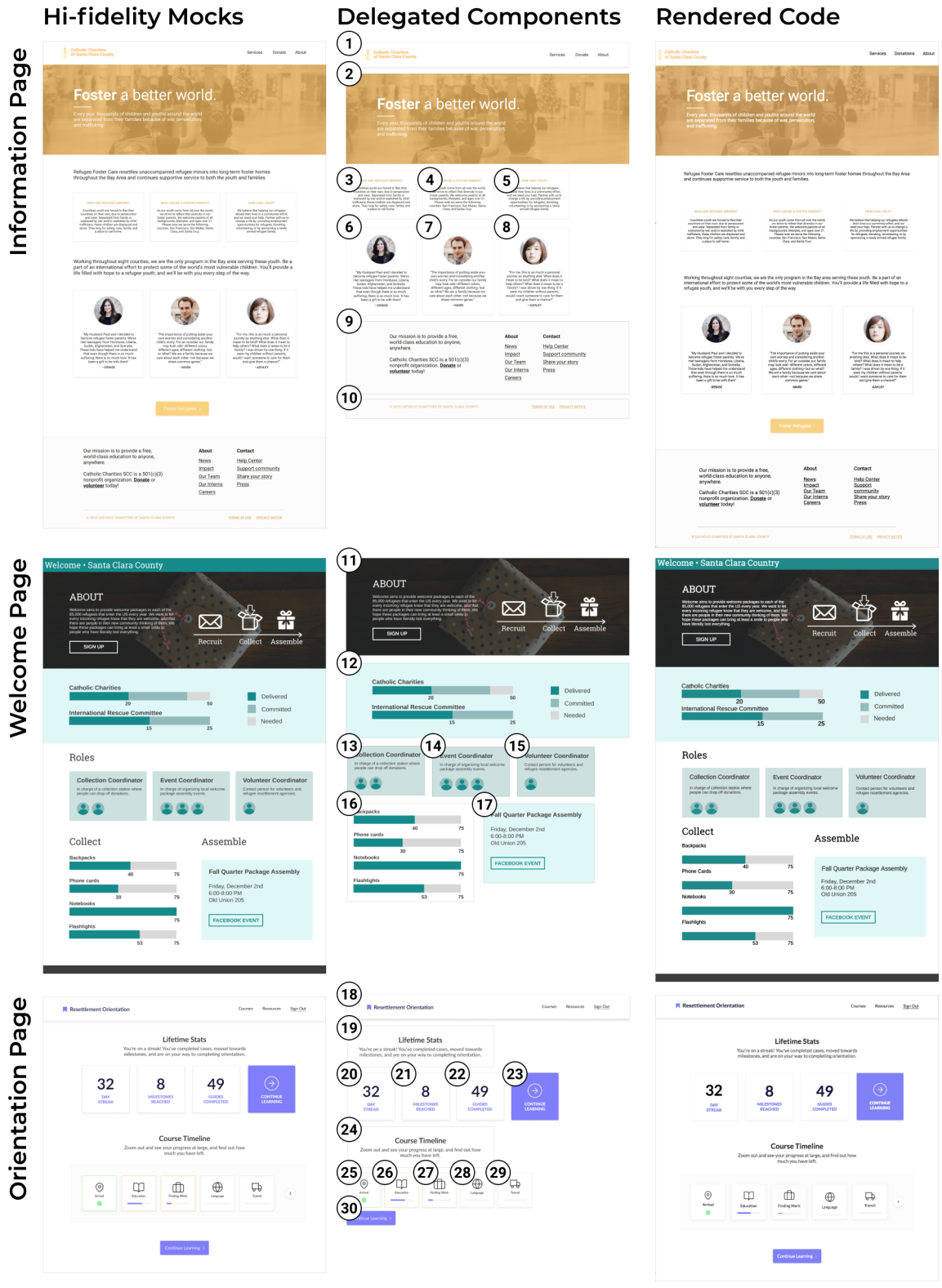

Figure 4: The left column shows the hi-fidelity mocks of the information page, welcome page, and orientation page. The middle column shows the hi-fidelity mocks of the delegated components. The last column shows the final rendered view resulting from participation coordinated using Causeway.

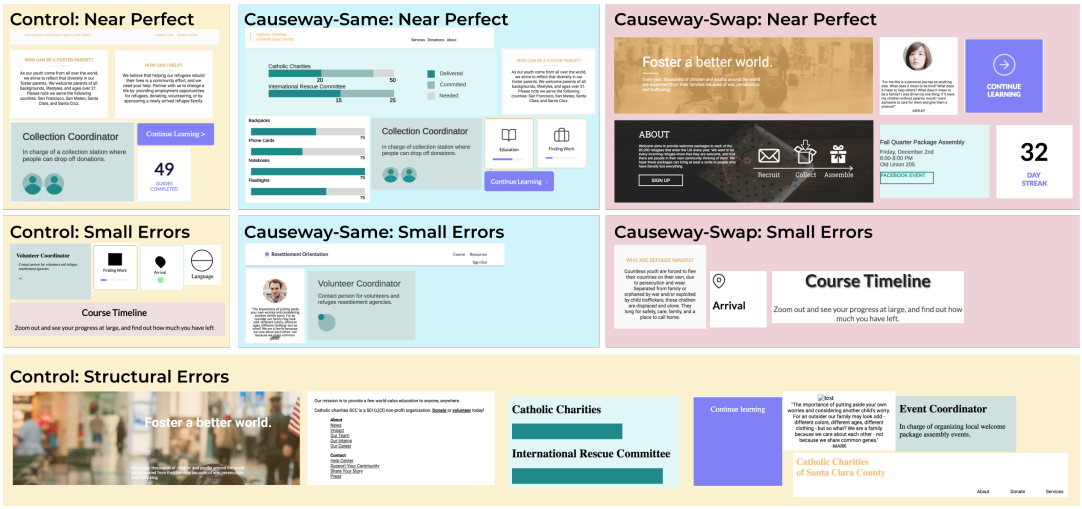

Figure 5: In the `control`, a small number of participants implemented their component perfectly or close to it, but most had either significant structural errors or small stylistic or resource-related errors. In the `causeway` treatment, none of the components had significant errors, though a few still had small errors.

in the same and swap conditions. These results demonstrate that: 1) most of our participants were *not* able to do even very simple components before using Causeway, 2) the vast majority of participants *were* able to implement small components near perfectly using Causeway, 3) quality does not degrade when a component is split between two people.

Completion of remaining components and full pages. The remaining components were implemented by the 10 participants who continued beyond the within-subjects study. These components were given to the page manager to accept or reject, who only ended up rejecting 2 of the components despite there being more with errors. She explained that since there were sets of components with the same template, she sometimes accepted a poorly done component when she knew she could copy and tweak the code from another component that was implemented well. The two rejected components were assigned to new participants. Eight participants wanted to do the final layout step, resulting in 3 per page. The right column in **Figure 4** shows the rendered pages for the best result. As can be seen, they are near perfect. There are some very minor spacing differences.

Time per micro-role and success of handoff. We measured the actual work time (not including breaks or waiting for child roles). Of the time logs we collected, we had to remove three due to recording errors. Participants spent more time in their first round of Causeway when they had to read through the tutorials, but this decreased significantly in the second round. In the first round, participants took an average of 1.33 hours (σ = 0.55) to complete the elements developer, 1.37 hours (σ = 1.12) to complete the layout developer, and 0.77 hours (σ = 0.43) to complete the page manager. By the second round, this decreased to an average of only 34 minutes (σ = 27) for the elements developer and 17 minutes (σ = 15) for the layout developer. No participants did the page manager more than once. As demonstrated by the resulting code for the three project pages, these small contributions were successfully integrated to create large, interdependent artifacts. Handoffs did not result in decreased quality. Out of the 22 initial participants, 11 were randomly buckets into the `swap` and `same` conditions, but 2 participants (both from the swap condition) did not complete the layout developer. As can be seen in Figure 5, both of these conditions had three cases with small errors, with the remainder as near perfect. The total time for completing all three pages was 55.2 hours.

Survey results: background, motivation, experience

For CSS, 68.2% of the 22 participants stated in the survey that they had close to no background at all or were still uncomfortable. For HTML, 59.1% said the same. 81.8% of the 22 participants said they would participate even if they were not building new websites for non-profits, but were just recreating existing websites, indicating that learning was the primary motivator. In the exit survey, people expressed they particularly liked learning in the context of doing real-world tasks. One participant said: *"I like the assignment of mini roles. It really helps in understanding and not feeling too overwhelmed."* When asked how likely they were to participate in the future (on a scale from 1-10, 1 = extremely unlikely, 10 = extremely likely), those who were new or only barely exposed responded with a mean of 8.8 (σ = 1.7). Those who were comfortable or well-versed responded with a mean of 7.2 (σ = 0.4). When asked how likely they were to recommend Causeway to their friends (on a scale from 1-10, 1 = extremely unlikely, 10 = extremely likely), those who were new or only barely exposed responded with a mean of 9.6 (σ = 0.6). Those who were comfortable or well-versed responded with a mean of 7.8 (σ = 1.5).

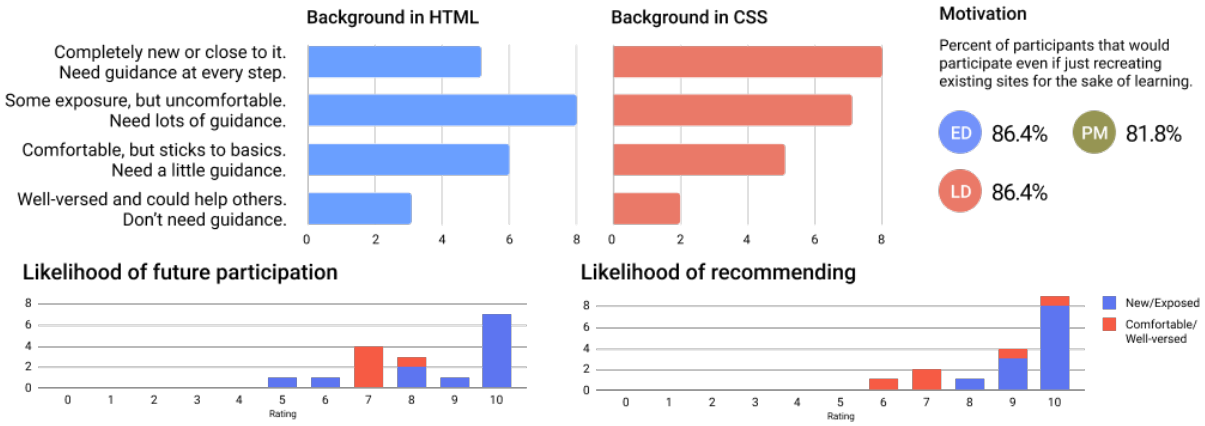

Figure 6: Most participants had little background, were primarily motivated by learning, and would recommend to friends.

7 DISCUSSION

Learning, work, and community engagement

While we have framed micro-role hierarchies primarily from the perspective of learning, we are equally interested in how they support community engagement through service learning. Nonprofits are resource-contrained, but cannot utilize volunteers interested in contributing their unique skills in small amounts of time. There are 16 million college students in the US. Suppose 10% contributed 2 hours of time a week as they learn. Suppose a website consists of around 10 distinct pages. Based on our field deployment, this would power the creation of 17,500 websites *every week*. Of course, that many static websites is not very useful, and there are better tools for doing this. The point is micro-role hierarchies have the potential to provide nonprofits with capacity in areas of expertise they do not traditionally have access to.

Scalability and Generalizability

Several important questions remain: how scalable are micro-role hierarchies given the heavy amounts of tutorial writing required? How are drop-offs handled? How generalizable are they beyond static websites? We see micro-role hierarchies as created through a peer production process to be reused and mixed by many. Developers of frameworks already spend large amounts of time creating learning resources, from API docs to toy examples to "real-world" examples. Micro-role hierarchies are a natural next step, as an even stronger way to teach a framework or approach and promote its adoption. It may be useful to think of micro-role hierarchies in analogy to frameworks like Angular, React, and Rails. Designing them requires tremendous time and expertise, but creating them in an open source manner is beneficial to companies in spreading adoption of (and contributions to) their practices.

While drop-outs might seem potentially problematic, we believe they may actually be easier to detect in a timely fashion given the small scope of work. Generalizability is still an open question. In the presented instantiation, micro-role hierarchies are better suited for execution as opposed to design. Within this bound, they should be transferable across domains in the same way workflows and hierarchies are. Design doesn't work as well due to the adaptivity required. However, we see future directions blending micro-role hierarchies with recent work on adaptive team-based hierarchies, e.g. via hybrid hierarchies where a primarily "adaptive" team can delegate to micro-role hierarchies, or adaptive evolution of micro-role hierarchies where a primarily "structured" team can deviate in ways that evolve the hierarchy.

8 CONCLUSION

In this paper, we advanced micro-role hierarchies, a model integrating algorithmic workflows, project hierarchies, and learning pathways to help scale situated learning. We argued for doing-by-learning as an important factor to scaling learning-by-doing. We demonstrated these ideas in a field deployment of Causeway that organized learners to code three websites for refugee resettlement agencies. Our hope is that this will open up future work in scaling opportunities for situated learning, and for deeper integration of learning with work and community engagement.

ACKNOWLEDGMENTS

The first author would like to thank Michael Bernstein, Rajan Vaish, Juho Kim, Liz Gerber, and many others for tremendous support moving into HCI at a stage of his career when moving into new fields is not recommended. We thank John Seely Brown for introducing us to situated learning and the whitewater world, and Kyle Qian and Colin Au for helping with different parts of the project. We thank the reviewers for their detailed and tremendously helpful comments.

REFERENCES

[1] 2017. Upwork. https://www.upwork.com/. (2017).

[2] Paul André, Aniket Kittur, and Steven P Dow. 2014. Crowd Synthesis: Extracting Categories and Clusters from Complex Data. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, New York, NY, USA, 989–998.

[3] John Seely Brown. 2018. Pardee RAND Commencement Speech 2018. http://www.johnseelybrown.com/JSBpardeerand.html. (June 2018). Accessed: 2018-9-10.

[4] John Seely Brown, Allan Collins, and Paul Duguid. 1989. Situated Cognition and the Culture of Learning. *Educ. Res.* 18, 1 (Jan. 1989), 32–42.

[5] G Chen, D Davis, M Krause, E Aivaloglou, C Hauff, and G Houben. 2018. From Learners to Earners: Enabling MOOC Learners to Apply Their Skills and Earn Money in an Online Market Place. *IEEE Trans. Learn. Technol.* 11, 2 (April 2018), 264–274.

[6] Lydia B Chilton, Greg Little, Darren Edge, Daniel S Weld, and James A Landay. 2013. Cascade: Crowdsourcing Taxonomy Creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1999–2008.

[7] Allan Collins, John Seely Brown, and Susan E Newman. 1989. Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. *Knowing, learning, and instruction: Essays in honor of Robert Glaser* 18 (1989), 32–42.

[8] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and Foldit Players. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466, 7307 (Aug. 2010), 756–760.

[9] John Dewey. 2004. *Democracy and Education*. Courier Corporation.

[10] John Dewey. 2007. *Experience And Education*. Simon and Schuster.

[11] Mira Dontcheva, Robert R Morris, Joel R Brandt, and Elizabeth M Gerber. 2014. Combining Crowdsourcing and Learning to Improve Engagement and Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3379–3388.

[12] K Anders Ericsson, Ralf T Krampe, and Clemens Tesch-Römer. 1993. The role of deliberate practice in the acquisition of expert performance. *Psychol. Rev.* 100, 3 (1993), 363.

[13] Thommy Eriksson, Tom Adawi, and Christian Stöhr. 2017. "Time is the bottleneck": a qualitative study exploring why learners drop out of MOOCs. *Journal of Computing in Higher Education* 29, 1 (April 2017), 133–146.

[14] Elena L Glassman, Aaron Lin, Carrie J Cai, and Robert C Miller. 2016. Learnersourcing Personalized Hints. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*. ACM, New York, NY, USA, 1626–1636.

[15] Daniel Haas, Jason Ansel, Lydia Gu, and Adam Marcus. 2015. Argonaut: Macrotask Crowdsourcing for Complex Data Processing. *Proceedings VLDB Endowment* 8, 12 (Aug. 2015), 1642–1653.

[16] Margeret Hall, Michelle Friend, and Markus Krause. 2018. Micro-internships on the Margins. In *Universal Access in Human-Computer Interaction. Virtual, Augmented, and Intelligent Environments*. Springer International Publishing, 486–495.

[17] Christoph Hannebauer. 2016. *Contribution Barriers to Open Source Projects*. Ph.D. Dissertation.

[18] Christoph Hannebauer, Vincent Wolff-Marting, and Volker Gruhn. 2010. Towards a Pattern Language for FLOSS Development. In *Proceedings of the 17th Conference on Pattern Languages of Programs (PLOP '10)*. ACM, New York, NY, USA, 15:1–15:10.

[19] Kate S Hone and Ghada R El Said. 2016. Exploring the factors affecting MOOC retention: A survey study. *Comput. Educ.* 98 (July 2016), 157–168.

[20] Kim, Juho, and Ph. D. Massachusetts Institute of Technology. 2015. *Learnersourcing : improving learning with collective learner activity*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[21] Joy Kim, Justin Cheng, and Michael S Bernstein. 2014. Ensemble: Exploring Complementary Strengths of Leaders and Crowds in Creative Collaboration. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, New York, NY, USA, 745–755.

[22] René F Kizilcec, Andrew J Saltarelli, Justin Reich, and Geoffrey L Cohen. 2017. Closing global achievement gaps in MOOCs. *Science* 355, 6322 (Jan. 2017), 251–252.

[23] Markus Krause, Doris Schiöberg, and Jan David Smeddinck. 2018. Mooqita: Empowering Hidden Talents with a Novel Work-Learn Model. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, CS14.

[24] Jean Lave, Etienne Wenger, and Etienne Wenger. 1991. *Situated learning: Legitimate peripheral participation*. Vol. 521423740. Cambridge university press Cambridge.

[25] Kurt Luther and Amy Bruckman. 2008. Leadership in Online Creative Collaboration. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (CSCW '08)*. ACM, New York, NY, USA, 343–352.

[26] Kurt Luther, Nathan Hahn, Steven P Dow, and Aniket Kittur. 2015. Crowdlines: Supporting synthesis of diverse information sources through crowdsourced outlines. In *Third AAAI Conference on Human Computation and Crowdsourcing*. aaai.org.

[27] W Maass. 2004. Inside an open source software community: empirical analysis on individual and group level. *Proceedings of the 4th Workshop on Open Source* (Jan. 2004), 65–70.

[28] J Maloney, L Burd, Y Kafai, N Rusk, B Silverman, and M Resnick. 2004. Scratch: a sneak preview [education]. In *Proceedings. Second International Conference on Creating, Connecting and Collaborating through Computing, 2004.* ieeexplore.ieee.org, 104–109.

[29] A Mockus, R T Fielding, and J Herbsleb. 2000. A case study of open source software development: the Apache server. In *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium.* ieeexplore.ieee.org, 263–272.

[30] A Pendleton-Jullian and J S Brown. 2016. *Pragmatic Imagination*. Vol. 1367563127. Blurb.

[31] Lee Rainie and Janna Anderson. 2017. The Future of Jobs and Jobs Training. http://www.pewinternet.org/2017/05/03/the-future-of-jobs-and-jobs-training/. (May 2017). Accessed: 2017-6-9.

[32] Christian Robottom Reis and Renata Pontin de Mattos Fortes. 2002. An overview of the software engineering process and tools in the Mozilla project. In *Proceedings of the Open Source Software Development Workshop*. ics.uci.edu, 155–175.

[33] Daniela Retelny, Michael S Bernstein, and Melissa A Valentine. 2017. No workflow can ever be enough: How crowdsourcing workflows constrain complex work. *ACM, New York, NY, USA* (2017).

[34] Daniela Retelny, Sébastien Robaszkiewicz, Alexandra To, Walter S Lasecki, Jay Patel, Negar Rahmati, Tulsee Doshi, Melissa Valentine, and Michael S Bernstein. 2014. Expert Crowdsourcing with Flash Teams. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 75–85.

[35] Niloufar Salehi and Michael S Bernstein. 2018. Hive: Collective Design Through Network Rotation. *Proc. ACM Hum. -Comput. Interact.* 2, CSCW (Nov. 2018), 151:1–151:26.

[36] Niloufar Salehi, Andrew McCabe, Melissa Valentine, and Michael Bernstein. 2017. Huddler: Convening Stable and Familiar Crowd Teams

Despite Unpredictable Availability. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. ACM, New York, NY, USA, 1700–1713.

[37] Robert Simpson, Kevin R Page, and David De Roure. 2014. Zooniverse: Observing the World's Largest Citizen Science Platform. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14 Companion)*. ACM, New York, NY, USA, 1049–1054.

[38] Igor Steinmacher, Marco Aurélio Graciotto Silva, and Marco Aurélio Gerosa. 2014. Barriers Faced by Newcomers to Open Source Projects: A Systematic Review. In *OSS*. 153–163.

[39] Lucy A Suchman. 1987. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press.

[40] Rajan Vaish, Snehalkumar (neil) S Gaikwad, Geza Kovacs, Andreas Veit, Ranjay Krishna, Imanol Arrieta Ibarra, Camelia Simoiu, Michael Wilber, Serge Belongie, Sharad Goel, James Davis, and Michael S Bernstein. 2017. Crowd Research: Open and Scalable University Laboratories. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 829–843.

[41] Rajan Vaish, Shirish Goyal, Amin Saberi, and Sharad Goel. 2018. Creating Crowdsourced Research Talks at Scale. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. rajanvaish.com, 1–11.

[42] Melissa A Valentine, Daniela Retelny, Alexandra To, Negar Rahmati, Tulsee Doshi, and Michael S Bernstein. 2017. Flash Organizations: Crowdsourcing Complex Work by Structuring Crowds As Organizations. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3523–3537.

[43] Luis von Ahn. 2013. Duolingo: Learn a Language for Free While Helping to Translate the Web. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces (IUI '13)*. ACM, New York, NY, USA, 1–2.

[44] Georg von Krogh, Sebastian Spaeth, and Karim R Lakhani. 2003. Community, joining, and specialization in open source software innovation: a case study. *Res. Policy* 32, 7 (July 2003), 1217–1241.

[45] M Mitchell Waldrop. 2013. Online learning: Campus 2.0. *Nature* 495, 7440 (March 2013), 160–163.

[46] Max Weber. 2009. *The Theory Of Social And Economic Organization*. Simon and Schuster.

[47] Sarah Weir, Juho Kim, Krzysztof Z Gajos, and Robert C Miller. 2015. Learnersourcing Subgoal Labels for How-to Videos. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 405–416.

[48] Etienne Wenger. 1998. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press.

[49] Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. 2016. AXIS: Generating Explanations at Scale with Learnersourcing and Machine Learning. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale (L@S '16)*. ACM, New York, NY, USA, 379–388.

[50] Yunwen Ye and Kouichi Kishida. 2003. Toward an Understanding of the Motivation Open Source Software Developers. In *Proceedings of the 25th International Conference on Software Engineering (ICSE '03)*. IEEE Computer Society, Washington, DC, USA, 419–429.

[51] Haoqi Zhang, Matthew W Easterday, Elizabeth M Gerber, Daniel Rees Lewis, and Leesha Maliakal. 2017. Agile Research Studios: Orchestrating Communities of Practice to Advance Research Training. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17 Companion)*. ACM, New York, NY, USA, 45–48.

[52] Sharon Zhou, Melissa Valentine, and Michael S Bernstein. 2018. In Search of the Dream Team: Temporally Constrained Multi-Armed Bandits for Identifying Effective Team Structures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 108:1–108:13.