Learning outside the classroom

Andrew Begel, Andrew J. Ko

1 Introduction

The history of computing education research is replete with studies about learning in *formal* contexts, i.e. students learning from teachers in school classrooms. In this chapter, we explore other contexts in which learning about computing occurs, for example, through reading books, working through online tutorials, competing in hackathons, or asking and answering computing questions on a Q&A website. These activities are all examples of *informal learning*—learning that is opportunistic, rather than planned; unstructured, rather than pedagogically created; self-directed, rather than teacher-centric; and integrated authentically into life activities (Marsick & Watkins, 2001), rather than taking place in a classroom environment. We collect and synthesize research about informal learning of computing and discuss open questions around where and how it occurs, and how to best support it.

1.1 Background

Since the 1970s, many education and learning science researchers have studied and described informal learning. However, all research in informal learning implicitly recognizes the centrality of the individual's learning context: the learner is in control of what is to be learned and when. This contrasts with most school-based and teacher-driven scenarios, where such decisions define the role, purpose, and authority imbued in teachers. In the settings we focus on in this chapter, learning is a central act of life, taking place in the most individual of circumstances on topics that may only be meaningful to the learner, for example, as Papert describes his beloved gears in *Mindstorms* (Papert, 1980).

Definitions of informal learning vary. A literature review by Marsick and Volpe finds six characteristics are intrinsic to informal learning: 1) integrates with life activities, 2) occurs when triggered, 3) is not always conscious, 4) can be haphazard, 5) involves repeated reflection and action, and 6) links to learning of other people (Marsick and Volpe, 1999). Many of these characteristics have long been studied, most notably by Knowles (1975). Knowles described and prescribed *self-directed learning*, "a process in which individuals take the initiative without the help of others in diagnosing their learning needs, formulating goals, identifying human and material resources, and evaluating learning outcomes." Some studies have investigated the "triggered" aspect of informal learning by investigating learning that occurred after unexpected events, such as nurses having to learn on the job while at war (Menard 1993). Whereas these viewed informal learning as a mostly solitary activity, more recent views have considered the integration with life and other people, leveraging social learning perspectives. For example, some argue that with the proliferation of information on the Internet, having knowledge is not as important as the process of discovering knowledge, which makes information resources such as people and technology more important than anything a learner already knows (Siemens 2005). Other recent views of informal learning argue that the divide between formal and informal learning is blurring, challenging the notion of the classroom as a “container” for learning (Leander et al., 2010). Within this view, the emergence of virtual spaces online has amplified the capacity for learning throughout one’s physical and social spaces, not just the classroom.

In this chapter, we view informal learning from Papert’s perspective, where the learner is in control instead of the teacher (Papert, 1980). For example, the only thing that makes a course “online” is the medium a student uses to access resources and feedback from a teacher. The only thing that makes “remote” learning remote is that a student is physically distant from a teacher. If you remove the teacher from the learning, or you view the teacher as just one of many resources for knowledge and feedback, then the lines between formal, school-centered learning and other types of learning become blurred. From this perspective, informal learning can occur anywhere, including work, play, and on-the-side, but also at school, for example in extracurricular activities, or in service of school goals, such as consulting a programming tutorial to prepare for a challenging course.

This same learner-centered view reshapes what it means to be motivated and stay motivated to learn. In a classroom-centric view of learning, a teacher is charged with motivating and engaging a student. A learner-centered view focuses on learners’ motivations, and acknowledges that learners’ motivation are likely to be more heterogeneous outside of a traditional educational institution. For example, a recent study of motivations to learn in MOOCs found that students had several distinct reasons for accessing MOOC content: fulfilling a knowledge gap relevant to their life, preparing for their future, satisfying a curiosity, or connecting with people (Zheng et al. 2015). This diversity of motivations means that “completing” a course cannot be viewed as the only notion of success: many learners may never have intended to complete it.

Since informal learning is so learner-centric, to an educator, it can appear quite haphazard. How then can informal learning be facilitated at all? Marsick and Watkins propose three conditions to enhance one’s informal learning: 1) encouraging critical reflection on what one already knows, 2) encouraging the learner to proactively identify missing skills and learn new strategies to facilitate learning, and 3) stimulating creativity to enable the learner explore a wide range of resources (Marsick & Watkins, 2001). One notable idea that attempts to support all three is the notion of a Personal Learning Environment (PLE), which is any constellation of tools, communities, and services that learners use to direct their learning and pursue education goals (Dabbagh & Kitsantas 2012). Recent studies have found that when students have the self-regulation skills needed to reflect on what they do and do not know, they create more socially-enriched PLEs and experience a greater sense of learning (Cho et al. 2010). Few works, however, have examined how to promote these self-regulation skills, or experimentally demonstrate that they are the cause of these richer experiences and learning outcomes.

In computing education, research has focused on numerous kinds of informal learning, but without the depth that one finds in the learning science research literature. Instead, research has largely explored the vast range of opportunities for informal learning and the systems needed to support it. This area is therefore full of open research questions that could bridge these literatures.

2 Environments for Informal Learning

In this section, we explore several contexts for informal learning. We begin with the primary modern informal learning environment, online learning, which is enabled by the ubiquity of the Internet. Online, learners must discover and use materials and resources, as well as learn to engage the online community. Recently, learners have been able to take advantage of digital textbooks while attending Massive Open Online Courses (MOOCs). Finally, we take a look at summer camps for coding, which expose learners to programming and computing concepts outside the structure afforded in a school environment. Note that we do *not* discuss other forms of informal learning that have not yet been studied, such as the use of books, magazines, and other media for self-study.

2.1 Online learning

Some of the first efforts to investigate informal learning in computing was in the form of *distance learning*. This phrase, which we now more commonly refer to as *online learning*, emerged from the goal of increasing access to computing education. The teaching and research community viewed this shift as one of essentially *translating* classroom activities to computer based media.

The earliest research on distance learning coincided with the proliferation of access to the Internet in the 1990s. This made it possible for students to attend class remotely. As with most new technologies, teachers attempted to translate existing teaching material such as lectures into new media on the web. Instructors teaching entirely online quickly found that teaching at a distance was not a simple matter of translating content (Gersting, 2000). Instructors wrote about the challenges in translating written classroom notes into recorded lectures that students watched on PCs (Gal-Ezer et al., 2009). Others investigated the challenges of translating synchronous in-person lectures into synchronous online lectures, discovering that engaging students at a distance was more challenging (Koppelman and Vranken, 2008). Some experimented with office hours through instant messaging and phone calls (Malan, 2009). Many instructors built robust, scalable courseware for packaging lecture content as web content (Dankell and Hearn 1997), created custom tutorials and tool support for writing and submitting programs online (Hitz and Kögeler 1997), and developed generic toolkits for synchronous chat and lectures (Pullen 2006). Some experimented with hybrid online courses that included both classroom and online activities, under the assumption that “independent learning” was an inherent part of learning computer science (Rosbottom 2001); such work continues, investigating blended online and in-person learning in MOOCs (Grover et al. 2015).

Throughout all of these efforts, attempts to evaluate effects on student learning were almost completely absent, with most evaluations simply reporting informally solicited, positive attitudes toward the new media. One of the only rigorous evaluations of learning computing online was performed by Carswell at the Open University in the UK, who found that communicating over the Internet via email had no significant effect on learning outcomes relative to other communication media such as phones (Carswell 1997).

Researchers were more experimental with the web, arguing that the medium had new affordances that needed to be understood (Carswell 1998), such as new opportunities for observation and experimentation on learning that classrooms do not (Howard et al., 2010). Instructors experimented with coding live in front of students, where students used instant messaging to provide a shared display of feedback and guidance on the instructor's programming decisions (Bower, 2008). Some instructors experimented with platforms like Second Life, a virtual environment that supported avatars and chat, embedding development environments and collaboration (Crellin et al., 2009). Others tried using video conferencing to facilitate large scale object-oriented design sessions in which a teacher and a student group developed and discussed solutions to systems design problems (von Wright 2000). As online courses increased in size with the proliferation of MOOCs, it became possible to experiment longitudinally and at scale with new techniques. For example, one study ran a 9-year experiment finding that gamification techniques caused a significant increase in engagement with online class activities (Lehtonen et al. 2015). Few of these studies investigated the informal learning skills required to support online learning in formal coursework.

While all of this work nominally occurred in formal learning environments, research throughout this period of experimentation revealed online learning required many of the same strategies found in informal learning environments. For example, a study of help-seeking in a web development course found that nearly all students sought help in unstructured discussion forums, from both instructors, and peers, and that they often relied on the Internet to learn independently (Park and Wiedenbeck, 2011). A study of help seeking in a user interface development course found that online documentation of APIs and development platforms were fundamental learning resources (Ko and Myers, 2004). These studies show that whatever materials a class provides, when there are more robust materials online, course materials have trouble rivaling the scope, scale, or relevance of content on the entire web. This has the effect of shifting a lot of the learning online, even when students are learning in collocated classroom settings.

2.2 Finding and using online resources

Another opportunity for learning occurs when informal learners struggle to find, assess, and use online information resources. Learners employ a variety of resources, including online Q&A websites like Stack Overflow, code search, digitals textbooks, MOOCs, and videos.

Online Q&A communities are key resources for developers seeking answers to programming questions about languages and APIs (Jones & Churchill, 2009). Not only do peers help diagnose programming bugs, they can also help learners avoid starting from scratch by enabling building projects based on one another's shared code (and introducing challenges around plagiarism, discussed in Chapter 3.3). Chambers et al. found that students depend on this online information; they frequently used code examples to overcome compilation errors and rarely referenced information sources that could have given them better success rates (Chambers et al., 2012). Other developers use code search to discover this kind of information. Sadowski et al. studied professional developers at Google and found that when they searched for code, they wanted to answer questions about how to use an API, for examples on how the code operates, and why it might be failing (Sadowski et al. 2015). Dorn and Guzdial found that graphic designers also engage with Q&A forums and other documentation sites in order to learn how to automate their work by programming scripts (Dorn & Guzdial 2006). From the perspective of Q&A site owners, it requires substantial design investment and community leadership to make forum designs effective at nurturing inviting, helpful discussions (Mamykina et al. 2011, Begel et al. 2013).

Several studies have found that there are critical information retrieval skills necessary to successfully use online resources about computing. For example, the use of Q&A sites and code search requires people to learn search and query reformulation skills, which are non-obvious to novices (Dorn et al. 2013). DiSalvo et al. discovered that parents looking for CS educational resources for their children had trouble obtaining good results from what they thought were reasonable search queries (DiSalvo 2014). Many researchers have found that novice searchers have trouble writing effective queries and recognizing good sources because they focus shallowly on the surface of a website and lack confidence in their awareness of appropriate online resources (Moraveji et al. 2011).

Other types of online resources can be useful orientations for students who need a first place to look. Hao et al. found that students who face difficult problems first look online, but as the difficulty level rises, they would rather seek out help from peers or other resources (Hao et al. 2016). One problem with online resources is that, because learners often have very specific personal goals, and resources are rarely tailored to those goals, learners struggle to assess the quality and relevance of those resources (Dorn & Guzdial 2010). Studies of professional software developers' use of API documentation have found that while they use documentation to learn, over time, they continue to rely on them as a form of external memory, stalling recall and deeper learning until just before they are needed (Brandt et al. 2009).

2.3 MOOCs

Informal learners often engage with content developed for formal instruction, such as digital textbooks, but doing so without the structure of a formal learning context can have benefits and drawbacks. For example, Warner et al. found that informal learners who accessed digital textbooks made extensive use of interactive components, such as executing code and answering multiple-choice questions, but rarely viewed textbook sections out of order (Warner et al. 2015). Guo and Reinecke also found that it was easy for learners to become disengaged with materials, as evidenced by a large-scale study of MOOC students' navigation history with course materials, showing that learners frequently skipped materials, read out of order, and read shallowly (Guo & Reinecke 2014).

There are also many tensions between the formats used in online media and the need to discuss code. For example, Zhu et al. noticed that text-based discussion forums were inefficient for teaching programming; in order to increase engagement, forums should integrate interactive and visual programming features (Zhu et al. 2015). Guo et al. found that engagement increases when MOOC videos are short, show talking heads, and use hand-made tablet-based drawings (Guo et al. 2014).

Research about the quality of online materials is still scarce. Researchers have partly tried to measure quality by measuring engagement, but measuring engagement can be complex, because of underlying factors of attitudes and motivation (Chapter 3.17). For example, in evaluating videos in MOOCs, people engage more by pausing or rewatching the same MOOC video segment; this can actually mean either that they are interested, or that they are simply confused (Kim et al. 2014). Kim and Ko conducted an evaluation of dozens of coding tutorials using a more principled, analytical method, finding that coding tutorials lack most of the key requirements for successful learning (Kim & Ko 2017), such as personalized feedback about problem solving, explanations about why concepts are important to larger problem solving guidance, guidance on common errors, and adaptation to learners' prior knowledge. Similarly, in a reflection on five years of MOOC education at Stanford, Cooper and Sahami felt that the lack of personalized instruction and feedback limits positive learning outcomes (Cooper & Sahami 2013).

2.4 Camps

Another widely-studied semi-informal learning context is camps. These can come in the form of after school programs, weekend programs, or week or multi-week summer programs. They are formal in that in there are often multiple instructors guiding learners' experiences and creative efforts. They are informal, however, in that learners, rather than teachers, are ultimately the ones in charge of what they learn, and how engaged they are, and even whether they attend regularly. After all, since camps are not compulsory, but often voluntary supplemental activities to formal learning, learners view them as a chance to explore their interests rather than satisfy a school requirement. This reduces the teachers' authority, which shifts them slightly (but not fully) from formal to informal learning

Computing camps are now ubiquitous in some countries. Some are run for profit, while others are non-profits. Some are supported by local colleges and universities and others \are run as research projects. And because they often occur outside of the context of a formal education institution, they can be structured in richly diverse ways. One computing camp followed a week-long summer curriculum for middle school girls, aiming to convey future careers, to connect students with invited speakers, and to use programming environments like Alice to tell stories by writing simple computer programs (Webb & Rosson 2011). Others used the App Inventor platform to scaffold the creation of mobile application development through daily support and guidance (Wagner et al. 2013). The Georgia Computes! project was perhaps one of the most extensive efforts at informal learning of computing, as it spanned the entire state of Georgia. It offered camps that leveraged a variety of platforms, including PicoCrickets, Scratch, LEGO NXT Kits, Alice, LEGO Textrix kits, LEGO WeDo Kits, and Pleo robots, which engaged a broad range of learner interests (Ericson & McKlin, 2012). Beyond these camps offered by researchers, there are countless non-academic organizations that offer camps as a way of engaging youth in computing. This variety of offerings and content is essential, as learners' interests are very diverse—without diverse content to serve those interests, many learners would lack the motivation to engage.

As a context for research on informal learning, camps are compelling because they offer more control than purely informal settings without teachers. Researchers can devise exactly the experience they want to test, or probe into experience in precise and systematic ways not usually possible in more constrained classroom environments. However, because they lack the compulsory nature of formal learning environments, they can be more dominated by learners' interests and motivation. This has meant that much research on camps focuses on changes in interest, motivation, and identity, rather than learning.

Research on camps has often lacked rigor. One analysis of published studies found that only 8% of them offered longitudinal evidence of impact of any kind. Most focused instead on measuring attitudes, interest, or programming skills, and reported positive or neutral findings (Decker et al. 2016). Another survey found that camps designed by researchers were significantly different from camps designed by practitioners. The research camps used different approaches, framed alternate outreach goals, and used more rigorous methods to analyze learners' experiences (DeWitt et al. 2017). Part of the challenge of conducting rigorous analyses of camps is their unstructured nature—learners in the same camp may do substantially different things based on their interests, making it difficult to systematically observe outcomes. The result is that many studies rely on short-term, self-reported changes in self-efficacy, learning, and other outcomes (e.g., McGill et al. 2015, Aritajati et al. 2015).

Some studies devised creative ways of observing impact without relying on self-report. For example, Kelleher et al. wanted to measure how a version of Alice that was designed for storytelling mediated middle-school learners' motivation to create Alice programs (Kelleher et al. 2007). Rather than asking learners to self-report their motivation to learn, the researchers structured the camp to hold numerous breaks with highly desirable treats like cookies. Researchers then measured how long learners continued to work after the breaks started, getting a continuous measure of *in situ* motivation relative to desire for snacks and food. Loksa et al. used another powerful idea, giving high school students in a web-development camp a list of requirements for a personal web site they were to create, but also encouraging them to devise self-defined requirements (Loksa et al. 2016). The researchers assessed the complexity and volume of self-defined requirements and analyzed the degree to which students implemented those requirements, and used these to measure productivity over an entire week.

Some research on camps has gone beyond the unit of analysis of a single camp, or even a constellation of camps, investigating entire systems and pipelines of informal learning opportunities. Most notably, the Georgia Computes! project investigated the role of state policy, the interaction between formal and informal learning, and the longitudinal effects of a pipeline of informal learning opportunities on identity and engagement (Guzdial et al., 2014). This type of policy research has led to recommendations about the requirements for success, suggesting the importance of the support of policy stakeholders and partners, of high quality portable resources, of an explicit goal to replicate success across outreach activities, and of multiple levels of details about the system.

3 Strategies for Informal Learning

In this section, we discuss several strategies that learners engage in when learning informally. We start off with self-directed learning, add a social component with peer learning, and move into large-scale community-based involvement in a learner's progression. We end by looking at how teachers can also be informal learners and can take advantage of the same environments and strategies that other learners enjoy.

3.1 Self-directed learning

Only a few works have considered truly self-directed independent learning of computing. McCartney et al., for example, investigated how computer science undergraduate students approach informal, self-directed learning to supplement their formal education (McCartney et al., 2010). They found that students were inspired to learn in order to complete personally meaningful projects, employing a variety of programming languages and technologies. These students chose to work on these projects because they would be relevant to their work,their home lives, or their careers (for example to prepare for future coursework, or to help out friends and family). Boustedt et al. built upon these findings, reporting that while students in school enjoy informal learning because they gain agency over the process, they believe that they miss important aspects of a topic, have difficulty assessing their learning, and miss the structure of school (Boustedt et al. 2011).

Zander et al. studied self-directed learning by focusing on computing professionals. They found that professionals were implicitly expected to learn on their own, and used a range of resources (e.g. Internet search), strategies (e.g. getting help from others, learning by trial and error, breaking problems into subproblems, etc.), and collaborators (for information gathering) to help them in the process. Professionals found their work-related learning to be enjoyable, and expressed a sense of confidence and pride. Yet, they often found informal, self-directed learning to be stressful, describing it as a never-ending process (Zander et al. 2012).

Many studies of informal learning concern adults. For example, many of the studies by Lee et al. (e.g., Lee & Ko, 2011; Lee & Ko, 2015) involved adults seeking opportunities to learn online through coding tutorials. Guo investigated the motivations behind this adult learning, finding that many people over age 60 want to learn to code, but get frustrated by their declining cognitive faculties, their lack of opportunities to interact socially with tutors, peers, and teachers, and their difficulties with constantly-changing software technologies (Guo 2017). Adults can also shift their attitudes about computing quickly. For example, a pre-post attitudinal survey of adults playing the Gidget game rapidly shifted their beliefs about the difficulty of programming from negative to positive after just 15 minutes of play (Charters et al., 2014). Few studies, however, have explored these issues longitudinally.

3.2 Peer learning

As Vygotzky proposed, a lot of learning happens in the company of and due to one's peers. Many studies have reinforced this theoretical claim. For example, while social interactions occur face-to-face and virtually, they all take place in contextually-linked places, such in tutoring centers, at whiteboards, in Facebook groups, or even at home where people can work on projects together (Knox & Fincher, 2013). Klomsri et al. found that South African youths took advantage of the ubiquity of Facebook's social networking to learn about one another's viewpoints, support one another, to share their own content with an audience of their design, and effectively achieve their own goals without much overhead due to any formal pedagogy or structure (Klomsri et al., 2013). Studies of mentoring around computing have found that many adolescents' interest in computing comes from informal peer mentors and not from classes (Ko & Davis, 2017).

Hackathons, large events where people gather to complete projects in collaborative programming teams, offer a significant amount of peer learning. Mentors from tertiary institutions and industry can provide round-the-clock hands-on support, troubleshooting, and advice. Nandi and Mandernach found that undergraduate students participating in hackathons spent quality time practicing the art of working together in teams (Nandi & Mandernach, 2016). Students are motivated to participate in these hackathons primarily for the social appeal of working in a fun environment with new people and new technology (Warner & Guo, 2017). After interviewing 6 hackathon participants, Warner and Guo found that hackathons were perceived by the student to be more authentic, intense, and democratic than classroom learning experiences. Hackathon activities motivate students to learn new skills because of their practical applicability, rather than because they are valued by academics. Working on hackathon projects helps reinforce students' communication skills, while catalyzing their personal motivations and self-confidence to work on personally-relevant projects.

3.3 Engaging with communities of practice

As we have discussed, a lot of informal learning is social. There is some evidence, however, that effective informal learning requires social engagement with not just peers, tutors, or strangers online, but whole communities (Lave & Wenger, 1991). Learners often engage with community members in the "real world" at work, during academically-sponsored service learning opportunities, co-ops, and internships (Fincher & Knox, 2013). Non-work-based contexts occur in many kinds of authentic communities, such as those that spring up around particular application domains, open source projects, and capstone course projects. The popularity and success of the Scratch programming environment (Resnick et al., 2009) has created and supported a community of young learners who “remix” one another’s projects to build their own. Dasgupta et al. found that learners who remix more often have larger repertoires of programming commands even after controlling for the numbers of projects and amount of code shared. They also find that exposure to computational thinking concepts through remixing is associated with increased likelihood of using those concepts (Dasgupta et al. 2016). Another study found that while building off one another’s project helps learners get started, it was not related to using more complex concepts in their Scratch projects (Fields et al., 2014).

Engaging in authentic communities of practice can have both positive and negative effects on learners. For instance, Ellis et al. found that students working on humanitarian-oriented open source projects increased their interest in computing, as they gain experience in developing software in a distributed environment (Ellis et al., 2015). Students improved their performance in attendant skills, such as communication and distributed teamwork. Hislop et al. found, however, that wile engaging in open source projects made students feel more comfortable interacting with professionals, it also made them feel that they knew much less than they thought they did before (Hislop et al., 2015).

Many institutions offer a capstone course to senior undergraduates, in which a team of students works with an outside for-proft or not-for-profit company (e.g., Cicirello, 2013, Stone et al. 2011, Stone et al. 2012). While project specifications come from the outside, students engage in authentic work experiences in the safe, monitored environment of tertiary institutions. The outside partners simultaneously monitor the students’ progress, as they anticipate and eventually receive delivery of the final product. In a report by Bloomfield et al. about the service-learning oriented capstone at the University of Virginia, students connect with local nonprofits to work on meaningful projects with real impact to the community, while learning teamwork, customer management, and organizational skills (Bloomfield et al., 2014). Working with outside partners takes real effort and administrative capabilities from tertiary instutitions, however. Venkatagiri found that implementing a service-oriented capstone in India required the instructor to negotiate appropriate contracts with outside partners to ensure appropriate expectations were communicated along the way. Instructors also had to train students in soft skills, such as effective brainstorming, presenting progress reports, and engaging with customers (Venkatagiri, 2006).

3.4 Teachers as informal learners

Teachers are learners too, of course, and because of the demands on their time, much of the learning they do to teach computing is informal. For example, researchers have created online communities with the goal of supporting informal learning by teachers struggling to master new concepts to deploy in their classrooms. These communities can also be used to share knowledge and support one another’s pedagogy development. Research in this area focuses on building effective communities of practice (Schlager and Fusco, 2003). Booth and Kellogg studied online communities for teachers and found that fostering a diverse population of members with various perspectives and levels of expertise helped one another co-construct new forms of meaning and understanding in ways that were individually and collectively valuable (Booth & Kellogg, 2015).

Designing online communities for promoting teacher learning is not easy. Fincher et al. studied the Nifty Assignments online resource and found that while *acquiring* contributions was effective (because they come from a special session at the yearly SIGCSE computer science education conference), teachers navigating the site had difficulty finding appropriate assignments to use because they preferred to find resources via general web search, rather than browsing through a forum organized by contribution year (Fincher et al., 2010). Teachers found it difficult to identify the pedagogical concepts taught in each assignment and also had to spend time to adapt assignments to their own classrooms. For an online community for the Greenfoot environment, Brown and Kölling compared their new sites' use with three different populations of educators, they found that each population behaved very differently (Brown & Kölling, 2013). Some shared information or announcements much more than others, and some asked domain-specific questions when others did not. Even the kinds of contributions and feedback varied among the populations in ways that the designers of the site could only identify, not influence. Leake and Lewis found similar differences in needs between novice and experienced secondary school computer science instructors (Leake & Lewis, 2017). Novice teachers wanted the ability to build off lessons and resources created by more experienced ones, but simultaneously reported difficulty in adapting those resources to their particular pedagogical contexts.

As Chapter 3.16 notes, engaging high school teachers in online communities is challenging. Howard and McKeown found that site designers found it difficult to engage communities of teachers because their teachers did not perceive the online community as an integral part of their normal work practice (Howard & McKeown, 2011). Leake and Lewis noted that informal learners who are teachers have a difficult time finding appropriate information resources, and do not contribute to them because it takes too much time away from what they perceive as their real job, teaching (Leake & Lewis, 2016). Mitchell and Lutters studied university professors in computer science and found similar results. While most were aware of repositories of instructional materials, only about half had ever used one, and of those who had, most expressed disappointment that the repositories did not meet their needs (Mitchell & Lutters, 2006). Clements et al. classified many different kinds of learning object repositories and suggest that teacher-generated, collaborative, quality instruments are the most sustainable (Clements et al., 2015). Beyond this work, however, there is little design guidance on creating useful informal learning repositories for teachers.

4 Supporting informal learning of computing

As we have discussed, prior work shows that people engage in a wide range of informal learning activities to learn computing, but that many struggle to learn independently.

Consequently, much of the research on informal computing education has focused on designing tools, resources, and experiences that promote longer engagement and better learning.

One form of improvement is offering *new genres* of instructional content. For example, early research, driven by the advent of the internet, explored new web-based multimedia tutoring environments that would provide richer explanations of computers, compilers, and circuits than were possible in a classroom, while also offering automated assessments that would allow learners to be self-paced and independent (Connelly et al., 1996). More recently, researchers have focused on a wide range of new experiences. Some have investigated case-based learning aids that embed instruction in tasks, contextualizing learning to the goals that an independent learner might be trying to achieve (Dorn 2011). Others have explored more interactive tools like PythonTutor that provide deeper visibility into notional machines (e.g. Guo, 2013), allowing students to independently explore the behavior of their own programs. Researchers have also explored a range of programming games that translate tasks in programming and debugging into interactive games that promote learning (Lee et al., 2014, Bishop et al., 2015, Miljanovic & Bradbury, 2017, Tillmann et al., 2011). Others have focused on developing interactive ebooks, including those with embedded program visualizations to contextualize program behavior with other instruction (Sirkia & Sorva, 2015), worked examples that support self-assessment (Ericson et al., 2015, Ericson et al., 2016), and granular interactive explanations of programming language semantics (Nelson et al., 2017). While teachers can use all of these novel genres of interactive instructional content to support formal learning, none of them *require* teachers to be used.

Some research is less focused on inventing new genres of instructional content and more on improving existing genres. For example, a series of studies on the Gidget programming game explored how different design decisions affect discretionary engagement in learning. For example, one study found that by visually representing the robot in the game with an anthropomorphic face and by rewriting error messages to use more collaborative personal pronouns such as “I,” “you,” and “we,” learners were more likely to attend to error messages, learn from them, and therefore master programming language concepts more quickly than learners who interacted with more conventional error messages and a robot with no face (Lee & Ko, 2011). This work was one of the first to frame error messages *as* instructional content. A follow up study found that by making the objects in the game vertebrate objects like cats and mice instead of inanimate objects like rocks, students spend more time learning and complete more exercises in the game (Lee & Ko, 2012). A third study found that incorporating formative assessments in the game led players to voluntarily play longer, and complete levels more quickly, suggesting more efficient learning (Lee et al., 2013). In MOOCs, some researchers have studied the effect of video, tutorial, and quizzes on dropout rates (Kim et al., 2014), finding that learners are deterred by long videos, abrupt transitions, and learning challenges without resources. These studies show that seemingly small factors in the design of materials can greatly impact the quality and duration of discretionary learning.

Because creating and designing effective instructional material for the wide range of concepts in computing can be challenging and slow, researchers have increasingly investigated techniques for automatically generating instruction using intelligent tutoring systems. For example, some have explored ways of semi-automatically generating API tutorials composed of code examples from open source projects on the web (Dahotre et al., 2011, Harms et al., 2013). Preliminary studies of these systems show that they can successfully promote learning, especially relative to fixed media such as textbooks. Others have explored end-user programmers who need to learn a little about programming to help automate a task, embedding end-user software engineering tools that generate context and task relevant instruction on design, reuse, integration, testing, and debugging (Ko et al., 2011). There are hundreds of such systems, each with the primary goal of helping people automate work, but the secondary effect of promoting some learning. For example, the Idea Garden concept explored opportunities to generate contextual problem solving instruction, helping people trying to write simple programs learn problem solving skills that helped them get unstuck on a programming task (Cao et al., 2011).

While some systems have explored generating instruction, others have focused on generating feedback about learners' skills. Cognitive tutors have focused on providing step-by-step feedback and guidance on problem solving (Jin & Corbett, 2011). Environments that gamify programming, inspired by how well video games promote skill mastery through feedback, show stronger learning outcomes than environments with no feedback or guidance (Lee & Ko, 2015). For decades, researchers have explored automated feedback in the context of online courses (Truong, 2005, Fitzpatrick et al., 2017). Unless learners can explain to themselves where this feedback comes from, many learners find automatically generated feedback to be untrustworthy (Kulkarni et al., 2014).

Rather than automate feedback, some researchers have explored ways of scaling peer feedback in informal settings. These include structured peer assessment in basic online forums (Warren et al., 2014), but also a range of new media. For example, Codeopticon let learners simultaneously chat with dozens of other learners, scaling peer feedback (Guo, 2015). Codechella let multiple people write code, visualize run-time state, debug, and chat in real time (Guo et al., 2015), creating a shared visual display of learning dialogue. Codepourri let anonymous learners create and share step-by-step coding tutorials for other learners (Gordon & Guo, 2015). These systems explore new ways to help learners support each other in their informal learning, without the aid or guidance of teachers or automatic feedback systems.

5 Open Questions

As we noted before, research on informal learning of computing is broad, but not deep. Researchers have explored many novel ways to support informal learning of computing, but only a few projects have deeply explored their impact on learning, and few have deeply leveraged theories of learning to inform design. There are also not yet clear best practices for doing research on these topics: the field still lacks robust, valid measures of many of the constructs it seeks to improve, such as learning, interest, and engagement.

Despite this lack of research infrastructure, there are still many urgent open questions about how informal learning unfolds and how to support it. Because of the inherently learner-centric nature of informal learning, many of the most important questions concern how to support learners, if not through a relationship with a teacher in a formal institution of education. For example, should learning technologies structure learning for learners or should learners be taught how to structure their own independent learning? What role can librarians play in helping learners navigate their informal learning? Since learners are often seeking online resources to learn to code, how can they be supported in searching, selecting, and effectively leveraging resources? These questions are important in every setting, whether after school, in a camp, in an online course, or completely separate from a formal learning setting.

Equally important are questions about informal learning resources themselves. How can we know whether a resource is effective? Is it possible to automatically personalize resources so they meet the goals of a specific learner? Is it possible to automatically generate resources to meet the wide range of things that people want to learn about computing, such as new APIs and platforms? How do informal learning materials need to be different from those used in formal education settings? How should resources be maintained and organized? Do they need to provide the same support as a teacher? Can they? Because so much about learning computing involves formal notations, it may be more amenable to automation than many other kinds of learning, but some things, such as a relationship with a trusted, supporting teacher, probably cannot.

Finally, as we noted throughout the chapter, much informal learning *does* involve teachers, framing them more as facilitators and resources rather than authority figures. In these learner-driven settings such as camps and online, is the kind of guidance and support that teachers need to provide different from those of formal classrooms, more akin to mentoring than instructing? And given the scarcity of people with expertise in teaching computing, how can we scale the guidance that teachers provide in formal learning?

Finally, we still know very little about the broader impacts of informal learning of computing. For example, widespread efforts such as Code.org's Hour of Code, and the dozens of online coding tutorials, are engaging hundreds of millions of people, but we still know very little about what anyone learns. Is this knowledge robust? Is it comparable to what is learned in formal settings? And is this informal learning more or less equitable than in formal settings?

We are just at the beginning of understanding how people learn computing outside of school. With further research, we may not only find ways of supporting learners in their self-directed learning more effectively, but also for those in school, how to better integrate their learning across formal and informal settings.

Bibliography

Aritajati, C., Rosson, M. B., Pena, J., Cinque, D., & Segura, A. (2015). A Socio-Cognitive Analysis of Summer Camp Outcomes and Experiences. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15). New York, NY: ACM. 581-586. doi:10.1145/2676723.2677292

Begel, A., Bosch, J., & Storey, M. A. (2013). Social Networking Meets Software Development: Perspectives from GitHub, MSDN, Stack Exchange, and TopCoder. *IEEE Software*, **30**(1) 52-66. doi: 10.1109/MS.2013.13

Bishop, J., Horspool, R.N., Xie, T., Tillmann, N. & de Halleux, J. (2015). Code Hunt: Experience with coding contests at scale. In Proceedings of the 37th IEEE International Conference on Software Engineering. New York, NY: IEEE, 398–407. doi:10.1109/ICSE.2015.172

Bloomfield, A., Sherriff, M., & Williams, K. (2014). A service learning practicum capstone. In Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14). New York, NY: ACM, 265-270. doi:10.1145/2538862.2538974

Booth, S. E., & Kellogg, S. B., (2015). Value creation in online communities for educators. *British Journal of Educational Technology*, **46**(4), 684-698.

Boustedt, J., Eckerdal, A., McCartney, R., Sanders, K., Thomas, L., & Zander, C. (2011). Students' perceptions of the differences between formal and informal learning. In Proceedings of the seventh international workshop on Computing education research (ICER '11). New York, NY: ACM. 61-68. doi:10.1145/2016911.2016926

Bower, M. (2008). The "instructed-teacher": a computer science online learning pedagogical pattern. In Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '08). New York, NY: ACM. 189-193. doi:10.1145/1384271.1384323

Brandt, J., Guo, P.J., Lewenstein, J., Dontcheva, M., & Klemmer, S.R. (2009). Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09). New York, NY: ACM. 1589-1598. doi:10.1145/1518701.1518944

Brown, N.C.C., & Kölling, M. (2013). A tale of three sites: resource and knowledge sharing amongst computer science educators. In Proceedings of the Ninth annual international ACM conference on International computing education research (ICER '13). New York, NY: ACM. 27-34. doi:10.1145/2493394.2493398

Cao, J., Fleming, S.D., & Burnett, M. (2011). An exploration of design opportunities for “gardening” end-user programmers' ideas. In Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), New York, NY: IEEE, 2011, 35-42.

Carswell, L. (1997). Teaching via the Internet: the impact of the Internet as a communication medium on distance learning introductory computing students. In Proceedings of the 2nd conference on Integrating technology into computer science education (ITiCSE '97). New York, NY: ACM. 1-5. doi:10.1145/268819.268821

Carswell, L. (1998). The “Virtual University”: toward an Internet paradigm?. In Proceedings of the 6th annual conference on the teaching of computing and the 3rd annual conference on Integrating technology into computer science education: Changing the delivery of computer science education (ITiCSE '98). New York, NY: ACM. 46-50. doi:10.1145/282991.283017

Chambers, C., Chen, S., Le, D., & Scaffidi, C. (2012). The function, and dysfunction, of information sources in learning functional programming. *Journal of Computing Sciences in College.* **28**(1). 220-226.

Charters, P., Lee, M. J., Ko, A. J., & Loksa, D. (2014). Challenging Stereotypes and Changing Attitudes: The Effect of a Brief Programming Encounter on Adults' Attitudes toward Programming. In Proceedings of the 45th ACM technical symposium on Computer science education, New York, NY: ACM. 653-658.

Cho, M. H., Demei, S., & Laffey, J. (2010). Relationships between self-regulation and social experiences in asynchronous online learning environments. *Journal of Interactive Learning Research*, **21**(3), 297-316.

Cicirello, V.A. (2013). Experiences with a real projects for real clients course on software engineering at a liberal arts institution. *Journal of Computing Sciences in College.* **28**(6), 50-56.

Clements, K., Pawlowski, J., & Manouselis, N. (2015). Open educational resources repositories literature review - Towards a comprehensive quality approaches framework. *Computers in Human Behavior* **51**(B), 1098-1106. doi:10.1016/j.chb.2015.03.026

Connelly, C., Biermann, A. W., Pennock, D., & Wu, P. (1996). Home Study Software: Complementary Systems for Computer Science Courses. *Computer Science Education* **7**(1). doi:10.1080/0899340960070104

Cooper, S., & Sahami, M. (2013). Reflections on Stanford's MOOCs. *Communications of the ACM*, **56**(2), 28-30.

Crellin, J., Duke-Williams, E., Chandler, J., & Collinson, T. (2009). Virtual worlds in computing education. *Computer Science Education*, **19**(4).

Dabbagh, N., & Kitsantas, A. (2012). Personal Learning Environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning. *The Internet and higher education*, **15**(1), 3-8.

Dahotre, A., Krishnamoorthy, V., Corley, M., & Scaffidi, C. (2011). Using intelligent tutors to enhance student learning of application programming interfaces. Journal of Computing Sciences in College. **27**(1). 195-201.

Dankell, D. D., II, & Hearn, J. (1997). The use of the WWW to support distance learning through NTU. In Proceedings of the 2nd conference on Integrating technology into computer science education (ITiCSE '97). New York, NY: ACM. 8-10. doi:10.1145/268819.268824

Dasgupta, S., Hale, W., Monroy-Hernández, A., & Hill, B. M. (2016). Remixing as a pathway to computational thinking. In Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing. New York, NY: ACM. 1438-1449.

Decker, A., McGill, M. M., & Settle, A. (2016). Towards a Common Framework for Evaluating Computing Outreach Activities. In Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16). New York, NY: ACM. 627-632. doi:10.1145/2839509.2844567

DeWitt, A., Fay, J., Goldman, M., Nicolson, E., Oyolu, L., Resch, L., Saldaña, J. M., Sounalath, S., Williams, T., Yetter, K., Zak, E., Brown, N., & Rebelsky, S. A. (2017). What We Say vs. What They Do: A Comparison of Middle-School Coding Camps in the CS Education Literature and Mainstream Coding Camps (Abstract Only). In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). New York, NY: ACM. 707-707. doi:10.1145/3017680.3022434

DiSalvo, B., Reid, C., & Roshan, P. K. (2014). They can't find us: the search for informal CS education. In Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14). New York, NY: ACM. 487-492. doi:10.1145/2538862.2538933

Dorn, B., & Guzdial, M. (2010). Learning on the job: characterizing the programming knowledge and learning strategies of web designers. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). New York, NY: ACM. 703-712. doi:10.1145/1753326.1753430

Dorn, B. (2011). ScriptABLE: supporting informal learning with cases. In Proceedings of the seventh international workshop on Computing education research (ICER '11). New York, NY: ACM. 69-76. doi:10.1145/2016911.201692

Dorn, B., Stankiewicz, A., & Roggi, C. (2013). Lost while searching: difficulties in information seeking among end-user programmers. In Proceedings of the 76th ASIS&T Annual Meeting: Beyond the Cloud: Rethinking Information Boundaries (ASIST '13), Silver Springs, MD: Association for Information Science and Technology.

Ellis, H. J. C., Hislop, G. W., Jackson, S., & Postner, L. (2015). Team Project Experiences in Humanitarian Free and Open Source Software (HFOSS). *Transactions on Computing Education.* **15**(4). doi:10.1145/2684812

Ericson, B. J., & McKlin, T. (2012). Effective and sustainable computing summer camps. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (SIGCSE '12). New York, NY: ACM. 289-294. doi:10.1145/2157136.2157223

Ericson, B. J., Guzdial, M. J., & Morrison, B. B. (2015). Analysis of Interactive Features Designed to Enhance Learning in an Ebook. In Proceedings of the eleventh annual International Conference on International Computing Education Research (ICER '15). New York, NY: ACM. 169-178. doi:10.1145/2787622.2787731

Ericson, B. J., Rogers, K., Parker, M. Morrison, B., & Guzdial, M. (2016). Identifying Design Principles for CS Teacher Ebooks through Design-Based Research. In Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16). New York, NY: ACM. 191-200. doi:10.1145/2960310.2960335

Fields, D. A., Giang, M., & Kafai, Y. (2014). Programming in the Wild: Trends in Youth Computational Participation in the Online Scratch Community. In Proceedings of the 9th Workshop in Primary and Secondary Computing Education (WiPSCE ’14). New York, NY: ACM. 2–11.

Fincher, S., & Knox, D. The Porous Classroom: Professional Practices in the Computing Curriculum. In *Computer*, vol. 46, no. 9, pp. 44-51, IEEE. 2013. doi:10.1109/MC.2013.261

Fincher, S., Kölling, M., Utting, I., Brown, N., & Stevens, P. (2010). Repositories of teaching material and communities of use: nifty assignments and the greenroom. In Proceedings of the Sixth international workshop on Computing education research (ICER '10). New York, NY: ACM. 107-114. doi:10.1145/1839594.1839613

Fitzpatrick, J. M., Lédeczi, Á., Narasimham, G., Lafferty, L., Labrie, R., Mielke, P. T., Kumar, A., & Brady, K. A. (2017). Lessons Learned in the Design and Delivery of an Introductory Programming MOOC. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). New York, NY: ACM. 219-224. doi:10.1145/3017680.3017730

Gal-Ezer, J., Vilner, T., & Zur, E. (2009). The professor on your PC: a virtual CS1 course. In Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education (ITiCSE '09). New York, NY: ACM. 191-195. doi:10.1145/1562877.1562938

Gersting, J. L. (2000). Computer Science Distance Education Experience in Hawaii. Computer Science Education, 10(1), 95-106.

Gordon, M. & Guo, P. J. (2015). Codepourri: Creating visual coding tutorials using a volunteer crowd of learners. In 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 13-21.

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, **25**(2), 199-237.

Guo, P. J. (2013). Online Python tutor: embeddable web-based program visualization for CS education. In Proceedings of the 44th ACM technical symposium on Computer science education. New York, NY: ACM. 579-584.

Guo, P. J. & Reinecke, K. (2014). Demographic differences in how students navigate through MOOCs. In Proceedings of the first ACM conference on Learning@ scale conference. New York, NY: ACM. 21-30.

Guo, P. J., Kim, J., & Rubin, R. (2014). How video production affects student engagement: An empirical study of mooc videos. In Proceedings of the first ACM conference on Learning@ scale conference. New York, NY: ACM. 41-50.

Guo, P. J. (2015). Codeopticon: Real-time, one-to-many human tutoring for computer programming. In Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology. New York, NY: ACM. 599-608.

Guo, P. J., White, J., & Zanelatto, R. (2015). Codechella: Multi-user program visualizations for real-time tutoring and collaborative learning. In 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). 79-87.

Guo, P. J. (2017). Older Adults Learning Computer Programming: Motivations, Frustrations, and Design Opportunities. ACM Conference on Human Factors in Computing Systems (CHI). New York, NY: ACM, 7070-7083. doi:10.1145/3025453.3025945.

Guzdial, M., Ericson, B., Mcklin, T., & Engelman, S. (2014). Georgia Computes! An Intervention in a US State, with Formal and Informal Education in a Policy Context. *Transactions on Computing Education*. **14**(2), Article 13, 29 pages. doi:10.1145/2602488

Hao, Q., Wright, E., Barnes, B., & Branch, R. M. (2016). What are the most important predictors of computer science students' online help-seeking behaviors?. *Computers in Human Behavior*. **62**(C), 467-474. doi:10.1016/j.chb.2016.04.016

Harms, K. J., Cosgrove, D., Gray, S., & Kelleher, C. (2013). Automatically generating tutorials to enable middle school children to learn programming independently. In Proceedings of the 12th International Conference on Interaction Design and Children (IDC '13). New York, NY: ACM. 11-19. doi:10.1145/2485760.2485764

Hislop, G. W., Ellis, H. J. C., Pulimood, S. M., Morgan, B., Mello-Stark, S., Coleman, B., & Macdonell, C. (2015). A Multi-Institutional Study of Learning via Student Involvement in

Humanitarian Free and Open Source Software Projects. In Proceedings of the eleventh annual International Conference on International Computing Education Research (ICER '15). New York, NY: ACM. 199-206. doi:10.1145/2787622.2787726

Hitz, M., & Kögeler, S. (1997). Teaching C++ on the WWW. In Proceedings of the 2nd conference on Integrating technology into computer science education (ITiCSE '97), New York, NY: ACM. 11-13. doi:10.1145/268819.268825

Jaccheri, L., & Osterlie, T. (2007). Open Source Software: A Source of Possibilities for Software Engineering Education and Empirical Software Engineering. In Proceedings of the First International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS '07). Washington, DC, USA, 5-5. doi:10.1109/FLOSS.2007.12

Howard, L., Johnson, J., & Neitzel, C. (2010). Reflecting on online learning designs using observed behavior. In Proceedings of the fifteenth annual conference on Innovation and technology in computer science education (ITiCSE '10). New York, NY: ACM. 179-183. doi:10.1145/1822090.1822142

Howard, S. & McKeown, J. (2011). Online practice & offline roles: A cultural view of teachers' low engagement in online communities. American Educational Research Association.

Jin, W., & Corbett, A. (2011). Effectiveness of cognitive apprenticeship learning (CAL) and cognitive tutors (CT) for problem solving using fundamental programming concepts. In Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE '11). New York, NY: ACM. 305-310. doi:10.1145/1953163.1953254

Jones, M. C., & Churchill, E. F. (2009). Conversations in developer communities: a preliminary analysis of the yahoo! pipes community. In Proceedings of the Fourth International Conference on Communities and technologies (C&T '09). New York, NY: ACM. 195-204. doi:10.1145/1556460.1556489

Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling alice motivates middle school girls to learn computer programming. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07). New York, NY: ACM. 1455-1464. doi:10.1145/1240624.1240844

Kim, J., Guo, P. J., Seaton, D. T., Mitros, P., Gajos, K. Z., & Miller, R. C. (2014). Understanding in-video dropouts and interaction peaks in online lecture videos. In Proceedings of the First ACM conference on Learning@ scale conference. New York, NY: ACM. 31-40.

Klomsri, T., Grebäck, L., & Tedre, M. (2013). Social media in everyday learning: How Facebook supports informal learning among young adults in South Africa. In Proceedings of the 13th Koli Calling International Conference on Computing Education Research (Koli Calling '13). New York, NY: ACM, 135-144. doi:10.1145/2526968.2526983

Knox, D., & Fincher, S.. (2013). Where students go for knowledge and what they find there. In Proceedings of the Ninth annual international ACM conference on International computing education research (ICER '13). New York, NY: ACM. 35-40. doi:10.1145/2493394.2493399

Knowles, M. S. (1975). *Self-Directed Learning: a guide for learners and teachers.* New York: Associated Press.

Ko, A. J., Myers, B. A., Aung, H. H. (2004). Six learning barriers in end-user programming systems. In 2004 IEEE Symposium on Visual Languages and Human Centric Computing, New York, NY: IEEE, 199-206. doi:10.1109/VLHCC.2004.47

Ko, A. J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaffidi, C., Lawrance, J., Lieberman, H., Myers, B., Rosson, M. B., Rothermel, G., Shaw, M., & Wiedenbeck, S. (2011). The state of the art in end-user software engineering. *ACM Computing Surveys*. **43**(3), Article 21, 44 pages.

Ko, A. J. & Davis, K. (2017). Computing Mentorship in a Software Boomtown: Relationships to Adolescent Interest and Beliefs. In Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17). New York, NY: ACM. 236-244. doi:10.1145/3105726.3106177

Koppelman, H., & Vranken, H. (2008). Experiences with a synchronous virtual classroom in distance education. In Proceedings of the 13th annual conference on Innovation and technology in computer science education (ITiCSE '08). New York, NY: ACM. 194-198. doi:10.1145/1384271.1384324

Kulkarni, C. E., Socher, R., Bernstein, M.S., & Klemmer, S.R. (2014). Scaling short-answer grading by combining peer assessment with algorithmic scoring. In Proceedings of the first ACM conference on Learning@ scale conference (pp. 99-108).

Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation.* Cambridge, UK: Cambridge University Press.

Leake, M., & Lewis, C. (2016). Designing a New System for Sharing Computer Science Teaching Resources. In Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion (CSCW '16 Companion). New York, NY: ACM. 321-324. doi:10.1145/2818052.2869109

Leake, M., & Lewis, C. M. 2017. Recommendations for Designing CS Resource Sharing Sites for All Teachers. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). New York, NY: ACM. 357-362. doi:10.1145/3017680.3017780

Leander, K. M., Phillips, N.C., & Taylor, K.H. (2010). The changing social spaces of learning: Mapping new mobilities. *Review of research in education*, **34**(1), 329-394. doi:10.3102/0091732X09358129

Lee, M. J., & Ko, A. J. (2011). Personifying Programming Tool Feedback Improves Novice Programmers' Learning. International Computing Education Research Workshop (ICER), New York, NY: ACM, 109-116.

Lee, M. J., & Ko, A. J. (2012). Investigating the Role of Purposeful Goals on Novices' Engagement in a Programming Game. IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), New York, NY: IEEE, 163-166.

Lee, M. J., Ko, A. J., Kwan, I. (2013). In-game assessments increase novice programmers' engagement and level completion speed. In Proceedings of the ninth annual international ACM conference on International computing education research. 153-160.

Lee, M. J., & Ko, A. J. (2015). Comparing the effectiveness of online learning approaches on CS1 learning outcomes. In Proceedings of the Eleventh Annual International Conference on International Computing Education Research. New York, NY: ACM, 237-246.

Lehtonen, T., Aho, T., Isohanni, E., & Mikkonen, T. (2015). On the role of gamification and localization in an open online learning environment: javala experiences. In Proceedings of the 15th Koli Calling Conference on Computing Education Research (Koli Calling '15). New York, NY: ACM. 50-59. doi:10.1145/2828959.2828973

Loksa, D., Ko, A. J., Jernigan, W., Oleson, A., Mendez, C. J., & Burnett, M. M. (2016). Programming, Problem Solving, and Self-Awareness: Effects of Explicit Guidance. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). New York, NY: ACM. 1449-1461. doi:10.1145/2858036.2858252

Malan, D. J. (2009). Virtualizing office hours in CS 50. In Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education (ITiCSE '09). New York, NY: ACM. 303-307. doi:10.1145/1562877.1562969

Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., & Hartmann, B. (2011). Design lessons from the fastest Q&A site in the west. In Proceedings of the SIGCHI conference on Human factors in computing systems. New York, NY: ACM, 2857-2866.

Marmorstein, R. (2011). Open source contribution as an effective software engineering class project. In Proceedings of the 16th annual joint conference on Innovation and technology in computer science education (ITiCSE '11). New York, NY: ACM. 268-272. doi:10.1145/1999747.1999823

Marsick, V. J., & Watkins, K. E. (2001). Informal and incidental learning. *New directions for adult and continuing education*, **2001**(89), 25-34. doi:10.1002/ace.5

Marsick, V. J., & Volpe, M. (1999). The Nature of and Need for Informal Learning. In V. J. Marsick and M. Volpe (eds.), *Informal Learning on the Job, Advances in Developing Human Resources, No. 3* (pp. 1-9). San Francisco: Berrett Koehler.

McCartney, R., Eckerdal, A., Moström, J. E., Sanders, K., Thomas, L., & Zander, C. (2010). Computing students learning computing informally. In Proceedings of the 10th Koli Calling International Conference on Computing Education Research (Koli Calling '10). New York, NY: ACM. 43-48. doi:10.1145/1930464.1930470

McGill, M. M., Decker, A., & Settle, A. (2015). Does Outreach Impact Choices of Major for Underrepresented Undergraduate Students?. In Proceedings of the eleventh annual International Conference on International Computing Education Research (ICER '15). New York, NY: ACM. 71-80. doi:10.1145/2787622.2787711

Menard, S. A. W. (1993). Critical Learning Incidents of Female Army Nurse Vietnam Veterans and Their Perceptions of Organizational Culture in a Combat Area. (Women Vetersans, Nurses). Ph.D. Dissertation. University of Texas, Austin.

Miljanovic, M. A., & Bradbury, J. S. (2017). RoboBUG: A Serious Game for Learning Debugging Techniques. In Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17). New York, NY: ACM. 93-100. doi:10.1145/3105726.3106173

Mitchell, S. M., & Lutters, W. G. (2006). Assessing the Value of Computer Science Course Material Repositories. In Proceedings of the 19th Conference on Software Engineering Education and Training Workshops (CSEETW '06). New York, NY: IEEE, 2-2. doi:10.1109/CSEETW.2006.6

Moraveji, N., Morris, M., Morris, D., Czerwinski, M., & Riche, N. H. (2011). ClassSearch: facilitating the development of web search skills through social learning. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). New York, NY: ACM. 1797-1806. doi:10.1145/1978942.1979203

Muilenburg, L. Y., & Berge, Z.L. (2005). Student barriers to online learning: A factor analytic study. *Distance Education*, **26**(1), 29-48.

Nandi, A., & Mandernach, M. (2016). Hackathons as an Informal Learning Platform. In Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16). New York, NY: ACM. 346-351. doi:10.1145/2839509.2844590

Nelson, G. L., Xie, B., & Ko, A. J. (2017). Comprehension First: Evaluating a Novel Pedagogy and Tutoring System for Program Tracing in CS1. In Proceedings of the 2017 ACM Conference on International Computing Education Research (pp. 2-11). New York, NY: ACM.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York, NY: Basic Books, Inc.

Park, T. H., & Wiedenbeck, S. (2011). Learning web development: challenges at an earlier stage of computing education. In Proceedings of the seventh international workshop on

Computing education research (ICER '11). New York, NY: ACM. 125-132. doi:10.1145/2016911.2016937.

Pullen, J. M. (2006). Scaling up a distance education program in computer science. In Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education (ITICSE '06). New York, NY: ACM. 33-37. doi:10.1145/1140124.1140136

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Milner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, **52**(11), 60-67.

Rosbottom, J. (2001). Hybrid learning - a safe route into web-based open and distance learning for the computer science teacher. In Proceedings of the 6th annual conference on Innovation and technology in computer science education (ITiCSE '01). New York, NY: ACM. 89-92. doi:10.1145/377435.377493

Sadowski, C., Stolee, K. T., & Elbaum, S. (2015). How developers search for code: a case study. In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2015). New York, NY: ACM. 191-201. doi:10.1145/2786805.2786855

Schlager, M. S., & Fusco, J. (2003). Teacher professional development, technology, and communities of practice: Are we putting the cart before the horse? *The information society*, **19**(3), 203-220.

Searle, K. A., & Kafai, Y. B. (2015). Boys' Needlework: Understanding Gendered and Indigenous Perspectives on Computing and Crafting with Electronic Textiles. In Proceedings of the eleventh annual International Conference on International Computing Education Research (ICER '15). New York, NY: ACM. 31-39. doi:10.1145/2787622.2787724

Siemens, G. (2005). Connectivism: A learning theory for the digital age. *International Journal of Instructional Technology and Distance Learning*. **2**(1). 3-10.

Sirkiä, T., & Sorva, J. (2015). How Do Students Use Program Visualizations within an Interactive Ebook?. In Proceedings of the eleventh annual International Conference on International Computing Education Research (ICER '15). New York, NY: ACM. 179-188. doi:10.1145/2787622.2787719

Stone, J. A., & Madigan, E. (2011). Experiences with community-based projects for computing majors. Journal of Computing Sciences in Colleges. **26**(6). 64-70.

Stone, J. A., MacKellar, B., Madigan, E. M., & Pearce, J. L. (2012). Community-based projects for computing majors: opportunities, challenges and best practices. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (SIGCSE '12). New York, NY: ACM. 85-86. doi:10.1145/2157136.2157166

Tillmann, N., De Halleux, J., Xie, T., Gulwani, S., & Bishop, J. (2013). Teaching and learning programming and software engineering via interactive gaming. In Proceedings of the 2013 International Conference on Software Engineering (ICSE '13). New York, NY: ACM and IEEE, 1117-1126.

Truong, N., Bancroft, P., & Roe, P. (2005). Learning to program through the web. In Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '05). New York, NY: ACM. 9-13.

Venkatagiri, S. (2006). Engineering the software requirements of nonprofits: a service-learning approach. In Proceedings of the 28th international conference on Software engineering (ICSE '06). New York, NY: ACM. 643-648. doi:10.1145/1134285.1134382

von Wright, J. (2000). Distance tutorials in a systems design course. In Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education (ITiCSE '00). New York, NY: ACM. 105-107. doi:10.1145/343048.343123

Wagner, A., Gray, J., Corley, J., & Wolber, D. 2013. Using app inventor in a K-12 summer camp. In Proceedings of the 44th ACM technical symposium on Computer science education (SIGCSE '13). New York, NY: ACM. 621-626. doi:10.1145/2445196.2445377

Warner, J., Doorenbos, J., Miller, B., & Guo, P. J. (2015). How High School, College, and Online Students Differentially Engage with an Interactive Digital Textbook. In Proceedings of the 8th International Conference on Educational Data Mining. International: International Educational Data Mining Society, 528-531.

Warner, J., & Guo, P. J.. (2017). Hack.edu: Examining How College Hackathons Are Perceived By Student Attendees and Non-Attendees. In Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17). New York, NY: ACM. 254-262. doi:10.1145/3105726.3106174

Warren, J., Rixner, S., Greiner, J., & Wong, S. (2014). Facilitating human interaction in an online programming course. In Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14). New York, NY: ACM. 665-670. doi:10.1145/2538862.2538893

Webb, H. C., & Rosson, M. B. (2011). Exploring careers while learning Alice 3D: a summer camp for middle school girls. In Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE '11). New York, NY: ACM. 377-382. doi:10.1145/1953163.1953275

Zander, C., Boustedt, J., Eckerdal, A., McCartney, R., Sanders, K., Moström, J. E., & Thomas, L. (2012). Self-directed learning: stories from industry. In Proceedings of the 12th Koli Calling International Conference on Computing Education Research (Koli Calling '12). New York, NY: ACM. 111-117. doi:10.1145/2401796.2401810

Zheng, S., Rosson, M. B., Shih, P. C., & Carroll, J. M. (2015). Understanding Student Motivation, Behaviors and Perceptions in MOOCs. In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15). New York, NY: ACM. 1882-1895. doi:10.1145/2675133.2675217

Zhu, J., Warner, J., Gordon, M., White, J., Zanelatto, R., & Guo, P. J. (2015). Toward a domain-specific visual discussion forum for learning computer programming: An empirical study of a popular MOOC forum. In IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). New York, NY: IEEE, 101-109.