Robotic System Design Principles for Human-Human Collaboration

Samantha Speer

CMU-RI-TR-25-105
December 2025

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Melisa Orta Martinez, Co-Chair
Illah Nourbakhsh, Co-Chair
James McCann
Kylie Peppler, University of California, Irvine

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Keywords: Robotics, Education, Collaboration, Collaborative Learning, Design

For Gigi, we have come so far.

Abstract

Robots possess unique affordances granted by combining software and hardware. Most existing research focuses on the impact of these affordances on human-robot collaboration, but the theory of how robots can facilitate human-human collaboration is underdeveloped. Such a theory could be beneficial in education. An educational device can afford collaboration in both assembly and use. This thesis will enumerate and validate the design principles of educational devices that facilitate collaborative assembly and collaborative work.

This research draws upon cognitive theories used in the disciplines of Computer-Supported Collaborative Work (CSCW), Computer-Supported Collaborative Learning (CSCL), Educational Robotics, and Human-Robot Interaction (HRI). Each discipline uses theories that align with its respective goals to model different pieces of cognition. However, they do not consider other factors outside their respective goals. Diverse analytical lenses are needed to understand the multiple dimensions of influence an educational device can have on human-human interaction to support collaborative assembly and collaborative work.

We explore these dimensions first through the development and assessment of RoboLoom, a robotic Jacquard loom kit designed for interdisciplinary, collaborative education. Through the study of RoboLoom's use and assembly in an undergraduate course, we extract design features that facilitate student-student collaboration during classroom activities. These features encompass task complexity, task parallelization, physicality, repetition of tasks, specificity of hardware, and familiarity with hardware.

We then explore these design principles through three studies: a comparison between two different looms, a study of devices designed for and against the principles, and a comparison of two versions of RoboLoom. We find five design principles that influence collaborative behavior: repetitiveness, specificity, parallelizability, physicality, and difficulty. These design principles were shown to causally change collaborative behaviors in controlled lab settings and in situ engineering education tasks. By evaluating these systems through multiple cognitive lenses, we determine that these design principles are effective in facilitating collaborative assembly and promising for collaborative learning.

Acknowledgments

I want to take this section to acknowledge all of those who helped me and supported me through the process of getting my PhD. Establishing myself as a researcher and building the foundations for my future career took a village of people helping me both professionally and personally. To all those who have supported me in ways both big and small, thank you for shaping my experience here at CMU. I list a few people and groups below, but know that if you're not in this list, if we talked for even five minutes, I appreciate your impact on my experience as well.

Firstly, I would like to acknowledge my committee members. Illah, thank you for being a wonderful advisor and mentor over these many years. I am grateful to have had someone to guide my research journey who shares my goals of meaningful impact on the local community. Melisa, thank you for financing my PhD and introducing me to the wonderful RoboLoom project, which formed the basis of my thesis work. Jim, thank you for welcoming me into your lab, inviting me to lunch, and discussing any and all research topics with me. I'm grateful for the diverse perspectives and advice you've given me along the way. Kylie, thank you for the wisdom and support you have given me. Thank you for so quickly welcoming me into the world of the learning sciences and making me feel at home there.

Next, I would like to thank my family. Thank you to my mom and dad for raising me to have such an incredible sense of curiosity, to value learning and sharing that knowledge with the world, and to value the scientific process and the search for truth. Josh, thanks for never being a perfect brother but always being a great one, and for listening and understanding my frustrations about the PhD program and the world. Gigi, thank you for always loving and supporting me, for all the talks about the bad old times and the hope to make them better, and thank you for raising me to believe that I could do or be anything I wanted, gender be damned. I could only have gotten through this program knowing that my family had my back if all else failed, so thank you. And to all of the islanders, thank you for being my community and making my escape my home.

And finally, I would like to thank my friends, who often are close enough to be family. Preksha, thank you for being here through everything, for always listening, and for letting me speak first draft. To Val and Angle, thank you for the entertainment (pending survival), for listening to me rant, for visiting, for never keeping me sane but always keeping me company, and for all the cat pictures. Ananya, thank you for being there when things were at their worst, for listening, and for always giving me a hug when I needed one. To Sarah, Bart, and Tejus, thank you for being my friends, for being my lunch buddies, and for keeping me going when I wanted to stop. To the pool gang, thanks for all the games and distractions to remind me there's always a way to procrastinate my work. To the rest of my CMU friends, Abby, Ben, Dan, Mrinal, Rayna, and Viv, thanks for making campus a friendlier place for me, and thanks for all the debauchery across the years to help me cut loose. To my normal friends, Simone, Madeline, Em, Bridget, Shinj, and Justine, thank you for reminding me that there is a world beyond my PhD. Thank you for letting me into your lives and for listening to me bemoan my own, for the trips and the escapes, and for being a safe space to escape to. To the Animal Alphabet Pals, Oscar, Madeline, Ben, and Carolyn, thank you for the “D&D” and for the D&D where we could pretend the world was nice, unions could exist, and police could be the good guys.

You all know that words are not my strong suit, so hopefully you understand when I say I could not fully capture how much all of you mean to me. This has been an incredibly formative experience, and I thank you all for being a part of it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In his book "The Design of Everyday Things", Don Norman observes that "bad design ... screams out its inadequacies, making itself very noticeable." [73] When this occurs with everyday objects, we may get frustrated or curse the designers and move on with our day. However, when this occurs in a learning setting, it can have a large negative impact on how the learner feels about their talents and capabilities [83]. A young girl may think she's bad at technology or math because a math problem website is confusing to use. If a group tries to use a tool designed for individual use, all but one participant in the group will become disengaged or resigned as they perceive themselves to be superfluous. These bad designs are screaming their inadequacies, but not in a language some learners can yet understand [73]. People tend to blame themselves for difficulties in interactions, especially "if the task feels simple or trivial" [73]. This can lead to discouraging emotional states that leave learners with negative impressions of the subject they are learning [83].

Collaboration is an important skill for students to learn to help prepare them for their future work in their careers in industry or academia [1, 2, 3]. Despite this, many students have a story about a bad experience with group work. These negative student affects can have a negative impact on the learning outcomes and readiness to collaborate in future work [75, 83]. Thus, the tools students use to collaborate need to be carefully designed so they facilitate collaboration and do not hinder it or leave students with a negative impression [75, 83].

The fields of computer-supported collaborative work (CSCW) and computer-supported collaborative learning (CSCL) recognize the effects of computing systems on the accomplishment of a shared goal by a group of people. While CSCW and CSCL are relatively new fields, much research has been done on how software and computing technologies can be designed with affordances to support good collaboration. However, the technological focus of CSCW and CSCL is primarily non-tangible. Within these fields, there is still a lack of exploration of how robotic systems can be designed with affordances to support collaborative work. Other fields like Educational Robotics and Human-Robot Interaction have investigated design principles for robotic systems, but lack the analytical lens to account for collaborative outcomes.

Hence, in this thesis, I will combine the cognitive theories and analytical lenses of all four fields of CSCW, CSCL, Educational Robotics, and HRI in order to ascertain design principles and guidelines to ensure assemblable, educational devices hold affordances to support collaborative work in educational settings. At a high level, designing for human interaction is about ensuring the artifact's design matches the needs and capabilities of the intended audience. Designing for interaction requires the ability to understand people's needs and capabilities, or to understand how they think. Different fields use different theories of cognition to model people's thoughts as an explanation for their interactions, each choosing cognitive theories that place their target interaction(s) at the forefront of the analysis of human thought. I will analyze data through the lens of these theories, allowing me to create different objectives, which, when positioned together, create a set of design guidelines.

My thesis answers three main questions:

RQ1. What features of a robotic system help facilitate collaborative assembly?

RQ2. What features of a robotic system help facilitate collaborative experiences in a classroom?

RQ3. What are the design principles for assemblable devices designed for human-human collaboration during assembly?

My thesis work centers around the design of a Jacquard loom kit for collaborative educational use (RoboLoom). Through this work, I have analyzed the collaborative building of the device and found that there are many ways in which hardware could be designed to support collaboration, including the difficulty and familiarity of the hardware components, the parallelizability and repetitiveness of the assembly tasks, the physical space, and the specificity of the hardware. Not only do these individual design guidelines have to be optimized within their own design space, but the interaction of each with the others must be considered when designing a hardware system.

We first studied RoboLoom in a pilot course to understand the ways in which students collaboratively interacted with the device. Through observation, student reflections, and interviews analyzed with thematic analysis, we found that the design elements of the loom supported students' abilities to assemble and work on RoboLoom. Chapter 4 discussed this design, and Chapter 5 presents the results of the collaborative experiences during assembly to begin to answer RQ1 and the collaborative educational experiences during the course to answer RQ2.

To further understand the hardware design principles that influence collaborative assembly, we conducted a comparative study between a loom designed for collaborative assembly (RoboLoom) and one designed for individual assembly (the Ashford Shaft Loom). I developed a coding scheme to capture the collaborative actions of participants as they assembled the looms. Using this coding scheme, I found there were differences in collaboration across the two looms. Using thematic analysis, I found that there were five categories of design features that differed across the two devices when collaboration differed. These five design features are: repetitiveness, specificity, parallelizability, physicality, and difficulty. These features answer RQ1, as they are the features that participants interacted with when collaboratively assembling the looms.

I validated these five design principles through two studies: one in a controlled setting and one in a real-world setting. The first study explored the causal relationship between small devices designed for and against the design principles and the collaborative behaviors between pairs of participants. The second study explored how the design of RoboLoom, when prioritizing different design principles, affected the collaborative assembly of undergraduate students in a classroom. We found that the less parallel and less specific version of RoboLoom led to less cooperation on different steps and more knowledge co-creation during assembly. From these studies, I show that there is evidence that these design principles influence collaboration. These studies provide causal evidence to support RQ3 and determine the design principles that can be used to create assemblable devices for human-human collaboration.

The main contributions of my dissertation are fourfold. First, I present the design of RoboLoom, a robotic Jacquard loom kit designed for collaborative education in the disciplines of mathematics, engineering, and weaving. Secondly, I present a pilot study deploying RoboLoom in an undergraduate course designed to teach mathematics, engineering, and weaving. Thirdly, I present a codebook to categorize and understand collaborative behaviors. Finally, I present a set of design principles validated through a set of three studies.

This dissertation presents the necessary background and studies to answer my research questions. Chapter 2 discusses the cognitive theories and their applications to the fields of CSCW, CSCL, Educational Robotics, and HRI. Chapter 3 elucidates the methods and approaches I use in my dissertation. Chapter 4 disseminates the initial design of RoboLoom and explains the interdisciplinary nature of the system. Chapter 5 answers RQ1 and RQ2 through the study of the collaborative assembly of and collaborative experiences in a classroom with RoboLoom during a pilot undergraduate class. Chapter 6 answers RQ1 by exploring the design features of RoboLoom and a commercially available loom and how they impact collaborative actions. Chapter 6 also disseminates the coding scheme used to analyze collaborative behaviors. Chapter 7 explores the design features causal relationship with collaborative actions to answer RQ3. Chapter 8 studies if this causal relationship holds in an educational setting to further add evidence to answer RQ3. Finally, Chapter 9 discusses the design principles that influence collaborative behavior and future studies to further understand the impact of design on collaboration. Chapter 10 concludes my dissertation and states the found design principles as well as final future work.

Chapter 2

Background and Related Work

This chapter serves to discuss the background research and related work that provides the theoretical basis for the thesis work, described in future chapters. I begin this chapter with an exploration of several cognitive theories that are used in related fields. It is necessary to understand human cognition in order to design for human interaction. Without a deep understanding of how people store information in their minds and act upon this information, the design of a system is destined to be faulty [73]. The different cognitive theories that I explore in the section below form the theoretical basis for four different fields: Computer-Supported Collaborative Work (CSCW), Computer-Supported Collaborative Learning (CSCL), Educational Robotics, and Human-Robot Interaction. I examine the prior work in these fields on the development of design principles for the intended goal of each field. This prior work provides a basis for understanding what design principles in other fields are and where there is a gap.

The different fields of Computer-Supported Collaborative Work (CSCW), Computer-Supported Collaborative Learning (CSCL), Educational Robotics, and Human-Robot Interaction all involve design for human interaction but vary in the intended goal of the interactions. CSCW focuses on the coordination of groups and ensuring smooth operation. CSCL focuses on learning through group interactions and focuses on communication and building knowledge together. Educational Robotics and HRI focus on physical interactions. To understand the design principles necessary for collaborative assembly, we must first understand the design principles for physical interactions and for collaborative interactions.

In this section, I explore the cognitive theories that form a basis for understanding how interactions shape our thoughts. I then explain the ways in which CSCW, CSCL, Educational Robotics, and HRI use these cognitive theories to form design principles. These design principles then form the initial basis for our understanding of the interactions and design principles that are found to answer our research questions. Finally, I provide a definition of collaboration to be used in this thesis.

2.1 Cognitive Theories

Throughout history, many different philosophical theories have been developed to understand human cognition, starting on the individual level and broadening to communities as philosophers began to recognize the impact of the zeitgeist native to the individual's community [105]. Individual theories of cognition, including social cognitive theory, sociocultural theory, grounded cognition theory, and psycho-linguistic contribution theory, focus on the individual's mental model as a result of internal thought and external factors contributing to one's thoughts. Different theories place different importance on different external factors. On the other hand, cognitive theories on the community level often acknowledge the external or societal influences as having a form of cognition themselves through the means of artifacts or shared knowledge. These theories, such as distributed cognition, situated learning in communities of practice, activity theory, conversation analysis, or coordination mechanisms and articulation work, often recognize co-constructed forms of meaning as an entity not belonging to just the individual, but to the entire community.

2.1.1 Individual Cognitive Theories

Individual cognitive theories focus on the individual's mental model as it is influenced by external factors. Different theories recognize and prioritize carrying external factors and thus place importance on different affordances of a system when considering how the individual will interact with that system. When analyzing collaboration or interaction with a group, there are different focuses on the unit of analysis and collaboration process when interacting within a group.

For example, ***Social Cognitive Theory*** is a theory of cognition that recognizes the social, historical, and cultural contexts of an individual's environment as variables that influence the outcome of their cognition [42, 77, 104, 105]. Through this theory, meaning is constructed and stored through individual thoughts [104, 105], specifically verbal representations of knowledge [42]. The unit of analysis of this theory remains as the individual's mental representations or understandings but acknowledges that these are influenced by social interactions. The focus of analysis is on how these social interactions are affecting the individual's cognition, rather than the social interactions themselves or other influences on the individual [77]. Some theorists claim that all cognition is social as the individuals non-direct interactions with society are still motivated by the society in which they are situated [77]. Along this vein, the analysis of a collaborative process through the lens of Social Cognitive Theory would focus on how the individual behaves as they cooperate with a group to complete a task and how this behavior gives insight into the individual's cognition.

A related cognitive theory is ***Sociocultural Cognitive Theory***. This theory of cognition is similar to Social Cognitive Theory but places more importance on the inclusion of culture as an environmental variable that will heavily influence an individual's mental model [40, 104, 105]. Like Social Cognitive theory, the sociocultural approach to analyzing cognition will use the individual's thoughts as the unit of analysis but recognizes the cultural influence over these thoughts. While analyzing mental models, this theory focuses on the influence of the culture rather than aspects of the culture itself. Analyzing collaborative processes through the lens of sociocultural cognitive theory focuses on the individual's behavior in the group as they cooperate to gain insights into the individual's cognition and the influence of the environment around them.

Another broad theory of cognition is ***Grounded Cognition Theory***. Grounded cognition theory strays from the usual view that cognition happens as amodal, modular thoughts and instead embraces the idea that an individual's cognition is dependent on modal stimulation, bodily states, and situated actions [9, 10, 11, 81, 82]. In this theory, meaning is created through emotional states, sensor experiences, kinesthetic states, and actions within a context. These different sensations within their context are then stored in a person's mind and create a mental model of a situation. Like Social and Sociocultural cognition theory, grounded cognition also identifies the individual's mental representation as the unit of analysis and recognizes there may be many influences including the physical and social environment [9, 10]. Unlike the other theories, however, grounded cognition theory recognizes the impact of modal signals and bodily states as additionally influencing those mental models [9, 11]. When analyzing a collaborative process through the lens of grounded cognition, the focus would be on the individual's behaviors and demonstrations of emotional states, sensory experiences, and kinesthetic states that the individual experiences throughout the process [9].

Finally, other smaller theories focus on just conversation. Usually, this is referred to as ***conversation analysis*** and falls into the category of community-level cognitive theories. However, Clark and Brennan [16] argue for psycho-linguistic contribution theory which analyses the conversation between a collective into the contributions made by an individual. These contributions are seen as individual expressions of belief or mental representations [16, 104, 105]. The form of meaning in this theory is then the shared mental model or universally accepted belief that is held by all individuals in the conversation rather than the shared knowledge being an artifact itself. In this theory, the unit of analysis is just the individual's contributions to a conversation and the analysis of a collaborative process would focus just on conversations that an individual held during the process of collaboration.

2.1.2 Community Cognitive Theories

Conversely to individual cognitive theories, community cognitive theories analyze the collective community as the unit of analysis rather than just an individual's thoughts. Community-level cognition theories recognize the community or situation that a group resides in not just as an environmental influence on an individual, but as a factor unto itself that can store knowledge and create meaning. Through these systems of analysis, an individual's thoughts can differ from the beliefs and knowledge of the group which is analyzed as a separate body of cognition from the individual.

One major community-level theory of cognition is ***Distributed Theory***. In distributed cognition theory, it is believed that cognition is not only held in an individual mind, but cognitive processes are distributed across sociocultural systems, the individuals that make them up, artifacts, and the relationships between all these [41, 44, 95, 104, 105, 113]. Here meaning is formed and stored in the artifacts of a sociocultural system, in the members of the system, and in the relationships within the system [44, 113]. The unit of analysis of distributed cognition is much broader than an individual's cognition and does not define cognition as a single person's thoughts. Instead, distributed cognition analyzes a broader category of cognition including all elements of a group process and the functional relationships they have with each other. This could be in the form of an individual brain, external artifacts, or interactions within a team or cultural group. Furthermore, when analyzing a collaborative process through the lens of distributed cognition, importance is placed on collective knowledge distributed across the group, tools or artifacts that represent this knowledge, the coordination that takes place between internal and external factors, and the change over time of these interactions and knowledge artifacts [44].

Another theory that puts importance on the community in the role of cognition is situated learning and the idea of ***communities of practice***. Situated learning is the idea that individuals learn and create knowledge through participation and membership in communities of practice (defined as a group of people sharing a broader goal and communicating information to collectively get better at achieving that goal) [53, 56, 105, 113, 115]. Here, meaning is formed through the routine behaviors co-developed by members of the community [56]. In this way, the cognition goes beyond the individual and focuses on the larger elements of the community. The unit of analysis in this case then becomes the community of practice and the shared practices that hold the tacit knowledge of the community. Analyzing collaboration through this lens focuses on the social network formed while transmitting and contributing knowledge [113] and the relationship between the individual and the system that they are operating in and contributing to [115].

Activity theory holds that cognition is influenced by the actions an individual takes including the subject, instruments, and objectives relating to said action [26, 27, 52, 67, 115]. In activity theory meaning is treated through shared goals and the actions taken to achieve those goals [27, 67]. In activity theory, the unit of analysis is the activity which is comprised of the subject of the action, the object of the action, and the mediating artifacts used in the action [67]. The object of the action can be defined as the motivation or objective of the action that stems from the needs or desires of the person or people involved in the activity system [27, 67]. When analyzing a collaborative process through the lens of activity theory, one would then analyze the actions taken within an activity system and how people interact with each other as they take action towards their shared object [48].

Coordination Mechanisms and articulation work is a specific theory that focuses on the analysis of the cognition of a group of people through the coordination mechanisms that the group uses and produces. A coordination mechanism is defined as an object that concertizes the results of articulation work (work done by a group of actors to ensure coordination in a collaborative setting) [100]. Through this theory, meaning is formed through the creation and updating of coordination mechanisms. The unit of analysis in this theory then becomes the coordination mechanism itself and the collaborative process is analyzed through the contributions to the articulation work as it becomes reified into the coordination mechanism.

The last theory I will discuss is ***Conversation analysis***. This theory focuses less on how people think, and more functions as a method of analyzing thoughts through observable phenomena, namely, communication with others. As mentioned in the previous section, conversation analysis can be focused on the individual's cognition and analyze just individual thought and contribution. However, other methods of conversation analysis usually include a study of the linguistic practices of an entire community [105]. This study of linguist practices investigates the mechanisms by which members of the community reach shared understanding and form meaning as a collective. Through this lens of analysis, communications become the unit of analysis and the method of meaning-making. When analyzing collaboration through the lens of conversation analysis, discourse between members of the community becomes the focus of the collaborative process.

2.2 Computer Supported Collaborative Work and Learning

For more than 30 years, Computer-Supported Collaborative Work (CSCW) and Computer-Supported Collaborative Learning (CSCL) have been topics of research with the goal of studying how computing systems can be used to accomplish a shared goal among a group of people as they work together [17, 78, 107]. CSCW focuses broadly on how people work on a task with each other using computing and communication technologies. Foundational cognitive theories for CSCW are often focused on the larger communities of the workplace as the social group and analyze the widespread work practices and the role of technological artifacts within the workplace ecosystem [105].

CSCL focuses more on researching how people interact with each other through the use of information and communication technologies with the larger purpose of learning, rather than just the accomplishment of a task [107]. CSCL research is dependent on a few alternative theories of cognition. Some theories focus on the individual as the agent of learning, learning through interactions with the group. Other foundational cognitive theories for CSCL view the agent of learning as the group itself, where learning is in the interactions rather than a product of interactions [107].

Each of these fields has different goals and different theoretical foundations with which they use to analyze data and draw conclusions. While there is overlap in these goals, namely collaboration, there are also differences, work versus learning, leading to distinct design principles for each field. These principles elucidate the design practice for shaping positive collaborative interactions with software systems. This provides a basis for us to understand how design can shape collaboration as we explore the idea of physical collaboration later in this thesis.

2.2.1 Design for Computer Supported Collaborative Work

Computer-Supported Collaborative Work (CSCW) is a relatively new field aimed at researching the role computers play in the carrying out of group work [94]. CSCW explores the ways software systems can be utilized to maximize the collaboration of groups and influence how collaboration in group work is defined. CSCW research generally analyzes work through the use of a community as the unit of analysis, looking at how the technological system influences the larger-scale working of the community. Different researchers in CSCW use different cognitive theories to analyze their work, highlighting different important features of the systems being analyzed.

For example, Zacklad analyses systems through the lens of communities of practice [114]. They identify two important goals: changing an external system to complete a shared goal and developing a community that allows the members to develop mutual knowledge. From these goals, the authors create a breakdown of activities: operational, strategic, relational, and integrative. They then propose these as four dimensions for the possibility of technological support. They propose a technological system to support a community of practice should be designed to support the short-term operation of the group, organize and plan the long-term operation of the group, support the construction of a social network on a small person-to-person scale, and support the organization of the larger group at hand.

Robinson [93] argues for the design of CSCW systems for unanticipated use through the design of common artifacts, aligning with the cognitive theory of distributed cognition, focusing only on the artifact as the unit of analysis and how it affects the other dimensions of meaning. They state four design principles for common artifacts to function in a distributed cognition system: Predictability, Peripheral Awareness, Implicit Communication, and Double Level Language.

Schmidt and Simonee [99] argue for coordination mechanisms as a cognitive theory for CSCW and posit six design principles that capture features that support the reification of work done by a group to ensure coordination. They argue a coordination mechanism must have a definable protocol that is editable to support changing organizational requirements. Furthermore, actors must be able to control execution and make local or temporary changes to fit smaller-scale needs and contingencies. The behavior of the technology must be specifiable during runtime to allow for updating specifications. The system must also allow for the establishment of relationships between protocol, specification, and execution. The whole system must be malleable, and finally, the system must be linkable to other coordination mechanisms within a system.

2.2.2 Design for Computer Supported Collaborative Learning

CSCL is similar to the field of CSCW as it looks at collaborations supported by computer systems, but focuses on the learning outcomes that come from collaborative situations rather than the product of work. The shared goal that defines a group's collaboration thus becomes learning itself. In pursuit of this, CSCL focuses more on individual learning and how collaboration impacts the learner, using individual cognitive theories as well as some community theories that take into account the social impact on cognition and the creation and storage of knowledge.

Kato et al. [49] argue for a sociocultural approach toward learning, ensuring that the emergent division of labor is supported throughout the process of group work in an educational setting. They state the division of labor can be emergent, being interactively negotiated with knowledge of others' situations and available knowledge. Systems can be designed to support this by being transparent with the knowledge each participant has. Furthermore, the division of labor must be maintained through continuous coordination being updated as the situation changes. Systems can support this by making changes in other's situations transparent in the system. Along this line, the division of labor must be reorganizable based on the monitoring of other's states. These design principles focus on the social interactions of the learners and ensure smooth management of labor as the learners engage in the task.

Kirschner et al. [50] discuss many different theories that inform CSCL design, such as situated learning and authentic tasks, cognitive apprenticeship, and social construction. Situated and authentic tasks align with the sociocultural perspective, noting that culturally authentic learning experiences affect the student's cognition heavily. Similarly, social construction theories of education align with the social cognitive theories, stating that knowledge is constructed through interactions with others. Finally, cognitive apprenticeship aligns with communities of practice in that both understand learning and cognition to happen through the lens of the completion of shared goals and the development of routine behaviors to do so. However, in the CSCL context, the individual's learning of the routine behavior is more focused than the development and storage of the behavior.

From these different lenses, Kirschner et al. list affordances that systems need in order to support collaborative learning including educational affordances, social affordances, and technological affordances. Educational affordances are broadly defined as artifact characteristics that can support certain learning behaviors. Social affordances are characteristics that can facilitate social behaviors between learners. Technological affordances are those properties that determine how the device can be used.

2.3 Educational Robotics

Educational Robotics is a field that studies the impact of robotic platforms on educational gains and environments. From the use of Turtle robots to Lego Mindstorms and beyond, much of early educational robotics focused on the effectiveness of the platforms. From this evidence, the field shaped and focused on the question of how to design effective robotic systems for education. Much of the field's design work is rooted in the theoretical frameworks of theories of individual cognition revolving around tangibility, embodiment, and action, pulling from cognitive theories of constructivism and grounded cognition. Some research in Educational Robotics also pulls on the theories of Social and Sociocultural Cognitive Theory and recognizes the importance of cultural identification and personalization in robotic systems.

Many of the design principles for education robotics pull from a constructivist theoretical framework. Constructivist design principles include the idea of allowing students to construct their knowledge through the use of hands-on design and execution of projects [61, 108]. Along with this, there is a focus placed on active learning through exploration, problem-solving, inquiry, and playful learning [35, 61, 108]. Giang et al. also include transparency as a heuristic for the design of robotic systems, stating that the system should be rich and open, allowing students to observe the underlying mechanisms [35]. This encourages students' exploration and construction of knowledge of the whole system.

In addition to constructing, many educational robotics design principles focus on tangibility, embodiment, and emotion, pulling from Grounded Cognition theory. Catlin and Blamires emphasize the importance of interaction and embodiment in their principles for educational robotics [15]. Interaction encompasses students actively learning through multi-modal interactions in various semiotic systems. Embodiment is afforded through the time- and space-situated physical interactions students have with the systems. These design aspects get at the core of how cognition functions according to grounded cognition theory. Additionally, Giang et al. note the importance of interactions meeting expectations of previous encounters and similar to future encounters. According to grounded cognition theory, similar stimuli will help with student recall and thus positively contribute to their learning [35]. Another aspect of grounded cognition theory is the emotional state of the individual. Design principles often include an aspect of student enjoyment or engagement [4, 15, 35]. This positive feeling then becomes one of the mental representations of the learning, which leads to positive learning outcomes [83].

Many design principles for educational robotics also come from social and sociocultural cognitive theories. These design principles stress the importance of designing affordances for communication and collaboration (social cognitive [35, 61, 108]) as well as personalization to afford meaningful and authentic experiences for individual students (sociocultural [15, 35, 61]). Often, design principles that state the importance of communication and collaboration do not expand on these definitions, nor go into the intricacies of their analysis. These principles are stated in accordance with Social Cognitive theory, with the unit of the analysis as the individual and thus focus only on how communication and collaboration will affect the individual's mental representations of the learning goals [35, 61, 108]. Design principles also call for meaningful and authentic projects with the robotic systems [61], personally relevant applications [35], and affordances for expression of student self-identity [15]. Similar to the socially oriented design principles, these principles focus on the impact of these larger cultural practices on the mental model of the individual, arguing that with a closer connection to their own culture, the student will have a more positive learning experience.

Many design principles also state affordances that support cognition regardless of the cognitive theory being used, such as supporting appropriate cognitive load and workflow [35], supporting adaptability to fit the user's cognitive needs [35], and being intelligent systems that strive towards a goal of supporting cognition (where these goals can be changed based on the cognitive theory being used) [15]. Educational robotics can also be designed to afford meta-cognition as the step after cognition. Systems can be designed to support student reflection [35, 61], debriefing [35], and life learning skills [15].

Finally, many design principles are requirements that bolster the adaptation of the platforms, including the educational relevance, practicality of the system, integration into pedagogy or the classroom, and user experience. Many of the heuristics listed by Giang et al. [35] touch on educational relevance, such as ensuring appropriate challenge level, not automating learning relevant tasks, providing feedback, and incorporating computational thinking. Other design principles around educational relevance include the incorporation of interdisciplinarity, a versatile system for the exploration of multiple learning goals, and support for the curriculum and assessment of students [4, 15, 108]. General design principles also include those of practicality, ensuring that the robotic system meets the practical needs of students, cost requirements, and necessary functional performance metrics [15]. Along this line, many researchers include user experience design principles to follow, such as pleasing aesthetics [35], ease of assembly and physical comfort during use [35], and student appeal [4]. Finally, the ease and appropriateness of adaptation into the teacher's pedagogy or classroom are design principles for good educational robotic systems. These principles include the age-appropriateness of the systems and teaching materials [108], the range of features offered by the system [4], and the versatility of applicable teaching methods for the robotic systems [15].

2.4 Design Principles for Human Robot Interaction

Design principles in Human-Robot Interaction (HRI) are generally focused on usability principles aimed at ensuring, broadly, that the robot is designed for ideal interaction with the user [57]. Further design recommendations are stated in very specific application areas, or even per robotic system that is developed. While recommendations and implications of design are prevalent in HRI research, there do not exist many design guidelines that generalize design [57]. This may be a result of the large application interaction that HRI must deal with. Some areas of HRI, such as social robotics, overlap with certain cognitive theories. In these instances, robots are designed for these applications without much use of theoretical frameworks for their design with a larger focus on the specific outcomes or technical implementation than the theoretical basis for the design. These design processes often result in intermediate knowledge artifacts rather than overarching theoretical frameworks [57].

2.5 Summary of Cognitive Theories as Theoretical Frameworks

The landscape of cognitive theories is diverse, with various schools of thought that complement and contradict each other in many ways. Different research fields tend to adhere to their preferred theories, shaping their analytical frameworks and influencing their scientific conclusions. In addressing phenomena that transcend traditional disciplinary boundaries, such as the design of robotic systems for human-human collaboration, it becomes imperative to consider the range of cognitive theories available.

Computer-Supported Cooperative Work (CSCW) uses theoretical frameworks that take into consideration knowledge outside the individual mind, choosing to look at community systems, including elements other than humans. However, CSCW primarily focuses on the affordances of computer systems rather than robotic systems and lacks focus on the area of education specifically.

Computer-Supported Collaborative Learning (CSCL) delves into task design and educational applications. Though CSCL primarily focuses on individual learning within group settings, research has begun to shift towards group learning and the impacts of knowledge creation through group interaction outside an individual mind. Similar to CSCW, CSCL neglects hardware design and does not currently investigate the available affordances robotic systems may provide.

Educational robotics prioritized the learning that can be facilitated by robotic hardware. The theoretical frameworks used are all in the individual unit of analysis, focusing only on singular learners, without considering the ability of the robotic system to provide affordances for the collaborative or social.

Human-Robot Interaction (HRI) concentrates on optimizing interactions with robotic systems but overlooks collaborative learning possibilities. A noticeable gap exists in HRI research concerning overarching design principles beyond application-specific recommendations.

Though each of these areas brings a unique perspective with different cognitive theories, no singular discipline covers all units of analysis necessary to analyze human-human-robot interactions. Thus, in this thesis, we explore how each of these cognitive theories can contribute to see which are most promising for future use.

2.6 Defining Collaboration

In this thesis, we will define collaboration as a process of two or more people collectively working towards a shared goal where the output of the group cannot be easily separated into individual contributions [21, 24, 65, 96]. We draw this definition from past theoretical frameworks for defining collaboration from the vast body of work in computer-supported collaborative learning, which often breaks collaboration down into the 3Cs model: Coordination, Cooperation, and Communication [25, 33]. Coordination is often defined as the process of organizing the people, activities, and resources necessary to accomplish a shared goal, ensuring shared understanding about the state of each [22, 25, 33, 65, 96]. Conversely, cooperation is the process of individuals working on achieving specific tasks that contribute to a shared goal [8, 33, 65, 96]. Baker distinguishes between cooperation and collaboration in that ”cooperation works on the level of tasks and actions, collaboration works on the plane of ideas, understanding, representations” [8]. We define communication as the sharing of knowledge between people. This can be verbal, written, visual, etc.

The 3C's model breaks collaboration down into specific observable behaviors that allow us to analyze different kinds of collaboration and see how they affect the group participants. For Coordination, these behaviors can be analyzed with more of the larger group theories that look at the higher-level management of people within a community collective. For example, coordination mechanisms and communities of practice shine a light on the articulation work necessary for a large community to run smoothly.

For cooperation, individual theories best explain how this cooperation in an environment influences the individual. For example, with Social and Sociocultural Cognition Theories, the actions of the individual in relation to the group influence how an individual thinks. Activity theory also places importance on the individuals that make up the activity system, and they perform their individual actions to create the larger system of activity together.

Finally, communication draws on many different theories that place importance on interactions between people for cognition. These theories include social, sociocultural, psycholinguistic, and conversation analysis.

Chapter 3

Approach

The multiple fields described in the previous section each take a unique approach to the design of systems based on the analytical lenses of cognitive theories. However, there is an under-researched overlap in these fields that involves the use of hardware systems in human-human collaboration. The field of Educational Robotics focuses on individual cognitive theories, including grounded cognition theory, social cognitive, and sociocultural cognitive theory, highlighting the importance of affordances of robotic technologies for the individual learner. Research in this field has a specific focus of analysis on the tangible affordances of the robotic system as they pertain to an individual's learning gains, but provides little guidance on the affordances of a robotic system for group learning.

Human-Robot Interaction (HRI) design principles focus on the individual and affordances, focused on reducing the cognitive load of users to ensure the ease of use of the system to accomplish the task as easily as possible. While this provides benefits for accomplishing the task, it does not take into account learning gains that can be provided through interactions with the system.

Computer-Supported Collaborative Work (CSCW) often analyzes group work through the lens of community-level cognitive theories, focusing on the design of artifacts within the collaborative system. This provides benefits for the analysis of the systems, but prior work in this field focuses mostly on software systems. Computer-Supported Collaborative Learning (CSCL) work focuses both on learning through the individual and the group lens, but like CSCW research, it focuses on software systems and often focuses on technology as a tool and not as the focus of the collaborative learning itself.

Designing robotic systems for human-human collaboration, both in educational contexts and beyon,d requires the use of cognitive lenses employed by these multiple disciplines, working together to provide an appropriate unit of analysis to capture all axes of potential affordances of robotic systems. In this thesis, I will analyze a robotic system designed for human-human collaboration and collaborative work from multiple cognitive lenses in order to distill a set of design principles. This thesis aims to tackle the issues of designing in the context of a three+-body-system (robot and two or more human learners), including issues pertaining to collaboration modelling, tangible interaction design, and interactive and collaborative educational experiences.

3.1 Research Questions

The series of questions that I aim to answer is:

1. What features of a robotic system help facilitate collaborative assembly?
2. What features of a robotic system help facilitate collaborative experiences in a classroom?
3. What are the design principles for assemblable devices designed for human-human collaboration during assembly?

In these research questions, I make two important semantic distinctions. The first distinction I make is between collaborative experiences in a classroom and learning in a collaborative environment. In this thesis, I will use collaborative experiences in a classroom to refer to the process by which a small group of people experience and complete different educational tasks together. I distinguish this from learning in a collaborative environment, which refers to an individual's learning process within an environment of influence that contains social interactions. The difference between these is the unit of analysis. Collaborative experiences in a classroom will be used to refer to a process that I will analyze by examining the small group, including each member, the collaborative process, and the tools involved, rather than analyzing just the individual (learning in a collaborative environment).

The second important distinction is in the purpose of the robotic system. For human-human collaboration, I use this phrase to mean the instances where the robotic system is the task that the human agents are collaborating on. The shared goal in this instance is to make the robotic system functional. In the case of "collaborative experiences in a classroom" the robotic system is a tool used to accomplish the learning goals.

In this thesis, I investigate the first two research questions through the analysis of an existing robotic system as it is built and used in a classroom environment by a group of students. This provides the initial hypothesis for my third research question. I use this analysis to propose a set of design principles for assemblable devices that impact human-human collaboration. I then present a set of devices designed for and against these design principles to test the causal impact of the designs on two-person assembly. Finally, I present a study comparing two versions of RoboLoom (see Chapter 4) that were designed with different principles and assembled in groups by students in a classroom setting. This validates my causal results in a real-world setting.

3.2 Contributions

Inspired by the fields of Educational Robotics, HRI, CSCW, and CSCL, I use multiple different cognitive theories to analyze how people interact with educational hardware in group settings as they work on the tasks of assembly and use of robotic systems. Using different theories allows me to build a foundation that encapsulates all the potential affordances that robotic systems can have in collaborative settings in educational contexts. Using this foundation for analysis, I examine the difference in collaboration between two pre-existing systems to understand which features led to different collaborative actions. Additionally, I examine a robotic system designed for collaborative use in an educational setting to hypothesize which design features contribute to collaborative experiences in a classroom. From this, I create a set of design principles for future systems that will encourage human-human collaboration in an educational setting. I then provide empirical evidence for the causal effect of these design principles, both on paired collaborative assembly in a lab setting and on group collaborative assembly in a real-world classroom.

From this work, I theorize and validate the following set of design principles for a robotic system to influence human-human collaboration on tasks related to building and using said systems. They are listed below:

1. **Intermediate task difficulty** The robotic system should be designed such that the difficulty of the task is perceived to be too difficult to achieve individually, but not too difficult to be achieved by the group. If the task is too easy, this encourages an individual to complete it alone, as they will assume the difficulty of the articulation work to be greater than the difficulty of completing the task alone. If the task is too difficult, students will fall back on asking the instructor to provide them guidance rather than beginning the articulation work and collaboratively problem-solving within their group. Robotic systems should also be designed with the familiarity of components in mind. When more familiar components are used, participants are more easily able to complete tasks. However, when less familiar components are used, participants tend to spend more time communicating and coordinating to understand the task before and during completion. When only some participants are familiar with the hardware, this leads to communication in the form of knowledge transfer.
2. **Parallelizability of tasks** When tasks to operate or assemble a robotic system can be done in parallel, group members will be more likely to cooperate on the tasks. However, coordination and communication happen more rarely during cooperative tasks across parallel subgroups. When designing collaborative hardware, parallel tasks can be used to increase cooperation, but hardware should require subgroups to rejoin to encourage coordination and communication as well.
3. **Physical space** The robotic system must be designed spatially for multiple participants to work simultaneously. The size of the system will affect the group's perception of how many can collaborate physically at the same time. Additionally, if the tasks require physically manipulating one contiguous piece of hardware, this will lead to more collaboration in the forms of communication and coordination. However, this can make the tasks more physically demanding and create frustration within the group.
4. **Repetition of tasks** Repetitive tasks in the robotic system can lead to different forms of collaboration. Initially, these tasks can take a form akin to a small, short-term community of practice where collaboration can be practiced through the establishment, continued improvement, and teaching of methodologies to complete the task. However, in simpler tasks requiring too much repetition, the collaborative affordance saturates after participants determine the ideal way to complete the task. Repetitive tasks can lead to higher cooperation, as participants will seek to split large amounts of work amongst all members to lessen individual task burdens.
5. **Specificity of the hardware** The robotic system should be carefully designed such that the specificity of the system is taken into account. Specificity refers to how open-ended the system is, or how many different ways it could be assembled. Open-ended design can

allow for more communication and coordination within groups, but can also lead to more errors during task completion. While errors can lead to positive learning experiences, they can also have negative long-term outcomes, such as frustration and increased time for task completion. The design of the system must be carefully balanced such that the system allows for some open-endedness to afford collaboration, but does not negatively impact user experience or learning.

In order to fully capture the collaborative behaviors of groups, we developed a coding scheme based on the 3Cs framework [25, 33] to analyze behaviors that might inform impact on cognition through the lens of various cognitive theories. The codes are:

1. *Coordination* - participants are engaging in behavior to manage others, activities, or resources needed for assembly. Coding for coordination can give insights into cognition based on coordination mechanisms and distributed theory.
   (a) *Coordination of People* - Participants are managing the distribution of labor.
   (b) *Coordination of Materials* - Participants are managing the materials for the task.
   (c) *Coordination of Specific Tasks* - Participants are managing how and when to accomplish their activities.
2. *Cooperation* - participants work separately on tasks that both contribute to the shared assembly goal. Looking at the physical acts of coordination can give insights into how participants are grounding cognition, distributing cognition, participating in communities of practice, or thinking in terms of activities.
   (a) *Cooperation - Different Steps* - Participants or participant groups work on two steps of assembly concurrently.
   (b) *Cooperation - Same Step, Different Hardware* - Participants are working on the same step of assembly, but interacting with distinct hardware pieces.
   (c) *Cooperation - Same Step, Same Hardware* - Participants are working on the same step of assembly and interacting with the same loom materials.
3. *Communication* - Participants discuss to share, create, and clarify knowledge of the assembly. Coding for communication can give insights into how participants are thinking through the lens of sociocultural theory, conversation analysis, or communities of practice.
   (a) *Helping: Knowledge Co-Creation* - Participants are creating knowledge about the loom's assembly with each other. This requires contributions from all participants involved to build the knowledge together.
   (b) *Helping: Knowledge Transfer* - One participant transfers knowledge to another, intending to aid the second in the assembly task.
   (c) *Communicating About Troubleshooting* - Participants talk through troubleshooting when there is an error in the assembly process.
   (d) *Communicating Instructions Confusion* - Participants communicate when instructions are unclear.
4. *Blocked Collaboration* - A participant attempted collaboration but did not succeed. This does not include participants who are off task.

Chapter 4

RoboLoom

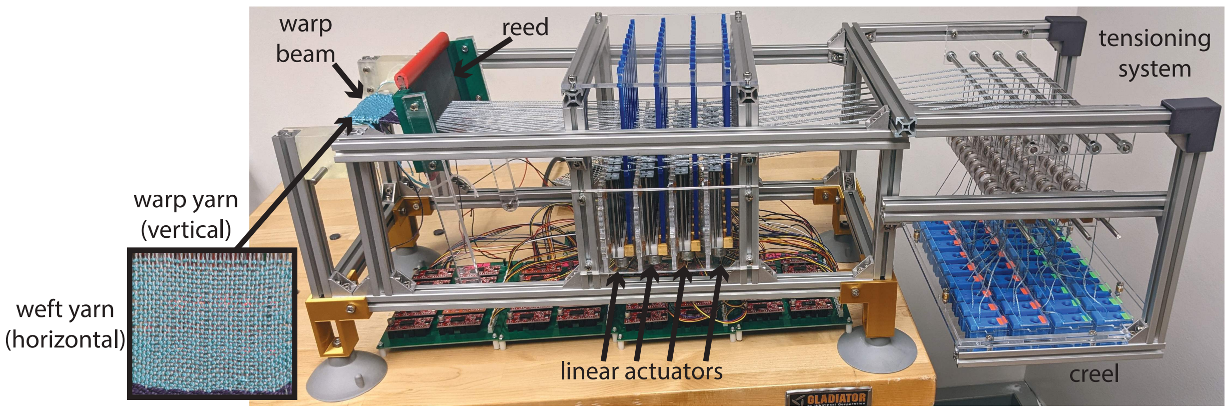


Figure 4.1: RoboLoom: An open-source robotic Jacquard loom kit for use in interdisciplinary collegiate classrooms

In this chapter, I disseminate the design of RoboLoom and demonstrate its adherence to basic functional design requirements. RoboLoom was chosen as a platform for collaboration because of the interdisciplinary nature of weaving. The connection between mathematics and cloth design and modeling, as well as the connection between engineering the devices made for weaving, allows for a diverse set of students to come together in the classroom when they otherwise would not. This diverse set of students allows for interesting opportunities and challenges for collaborative work and collaborative experiences in a classroom. In this chapter, I discuss the background of weaving, mathematics, and engineering that functions as the foundation of the course. Then I discuss how this influences the design of RoboLoom. In future chapters, I discuss how this interesting overlap of disciplines can influence collaboration.

4.1 Introduction

Weaving is a fabrication process that is shaped by art, mathematics, and engineering. For centuries, humans have used woven cloth to create artistic expressions through material, color, pattern, and weave [88]. These artistic expressions offer an opportunity to explore mathematical representations and models for patterns and textiles. For example, weaving patterns can be represented mathematically through the binary, matrix-like nature of the pattern [39]. The feel and drapability of the cloth can be analyzed geometrically through an understanding of the yarn tension, weight, and how the weaver interlaces the yarns together [23, 87]. Even the layering of the cloth can be defined mathematically through the use of set theory to group yarns into layers [23].

In order to create high-quality, complex cloth, weavers follow a process that mirrors the engineering design process [58]. They start by designing or choosing a desired pattern and analyzing the factors that will determine the feel and quality of their final product (drapability, the tension of the loom, and the quality of the yarn). They then plan a weaving strategy to achieve their desired final product, given the constraints of the tools available and iterate on their design [62]. Weavers' desire to create more complex patterns and the industry's desire to mass produce these products have led not only to advancements in processes but also to multiple engineering innovations [14]. For example, the development of modern automation was driven by the introduction of punch cards to program the first Jacquard looms [28], which led to modern-day computers.

This connection between weaving, math, and engineering presents an opportunity to bring interdisciplinary learning into the classroom [85, 86]. Interdisciplinarity brings together different disciplines, providing an opportunity for students from different backgrounds to collaborate toward a shared goal. Interdisciplinary curricula can also improve student outcomes in education as well as support the learning of critical skills to bolster student success in future careers [46].

Our goal in designing RoboLoom is to take advantage of the complex mathematical and engineering relationships with weaving to create interdisciplinary instruction for post-secondary classrooms. Toward this goal, we developed RoboLoom as an open-source Jacquard loom kit for supporting arts, mathematics, and engineering learning. RoboLoom supports the exploration of mechatronic concepts and engineering design principles through its open-source design and assembly. RoboLoom's Jacquard capabilities allow it to create complicated design patterns, affording the instruction of complex mathematical concepts (e.g.linear algebra, vector calculus, and set theory). All of our designs, assembly instructions, and software can be found at: *https://sites.google.com/view/RoboLoom*.

4.2 Weaving, Mathematics, and Engineering

Both mathematical and engineering principles can be used to define cloth, categorize its properties, and shape its fabrication. Weaving machines afford the re-contextualization of digital and computation in a non-typical application [29]. Recently, weaving has been explored as a way of fabricating electronics [20]. Applications have explored the ability to weave conductive thread into cloth with applications in sensing [111], actuation [110], and design [18, 32, 47]. Not only can weaving be used in engineering, but engineering is also a necessary component of weaving. We explore three applications of weaving and math: matrix multiplication and pattern generation, mathematical drapability, and mathematically modeling cloth separability.

Once a cloth is designed, the weaver can use these mathematical models, to iterate on their design to achieve the desired final cloth, following a process like the engineering design process [58]. Once their design is finalized, weavers use a loom to fabricate their textile product.

Weavers must carefully ensure tension is evenly held across warp yarns as the weft is interwoven into them to create cloth. Looms have been expertly engineered over centuries to precisely achieve the perfect cloth [14]. Engineers must use systems engineering skills when considering the textile constraints and system interactions. Furthermore, they must use system construction skills when considering the loom's robustness and durability enabling long-term use under tension.

4.2.1 Weaving and Matrix Multiplication

Cloth is fabricated by interlacing vertical warp yarns with horizontal weft yarns (Figure 4.6, Figure 4.2). Warp yarns run vertically through the loom and are raised or lowered to control how the horizontal weft yarn is interlaced by the weaver. Many mathematical principles are illustrated in weaving paradigms, including the matrix representations of pattern design. Weaving patterns are often represented as weaving drafts (Figure 4.3) consisting of four major components: Threading (which warp yarns are actuated by which shaft), Tie-up (which shafts can be raised together by a single pedal), Treadling (which pedal is pressed at a given time step), and Draw Down (final cloth pattern). Given the binary nature of weaving [39], we can write each of the parts of the weaving draft as a binary matrix. Multiplying these matrices results in the matrix representation of the drawdown, represented as:

$$D = Tr \times Tu^T \times Th. \tag{4.1}$$

Where D represents the draw down, Tr the treadling, Tu^T the tie-up transposed, and Th the threading (Figure 4.3).

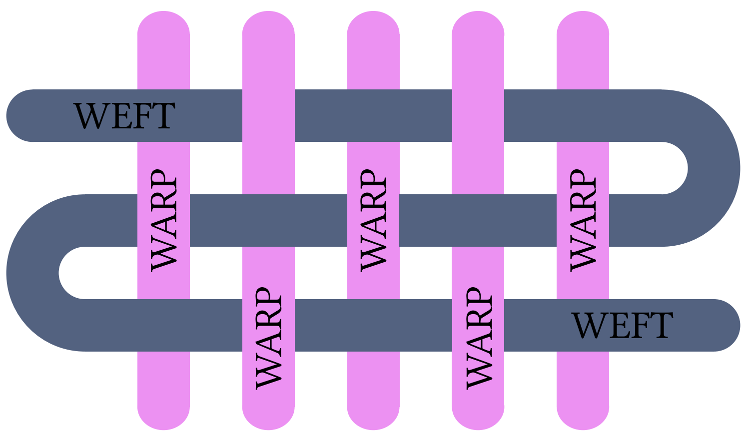


Figure 4.2: An illustration of plain weave cloth showing the warp and weft yarns.

A visualization of the multiplication is shown in Figure 4.4. Multiplying the treadling with the tie-up ($A = Tr \times Tu^T$) yields a matrix that describes which shafts will be raised at each time step (represented as a row in the A matrix). Multiplying this result with the threading ($D = A \times Th$) then describes what warp yarns will be raised at each time step, telling the weaver for each row of their cloth which warp yarns show and which are covered by weft yarns.

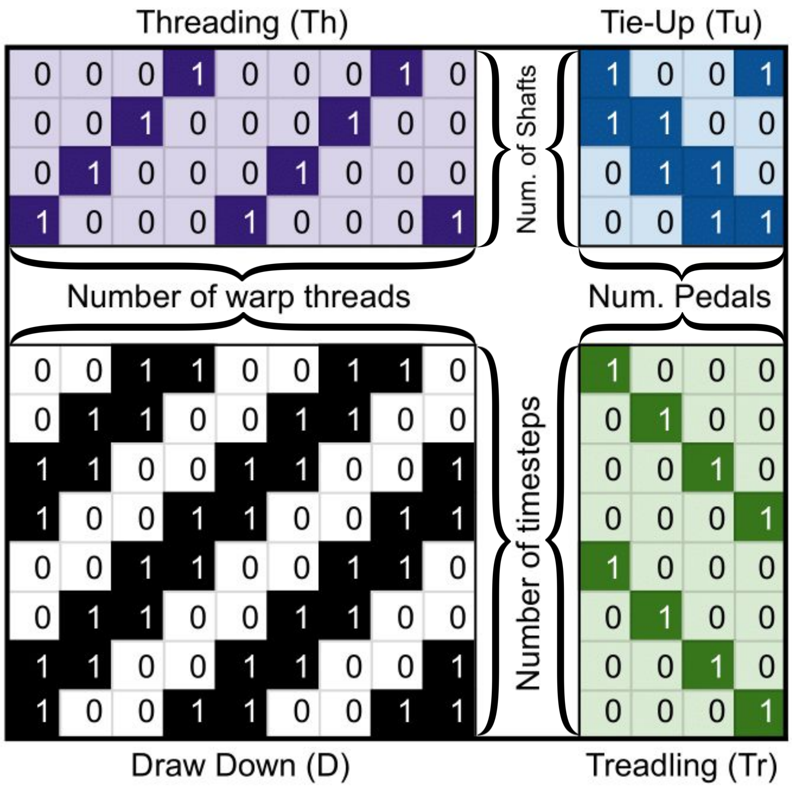


Figure 4.3: A weaving draft for a typical shaft loom. Drafts describe how warp yarns connect to shafts (threading), how shafts connect to pedals (tie-up) and how pedals are actuated (treadling) to create the weaving pattern (drawdown).

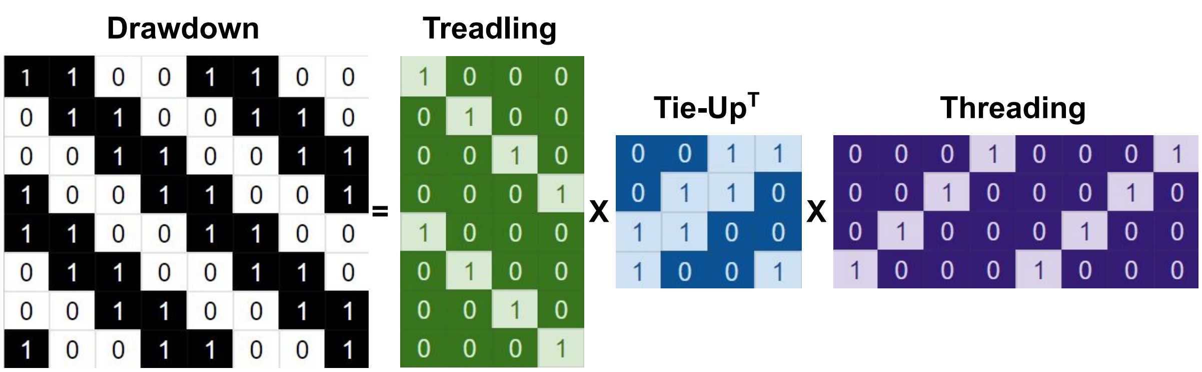


Figure 4.4: An image showing the matrix multiplication of weaving draft elements for an example 2x2 twill pattern.

4.2.2 Mathematical Modeling of Cloth Properties

Cloth properties such as layering and drapability can be modeled mathematically. The layering of the cloth is defined through constructing a set of yarns in the layer through set theory and referenced as cloth integrity. The drapability of the cloth can be mathematically modeled through the weave factor. These definitions are discussed in the sections below.

4.2.2.1 Cloth Integrity

Grunbaum and Shephard [38] define a key requirement that a valid weaving pattern must not "fall apart", meaning that one cloth forms a bound layer. If there exists a set of yarns, A, that always go over a set of yarns, B, it will separate from the B yarns and thus A and B are not the same layer of cloth. If a cloth falls apart it does not have "cloth integrity"

We define the A set as containing two subsets: the columns of the A set (A_c), and the rows of the A set (A_r). We define B_c and B_r similarly. Representing the pattern as a binary matrix, we can state this definition mathematically by saying a cloth, P, will "fall apart" if and only if there exists sets $A = A_c \cup A_r, A_c \neq \varnothing, A_r \neq \varnothing$ and $B = B_c \cup B_r, B_c \neq \varnothing, B_r \neq \varnothing$ such that $A \cap B = \varnothing$ and $\{P(b_r, a_c)\}_{a_c \in A_c, b_r \in B_r} = \{1\}$ and $\{P(a_r, b_c)\}_{b_c \in B_c, a_r \in A_r} = \{0\}$, where the function $P(r, c)$ is accessing the value of the pattern matrix at row r and column c. An example pattern fitting this definition can be found in Figure 4.5.

	$b_{c1} = 0$	$a_{c1} = 1$	$a_{c2} = 2$	$b_{c2} = 3$
$a_{r1} = 0$	0	0	1	0
$b_{r1} = 1$	0	1	1	1
$b_{r2} = 2$	1	1	1	0
$a_{r2} = 3$	0	1	0	0

Figure 4.5: An example of a pattern that fits the mathematical definition of "falling apart". In this case $A_r = \{a_{r1}, a_{r2}\}$, $A_c = \{a_{c1}, a_{c2}\}$, $B_r = \{b_{r1}, b_{r2}\}$, and $B_c = \{b_{c1}, b_{c2}\}$. Blue highlighted squares are instances of $P(b_r, a_c)$ for $a_c \in A_c, b_r \in B_r$. Orange highlighted squares are instances of $P(a_r, b_c)$ for $b_c \in B_c, a_r \in A_r$.

4.2.2.2 Weave Factor

The cloth's sturdiness and drapability can be described through the cloth's weave factor [12]. The weave factor of a cloth accounts for the number of interlacings of warp and weft yarns and is expressed as $M = \frac{E}{I}$, a ratio of the number of yarns per pattern repeat (E) to the number of times the pattern changes value (I). When the warp and weft interlacings are different, the weave factor must be calculated for each warp and weft as M_1 and M_2 respectively. M_1 is calculated by the ratio of the number of warp yarns (E_1), to the number of times a row switches values

(I_2). M_2 is then the complement of this as the ratio of the number of weft yarns (E_2), to the number of times a column switches values (I_1). When the number of interlacings is not equal in a pattern's repeat, as is usual for Jacquard patterns, the irregular weave factor must be calculated as $M = \frac{\sum E}{\sum I}$.

4.3 Looms as Classroom Tools

Weaving cloth using the concepts discussed in section 4.2 requires a versatile loom. Different loom types offer varying versatility at the expense of increased cost. In this section, we explore this trade-off and discuss the benefits and detriments of three loom types: rigid heddle, shaft, and Jacquard.

Rigid heddle and shaft looms are less costly than Jacquard looms but offer less versatility with less control over individual warp threads. Loom cost is proportional to the quality of its construction and the number of heddles and shafts offered. These looms range from tens to several thousands of USD. While the less costly versions are monetarily feasible for a collegiate classroom, they require significant expertise and time to warp and thread (described in Section 4.4.1). Changing patterns to explore different mathematical and engineering concepts means repeating this lengthy process, yielding low versatility and thus low classroom feasibility.

Jacquard looms offer the most weaving versatility by actuating each warp thread individually. Here we discuss two Jacquard loom types: commercial and DIY. Commercial Jacquard looms provide the highest quality cloth, but are costly. These looms are usually covered machines designed to be plug-and-play limiting the ability to ”tinker” with them, thus limiting instructional support of engineering design skills. DIY looms are significantly less costly and allow for deeper exploration of engineering skills but produce lower-quality cloth. These trade-offs between cloth quality, educational potential, and cost are important classroom considerations.

Two popular commercially available Jacquard looms are the TC2 [74] and the Jacq3g [45]. Their cost is high – tens of thousands of dollars – making them infeasible as classroom tools. While the commercial availability of these looms affords more access to exploring the mathematical principles of produced cloth, it restricts the engineering skills that can be explored due to the opaqueness of the product and the legal protection of novel design advancements.

To address the cost issue, many hobbyists and researchers have made affordable, personal Jacquard looms [5, 55, 66, 71, 79, 97]. Some [5, 71, 97] use serial actuation, reducing cost but increasing the warp actuation time (shedding time) which must be done hundreds of times to produce a single cloth. Serial actuation looms range from 32 [71] to 60 [97] warp yarns. Other DIY Jacquard looms use parallel warp yarn actuation, decreasing shedding time, but increasing cost [55, 66]. To reduce their cost, these looms typically have fewer warp yarns (14 [55]–24 [66]), reducing cloth quality. These DIY looms are optimized for personal use, sacrificing quality and efficiency for lower cost. A loom specifically designed for classroom use needs to balance enough cloth quality to teach the desired course topics, whilst being efficient, robust, and reasonably priced.

DIY loom designs are openly available unlike their commercial counterparts, often with websites describing the engineering processes [55, 66, 71, 97]. However, recreating the devices require specific expertise, restricting the ability of students to be active participants in creating their own loom.

4.4 RoboLoom Functional Design Requirements

A loom kit designed for interdisciplinary education in art, math, and engineering must facilitate time- and labor-efficient interactions. The loom must be robust, moderate cost, and relevant to weaving, math, and engineering. These requirements are delineated below.

4.4.1 User Interaction

Ease-of-use and efficiency are important design considerations for human-tool interaction [63, 72, 112]. Classroom technologies must also have these qualities to not distract from learning [89]. There are two typical human interactions with looms: warping and weaving. Each should be efficient, reducing non-educational work time.

4.4.1.1 Warping Efficiency

Warping a loom is a lengthy process that consists of two stages: winding yarn onto the back warp beam, and threading the yarn through the heddles of the loom [62]. Winding requires the weaver to hold manual tension while stretching the yarn across pegs of a warp frame. Then the yarn can be transferred to the back warp beam where tension must be held manually as the yarn is rolled on. From here, the back warp beam is attached to the loom and the threading process can begin. Threading requires taking each warp yarn through the correct heddle carefully so as to not make mistakes, or the process must be repeated. To reduce the expenditure of classroom time on non-learning related tasks, a well-designed classroom loom should be easy and quick to warp and allow for corrections in the process should errors occur.

4.4.1.2 Weaving Efficiency

A loom designed for classroom use should ensure there are as few as possible interruptions during the weaving process to lessen distractions from learning. Weaving time on the loom should feel productive and efficient, requiring the shedding time be as quick as possible. In an interdisciplinary classroom, student weavers will be novices and will inevitably make mistakes, e.g., a single warp yarn losing tension or breaking. These problems should be quick and easy to correct.

4.4.2 Accessibility

To be accessible for classroom use, the cost of the loom must remain low enough that multiple looms could be purchased by schools [34]. The accessibility of a device can also be increased through open-sourcing the design [80], allowing users to customize the device to fit their specific needs.

4.4.3 Interdisciplinary Relevance

As an educational tool for textiles, engineering, and math learning, the loom should be designed to aid in combining these interdisciplinary concepts without becoming a distraction [89]. Furthermore, the loom must support beginner- through higher-level concepts as students will have various backgrounds in each discipline.

4.4.3.1 Weaving

To support novice student weavers, the loom should be able to produce a high enough quality cloth to weave beginner projects such as coasters, wall hangings, small pouches, scarves, and headbands [62]. To pattern these cloths with high enough fidelity, the loom should have at least 24 warp yarns [62]. Sufficient-quality hand-woven cloth is usually in the range of ≈8-36 ends per inch (EPI) [62], so the loom must support this warp density. For the purposes of this paper, we will describe a cloth with at least 24 warp yarns and at least 8 EPI as quality cloth. The loom must be able to weave with minimal warp yarn breaking while keeping tension at ≈50g-250g [62].

4.4.3.2 Mathematics

To facilitate the interdisciplinary learning of post-secondary math concepts through weaving, the loom should be able to weave patterns designed using mathematical concepts such as matrix algebra [39], weave factor [12], and cloth integrity [38], [87] (Section 4.2.1). For students to see the results of matrix operations in their cloth, the loom should also be able to weave patterns with high enough fidelity. The definition of quality cloth in the above section satisfies this requirement as 24 warp yarns at 8 EPI is high enough fidelity to see complex cloth patterns clearly [62]. Additionally, the loom should allow students to explore weave factor and cloth integrity (Section 4.2.1) through the comparison of values for different weave structures (e.g., plain weave, twill weaves, satin weaves) and more complex weaves (e.g. Jacquard patterns) in the design and production stages.

4.4.3.3 Engineering

The loom should support students as they explore the engineering design process [58]. It should allow them to consider systems engineering principles (designing under constraints and understanding system behaviors and interactions) [101], system construction principles (robustness and durability) [43], and engineering validation methods (modeling and testing) [43].

For students to explore concepts of systems engineering and construction, the loom should be uncovered. An uncovered design allows students to see the mechanisms, components, and their interactions. For example, students will be able to see an actuator's behavior, consider what constraints lead to the selection of that actuator (e.g. cost, force), and see how that actuator interacts with other components (e.g. electronics, warp yarns).

Designing a loom to be manufactured and assembled by students gives students the opportunity to see how system construction principles (i.e. robustness and durability) affect material choice and performance. For example, weaving requires the loom to hold a considerable amount of tension between warp beams, requiring sturdy materials to support this force.

To support validation methods the loom should allow students to model and test different weaving drafts. Iterative testing of design will help students rapidly evaluate whether the final product will meet the intended form, fit, and function.

4.5 System Design

To our knowledge, RoboLoom (Figure 4.6) is the only open-source robotic loom kit created for and tested in higher education settings. In the following sections, we explain how the design of our loom meets our aforementioned requirements.

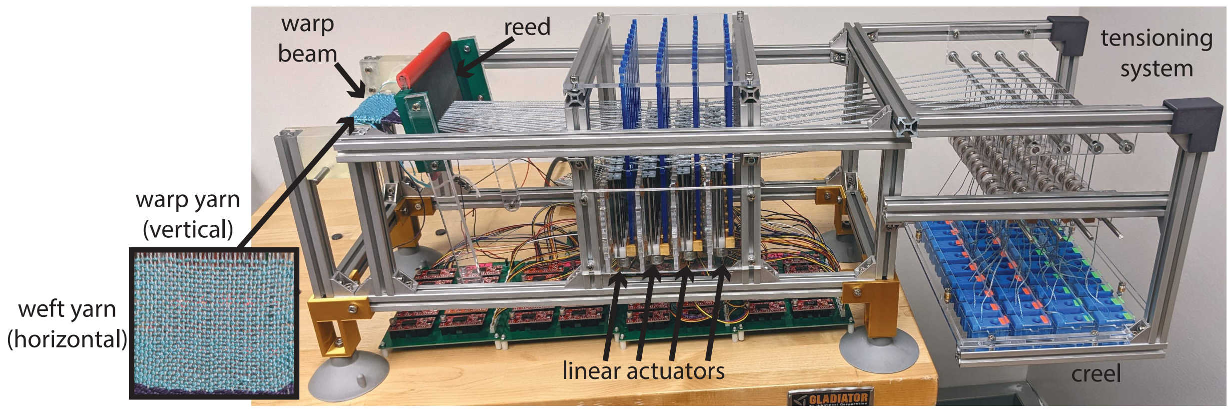


Figure 4.6: RoboLoom: An open-source robotic Jacquard loom kit for use in interdisciplinary collegiate classrooms

4.5.1 Hardware

We designed RoboLoom (Figure 4.6) in accordance with the design requirements outlined in Section 4.4.3.3. We chose to make RoboLoom a Jacquard loom that individually actuates each warp yarn to increase the flexibility of possible weaving patterns and allow for the exploration of more mathematical concepts (Req. 4.4.3.2). Although matrix multiplication only relates to shaft loom weaving (as defined in 4.2.1), artificial constraints can be created through software to simulate a shaft loom using the Jacquard mechanism. This setup allows students to switch shaft loom patterns with no re-threading and minimal re-warping (Req. 4.4.1.1).

RoboLoom's frame is made of t-slotted aluminum, ensuring that it is light, robust, and easy to assemble by novices (Req. 4.4.3.3). There are three main components of the loom: the front warp beam, the heddles, and the tensioning system and creel, shown in Figure 4.6. Aside from the t-slotted aluminum, components consist of 3D-printed and laser-cut parts so the kit can be open source, easily manufactured, and lower cost (Req. 4.4.2).

RoboLoom is capable of individually actuating 40 warp yarns, balancing the cost of the actuators with the ability to produce quality cloth (Req. 4.4.3.1 and 4.4.2). The larger the number of warp yarns, the more complex a pattern can be. We chose to use more warp yarns than required by Req. 4.4.3.1 to allow more pattern exploration by students. Each warp yarn is threaded through a heddle which is rigidly attached to a linear actuator allowing simultaneous warp yarn movement, and decreasing shedding time and mechanical complexity over serial actuation designs (Req. 4.4.1.2). The cost of the linear stepper motor is lower than that of counterparts used in professional Jacquard looms but, due to its size, the heddles cannot be spaced as closely together as they would on a commercial loom. To overcome this issue, we divide the actuators into different planes in the frame design and offset them to decrease the gap between heddles, achieving 12 EPI (Req. 4.4.3.1).

From the heddles, the warp yarns pass into RoboLoom's tensioning system and creel, described in the following sections.

4.5.1.1 Tensioning System

To produce quality cloth it is important to maintain uniform tension in the warp yarns (Req. 4.4.3.1). In most looms, uniform tension is established by the weaver feeling the tension on yarns by hand. Correcting uneven tension is usually very time-consuming and the fixes can range from having to re-thread portions of the warp to having to place weights or cardboard pieces in parts of the creel. To optimize the warping process and minimize error recovery time for beginners (Req. 4.4.1.1 and Req. 4.4.1.2), we designed a novel tensioning mechanism that allows for individual setup, tensioning, and adjustment for each warp yarn.

RoboLoom's tension system uses a passive mechanism to keep cost low (Req. 4.4.2). Each warp yarn passes through a set of tensioning disks forced together by a spring and held in place by a rod and spacer (Figure 4.7). The tension on the yarn is then dependent on the coefficient of friction between the yarn and the disks, μ_1, the coefficient of friction of the yarn on the stainless steel rod, μ_2, the force of the spring, N, and the angle of the yarn around the rod, θ_1. We approximate the tension on the yarn by modeling the system as in Figure 4.8. The normal force of the system is dependent on the spring constant, k, and the compression of the spring, Δx. The tension on the yarn after passing through the tensioning device can then be expressed as:

$$T_m = (T_i - \mu_1 k \Delta x) e^{\mu_2 \theta_1} - \mu_1 k \Delta x \tag{4.2}$$

The warp yarns are then redirected by a rod to align them horizontally with the front warp beam, increasing the tension to produce a final tension, T_f, dependent on the initial tension, T_i, of the yarn and the compression of the spring, Δx:

$$T_f = T_i e^{\mu_2(\theta_1+\theta_2)} - \mu_1 k \Delta x (e^{\mu_2 \theta_2} + e^{\mu_2(\theta_1+\theta_2)}) \tag{4.3}$$

After the tensioning system, the yarn passes through the heddles. The guiding rods and front warp beam are horizontally aligned, and the heddle frame is positioned such that the raised and lowered heddle configurations are vertically equidistant from the front warp beam (Figure 4.9). This means that the total yarn length is nearly[1] equivalent regardless of the heddle position (up

[1] Distances vary slightly at the back, because the front-most guide rod contacts all warp yarns when heddles are lowered, and only the front-most yarns when heddles are raised. We considered adding an upper guide rod but found that, in practice, the tension was uniform enough (see Section 4.6.2.1).

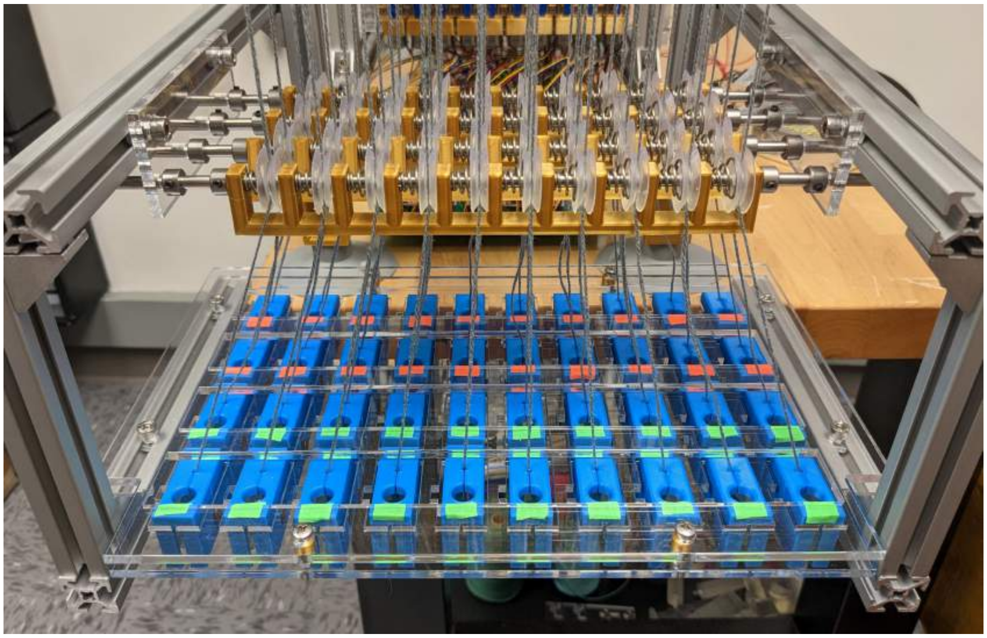


Figure 4.7: RoboLoom's novel tensioning system and creel. The creel has 40 individual cases with bobbins holding $\approx$ 6 meters of yarn. The yarns are then passed through the tensioning system. Each frame has its own tensioning rod.

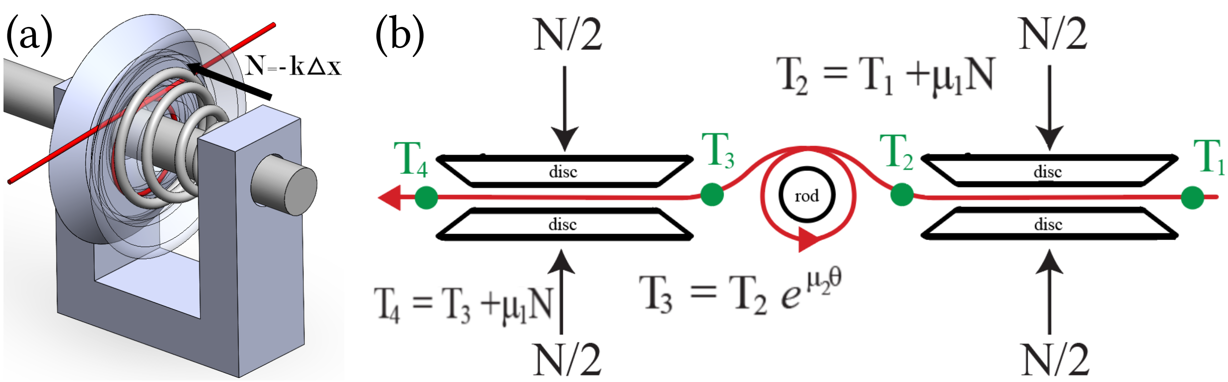


Figure 4.8: RoboLoom's tensioning mechanism consisting of (a) two tensioning disks, a conical spring, and a spacer with the yarn (red) wrapped around the rod. (b) Model of the tension of the system.

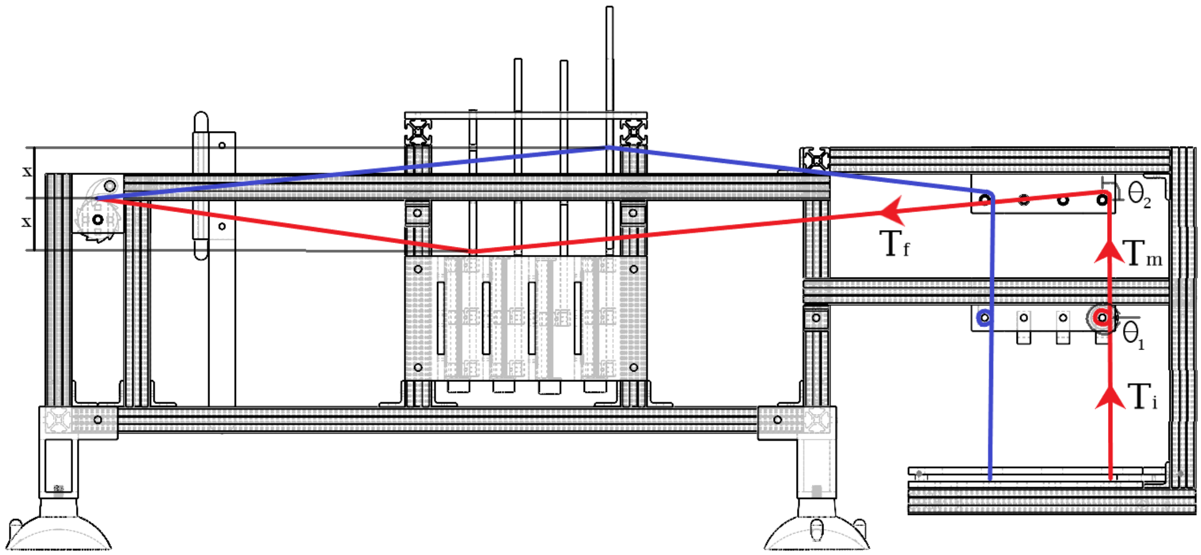


Figure 4.9: Different warp yarn positions in RoboLoom. Each heddle can be either raised or lowered, with the same amount of tension pulled on the warp yarn in either position. Yarns (blue and red) pass through the tensioning system and over the guiding rods before going through the heddles..

or down), ensuring reasonably uniform tension in the raised and lowered positions. Note that the total bobbin-to-beam lengths differ between individual warp yarns (owing to both heddle frame position and lateral deflection to reach the reed); but this is not an obstacle to uniform tensioning because of our individually-tensioned creel.

The passiveness of the system means yarns could lose tension while students are weaving on the loom (e.g., if they pull on a warp yarn accidentally when inserting the weft). This problem is easily fixed by novice weavers (Req. 4.4.1.2) requiring minimal effort (pulling the yarn and reeling it back into its bobbin) and minimal time ($\approx$ 1 sec.). To recreate proper tension, the weaver need only reel the yarn back into its bobbin.

4.5.1.2 Creel and Warping Routine

We designed RoboLoom with a creel system (individual bobbins) rather than a warp beam (one unified spool for all the warp yarns) that is typical for looms and used in all looms listed in Sec. 4.2. The creel eliminates the need to wind a back warp beam and makes threading easier to change (e.g., to fix mistakes) (Req. 4.4.1.1). To warp RoboLoom, the weaver winds individual bobbins[2], places them in cases, and installs the cases into the creel. Then to thread the loom, the weaver simply unwinds yarn from the bobbins one-at-a-time and threads it through the tension system, heddles, and reed, and then ties it down on the beam. Individual bobbins of yarn allow for quick *partial* warp exchanges if the weaver wishes to (e.g.) change half of the warp yarns for double cloth, makes a mistake in the threading process, or if a yarn breaks during weaving.

[2]In practice, one can do this in advance of a class.

This avoids a large potential source of discouragement for novice weavers (Req. 4.4.1.1 and Req. 4.4.1.2).

4.5.2 Electronics

RoboLoom is equipped with 40 linear actuators [6], which are driven individually in parallel. RoboLoom uses an Arduino Mega[3] [7] which is easily programmable by novices [19] (Req. 4.4.3.3). The Arduino commands 40 EasyDriver stepper motor drivers [98] through a series of MCP23017 port expanders [60] communicating over I2C. The firmware uses an interrupt system for driving the stepper motors with custom commands for running the motors designed for use by students from any background while remaining open for more advanced programming exploration (Req. 4.4.3).

4.5.3 Software

RoboLoom's graphical user interface (GUI), shown in Figure 4.10, is a Python program [90] that allows the user to create or load a pattern, visualize and explore the integrity and weave factor of their pattern (Req. 4.4.3.2), and control the loom (Req. 4.4.3.3). We chose Python to program the GUI in because it is an accessible programming language which then allows more advanced students to explore RoboLoom's algorithms (Req. 4.4.3).

RoboLoom currently has the capability to read in patterns as matrices stored in CSV files. However, many weaving draft softwares use the WIF file type that must first be converted to a CSV before use.

After uploading a pattern, RoboLoom's GUI provides pattern drafting feedback (Req. 4.4.3.2) through an illustration of the weave factor of a given row or column (Figure 4.10) which helps novice weavers notice long stretches of yarns without interlacement that create less sturdy cloth (Req. 4.4.3.3). The weave factor is calculated using horizontal and vertical pixel difference edge detection on the pattern matrix. This process is described in more detail in Appendix 4.2.2.

Additionally, RoboLoom's software allows students to explore cloth integrity employing a novel algorithm to assess if a cloth meets the mathematical criteria for "falling apart". This enables students to explore custom, complex patterns not guaranteed to have good cloth integrity (Req. 4.4.3.2). To our knowledge, our algorithm is the only real-time algorithm to calculate cloth layering using Grunbaum and Shepard's definition [38].

RoboLoom's software uses a novel algorithm to determine a cloth's integrity. We use the definition of cloth integrity as defined in section 4.2.2. The brute force method to determine if a pattern has integrity is to examine all possible sets of A yarns and B yarns and determine if any satisfy the definition. This algorithm runs in $O(2^{NM})$ time, for M columns and N rows. As shown by Figure 4.11, this works for smaller patterns (number of warp yarns less than 12) but takes too long to run for larger patterns, making it unfeasible for students to use to explore RoboLoom patterns of 40 warp yarns.

RoboLoom's algorithm, explained in Algorithm 1, examines each column and iteratively attempts to find an A and B set, containing the current column, that are not in the same layer. If

[3] Though an Arduino Uno would be sufficient.

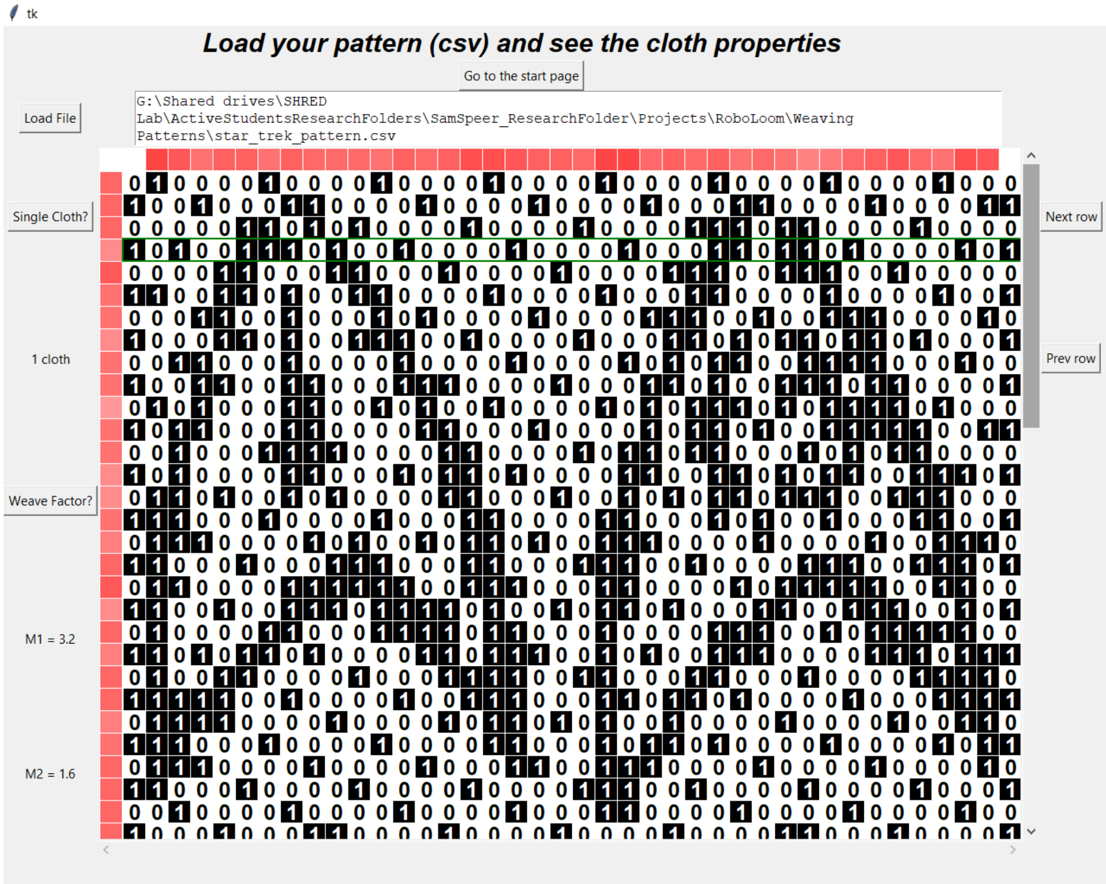


Figure 4.10: RoboLoom's graphical user interface. The interface allows the user to load a pattern, visualize the cloth, explore the cloth properties by using the 'Cloth Integrity' and 'Weave Factor' buttons, and weave the cloth by using the 'Next Row' and 'Previous Row' buttons.

no such set exists for any of the columns, we conclude the cloth is a singular layer. This algorithm is able to run at interactive rates as it has complexity $O(M(M+N))$. Even for large numbers of warp yarns, the algorithm is able to run in less than one second, as shown in Figure 4.11.

4.6 System Evaluation

To evaluate RoboLoom's weaving quality, warping and weaving efficiency, and cost we compared RoboLoom against other commercial and hobbyist looms. Our methods and results are discussed in the sections below.

4.6.1 Methods

We evaluated RoboLoom on weaving quality (number of warp yarns, EPI, and tension), warping efficiency (winding and threading time), weaving efficiency (shedding time), and cost require-

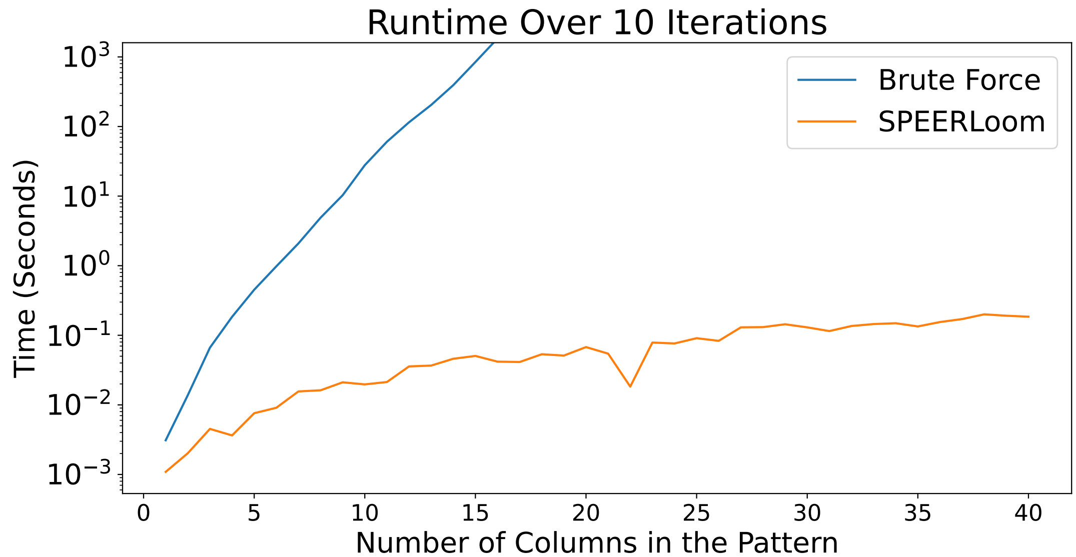


Figure 4.11: The graph shows the run time of each algorithm over 10 iterations of the algorithm as the number of columns in the pattern is increased.

ments (Section 4.4.3.3). We compare these results to other Jacquard looms (two commercial looms (the TC2 [74] and the Jacq3G [45]), one DIY loom (Albaugh's loom [5])) and a shaft loom (the Ashford Katie Table Loom [30]). Warp winding and threading time were estimated based on the time taken by non-experts (RoboLoom, Albaugh's loom, Jacq3g, and Ashford) and estimated by loom experts for the case of the TC2. Shedding time was timed for weaving basic patterns where 50-58% of the warp yarns were raised.

For evaluation of RoboLoom's tension, we compare the measured tension to the Ashford shaft loom to ensure comparable variance in per-yarn tension to a standard two-warp beam tensioning mechanism. Additionally, we evaluate our tension model through empirical measurements, ensuring the equation estimates the final tension properly. We estimated μ_1 and μ_2 through averaged measurements of tension at different stages of the system in Figure 4.8. We first varied N and measured T_2 and T_i to find μ_1 by averaging calculated values. We then varied T_2 and measured T_3 to find μ_2 by averaging calculated values. All values of tension were measured with a tensiometer[4] for different stages of RoboLoom's tensioning system and the Ashford loom's yarns.

4.6.2 Results

The results of the quantitative measurements taken for various looms are shown in Table 4.1 and discussed in the following sections.

[4] A Checkline Tensiometer Model TX SP-30 was used

Algorithm 1 RoboLoom's algorithm to determine if a cloth is a singular layer.

$A_r, A_c, B_r, B_c \leftarrow \{\}$
for each column c in pattern **do**
  insert c into A_c
  $A_r \leftarrow \{r \mid \text{pattern}(r, c) == 0\}$
  append $\{c \mid \text{pattern}(r, c) == 0 \text{ for } r \in A_r\}$ to A_c
  $B_c \leftarrow \{c \text{ for } c \in \text{pattern} \backslash A_c\}$
  $B_r \leftarrow \{r \text{ for } r \in \text{pattern} \backslash A_r\}$
  while B_c and B_r are not empty **do**
    if all elements of pattern(B_r, A_c) are 1 **then**
      if all element of pattern(A_r, B_c) are 0 **then**
        return falls apart
      end if
    end if
    append $\{r \mid 0 \in \text{pattern}(r, A_c)\}$ to A_r
    append $\{c \mid 1 \in \text{pattern}(A_r, c)\}$ to A_c
    $B_c \leftarrow \{c \text{ for } c \in \text{pattern} \backslash A_c\}$
    $B_r \leftarrow \{r \text{ for } r \in \text{pattern} \backslash A_r\}$
  end while
end for
return single layer

4.6.2.1 Weaving Quality

As shown in Table 4.1, RoboLoom meets or exceeds the cloth properties (warp yarns and EPI) of other DIY looms. While RoboLoom produces cloth of lower quality than commercial looms, we found that RoboLoom is able to weave cloth meeting design requirements 4.4.3.1 and 4.4.3.2, specifically meeting the definition of quality cloth, as defined in Section 4.4.3.1 with regards to EPI, number of warp yarns, and tension.

Figure 4.12 shows different cloths woven on RoboLoom. RoboLoom is able to weave basic patterns as well as more complex Jacquard patterns (Req. 4.4.3.2). These patterns were woven at 12 EPI, giving a sufficient quality of cloth (Req. 4.4.3.1). While this EPI is not as fine as commercial looms, it is more suitable for classroom use than other DIY options(Table 4.1). The lower EPI of Albaugh's loom and other DIY looms results in lower fidelity of patterning, reducing the complexity and visibility of patterns produced cloth.

Design requirements in Section 4.4.3 require RoboLoom to be suitable for novice and experienced weavers. Student weavers in a class taught with RoboLoom (see Section 5) had a range of background experience with textiles, but were all able to accomplish weaving cloth on RoboLoom. The students created custom patterns with matrix multiplication which can be clearly seen in Figure 5.3.

Figure 4.13 shows the tension on RoboLoom and the Ashford loom, demonstrating that RoboLoom's variance in tension is comparable to that of the Ashford loom. RoboLoom has an average of ≈199g of tension with a standard deviation of ≈10g (Req. 4.4.3.1). The Ashford

Table 4.1: Comparison of the different quantitative design requirements across a number of looms. All looms except the Ashford Shaft Loom are Jacquard looms. In this work we use max EPI to mean the maximum achievable EPI of the loom with each warp thread possibly individually actuatable. Winding and threading time are reported as minutes per warp yarn to account for differences in number of warp yarns.

Loom	Cloth		Efficiency of Use			Cost (USD)
	Warp Yarns	Max EPI	Winding (min/warp)	Threading (min/warp)	Shedding (sec)	
RoboLoom	40	12	≈0.25	≈0.75	6	$1097.17
TC2 [74]	440	180	≈0.5	≈0.75	1	$36,000.00
Jacq3G [45]	120	80	≈1.5	≈3	1	$31,449.50
Albaugh et al.'s Loom [5]	40	4	≈1.5	≈0.75	14	<$200.00
Ashford Shaft Loom [30]	320	40	≈2.25	≈2.25	5	$ 1,150.00

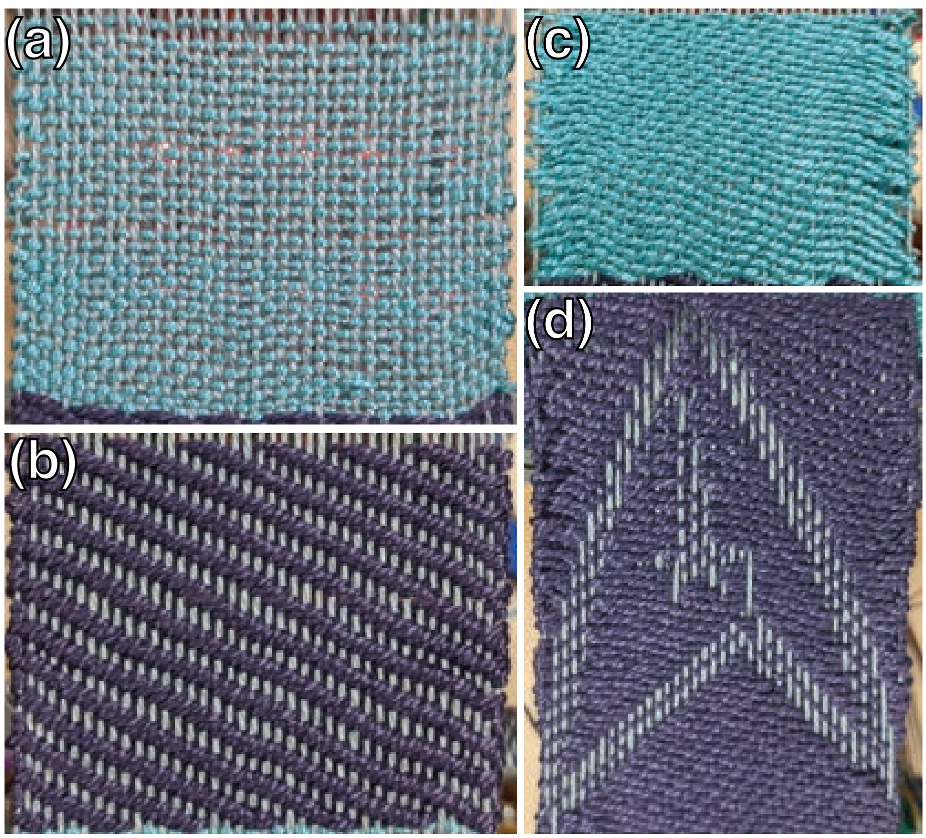


Figure 4.12: Cloth woven on RoboLoom: (a) Plain Weave, (b) Twill Weave, (c) Satin Weave, (d) Custom Jacquard Weave.

loom has ≈63g of average tension with a standard deviation of ≈28g. While the average tension of the two is different, this can be adjusted on either RoboLoom (by changing the compression of the spring) or the Ashford loom (by adjusting both warp beams) in order to fit the needs of

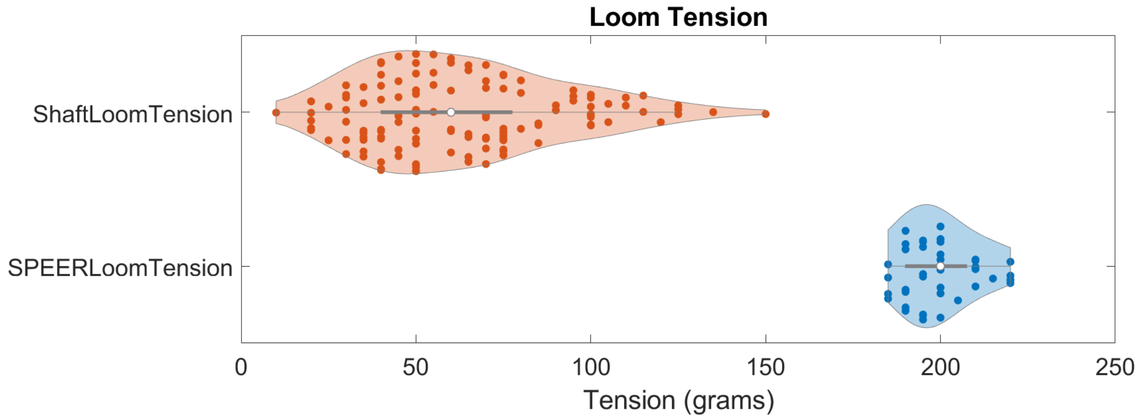


Figure 4.13: A violin plot showing the measured tension of two looms. The y-axis shows the tension on each warp yarn, the x-axis shows the probability density of the measurements. RoboLoom's variance in tension is smaller than that of the Ashford loom.

a specific weaving project. The variance of tension from yarn to yarn is something that cannot be easily adjusted on a two-warp beam style loom, as it would require re-winding the back warp beam. RoboLoom's yarns however can be individually adjusted to create a more consistent tension across all yarns. This shows RoboLoom's novel tensioning system is as consistent as an example two-warp beam tensioning mechanism, while saving time when adjusting individual yarn tension (Reqs. 4.4.3.1 and 4.4.1.1).

Warping and Weaving Efficiency We found that RoboLoom was more efficient than all other looms with regards to warping efficiency (Req. 4.4.1.1). Weaving efficiency on RoboLoom exceeded that of other the other DIY loom (Req. 4.4.1.2).

RoboLoom's tension system and creel were designed to eliminate the need for winding a back warp beam to satisfy requirement 4.4.1.1. This process can take ≈3-5 hours depending on experience. RoboLoom's creel was assembled in 20 minutes by a researcher. As shown in Table 4.1, RoboLoom's per warp time is quicker than that of other looms satisfying requirement 4.4.1.1. This saves students hours of warping time for each warp pattern they wish to weave.

To further satisfy requirement 4.4.1.1, RoboLoom is more efficient or as efficient as other looms with regards to threading. When measuring threading time, beginners threaded the Ashford loom [30], Jacq3g [45], and RoboLoom. Threading time for the TC2 [74] was reported by Digital Weaving Norway. All threading time is reported per warp yarn to account for the difference in number of warp threads. An important aspect of the loom threading process for beginners is that a large amount of time is spent correcting mistakes such as threading yarn onto the wrong heddle or in the wrong order. During the threading process for each of the looms, users made several mistakes. The difference we observed was in the time it took to recover from those mistakes. Threading yarn in the wrong heddle for the Ashford loom [30] or Jacq3g [45] meant having to redo most of the threading process.

We observed novice student weavers threading RoboLoom (see Section 5) and saw that when students made a mistake in threading their loom, it took them on the order of seconds to recover from their mistake. This was due to RoboLoom's ability to control, place, and tension each warp yarn individually, which enabled the students to swap and re-tension the affected yarns without having to re-thread any other warp yarns (Req. 4.4.1.1). In this regard RoboLoom is an improvement over the commercial and DIY alternatives.

As shown in Table 4.1, RoboLoom's shedding time is on par with other looms and, while it is slower than commercial looms, still satisfies requirement 4.4.1.2. The increase in shedding time over commercial alternatives is a direct result of the reduction in cost by a factor of 30. As compared to a serial mechanism in Albaugh et al.'s loom, RoboLoom has a much decreased shedding time. This decreased shedding time is a direct result of the increased cost for parallel actuation, but allows students to weave twice as fast.

RoboLoom's shedding time was not detrimental to students' ability to weave quickly. Students weavers in a collegiate class were able to weave the projects shown in Figure 5.3 over the course of a single week (see Section 5). This duration of weaving is comparable to other looms. Additionally, students commented that they feel as if they saved time weaving on RoboLoom by having the opportunity to mathematically explore their cloth properties, allowing for faster testing without requiring weaving time.

Accessibility RoboLoom meets the accessibility requirements stated in Section 4.4.2 through its moderate cost and open-source design. RoboLoom is much less costly than the commercial options, and somewhat more costly than Albaugh et al.'s Jacquard loom [5] (Table 4.1). The cost differential from the other Jacquard looms comes at the trade-off of quality and efficiency. RoboLoom has higher-quality cloth than Albaugh et al.'s Jacquard loom [5], but lower than that of the TC2 and Jacq3G. Additionally, RoboLoom has a higher weaving efficiency than Albaugh et al.'s loom which comes at the expense of higher cost.

We designed RoboLoom at a slightly higher price point to ensure the kit components would be durable, reusable, and reliable. Additionally, the open-source nature of RoboLoom allows users to swap components, potentially decreasing overall price and allowing for singular components to be easily replaced. We also designed RoboLoom to have more warp yarns and EPI allowing for more complex pattern exploration. Reducing the number of warp yarns and EPI to the minimal viable setup as stated in requirement 4.4.3.1 would reduce the cost of RoboLoom by $\approx \$250$ USD.

In order to make RoboLoom more accessible, we are currently working on reducing the cost of the frame (≈$270) by using more laser-cut and 3D-printed components and the electronics ($\approx \$300$) by using different motor drivers.

4.7 Conclusion

In this chapter, we presented the design of RoboLoom, an open-source, Jacquard loom kit for classroom use. RoboLoom's designs and other materials are available at: `https://sites.google.com/view/roboloom`. We have listed a set of design requirements necessary for a robotic loom to be an effective classroom tool for supporting interdisciplinary learning including efficient user interaction, accessibility, and interdisciplinary relevance. RoboLoom satisfies these requirements through its novel tensioning system and creel which create efficient warping and weaving interactions for beginner weavers. RoboLoom is accessible to classrooms with its moderate cost and open-source design. Finally, RoboLoom was used in a collegiate classroom to support students' interdisciplinary learning in textiles, math, and engineering, details of this study are provided in Section 5. From this, we conclude that RoboLoom supports efficient user interaction, is accessible in the classroom, and supports interdisciplinary engagement. In future chapters, we explore how RoboLoom encouraged collaboration during assembly and learning in a collegiate interdisciplinary classroom.

Chapter 5

RoboLoom Undergraduate Course Pilot

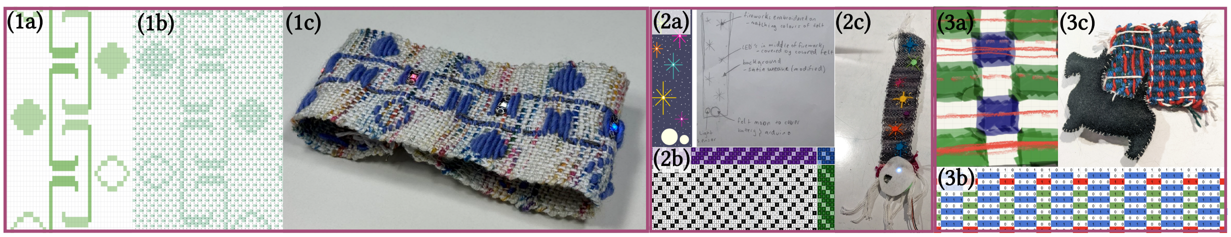


Figure 5.1: Student Weavings from the Pilot Undergraduate RoboLoom Course

5.1 Introduction

Past work has explored interdisciplinarity in primary and secondary education using weaving to teach mathematics and computational thinking (e.g. [54, 64, 92]). However, the concepts being taught and the supporting weaving technologies are limited. Most used simple cardboard looms or even construction paper for student pattern creation, restricting the complexity of the patterns and concepts that can be taught. Additionally, these pedagogical approaches to interdisciplinary learning do not include collaboration.

Our goal is to take advantage of the complex mathematical and engineering relationships with weaving to create collaborative interdisciplinary instruction for post-secondary classrooms. Toward this goal, we developed RoboLoom: an open-source Jacquard loom kit for supporting arts, mathematics, and engineering learning. We deployed RoboLoom in an undergraduate class and observed how the loom supported and hindered students' collaboration and interdisciplinary experiences through collaboration. In this section, I relay the findings and conclusions from this study.

5.2 Methods

To evaluate our hypotheses that RoboLoom supports post-secondary students' interdisciplinary experiences, we designed a course whose curriculum teaches concepts in weaving, engineering, and matrix math through the use of RoboLoom. Course materials can be found at: *https://sites.google.com/view/Roboloom*. We designed the course with input from engineering, math, and textiles instructors at Carnegie Mellon University and the University of California, Irvine. Weaving curricula mirrored other courses [13, 69, 91], but focused on the relationship of engineering and math principles to weaving.

The seven-week course was presented as a flipped classroom in a collegiate-level class. The course consisted of five synchronous in-person class sessions, each lasting three hours, and five sessions of asynchronous lecture videos, lasting less than two hours each. During in-person class sessions, students worked in interdisciplinary groups of three or four based on their background (i.e. textiles, math, or engineering expertise).

The course covered basics in weaving, math, and engineering, aligning with requirement 4.4.3, and used RoboLoom to support instruction. During the first week of class, lectures (1.3 hours) covered the basics of textiles including weaving drafts, weave structures, basic loom components, and culturally significant weaving. The second week, lectures (1.5 hours) focused on mechatronic principles including electronics and electromechanical actuation and their applications to RoboLoom. During the in-person session, students began building their RoboLoom kits. Week three of the course, lectures (2 hours) covered basics in linear algebra including vectors, matrices, basic operations with matrices (addition, subtraction, multiplication), and their applications to weaving drafts. Students continued building RoboLoom during the corresponding in-person session. Week four (lectures totaling 1.5 hours) built upon concepts taught in week three covering vector differentiation and its application to weave factor and cloth integrity. The in-person activities included weaving on RoboLoom and getting familiar with its operation and basic weaving paradigms. The final two weeks of the class were dedicated to the student's final projects.

As part of the assessment of students' understanding of the concepts, students completed a final project that required them to design an interactive textile and weave their design on RoboLoom. These final textiles had to be designed to be interacted with in some way, but the students chose the interactions themselves. Groups were observed during synchronous in-person meetings, while they worked on projects, and during final presentations.

5.2.1 Participants

We recruited students from the class to participate in research approved by Carnegie Mellon University's IRB. Each student gave written informed consent to participate. The study consisted of thirteen students, seven of which participated in a post-interview.

Out of 13 participants, 2 were male, and 11 were female. Students came from a variety of backgrounds: Fine Arts (n=7), Humanities (n=2), Engineering (n=2), Computer Science (n=1), and Information Systems (n=1), as well as class levels: Doctorate (n=2), Masters (n=2), Undergraduate (Senior (n=3), Junior (n=1), Sophomore (n=5)).

5.2.2 Data Analysis

We gathered data from observations during class, post-interviews, and anonymized classwork (detailed in Appendix C). Observations of students were collected as semi-structured field notes focusing on group dynamics, engagement, student expression of affect or ability, and physical interactions. Observations were collected during each in-person class session by trained researchers. Only groups where all students participated were observed (groups 2-4). Observational notes were then collected, and thematic analysis was performed on the notes. Observations were categorized into groups based on course activity (loom assembly, weaving, other course activities) and topic of student expression (efficacy, educational experience, group dynamic). Each category was summarized, and recurring themes were noted. We found themes pertaining to student engagement, student expression of confidence or learning of art, math, or engineering skills, and student perception of groupmates' efficacy.

Students participated in 20-minute post-course interviews reporting on effects on self- and other-efficacy, learning of art, math, and engineering skills, and RoboLoom interactions. Questions focused on if and how the assembly of and interactions with RoboLoom affected their skills in art, math, and engineering ("Did building your loom affect your engineering skills?") and if they felt they learned from these experiences ("Did you feel you learned anything about math, engineering, or art throughout the class?"). Thematic analysis was performed on the interview data using the same categories and themes as the observational data.

Collected coursework included surveys of student background, student reflections on activities, post-lecture quizzes, and final project assessments.

Themes from observations, interviews, and assignments (including the final project) were categorized into engineering experiences from assembly and interdisciplinary experiences from other course activities. These findings are discussed in the sections below.

5.3 Collaborative Assembly

Through our observations of the assembly process, we found there were features of RoboLoom's design that encouraged students to work together to achieve the assembly and features that did not encourage collaboration and allowed students to work individually.

5.3.1 RoboLoom Features Encouraging Collaboration

During the assembly process, two of the three groups chose to attempt to parallelize their assembly process, splitting their group of four into two groups of two and working on different subsystems of RoboLoom. The third group, group three, chose to work together on their assembly. Group three was able to collaborate with four people all working on a single task. They chose a group dynamic where one member would read instructions and disseminate the information to other group members while they worked on various smaller tasks of assembly. This allowed for collaboration but was a slower method of assembly.

Groups two and four chose to split into pairs to parallelize task accomplishment, with the more experienced student in each pair taking the lead. We observed that RoboLoom's features allowed students to do this without decreasing the amount they needed to collaborate to complete tasks. RoboLoom's subsystem design allowed groups to collaborate in pairs. When it was time to assemble subsystems together, however, groups were forced to come back together and collaborate in groups of four. This process allowed students to more quickly assemble RoboLoom, while still having important collaborations with group members.

We observed across all groups that students would always collaborate specifically on physical assembly tasks that were made easier when students worked together to complete them. For example, we observed two students assembling frame pieces together with one student holding the piece in place, while the other secured the pieces with the provided brackets. The need for multiple sets of hands led to collaboration on those specific tasks and future tasks. In one instance, a pair was working disjointly at first, when one student asked for an extra set of hands from another student. After completing that task, the two students continued to work together on the next steps. Two students reported in their interviews that "you needed another set of hands to hold things" (S2), with one student noting "sometimes they needed two people together" (S5) and would schedule more collaborative sessions to build the loom outside of class because of this.

Another instance of encouraging collaboration was when students needed more guidance than the provided instructions gave. One student reports the instructions were "sometimes not specific so they had to talk with each other" (S5). In these cases where students had to problem-solve to reach the right assembly, they collaborated within their group and with others in the class.

Some features of RoboLoom allowed for collaboration, offering a decrease in effort when collaborating. After finishing the assembly of RoboLoom, we observed how students chose to weave on it. All groups chose to weave with two students controlling the loom, one controlling the UI, and the other controlling the weft yarn. Students would communicate the desired state of the loom's heddles to ensure proper positioning. While these tasks could theoretically be completed by a single student, all groups recognized it was easier to do with multiple people and chose to collaborate to do so. This highlights that even with those groups less inclined towards collaboration, they would collaborate when they perceived it would alleviate work.

5.3.2 RoboLoom Features Not Encouraging Collaboration

Conversely, we observed a few features of RoboLoom that did not require collaboration from students. These features were commonly repetitive, individual tasks such as sanding down 3D-printed parts. Groups with collaborative group dynamics at this point in the class assigned one or two group members to complete the task. Some took the opportunity to talk with their group to get to know their background and experiences better. However, in group three a disagreement regarding the assembly of the loom created a small rift between the members. One group member took the opportunity to do the individual tasks as a way to leave her group briefly and talk with other students in the class. Additionally, we observed the group with the most congenial group dynamic completed their threading task fully collaboratively, despite the task being completable by an individual. However, the teams with less inclination towards collaboration completed this task more individually. While these design elements of RoboLoom can be collaborative, there is a need for more encouragement of collaboration during these tasks, whether from the design of the loom itself or from the instructor or other course materials.

5.4 Collaborative Experiences

Students of the class worked in groups throughout the course. The following sections discuss students' educational experiences as they went through the course.

5.4.1 Engineering Experiences During Assembly

We predicted students would engage with system construction skills and would consider systems engineering concepts while assembling RoboLoom. From our observations of student assembly and students' reflections on the assembly process during interviews, we found that they did explore these concepts. Six of the seven students reported that their engineering skills increased through the building interaction with RoboLoom. The remaining student reported RoboLoom's assembly was a new application of their skills.

Four students reported considering the systems engineering principles regarding the design of the loom during assembly (Req. 4.4.3.3). Some students did not initially look at the assembly process as learning because they were simply following instructions. However, upon diving deeper into their interaction, they recall applying problem-solving skills and considering design elements when they struggled with the instructions.

> It was more putting parts together rather than actually messing with any of the things themselves... Can I take back what I said about the engineering earlier? The whole process helped with engineering and thinking about interactive design. (S8)

RoboLoom's open design allowed students to reflect on other systems engineering aspects, i.e., the requirements of a loom, the design decisions made to satisfy them, and component design within the system (Req. 4.4.3.3). Students were observed misassembling the loom, breaking components in the process. Students used these points of friction as a learning exercise to understand the function of the broken or misassembled part and brainstorm components that achieve the same functionality. For example, one group struggled to mount the rods on the back of the loom, but S10 was able to find a different mounting method that could hold the same amount of force.

Students also reflected on RoboLoom's weaving capabilities as well as design modifications that could be made to increase its capabilities (Figure 5.2). Students reflected on component design considering how different components could be redesigned within the parameters of the machine. For example, we observed that students in group 2 struggled with assembling the feet of the loom and discussed changes that could be made to the 3D-printed parts to improve the assembly process while still providing a sturdy base for the loom. These reflections increased their understanding of engineering concepts and helped them complete the assembly.

Four students reported engaging with system construction skills (Req. 4.4.3.3). Some groups split into pairs during the assembly process, led by students more experienced in engineering. We observed less experienced students gathering information about system construction from

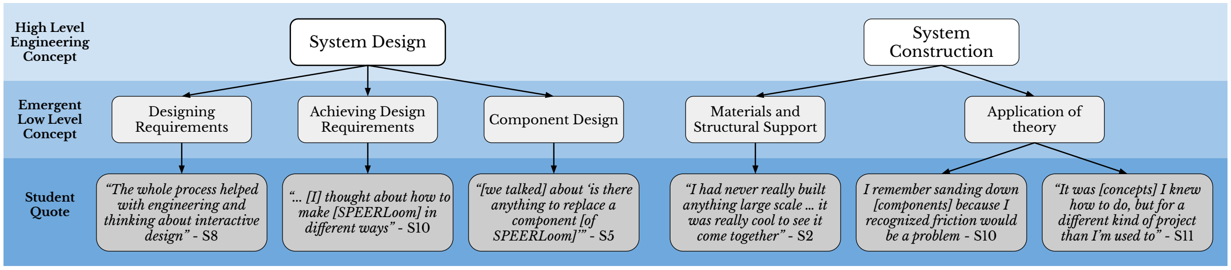


Figure 5.2: A hierarchical organization of the concepts students reported learning in engineering while building RoboLoom. The top layer describes high-level concepts targeted by the curriculum, the middle level presents low-level concepts that emerged through data analysis, and the bottom layer shows quotes supporting the low-level concepts.

their assembly partners. S5 reported learning a lot from a group member with a mechanical engineering background because "[they] knew a lot about how to build the loom, so there were things [they were] able to explain". More experienced students would note the application of different theoretical knowledge in engineering and how it could apply to the real-world example of RoboLoom, for example, a student's application of their knowledge of friction (Figure 5.2). We observed more experienced students in group 4 teaching S2 the basics of fastening pieces together with screws and nuts at the beginning of the assembly. S2 commented on their lack of experience (Figure 5.2) but by the end of the assembly they were completing these tasks without help and reported they "were more confident in [their] ability to learn things".

During observations and interviews, students often referenced exploring art, engineering, or math in the context of other disciplines. For example, students considered how the systems engineering of RoboLoom influenced its creation of textile art and considered how they could apply the engineering design process [58] to their artistic designs (Figure 5.4). Additionally, students discussed exploring math and weaving together, speaking to the linear algebra skills used to create and describe artistic expression in the form of textile patterns. These findings are further discussed in the sections below.

5.4.1.1 Engineering and Art

In accordance with requirement 4.4.3.3, students were able to trace the weaving process from the computer input to the electronics to the mechatronic actuators. In week 4, when group 3 ran their RoboLoom for the first time, students came together to use learned knowledge and prior experience to holistically analyze the weaving process of RoboLoom. S7 explained to their group how the computer commanded the Arduino, which controlled each motor. S11 then explained how the motors are creating the shed on the loom by raising the warp yarns, allowing them to pass a shuttle through and create a row of weaving. The team then discussed how the moving motors were impacting their cloth design. In this interaction, novice students explored how mechatronic elements serve specific purposes, applying general engineering skills to weaving.

During the final project, students reported exploring weaving through the context of engineering challenges (Req. 4.4.3). Due to the physicality of their produced cloth, students had to consider constraints and change their artistic design through the use of their engineering problem-

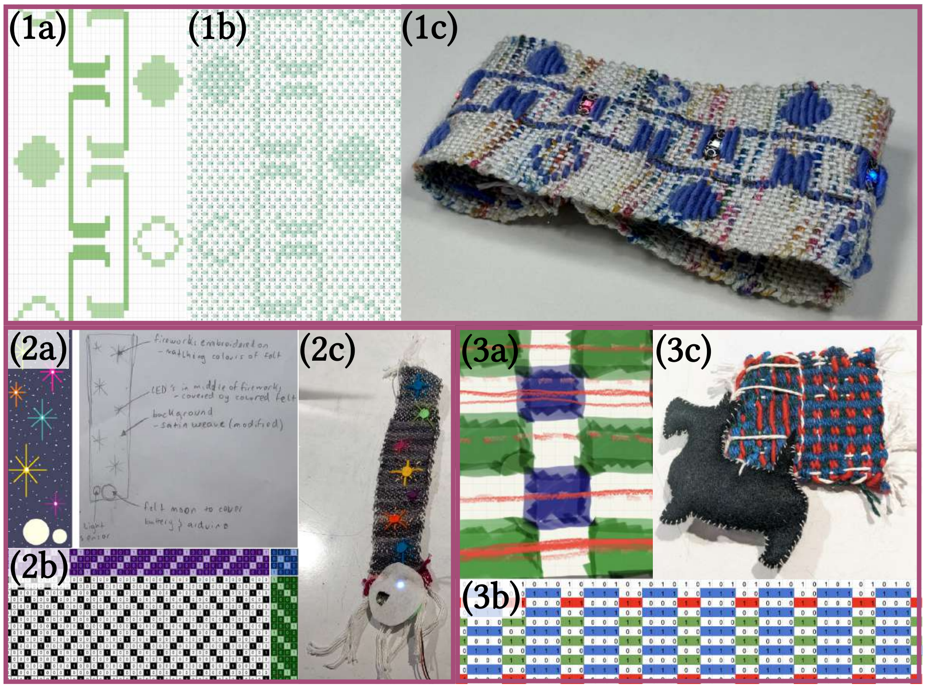


Figure 5.3: Participating students' final class group projects' designs and prototypes (1,2,3). (a) Initial feasible designs or models of the projects. (b) Revised design after testing the mathematical principles of their design. (c) Final prototype.

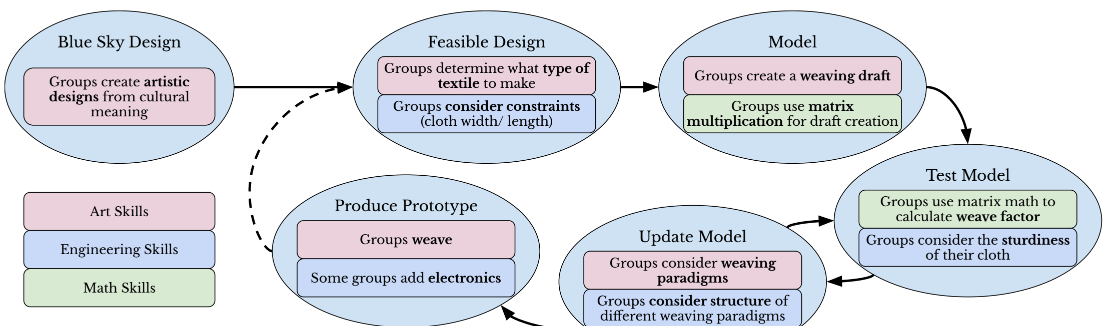


Figure 5.4: Groups' workflow for the interdisciplinary course's final project. This workflow resembled the engineering design process steps (larger blue bubbles). At each step of the larger process, students considered principles from art (pink), math (green), and engineering (blue) and how they interacted in the context of their project.

solving skills. This happened in both the feasible design steps and test model steps shown in Figure 5.4.

> Our [final project] was iterated on in a way that felt like engineering to me. It was kind of iterative, [we would] talk about a solution and then pick it apart in conversation and then go back to a new idea, iterating in a problem-oriented way of thinking that usually happens in engineering. (S7)

Another student reported that when designing their cloth for the final project, their team originally designed something too large and over ambitious for the timeline and cloth size constraints, leading them to scale their project down to optimize for the physical constraints they had. This is captured in Figure 5.4 as the transition from blue sky design to feasible design.

Students also discussed other engineering considerations affecting weaving interactions and the "potential for the loom to make more complicated things for them" (S15). S2 expressed that using RoboLoom's software to visualize their final design enhanced their artistic skills and allowed them to better picture how the cloth would come together.

5.4.1.2 Math and art

In order to satisfy design requirement 4.4.3.2, the curriculum included matrices and their basic operations as they apply to weaving. All but one student reported in the interviews that they experienced math skills in the class. The singular student who did not mention experiencing new math skills reported having a strong background in math. Many students reported that the contextualization of matrix multiplication in weaving was more meaningful to them than previous experiences learning linear algebra. S11 stated the interactions with RoboLoom gave them "a more nuanced understanding of the math". S15 even reported

> The transfer of calculations back to a physical fabric helps stimulate my brain in a different way and see math in a physical fabric.

Students echoed this sentiment through their interviews saying that the calculation of weave factor "refreshed those [matrix concepts]" (S5) and after "understanding the matrices and how it makes [patterns] ... calculating the [weave factor] then calculating the matrices refreshed how to do those things" (S10). Although the weave factor could be automatically calculated, students still made smaller pattern calculations by hand, using the automatic calculation only for verification. Students reported "learning about weaving drafts was good for [their] math skills" (S8) and it was "cool to see how math can make patterns" (S2).

The students demonstrated their experiences and solidified direct connections between textile art and matrix math during the final project. Students used matrices to create their final patterns (Figure 5.4, model step) and the weave factor to determine the physical properties of their cloth (Figure 5.4, test model step). Student calculated the weave factor prior to weaving their cloth realizing their cloth may not have the desired physical properties for the designed application. For example, one group made a woven book with a weaving pattern that would yield a loose cloth unsuitable to bind into a book. They discovered this by calculating the weave factor and then applied a different weaving paradigm to increase the weave factor and strengthen their cloth.

Overall, interactions with RoboLoom were reported to have supported the student's educational experiences in the class. S8 captured the course's interdisciplinarity, reporting "Art-wise [interactions with RoboLoom] helped with thinking about patterns and how models and prototypes are represented as actual things. With the weaving drafts [RoboLoom helped with] how they translate the 0s and 1s into an actual design".

5.5 Conclusions

In this Chapter, I presented a study done on the implementation of RoboLoom in an undergraduate course taught at CMU. The course covered basics in weaving, math, and engineering. Thirteen students participated in the study. Through observations and interviews, we collected data on how students assembled and used RoboLoom throughout the course. We found that there were some features that allowed and perhaps encouraged students to work collaboratively in their groups. The parallelization of RoboLoom allowed for pairing off to work and increased assembly speed. We also observed some physical elements, like size and number of moving parts encouraged collaboration. However, we also observed opportunities for individuals to split from the group during assembly on tedious and repetitive tasks and not work collaboratively. These observations begin to form an answer to our first research question ("What features of a robotic system help facilitate collaborative assembly?"). Further study on these features is conducted in future chapters.

We also observed and recorded self-reported collaborative experiences from the undergraduate course. We observed students working together to understand the assembly, self-reporting that the process helped them explore engineering concepts. We also observed and corroborated with student interviews that students explored interdisciplinary skills in weaving, art, math, and engineering through their collaborative work on RoboLoom throughout the course. The interdisciplinary relevance (Section 4.4.3)features supported the students collaborating on RoboLoom, providing an answer to our second research question ("What features of a robotic system help facilitate collaborative experiences in a classroom?").

5.5.1 Future Work

These observations result in the initial basis for future work studying the specific hardware features that provide opportunity for and encourage collaboration. Future work deepens the understanding of collaborative behaviors to classify specific actions as collaborative or not. We also rigorously record and determine the cause of collaborative behaviors in our future studies. This work is described in the next Chapter of the thesis.

Chapter 6

Loom Assembly Study

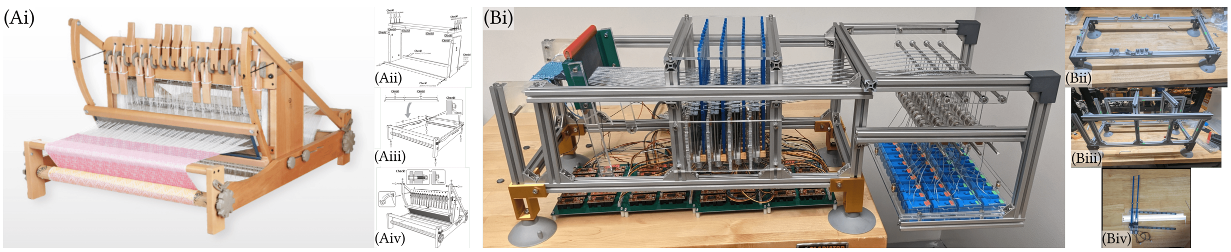


Figure 6.1: The Ashford Loom [109] (Ai) and RoboLoom [102] (Bi) in their built and partially built stages. For the Ashford loom, (Aii) shows the castle assembly, (Aiii) shows the frame assembly, and (Aiv) shows the beater assembly. For RoboLoom, (Bii) shows the base frame assembled, (Biii) shows the final frame assembled, and (Biv) shows partial assembly of the motors.

6.1 Introduction

We have established in previous sections that collaboration is a core skill for future engineers as they navigate their careers in industry or academia [1, 2, 3]. In order to prepare students for their future, collaboration needs to be taught in higher education engineering curricula [70]. Collaborating on engineering projects helps future engineers learn through doing and prepares them for their work. In order to do this, it is necessary to have a project that facilitates good, natural collaboration between students as they work together to solve engineering tasks.

From out initial observations of RoboLoom's collaborative assembly by students, we hypothesized there are specific design features that can hinder or support collaborative assembly when building a device in the classroom. We designed a study to explore and enumerate these features, not just in our design, but in the design of commercial products that could be substitutes for RoboLoom in the weaving class. Through this study we aim to understand and enumerate the opportunities for coordination, cooperation, and communication in devices designed for assembly.

Toward this goal, we studied the assembly of two weaving machines: RoboLoom and an Ashford Table Loom. RoboLoom is a robotic Jacquard loom kit designed for collaborative assembly in higher educational classrooms. The Ashford table loom is a commercial loom that is designed for assembly and use by the individual weaver. We compare the collaboration during the process of assembling each of the looms, finding that certain design features and decisions lead to collaboration across both looms, while other features did not.

To analyze participants' collaborative behaviors during building, we developed and applied a coding scheme based on the 3C's model of collaboration [25, 33] (Coordination, Cooperation, Communication). We then used thematic analysis of these actions within codes to find hardware features that promoted and hindered collaboration. Particularly, we found five categories of hardware features that created the opportunity for collaborative actions: repetitiveness (DP1), specificity (DP2), difficulty (DP3), parallelizability (DP4), and physicality (DP5).

We found that collaboration was influenced by these hardware principles in the following ways:

C1. Coordination can be facilitated by hardware that is repetitive (DP1), semi-specific (DP2), physically and mentally challenging (DP3), parallelizable (DP4), and requires simultaneous manipulation (DP5).

C2. Cooperation is possible with hardware that is repetitive (DP1), physically challenging (DP3), parallelizable (DP4), or requires simultaneous manipulation (DP5).

C3. Communication is encouraged by hardware that is repetitive (DP1), is semi-specific (DP2), is physically and mentally challenging (DP3), or requires simultaneous manipulation (DP5).

C4. Collaboration can be blocked by hardware that is too mentally challenging (DP3), not parallel (DP4), is small (DP5), or does not require simultaneous manipulation (DP5).

6.2 Related Work

Based on the definitions of collaboration in section 2.6, technology that supports collaboration must be designed for supporting communication, coordination, and cooperation [65] and thus designed for the support of shared meaning across different contexts, of organization and shared understanding of state space between actors, and of joint operations in the workspace [65]. Designing effective collaborative problems requires that tasks make working together necessary for the achievement of the task and that goals cannot be achieved by the individual [8]. Kirschner et al. provide further guidelines for the design of educational collaborative technologies [51] including technological usability and educational support stating that the technology should fulfill the learning intentions of the group instantaneously and guide members to use the learning interventions when needed.

6.3 Methods

In this exploratory study, participants collaborated in groups to construct two hardware kits: RoboLoom and the Ashford Loom. The aim of the study was to gain insight into 1) what design features of the RoboLoom/Ashford loom support collaboration, 2) conversely, what design features of the RoboLoom/Ashford loom hinder collaboration, and 3) how interactions with the RoboLoom/Ashford hardware and components support collaborative actions.

6.3.1 Loom Kits

We chose to use looms as an initial starting point for studying hardware design features that can eventually be studied more generally. This choice was made due to the existence of decades of design for looms for end-user assembly, as well as the existence of do-it-yourself kits that have begun to be used for higher education engineering instruction [102] and our prior results from Chapter 5.

We chose RoboLoom due to the prior research showing initial differing interactions with assembling the RoboLoom [102]. We chose the Ashford Loom (Figure 6.1) because it is a commercial counterpart designed for end-user assembly. As they are both looms, they share common design features for loom functionality, but differ in the specific design and assembly of these features. For example, both looms have a base frame that holds the mechanism for yarn manipulation in the middle. We use this as an opportunity to observe collaborative actions with hardware designed in different ways for similar functionality.

Instructions for both loom kits are explained in Appendix B.

6.3.2 Participants

Eight students participated in the study, recruited via email lists, Slack, etc., including informatics graduate students. Seven were graduate students in Informatics or Education, and one was an undergraduate student in Computer Science. Participants were grouped into two groups of four and distributed so that each group had a diverse disciplinary background. All participants had diverse sets of expertise in the arts, design, science, technology, engineering, and mathematics (STEM), programming, and human-computer interaction (HCI). Both groups engaged in the construction of both looms – Group 1 built RoboLoom first, then the Ashford Loom, while Group 2 did the reverse. Assembly sessions were ≈3-6 hours per loom (Table 6.1) over four days. Participants were provided with instruction manuals[1] and the necessary tools to complete the assembly. Facilitators were available to answer questions along the way, without intervening in group work. This study focuses on investigating the hardware features that support or hinder group collaboration. The aim of the analysis is exploratory, rather than to determine causation.

6.3.3 Data Sources and Analysis

We collected audio and video data from the assembly sessions and constructed one data log per group per loom (four data logs total). Each log documented the assembly process in increments of 5 minutes. Each 5-minute section was summarized and labeled with the assembly step(s),

[1]RoboLoom instructions found at: `https://sites.google.com/view/speerloom/assembly-guide`. Ashford Loom instructions found at `https://www.ashford.co.nz/instructions/SS610.pdf`.

Table 6.1: Time spent on assembly per group per loom given in the format of h:mm:ss as well as the number of data points (5-minute segments) in the data log.

Group	RoboLoom		Ashford Loom	
	Time	Data Points	Time	Data Points
1	5:47:12	70	4:36:20	56
2	5:02:36	61	2:54:12	35

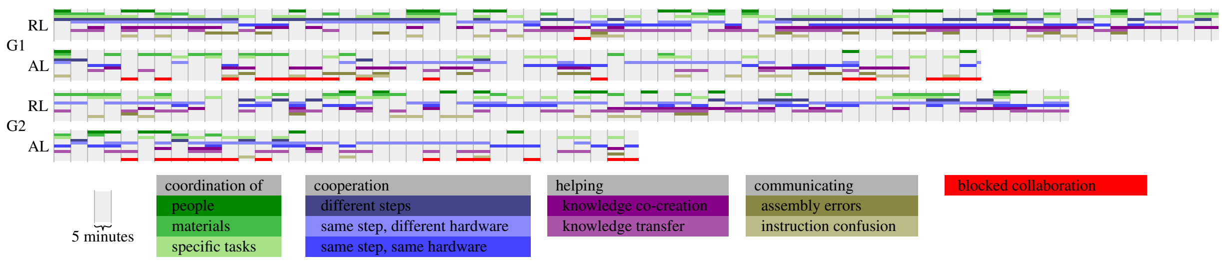


Figure 6.2: A graphical summary of the codes assigned to each 5-minute block of each loom construction session. "RL" and "AL" denote the RoboLoom and Ashford Loom, respectively.

engaged hardware, and group members. Table 6.1 shows the number of 5-minute segments (Data points) as well as the total assembly time for each loom. Figure 6.2 gives a graphical summary of the coded data.

In our subsequent iterative analysis, we selected the joint collaborative actions as the unit of analysis [105] to understand the actions taken by the group as a whole, rather than examining individual contributions. We developed our coding scheme by building on existing observational data of students building the RoboLoom in a prior classroom study [102], which revealed different group dynamics during and after assembly. Cooperative hardware interaction on different steps may have led to disjointed group dynamics. Cooperation on the same hardware may have led to more group cohesion.

We then combined these emergent observations with the 3Cs model of collaboration to specifically capture the nuance between the modes of collaboration (coordination, cooperation, and communication). From this, we defined four code categories with specific codes in each grounded in both data and theory. These codes are not mutually exclusive and often appear together as coordination, cooperation, and communication build together to create collaboration as a whole. Additionally, we code for instances where collaboration was attempted, but not completeable due to hardware design. This blocked collaboration could lead to adverse effects, like unwillingness to participate or negative affect [75, 83]. The codes are:

1. *Coordination* - participants are engaging in behavior to manage others, activities, or resources needed to assemble the loom (e.g. defining group roles, gathering loom materials, etc.)
   (a) *Coordination of People* - Participants are managing the distribution of labor (e.g. the group decides to have participant 1 do task A and participant 2 do task B).
   (b) *Coordination of Materials* - Participants are managing the materials for the task (e.g. participant 1 calls out materials while participant 2 finds them).

   (c) *Coordination of Specific Tasks* - Participants are managing how and when to accomplish their activities.

2. *Cooperation* - participants work separately on tasks that both contribute to the shared assembly goal (e.g. building separate parts of the loom concurrently)
   (a) *Cooperation - Different Steps* - Participants or participant groups work on two steps of assembly concurrently (e.g. building the loom frame and mounting the loom's motors).
   (b) *Cooperation - Same Step, Different Hardware* - Participants are working on the same step of assembly, but interacting with distinct pieces of the loom (e.g. participant 1 builds a shaft of the loom, and participant 2 builds another).
   (c) *Cooperation - Same Step, Same Hardware* - Participants are working on the same step of assembly and interacting with the same loom materials (e.g. participant 1 holds pieces together while participant 2 secures them).
3. *Communication* - Participants discussing ideas and confusion during the assembly process.
   (a) *Helping: Knowledge Co-Creation* - Participants are creating knowledge about the loom's assembly with each other. This requires contributions from all participants involved to build the knowledge together.
   (b) *Helping: Knowledge Transfer* - One participant transfers knowledge to another with the goal of aiding the second in the task.
   (c) *Communicating About Troubleshooting* - Participants talk through troubleshooting when there is an error in the assembly process (e.g. participant 1 determines a part is backward and the group discusses how to fix it).
   (d) *Communicating Instructions Confusion* - Participants communicate when the instructions are unclear (e.g. Participants determining which pieces the instructions refer to).
4. *Blocked Collaboration* - A participant attempted collaboration but did not succeed. This does not include participants who are off task.

This coding scheme is summarized, and examples for each code are provided in Appendix A.

For each five minutes of assembly, we coded joint actions with the applicable code(s) and the engaged design feature. Two coders coded the data, with 20% of the data coded by both to determine Cohen's Kappa for inter-rater reliability (Table 6.2). As shown in Figure 6.2, the results from our coding show coordination, cooperation, and communication co-existing within the same 5 minutes. This is expected as all three are often a part of a successful collaborative action. We then performed thematic analysis for each code to understand what hardware features supported or hindered collaboration through cooperation, coordination, and communication.

6.4 Results

We qualitatively analyzed the observation data to determine the percentage of time groups spent collaborating and which design features they interacted with while collaborating. We summarize our findings in Figures 6.3, 6.4, 6.5, and 6.6, which report the percentage of data points (5-minute

Table 6.2: Inter-rater Reliability for each code.

Code	Cohen's Kappa (κ)
Coordination of People	0.88
Coordination of Materials	1.00
Coordination of Specific Tasks	0.91
Coop - Different Steps	1.00
Coop - Same Step, Different Hardware	1.00
Coop - Same Step, Same Hardware	1.00
Com - Helping: Knowledge Co-Creation	0.88
Com - Helping: Knowledge Transfer	0.94
Communicating About Troubleshooting	0.85
Communicating Instruction Confusion	0.93
Blocked Collaboration	1.00

segments) that received each code. Percentages were calculated by counting the number of data points marked with that code, and then dividing by the total number of data points (found in Table 6.1).

6.4.1 Coordination

The results for the coordination codes are shown in Figure 6.3. Notably, the ideal amount of coordination during a collaborative task should be in the mid-range of percentage of time [33]. Too little coordination could indicate a lack of collaboration, and too much coordination could indicate that the coordination is not working as intended [33].

6.4.1.1 Coordination of People

Figure 6.3 shows that Group 1 coordinated people 19% more during their RoboLoom build. Coordination of people occurred when tasks (1) were parallel (DP4), (2) were physically difficult (DP3), (3) had physicality requiring simultaneous manipulation (DP5), (4) had non-specific hardware (DP2), and (5) were repetitious (DP1).

Group 1 began by splitting into pairs to parallelize the assembly of RoboLoom's frame and motor assemblies, necessitating coordination of pairs and task assignments. As they continued, they coordinated people for the new set of tasks, shown in Figure 6.2. The Ashford Loom has a more serial approach to the frame and yarn manipulation portions of the machine. We hypothesize that this difference in serial assembly method led to fewer opportunities for coordination of people.

Physically difficult tasks allowed for coordination of people through specialization. For example, P1 struggled with dexterous tasks (aligning a T-Nut and attaching RoboLoom's length beam) and P4 took over. Additionally, tasks requiring larger forces provided the opportunity to coordinate people. For example, P2 took over for P3 as she tried, but failed to lift the castle of the Ashford Loom. Both looms had difficult hardware, but we observed more instances in

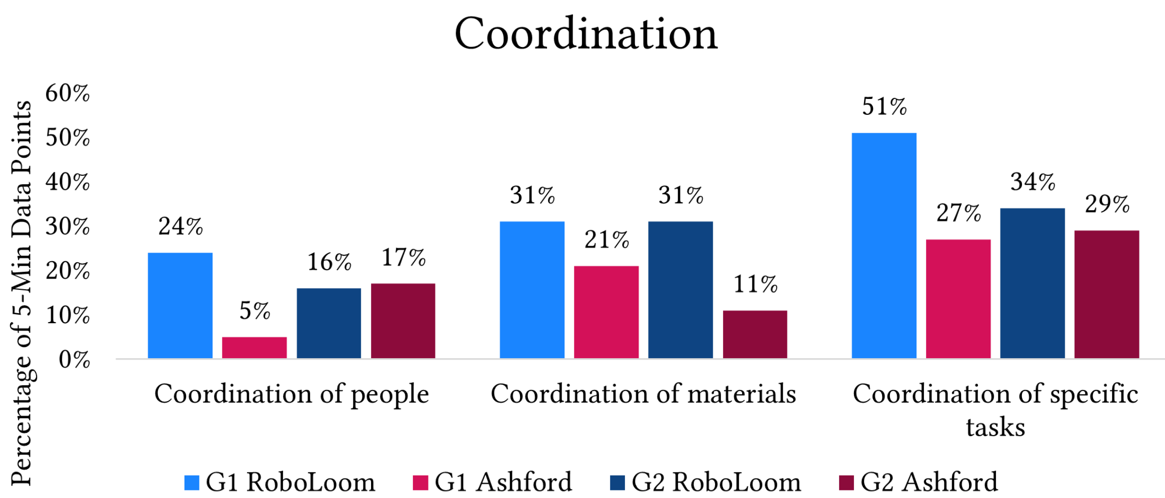


Figure 6.3: The percentage of 5-minute data points coded for the coordination codes for each of the four test cases. The bars show higher coordination across codes in at least one RoboLoom build.

RoboLoom.

Physicality requiring simultaneous manipulation prompted coordination of people as well. For example, aligning RoboLoom's heddle guide requires 40 heddles to be manipulated simultaneously, leading P1 and P4 to coordinate. A similar event occurred on the Ashford Loom, when P2 asked for help holding the beater. Fewer such moments were observed on the Ashford due to less simultaneous manipulation.

Non-specific hardware introduced uncertainty, so participants coordinated to have group members check their work. For example, P1 asked P4 to double-check a measurement she took, requiring coordination of people.

Finally, repetitive hardware created the possibility for coordinating people. RoboLoom requires 46 corner assemblies, so P2 coordinated with her group to split the work for speed. While individually accomplishable, the potential for optimization was reason enough for P2 to coordinate more help.

6.4.1.2 Coordination of Materials

Figure 6.3 shows that Group 2 increased their coordination of materials by 20% during RoboLoom assembly. This occurred when tasks were mentally difficult (DP3) or during parallel (DP4) tasks to share tools.

RoboLoom uses many unfamiliar hardware components. To reduce mental difficulty, P6 coordinated the organization of the materials. This happened frequently in the RoboLoom assembly (15/20 data points). This occurred less in the Ashford Loom assembly (2/4) as participants were more familiar with the materials, and materials were labeled for each step.

Group 2 also coordinated materials when they had multiple, short, individual, parallelizable tasks that required the use of the same tool during RoboLoom's assembly. The Ashford Loom's task required more time, and participants did not coordinate sharing the tool; instead, they serialized their assembly.

6.4.1.3 Coordination of Specific Tasks

Group 1 increased coordination for specific tasks during their RoboLoom build by 24% (Figure 6.3). Group 1 coordinated specific tasks during their RoboLoom build because tasks (1) were non-specific (DP2), (2) were mentally difficult (DP3), (3) were repetitive (DP1), (4) could be optimized for time or ease through parallelizability (DP4) or non-specificity (DP2), (5) had physicality requiring simultaneous manipulation (DP5), or (6) had physicality involving manipulating the same object (DP5). During Group 1's Ashford Loom assembly, the participants coordinated on specific tasks where tasks (1) were non-specific (DP2), (3) were repetitive (DP1), or (5) had physicality requiring simultaneous manipulation (DP5).

Coordination of specific tasks was observed when hardware was non-specific and could be assembled incorrectly. For example, RoboLoom's frame requires that the corner assemblies be built in a specific orientation, but this is not enforced during their assembly. This was also observed in the Ashford Loom assembly during the calibration of the shafts. The hardware did not specify correctness of assembly, leaving participants unsure and seeking confirmation from the group. These instances of non-specific hardware created opportunities for coordination of specific tasks.

Coordination of specific tasks was also observed with unfamiliar hardware, causing mental difficulty. For example, P2 and P3 were unfamiliar with mounting RoboLoom's heddles to the motors, so they coordinated while they figured out how to achieve the task. Once they had a solution, they stopped coordinating. The Ashford Loom's hardware was familiar already (due to hardware choices like screw selection, etc.), so no instances of coordinating specific tasks occurred for this reason.

Similarly, when tasks were repetitive, participants would split the work and initially coordinate how to complete the task, then stop coordinating once they had completed the first few instances of the repetitive task (e.g. making corner assemblies or making shafts).

Optimizable assembly provided another opportunity for coordination. Group 1 chose to parallelize RoboLoom's assembly, allowing them to coordinate specific tasks and their scheduling. This occurred multiple times throughout the build as tasks were finished, as shown in Figure 6.2. Task scheduling was not observed in the Ashford Loom assembly, as the task order was more constrained. Another instance of optimization was non-specific hardware that allowed for multiple approaches. For example, P1 and P4 coordinated the beam attachment order in RoboLoom's assembly.

Participants also coordinated on specific tasks whose physicality necessitated the simultaneous manipulation of multiple objects. For example, when P1 and P4 were attaching the heddle guide to the RoboLoom, they coordinated the manipulation of multiple heddles into place while also holding the guide steady (Figure 6.11).

Finally, participants coordinated on specific tasks whose physicality involved manipulating the same object. For example, when P1 and P4 were assembling RoboLoom's height beams, they both needed to manipulate the base frame. This required coordinating the loom's position and managing the forces exerted on the frame.

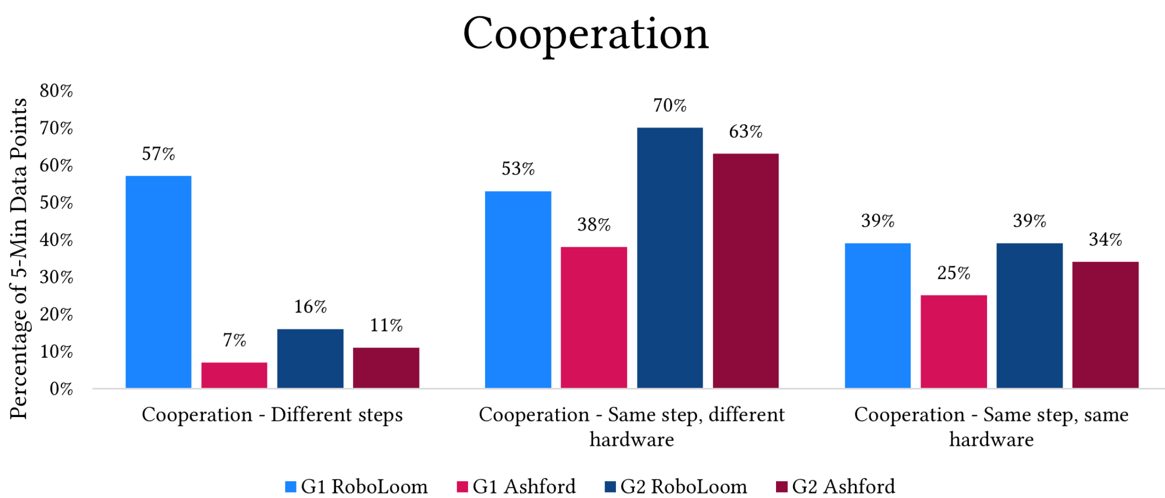


Figure 6.4: The percentage of 5-minute data points coded for the cooperation codes for each of the four test cases. G1 RoboLoom shows an increase in all cooperation codes over the Ashford loom, especially in different steps.

6.4.2 Cooperation

The results of the coding for cooperation codes are shown in Figure 6.4. We separate cooperation into three categories: different steps, same step, different hardware, and same step, same hardware. We chose these categories as a way to describe how closely participants were working together. When working on different steps, the need for cooperation may be lessened. However, during the same step, when working with different hardware, participants may cooperate to share their knowledge of the task. Finally, when physically interacting with the same hardware during the same step, more forms of cooperation may be needed.

6.4.2.1 Different Steps

Group 1 cooperated on different steps 50% more during their RoboLoom assembly (Figure 6.4) due to parallelization (DP4) of the frame and yarn manipulation subsystems. They split into pairs, causing the large continuous segments of the 'cooperation-different steps' code in Figure 6.2 without sacrificing cooperation on the same step. Group 2 discussed parallelizing their RoboLoom assembly only at the end of the build, resulting in less cooperation on different steps. Groups did not split assembly with the Ashford loom due to its serial nature.

6.4.2.2 Same Step, Different Hardware

Group 1 was observed spending more time (15%, Figure 6.4) cooperating on the same step, different hardware during RoboLoom's assembly. Participants cooperated on the same step, different hardware when tasks were repetitive (DP1). Both looms had repeated assembly tasks due to loom symmetry. For example, when adding the length beams on RoboLoom, participants split the work and added them simultaneously. Other repetitive tasks (e.g. building 16 shafts for the Ashford Loom, making 46 corner assemblies, or 40 motor assemblies for RoboLoom) were also split. RoboLoom had more repetitive tasks, which could have led to more observed instances of 'cooperation - same step, different hardware'.

6.4.2.3 Same Step, Same Hardware

Group 1 increased cooperation on the same task with the same hardware by 14% (Figure 6.4) when (1) the hardware's physicality required simultaneous manipulation (DP5) and (2) tasks were physically difficult (DP3). During both RoboLoom and Ashford Loom assemblies, tasks would require one participant to hold a part while another fastened it. RoboLoom had additional tasks where simultaneous manipulation of multiple parts was necessary, e.g. attaching the heddle guide or mounting the motor frames.

Additionally, we saw more cooperation on the same hardware during RoboLoom's assembly when parts were difficult to manipulate. For example, when participants were attaching the length beam, they often needed two people to guide the beam and align the corners simultaneously.

Some instances of this code were not attributed to aspects of the hardware, e.g. participants handing each other hardware (which happened for both looms).

6.4.3 Communication

The results for the communication codes are shown in Figure 6.5. We observed a difference in helping behaviors across loom hardware through knowledge co-creation and transfer. Though we coded for communication about troubleshooting and instruction confusion, we did not observe differences across the loom hardware in these behaviors.

6.4.3.1 Helping: Knowledge Co-creation

Both groups engaged in co-creating knowledge more during the RoboLoom assembly than the Ashford loom assembly. We observe that knowledge co-creation occurred when (1) hardware was non-specific (DP2), (2) hardware was mentally difficult (DP3), and (3) hardware was physically difficult (DP3). In the case of RoboLoom, there were more instances of these hardware features than the Ashford Loom, leading to more co-creation of knowledge.

Some hardware has a non-specific design, meaning it can be assembled multiple ways, only one of which is correct. Participants would engage in knowledge co-creation when hardware could be assembled wrong to discuss the function of the hardware, the correctness of their assembly, and mistakes made during the assembly. For example, participants communicated about the mounting of RoboLoom's warp beam, using their knowledge of loom function to determine the correct way to assemble the beam. Additionally, Group 2 built their base frame incorrectly and collaboratively created the solution to fix their error. Finally, Group 1 communicated using prior loom knowledge to co-create the knowledge of their final assembly's correctness. These opportunities for uncertainty due to a lack of hardware specificity lead to opportunities to co-create knowledge.

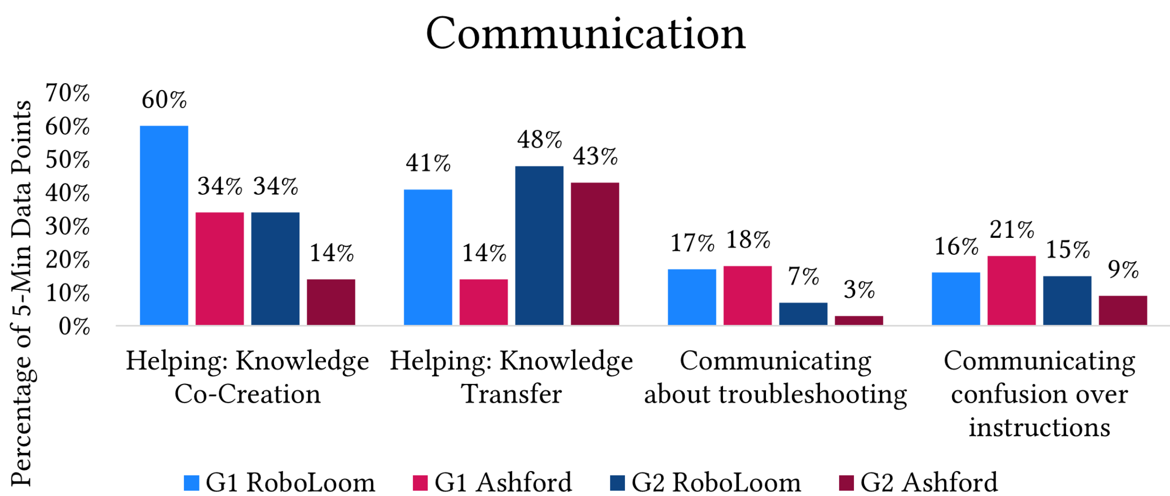


Figure 6.5: The percentage of 5-minute data points coded for the communication codes for each of the four test cases. The graph shows a difference in helping codes for G1's RoboLoom build.

Another instance of co-creating knowledge was during the mentally difficult task of using unfamiliar hardware. RoboLoom's hardware components were unfamiliar to the participants, leading to discussion on assembly and function. For example, fastening methods (e.g. corner assemblies, heddle-to-motor attachment, bobbin-to-holder attachment, and shaft collars) attaching length beams, and the ratchet and pawl of the warp beam were novel and led to knowledge co-creation. The Ashford loom also had unfamiliar tasks, though fewer than RoboLoom. Participants, while familiar with the use of Philips head screws, were unfamiliar with screwing into hardwood, prompting discussion on why the task differed from previous experience.

Physically difficult tasks caused participants to struggle, creating the opportunity for group problem-solving. These tasks included assembling RoboLoom's length beam, aligning and attaching RoboLoom's motor frames, aligning and attaching RoboLoom's heddle guide, mounting RoboLoom's tension rods, squaring both loom frames, and screwing into the Ashford Loom's hardwood. These struggles prompted the groups to discuss reasons and solutions for the struggles, co-creating the knowledge of how to help.

6.4.3.2 Knowledge Transfer

Group 1 showed an increase in knowledge transfer from one participant to another during their RoboLoom assembly by 27% (Figure 6.5). We observed some common themes when participants transferred knowledge, including sharing understanding from the directions and sharing experience with the hardware. When participants shared knowledge from the directions, one group member would read the instructions and disseminate the information. This was not observed to be influenced by the hardware. When participants transferred knowledge about hardware, it was when hardware was (1) repetitive (DP1), (2) mentally difficult (DP3), (3) physically difficult (DP3), (4) non-specific (DP2), and (5) physically requiring simultaneous manipulation (DP5).

In the case of repetitive tasks like making corner assemblies (RoboLoom) or shafts (Ashford Loom), participants would share tips and tricks with other participants to help with the assembly.

In the case of physically difficult hardware like the length beam assembly (RoboLoom) or tightening screws into hardwood (Ashford loom) the participant who had figured out how to do the task would share that information to reduce group effort. This would happen in cases where one participant was able to figure it out fully on their own. In the cases where this was not possible, participants would co-create the knowledge as discussed in the previous section.

When tasks were mentally difficult due to the unfamiliarity of one participant but not another, participants would transfer knowledge to ensure more group members could complete the assembly. This was seen for RoboLoom when using a ball-end Allen key to tighten screws. This was not observed during the assembly of the Ashford Loom.

When hardware features were non-specific regarding assembly methods, participants with the knowledge of correct assembly methods would transfer this information to other participants working on the same task. For example, when P1 figured out the arrangement of 8020 in the foot holder of RoboLoom, they shared this information with P4, who was working on this task with them. Similarly to difficulty, the instances of knowledge transfer happened when one participant created the knowledge on their own, but there were instances where knowledge was co-created (discussed previously).

Participants shared ideas with their group when hardware had to be simultaneously manipulated. For example, in both RoboLoom and Ashford Loom assemblies, one participant had their hands full with a portion of the task and knew how to complete the rest of the task, but was physically unable to do so. Participants then transferred the knowledge necessary to complete the task to another group member. When tasks (mounting motor frames or main frame assembly) required more than one set of hands, participants would communicate more.

6.4.4 Blocked Collaboration

Our results for the ‘blocked collaboration’ code are shown in Figure 6.6. Collaboration was blocked only once in both RoboLoom builds when a participant did not see the next available task. However, after discussion with the group, the participants found a task to do and continued collaborating.

Conversely, both groups had multiple instances where collaboration was attempted but not successful for the Ashford loom assembly. These instances happened for several reasons, including (1) a lack of tools, (2) hardware physicality designed for a single builder (DP5), (3) non-parallel hardware (DP4), (4) small physicality (DP5), and (5) too much mental difficulty (DP3). When participants lacked the tools to complete tasks, they would build without collaborating. This was not due to hardware design.

When the physicality of the Ashford loom was designed such that tasks did not allow for multiple builders, the participants would not collaborate on the assembly. For example, when a single bolt needed to be hammered in, P5 completed this task while P6 and P7 watched him. This task does not require multiple sets of hands or simultaneous manipulation of two or more objects and thus requires only one person to complete it.

These instances were often coupled with the inability to move to the next task without first

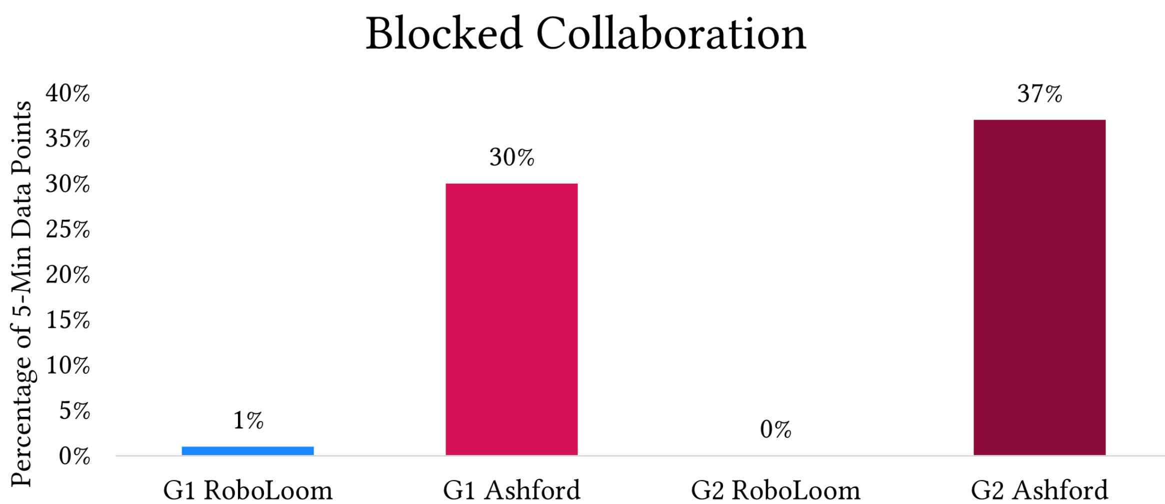


Figure 6.6: The percentage of 5-minute data points coded for the Blocked Collaboration code for each of the four test cases showing more blocked collaboration during the Ashford Loom building.

completing the current step of assembly (i.e., non-parallel hardware design). Without another task to complete, participants could not cooperate. For instance, we observed participants of Group 2 attempting to cooperate by going forward in the instructions while P6 attached the ratchet gears. However, upon P7 reading the next step, P5 and P7 saw they could not attach the handle to the ratchet gears until they had been attached to the loom. They then decided they could not cooperate and instead waited for P6 to finish her task.

Collaboration was also blocked by small physicality, creating a lack of space for cooperation. For instance, when inserting shafts into the castle of the Ashford Loom, P6 inserted the shafts while P7 held them in place. P5 stood by and watched as there was no room or need for his physical help.

Finally, we observed blocked collaboration when tasks were too mentally difficult, e.g., a participant did not contribute to problem-solving or troubleshooting due to a lack of building knowledge.

6.5 Discussion

We observed that hardware design features did provide opportunities for participants to engage in collaborative behaviors. These hardware features can be summarized into 5 categories: repetitiveness (DP1), specificity (DP2), difficulty (DP3), parallelizability (DP4), and physicality (DP5).

6.5.1 Repetitiveness

Repetitive hardware elements – e.g., Figure 6.7 – can create opportunities for coordination of people and specific tasks, cooperation on different hardware, and communication to co-create and transfer knowledge.

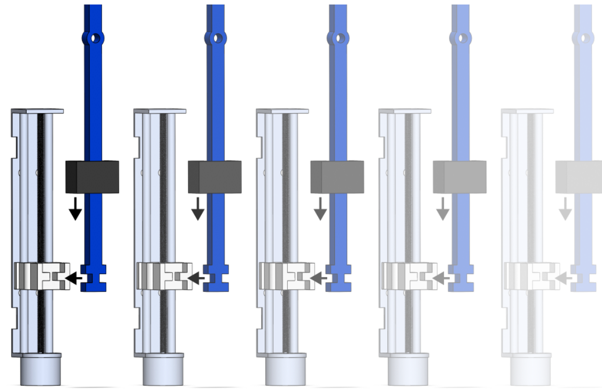


Figure 6.7: A visual of a task whose repetitive nature can provide the opportunity for a divide and conquer approach to assembly, allowing for coordination, cooperation, and communication. The image shows the repeated assembly of a motor for building RoboLoom.

When tasks are repetitive, participants can coordinate to divide the labor of a repetitive task and then coordinate during the specific task. Participants can then cooperate on their shared task to decrease individual effort and overall assembly time. Throughout the assembly, participants can communicate to co-create the knowledge of how to assemble their shared repetitive task. When a new participant joins the repetitive task, or one participant discovers something about the task, there is opportunity for knowledge transfer as the participants share tips.

Longer repetitive tasks increased the number of cooperation-coded data points, as participants spent longer on the task. However, we did not observe more coordination or knowledge transfer during these instances. Further study is needed to specifically determine the effect that the length of a repetitive task has on cooperation, coordination, and knowledge co-creation and transfer, specifically in the existence of a saturation point for coordination and knowledge transfer and co-creation.

Designing educational kits to have repetitive tasks can provide the opportunity for participants to coordinate, cooperate, and transfer and co-create knowledge. Splitting repetitive tasks allows for physical cooperation while still encouraging communication about the shared goal. When designing for collaborative assembly, the length and number of repetitive tasks should be considered, as longer, more numerous tasks can provide more opportunities for cooperation; however, they may not provide more opportunities to coordinate, transfer knowledge, or co-create knowledge.

6.5.2 Specificity

The specificity of hardware – e.g., Figure 6.8 – can influence the opportunity to coordinate people and specific tasks and communicate to co-create and transfer knowledge. We define specificity as how pre-determined the hardware is for its intended purpose, similar to the concept of poka-yoke [76], or “error proofing.” Specificity can refer to the directionality of a piece (i.e., can it be assembled backward), the flexibility of a piece (i.e., how many uses it has), or where it can be mounted. We observed that the amount of specificity of the hardware components affected how participants collaborated through their knowledge co-creation, knowledge transfer, and coordination of people and tasks.

Non-specific hardware features provide opportunities for participants to discover the knowledge of how the part should be assembled (e.g., does the direction matter?). The participants can co-create this knowledge using the hardware's intended function (e.g., this part should only rotate clockwise) or through trial and error in assembly. They can also transfer this knowledge to others in their group as they build. Non-specificity of hardware can cause participants to have questions as they assemble, leading to coordination of specific tasks and potentially asking group members to check their work (and coordinating people to do so).

Though in commercial applications hardware is usually designed to be as error-proof and as easily assemblable as possible [84], as is the case with many hardware features of the Ashford Loom, this does not provide the opportunity for participants to collaborate. Poka-yoke hardware features are designed to be easily assembled by one person, making collaboration unnecessary. On the other hand, if hardware is too non-specific with little way to determine correctness, participants may turn to facilitators for help, thus preventing collaboration.

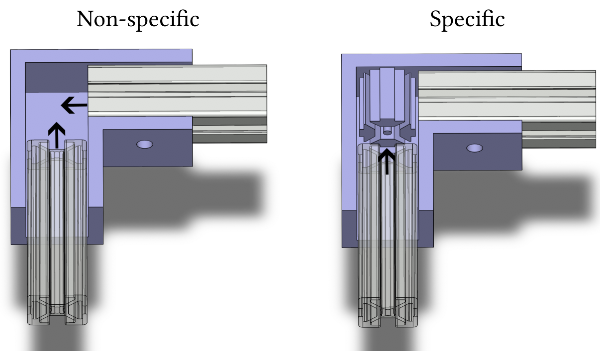


Figure 6.8: An example of specificity showing a design feature that can be added to a corner bracket, forcing the way aluminum extrusion mounts onto it. The specificity directs the assembly, but the non-specificity can allow opportunity for coordination and communication.

When designing for collaboration, non-specific hardware features can be used to give opportunities for more communication in the form of knowledge co-creation and knowledge transfer. However, this provides more opportunities for assembly errors. While this in itself can be used as a learning opportunity, it must be carefully balanced to prevent frustration from discouraging participants. When designing an assemblable educational toolkit, specificity can be used to increase opportunity for communication (knowledge co-creation or transfer) and coordination (people and tasks), but must be carefully considered with time and frustration in mind.

6.5.3 Difficulty

Difficult hardware – e.g., Figure 6.9 – can create opportunities for coordination, cooperation, and communication, but when hardware is too difficult, it can inhibit collaboration. The difficulty of the hardware refers to how difficult it is for participants to work with the hardware. This refers to mental difficulty in how familiar a participant is with the hardware or physical difficulty in how much effort or dexterity the task takes. Mentally difficult tasks affected the coordination of specific tasks and knowledge co-creation and transfer. Physically difficult tasks affected the coordination of people, the cooperation on the same hardware, and knowledge co-creation and transfer.

Physically difficult tasks like attaching the length beam of RoboLoom gave participants an opportunity to coordinate people and cooperate on the task. Participants found the length beam of RoboLoom difficult to maneuver, align with the corner fasteners, and fasten down. This allowed for the opportunity to coordinate multiple people to work on the task, which then led to cooperation on the task. Conversely, when some tasks were too physically difficult for participants, such as aligning height beams, they would take the opportunity to coordinate their team and assign the most physically skilled at this portion of the assembly to the task, while finding other roles for the other team members.

When tasks were mentally difficult, we observed the coordination of specific tasks and knowledge co-creation and transfer. For example, participants did not coordinate when they were working with Philips head screws. Commercial designs for assemblable devices privilege ease of assembly to capture a wider audience of abilities. This leads to the use of familiar fastening devices. However, when using unfamiliar fastening methods, participants would coordinate about the task, as they did not already have a mental model of the necessary steps. This increase in difficulty, while non-optimal in a commercial design, provided the opportunity for collaboration. However, when tasks were too difficult and no participants could find a solution, they would stop collaborating and ask a facilitator for help.

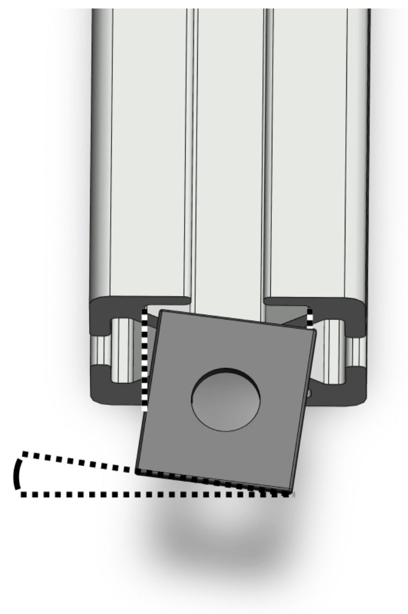


Figure 6.9: An example of difficulty showing the small tolerance for inserting a t-nut into aluminum extrusion, requiring dexterous manipulation to successfully assemble. Difficult hardware can create opportunity for coordination, cooperation, and communication, but can also prevent collaboration if the task is too difficult.

6.5.4 Parallelizability

When designing assemblable hardware, features can provide the opportunity for coordination of people and specific tasks, cooperation, and co-creation and transfer of knowledge. Hardware features that are physically and mentally difficult provide these opportunities. However, there are other effects of difficulty that should be considered when designing such hardware. For example, the possible frustration participants could face may discourage them from completing the task, or a too difficult task could result in participants needing help rather than being able to collaborate to find the answer. When designing purposeful difficulty, the level of difficulty must be balanced so that participants are not discouraged and can collaboratively solve the problem.

When hardware elements were parallelizable – e.g., Figure 6.10 – cooperation on different steps and cooperation on the same step were possible. There was also the opportunity for coordination of people, materials, and specific tasks.

When hardware was parallelizable, groups of four would split into pairs to optimize the assembly for time. This provided the opportunity to not only cooperate on different steps, but also to cooperate on the same step within pairs (when allowable by the hardware, see physicality). This split also allowed coordinating people and specific tasks. In groups of three, parallelizable hardware led to groups splitting into one pair and one individual. The individual would then be cooperating on a different step, but not directly communicating with group members.

When parallel tasks had shared steps or materials, it created the opportunity for the pairs to come back together and coordinate the shared materials, and also to transfer knowledge they had gained during their task building.

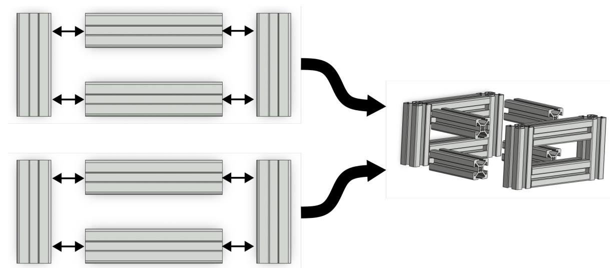


Figure 6.10: An example of a box with the ability to be built in parallel, one side at a time, then the sides are assembled together. This parallelizability allows the opportunity for coordination and cooperation.

When designing educational kits, parallelization can be used to increase cooperation, but planned dependencies can create opportunity for groups to come back together to coordinate and communicate. While fully parallelizable tasks increase cooperation and speed of assembly, they could limit communication between group members if tasks do not overlap at all. In an educational setting, this could lead to inaccurate beliefs about the contribution of others and less co-creating knowledge. This may have adverse effects on group dynamics, attitudes, and collaborative educational experiences. Further study is needed into the effect of parallelizability on cooperation and communication to determine the trade-offs of speed and perceived collaboration.

6.5.5 Physicality

The physicality of hardware – e.g., Figure 6.11 – can provide the opportunity for coordination, cooperation, and communication to transfer knowledge. The physicality of a device is a feature that we use to refer to the size of the hardware, how many non-fixed parts the hardware has while it's being assembled, and how many components interact with each other during an assembly task.

Figure 6.11: An example of physicality, where the alignment of heddles with the guide requires manipulating multiple objects simultaneously to keep them aligned as the guide is put over the top. This can provide opportunities for coordination, cooperation, and communication.

he size of hardware can affect cooperation on the same step, same hardware. When the hardware is too small for multiple people to work, it can block attempts at cooperation, potentially leading to adverse effects on collaborative attitudes.

When assembly tasks require simultaneous manipulation or multiple sets of hands to complete, this creates the opportunity to coordinate the specific task and cooperate on the same step, same hardware. Multiple moving parts add complexity and are thus usually avoided in designing commercial assemblable devices. However, this added complexity provides the opportunity for participants to manage what was going on to ensure smooth assembly. When more parts need to be manipulated than one participant can manage alone, participants can cooperate on these steps to fill the need for help. Designing hardware features to require multiple simultaneous manipulations can encourage assemblers to coordinate and cooperate.

Physicality, where different pieces attach to the same piece of hardware, provided the opportunity for participants to coordinate specific tasks as they worked on the same piece, communicating their various needs for orientations, movements, or forces exerted at different times. During this process, the opportunity to communicate to transfer knowledge about tasks and assembly also arises. This additional complexity is usually avoided in commercial assemblies as it adds effort to the assembly process. However, this effort increases the opportunity for coordinating tasks and transferring knowledge.

When designing assemblable kits, the physicality of the hardware should be considered, as it can provide the opportunity for coordination, cooperation, and communication to transfer knowledge. Increasing simultaneous manipulations and requiring manipulation of the same piece for different tasks adds complexity to the assembly, shifting it away from commercial design principles, which optimize for ease of assembly. These features can provide the opportunity for collaboration, but could also increase frustration, potentially to the point of detriment to collaboration. Further study is needed to understand the trade-off in designing for physicality and the impacts on collaboration and frustration.

6.5.6 Summary

We have discussed five design principles for nuanced interactions during collaboration, which should be carefully considered for the context of the assembly. These principles (repetitiveness, specificity, difficulty, parallelizability, physicality) have been studied in individual building contexts for error reduction, ease of building, or speed of building, but have not been studied in a collaborative context. Entirely parallel building may be faster and easier, but if students do not interact with each other, they may have negative learning outcomes [**?**]. Depending on the end goal of the assembly (speed, educational aspect, cooperative building, system integration learning), designers may want to change how much parallelization is used in design. In a collaborative educational experience, if the designs are too repetitive, they might be less difficult and thus decrease the communication among participants. We argue that each of these design principles is a continuous axis to consider when designing hardware. These principles can influence collaborative behaviors in the initial ways we present here, but may also have more nuanced interactions with each other that require further study.

There are potential effects of these hardware design principles on frustration (difficulty, repetitiveness, and physicality) and how this could influence collaboration and participation. We saw that these design principles, when made more challenging, lead to increased collaboration. However, with the increase of challenge comes the risk of frustration, which could adversely affect a student's collaborative experience. For example, designers should not make tasks so repetitive that they frustrate users to the point of giving up; tasks should be difficult enough that users ask each other questions and collaborate, but not so difficult that they cannot work. Future work should examine the frustration-challenge balance with these design principles.

6.6 Conclusion

In this Chapter, we presented a crossover study involving two groups of participants building two looms: RoboLoom and the Ashford Loom. These devices were specifically chosen to investigate the hardware design features that influence collaborative actions during the assembly process. Rather than focusing on task design, we utilized the 3Cs framework to observe and categorize collaborative behaviors, examining the engaged hardware to determine its potential impact on group collaboration in a collaborative engineering setting.

Our observations and analysis revealed that different hardware designs facilitated distinct collaborative actions in the loom-building process. We identified five key categories of hardware features: repetitiveness, specificity, difficulty, parallelizability, and physicality.

DP1. **Repetition** of hardware elements provided opportunities for coordination of people and tasks, cooperation on the same step but different hardware, and communication to transfer knowledge.

DP2. **Specificity**, or features to indicate or error-proof assembly, influenced the opportunity for communication to co-create or transfer knowledge. Hardware that was highly "error-proof" reduced the need for assistance, whereas open-ended hardware often required external help from instructors, limiting peer-to-peer assistance. Mid-range specificity fostered more collaborative actions through co-creation or transfer of knowledge.

DP3. **Difficulty**, both mental and physical, mirrored the effects of specificity. An optimal level of difficulty encouraged communication to co-create or transfer knowledge and the coordination of people, materials, and tasks. However, tasks that were too easy or too challenging reduced such interactions.

DP4. **Parallelizability** in hardware created the opportunities for cooperative behaviors.

DP5. **Physicality**, both in size and the requirement of simultaneous manipulation, created opportunities for coordination of people and tasks, cooperation on the same hardware pieces, and communication to transfer knowledge. Smaller hardware was observed to prevent physical collaborative actions.

Collaborative work is beneficial for engineering education [36, 59, 68, 106]. One opportunity to work collaboratively in engineering education is through the assembly of hardware kits. We posit that by viewing the assembly of the kit as its own sort of tangible user interface, the above-presented design features, when used as a framework to design hardware for assembly, can facilitate a collaborative user experience. By designing hardware with these features in mind, educators can create opportunities for coordination, cooperation, and communication. Specifically:

C1. Coordination can be facilitated by hardware that is parallelizable, physically and mentally challenging, requires simultaneous manipulation, semi-specific, and repetitive.

C2. Cooperation can be enhanced by hardware that is parallelizable, is physically challenging, requires simultaneous manipulation, or is repetitive.

C3. Communication can be encouraged by hardware that is physically and mentally challenging, requires simultaneous manipulation, is semi-specific, or is repetitive.

C4. Conversely, hardware that is non-parallel, is overly mentally challenging, is small, or does not require simultaneous manipulation of more than two objects can block collaboration when it is attempted and possibly discourage future attempts to collaborate.

6.6.1 Future Work

In this Chapter, we presented our initial findings on these hardware feature categories as a foundation for design recommendations and further research on facilitating collaboration through interactions with hardware kits. As our study was limited in number and diversity of participants (most coming from Informatics), future work should include further study of these design features and interactions with them from different populations. Future work should also include a causal study of these design features, designing for and against them to assess their impact on collaboration. Finally, this Chapter presents an initial point for exploring design features for collaboration with many open avenues for further exploration into different collaborative contexts, further study into the interplay of these features and their effects on collaboration, and studies on challenge-frustration trade-offs.

In the following chapters of this thesis, I explore the causality of these design features on collaborative behaviors, both in a controlled lab study and also in a real-world setting. Participants are recruited generally across Carnegie Mellon at the undergraduate and graduate level and across majors.

Chapter 7

Collaborative Assembly Study

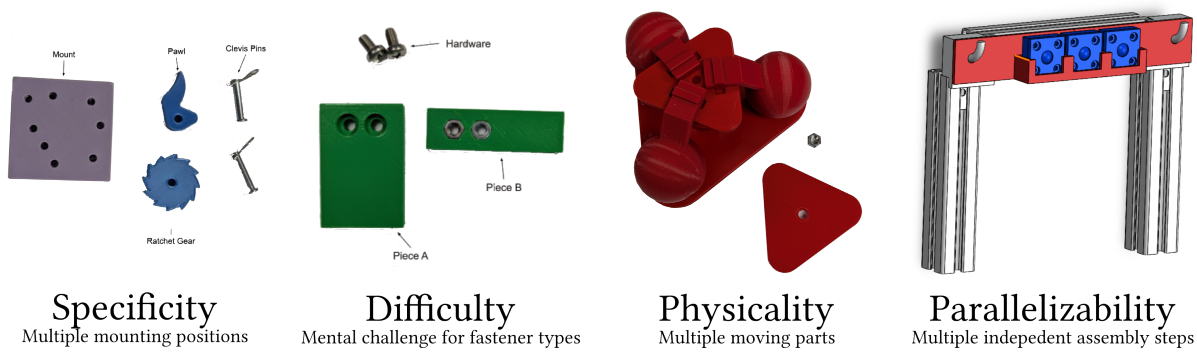


Figure 7.1: Four different types of devices to elucidate four design principles to study their effect on collaborative assembly.

7.1 Introduction

Collaboration is a critical skill for engineers, scientists, and designers as they prepare for work in both industry and academia. Prior research in computer-supported collaborative work (CSCW) and computer-supported collaborative learning (CSCL) has shown that software can be intentionally designed to support collaboration. However, much less is known about how hardware—particularly educational and robotic systems—can be designed to afford collaborative action. This gap is especially important in classroom contexts, where poorly designed tools can either limit or undermine opportunities for students to engage in meaningful group work.

Building on the findings from the earlier studies presented in this thesis, I now turn to testing causation more directly. In previous work, I identified design features of RoboLoom and other devices that appeared to influence students' collaborative behaviors during assembly. These features included specificity, parallelizability, physicality, and difficulty. In this study, I sought to move beyond observational findings toward experimental testing of these design principles.

To do so, I designed twelve devices that varied systematically along these four dimensions—three variations for each design principle. Participant dyads assembled these devices, and their behaviors were analyzed using the coding scheme developed in Section 6.3.3 and Appendix A. This approach isolates each design principle, allowing for the comparison of the effect of specific hardware design choices on coordination, cooperation, and communication. By examining how differences in design led to differences in collaborative practice, I aim to provide stronger evidence for the causal role of hardware design in shaping human-human collaboration.

7.2 Related Work

7.2.1 Collaboration

As in Section 2.6, we define collaboration as a process of two or more people collectively working towards a shared goal where the output of the group cannot be easily separated into individual contributions [21, 24, 65, 96]. We use the 3Cs model: Communication, Coordination, and Cooperation [25, 33] to break down collaborative behaviors more specifically. Coordination is often defined as the process of organizing the people, activities, and resources necessary to accomplish a shared goal, ensuring shared understanding about the state of each [22, 25, 33, 65, 96]. Conversely, cooperation is the process of individuals working on achieving specific tasks that contribute to a shared goal [8, 33, 65, 96]. Baker distinguishes between cooperation and collaboration in that ”cooperation works on the level of tasks and actions, collaboration works on the plane of ideas, understanding, representations” [8]. Communicating, we break down and focus on the ideas of communication used by participants to help solve problems encountered in trying to complete the collaborative task. We break this further into helping with knowledge co-creation, the exchange of ideas back and forth between two or more people, and knowledge transfer, the exchange of knowledge from one person to another without reciprocal knowledge exchange.

7.2.2 Designing Hardware for Collaboration

Based on our definition of collaboration, technology that supports collaboration must be designed to support communication, coordination, and cooperation [65]. Designs must therefore support shared meaning across different contexts, of organization and shared understanding of state space between actors, and of joint operations in the workspace [65]. Designing effective collaborative problems requires that tasks make working together necessary for the achievement of the task and that goals cannot be achieved by the individual [8]. The design principles put forth earlier in this thesis (Section 6.5) each theoretically contribute to creating a task that requires multiple actors to accomplish the goal. They also provide opportunities in which creating a shared context and having a shared organization become beneficial for the completion of the task as well. However, the prior work establishing these five design features does not show causation, as it was purely observational.

This study builds directly on these insights by systematically varying hardware design features in controlled conditions. Whereas earlier work identified potential principles through observational and comparative methods, the current study isolates and tests these principles experimentally, aiming to establish causal links between design features and collaborative outcomes.

7.3 Methods

To determine whether hardware design influences various collaborative behaviors, we recruited 12 participants to build six different devices (one each from the sets of devices for specificity, parallelizability, and physicality, and all three from the set of devices for difficulty). The assembly of the specificity, parallelizability, and physicality devices was found to have an ordering effect on how the assembly was approached, so participants were asked to only build one object from each of these groups. No ordering effect was found for difficulty devices due to the inherently different nature of the designs (see Section 7.3.2.4).

Participants worked in pairs and were told to assemble all the devices they were given collaboratively. The researcher running the study answered no questions. Participants determined when they were done building a device and alerted the researcher. Participants were not given a time limit on building; however, all groups finished the study within an hour. Participants were audio- and video-recorded and gave full written consent before participating. The study was approved by Carnegie Mellon University's IRB (Protocol STUDY2024_00000197).

7.3.1 Participants

Participants were recruited via email, flier, and various messaging apps such as Slack. All participants were graduate or undergraduate students at Carnegie Mellon University. Participants were between the ages of 20 and 36 (mean = 26, standard deviation = 4). Half the participants were women (n=6), and half were men. Eight participants were robotics majors, two were from mechanical engineering, one was from material science and engineering, and one was from human-computer interaction. Participants ranged from undergraduates to graduate students. Two pairs were familiar with each other before the study, one pair was moderately familiar with each other, and three pairs were not familiar with the other person at all.

7.3.2 Devices

We developed versions of the same device for each of the four design principles: specificity, parallelization, physicality, and difficulty. When designing the devices, we controlled for other design factors, changing only one axis of the hardware design and creating a least, mid, and most version of each device for each design principle. Each device came with a written set of instructions that showed images and had written instructions for assembling the device. Instructions can be found here: `https://drive.google.com/file/d/1BBtV3BJW_0NXXab_DQ1-nTKZOniXIPwg/view?usp=sharing`.

Participant	Age	Gender	Major	Program Year	Familiarity with Partner
P5	25	Female	Robotics	PhD 4th Year	Very Familiar
P6	26	Male	Robotics	PhD 4th Year	Very Familiar
P7	23	Male	Robotics	Masters 2nd Year	Familiar
P8	36	Female	Mechanical Engineering	PhD 6th Year	Moderately Familiar
P9	20	Female	Mechanical Engineering	Undergraduate 3rd Year	Not Familiar
P10	26	Male	Robotics	PhD 5th Year	Not Familiar
P11	27	Male	Robotics	PhD 5th Year	Not Familiar
P12	23	Female	Robotics	PhD 1st Year	Not Familiar
P13	27	Female	Robotics	PhD 5th Year	Not Familiar
P14	21	Female	Materials Science and Engineering	Undergraduate 4th Year	Not Familiar
P15	27	Male	Robotics	PhD 4th Year	Very Familiar
P16	27	Male	Human-Computer Interaction	PhD 5th Year	Very Familiar

Table 7.1: Participants in the Collaborative Assembly Study. Participants were paired consecutively as listed in the table (P5&P6, P7&P8, etc.).

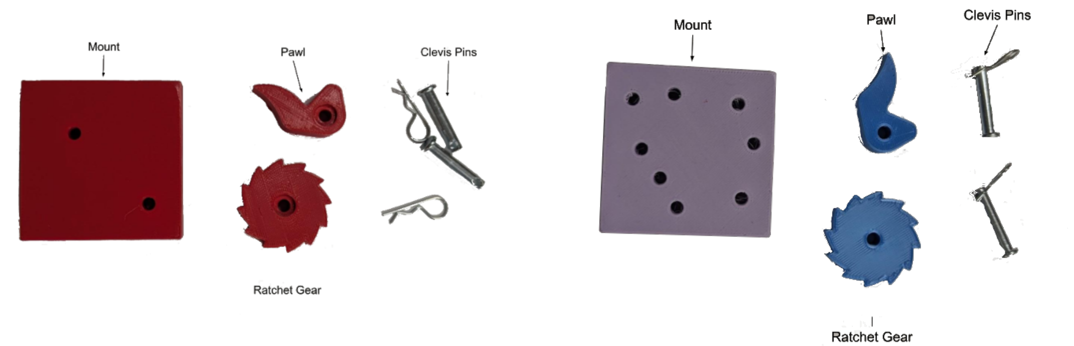
7.3.2.1 Specificity

The specificity devices are ratchet and pawl mechanisms 7.2, with the assembly task being to attach the ratchet and pawl to the mount, such that the mechanism rotates in one direction and not the other. Instructions were presented with a labeled parts list (Figure 7.2) with the instruction "Attach the ratchet gear and pawl to the mount using the clevis pins. Attach them so that you have a working ratchet and pawl mechanism."

The most specific ratchet and pawl, Figure 7.2a, has only two holes for mounting, giving relatively few combinations for mounting the ratchet and the pawl. The mount also has a guide printed into the mount to show where the ratchet and pawl go. These features all specify the assembly of the device, thus leading it to be the most specific.

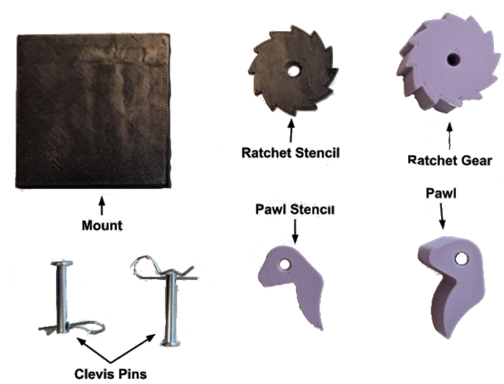
The mid-specific ratchet and pawl, Figure 7.2b, has eight possible holes to mount the ratchet and pawl in, only two of which provide a correct configuration for a working ratchet and pawl mechanism. This feature decreases the specificity of the assembly, with no indication of which option is correct in the hardware.

The least specific ratchet and pawl, Figure 7.2c, has no holes. In this assembly, the participants must determine where to put holes in the mount to mount the pawl and ratchet gear. This feature further decreases specificity, with no hardware feature determining if the assembly is



(a) The most specific device with a mount with two holes and features displaying the placement of the ratchet gear and pawl.

(b) The mid specific device with a blank mount with eight holes.



(c) The least specific device with no holes, requiring holes to be added.

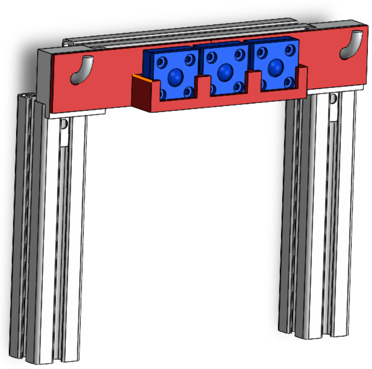
Figure 7.2: A ratchet and pawl device designed with different levels of specificity.

correct.

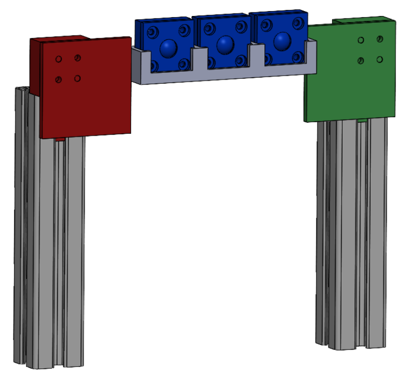
7.3.2.2 Parallelization

The parallelization devices (Figure 7.3) consist of three elements: a frame, widgets, and a widget holder. The frame is made from aluminum extrusion and 3D printed parts. The widgets are designed to simulate a possible mechanism assembly, and require putting spheres in holders such that they freely rotate. The widget holder is designed to mount the widgets onto the frame. The frame is designed to have the object stand up.

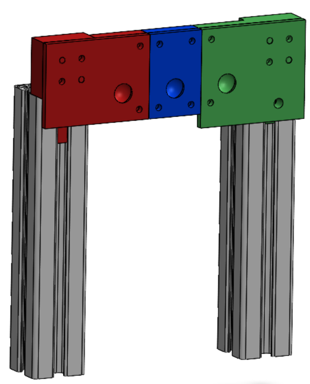
Instructions for the completion of the object contained a parts list, an instruction page with an overview of each object listing the three different parts, and three separate pages detailing the completion of each part. Parts of the assembly were broken down by color as shown in Figure 7.3. For all three versions of the device, participants were asked to assemble the device "optimally and quickly" to encourage them to attempt parallelization, as previous research shows that even when able, participants don't always parallelize assembly [102, 103].



(a) The most parallel device with a frame (gray), widget mount (red), and widgets (blue) all separate.



(b) The mid parallel device with a combined frame and widget mount (red, green, and gray) and separate widgets (blue).



(c) The least parallel device with a combined frame, widget mounts, and widgets.

Figure 7.3: The parallelization devices.

The most parallel device (Figure 7.3a) consists of three separate elements: a frame, a widget mount, and widgets. Each of these elements is designed to be assembled separately. The widgets can be built irrespective of the other elements and simply sandwich a sphere into an open face 3D printed part that allows the sphere to rotate. The widget mount is a series of buckets that the widgets get slotted into. The mount requires assembly of the three pieces together. The mount is designed to hook onto the frame with relative ease. The frame can be built separately from the other two elements and consists of aluminum extrusion and 3D printed corners with hooks for the widget mount to fit onto. This device is most parallel as all the elements are separate from each other and can be built simultaneously.

The mid parallel device (Figure 7.3b) combines the frame and the widget mount. The widgets can still be built irrespective of the other elements. However, the frame is now built with the widget mount. The corners of the frame (shown in green and red in Figure 7.3b) sandwich the buckets that hold the widgets (shown in gray), adding a dependency on the order of building the frame and widget mounts. This restricts the parallelization of the device slightly, but not fully, as the widgets remain separate.

The least parallel device (Figure 7.3c) combines the frame, widget mounts, and widgets. The frame is built with the widgets inside it, so all the elements must be built simultaneously. Furthermore, the left corner must be assembled before the middle section (which sandwiches over the left corner), and the middle section must be built before the right corner (which sandwiches over the middle section). This restricts the parallelization of the device fully, forcing which elements are constructed in which order.

7.3.2.3 Physicality

The physicality devices consist of a number of spherical widgets that are designed to be mounted between two mounts, held together by a single screw and nut. Participants were instructed to "attach the widgets to the mounts by sandwiching them in place between the mounts and securing the mounts with the screw and nut."

The most physical device has six spherical widgets to be mounted between two mounts with a nut and screw. The widgets are not fully fixed in place until all six are in the slots of the mount, and the mounts are closed down with the nut and screw, which leads to eight moving parts to be coordinated during assembly.

The mid physical device has three spherical widgets to be mounted between two mounts with a nut and screw. Again, the widgets are not fully fixed until the mounts are screwed together. This led to five moving parts to be coordinated during assembly.

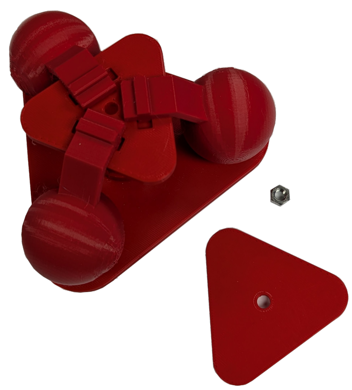
The least physical device also has three widgets to be mounted, but also comes with a stand to be used during assembly to rest the widgets and mounts on. This allows for the possibility of only one moving part at a time during assembly.

7.3.2.4 Difficulty

The difficulty devices (Figure 7.5) are designed to fasten two mount pieces together using the provided hardware. The hardware ranged in familiarity to the participants and the number of mechanisms used. The participants were given a labeled image of the parts labeling piece A,

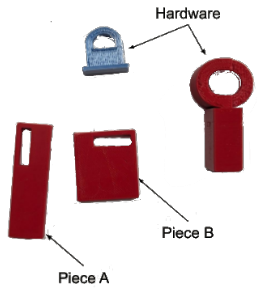
(a) The most physical device with six widgets to mount.

(b) The mid physical device with three widgets to mount.

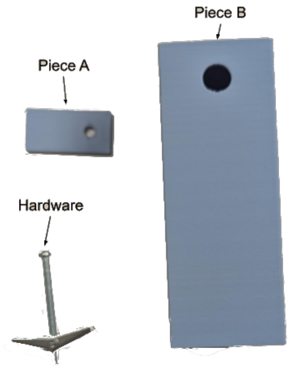


(c) The least physical device with three widgets to mount and a stand to hold the device as it was assembled.

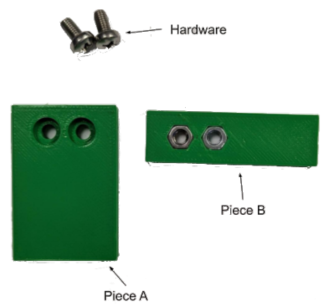
Figure 7.4: The physicality devices.



(a) The most difficult device, which uses a clevis pin (blue hardware) and spring-loaded shackle (red hardware).



(b) The mid difficult device which uses a toggle bolt.



(c) The least difficult device which uses a screw and nut.

Figure 7.5: The difficulty devices.

piece B, and the hardware. They were also given the instruction: “Attach the two mount pieces using the provided hardware.”

The most difficult device uses two uncommon mechanisms: a clevis pin and a spring-loaded shackle. The shackle mechanism is often only used in sailing and requires the users to slide a spring-loaded collar down to release the shackle, then hook the shackle into place and slide the collar down again to close and lock the shackle. This being the most unfamiliar and mechanically complicated makes it the most difficult device.

The mid difficult device requires the participants to use two mechanisms: a screw and nut, and a toggle bolt. During assembly, the toggle bolt must be used as a screw and nut to attach one piece, then used as a toggle bolt to attach the second piece. The toggle bolt is often used in mounting to drywall and thus could be more familiar to participants. This familiarity leads this to be the mid difficult object.

The least difficult object has only one very common mechanism: a screw and a nut. Participants are most likely all familiar with this mechanism, making it the least difficult.

7.3.3 Data Sources and Analysis

We collected audio and video recordings of the sessions and constructed a data log for each object that the participant group constructed. Each data log was a descriptive transcription of each video of the device assembly, where actions and speech were captured in 5 or 10-second segments. Table 7.2 shows the length of each group's build time for each device as well as the number of 5 or 10-second segments that comprise the data log for that device build. Figure 7.6 gives a graphical summary of the coded data. The time off task was not counted in the total time; for example, the least specific device required holes to be drilled. In our analysis, we use the coding scheme previously described in Section 6.3.3 to evaluate the collaborative actions based on the 3Cs framework.

Participant Group	Specificity Device (5s seg)			Parallelizability Device (10s seg)			Physicality Device (5s seg)			Difficulty Device (5s seg)		
	Most	Mid	Least	Most	Mid	Least	Most	Mid	Least	Most	Mid	Least
P5P6		2m48s (34 seg)			8m00s (48 seg)				1m55s (23 seg)	1m18s (16 seg)	2m07s (26 seg)	1m05s (13 seg)
P7P8	1m34s (19 seg)			7m35s (46 seg)			2m40s (32 seg)			2m00s (24 seg)	2m35s (31 seg)	0m45s (9 seg)
P9P10			2m45s (33 seg)			9m06s (55 seg)		0m53s (11 seg)		4m05s (49 seg)	1m27s (18 seg)	0m40s (8 seg)
P11P12			4m36s (56 seg)	6m22s (39 seg)					3m23s (41 seg)	2m27s (30 seg)	1m28s (18 seg)	1m15s (15 seg)
P13P14		2m15s (27 seg)			7m13s (44 seg)			1m18s (16 seg)		0m38s (8 seg)	2m21s (29 seg)	0m45s (9 seg)
P15P16	2m10s (26 seg)					13m40s (82 seg)	2m39s (32 seg)			5m13s (63 seg)	2m51s (35 seg)	1m29s (18 seg)

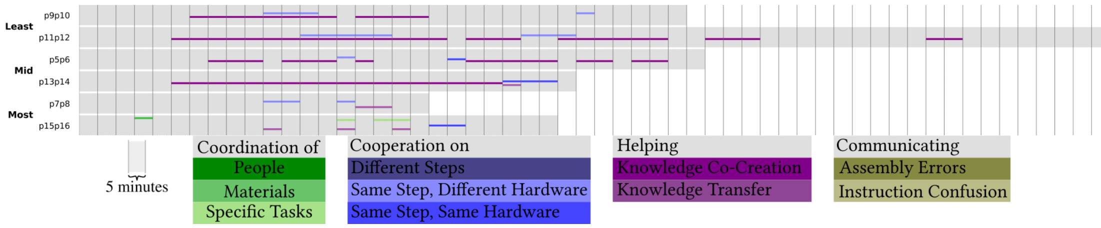
Table 7.2: Time spent by each participant group with devices. Each entry shows minutes:seconds on the first line and the corresponding number of segments on the second line.

7.4 Findings

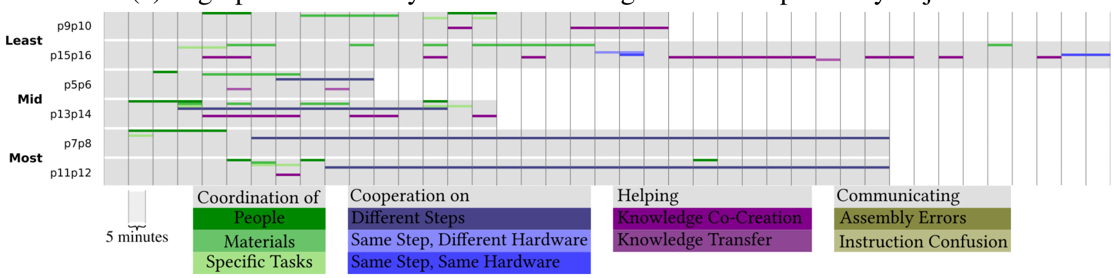
We qualitatively analyzed the observation data to determine the percentage of time groups spent collaborating for each object. We summarize our findings in the sections below, which report the percentage of data points (5 or 10-second segments) that received each code. Percentages were calculated by counting the number of data points marked with that code, and then dividing by the total number of data points (found in Table 7.2).

7.4.1 Specificity

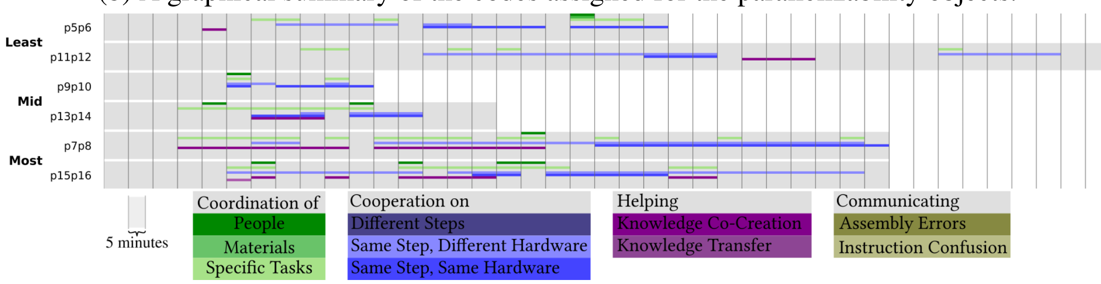
Based on prior research [103] (Chapter 6), we hypothesized that the specificity of the device would affect how participants coordinated people, coordinated specific tasks, helped with knowledge co-creation, and helped with knowledge transfer. The percentages of coded data are shown in Table 7.3. The least specific device required the facilitator to drill holes during the assembly.



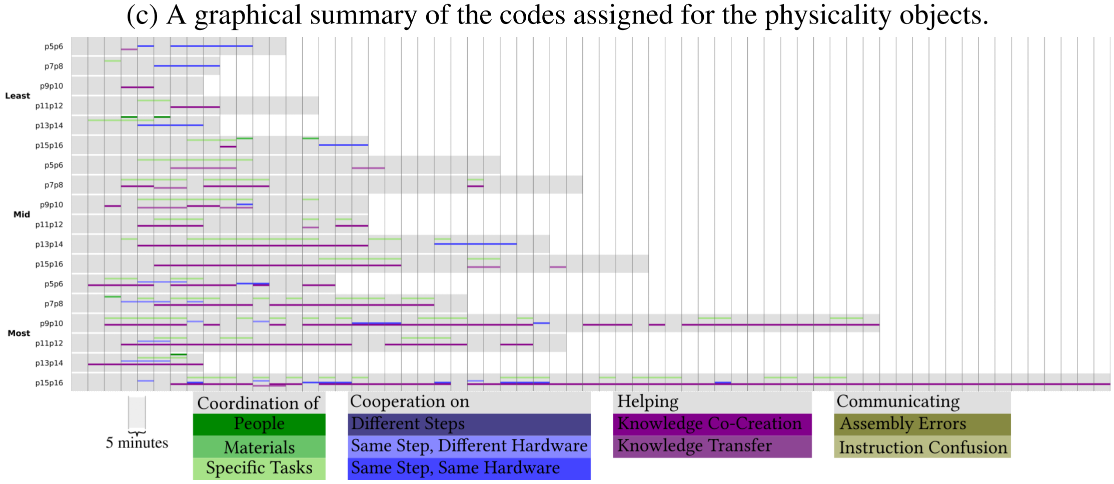
(a) A graphical summary of the codes assigned for the specificity objects.



(b) A graphical summary of the codes assigned for the parallelizability objects.



(c) A graphical summary of the codes assigned for the physicality objects.



(d) A graphical summary of the codes assigned for the difficulty objects.

Figure 7.6: The codes assigned to different device builds for each 5-second or 10-second (parallelizability) block.

To ensure that this did not skew the percentages, drilling time was removed from the total assembly time used to calculate percentages. No work on the assembly occurred while the holes were being drilled.

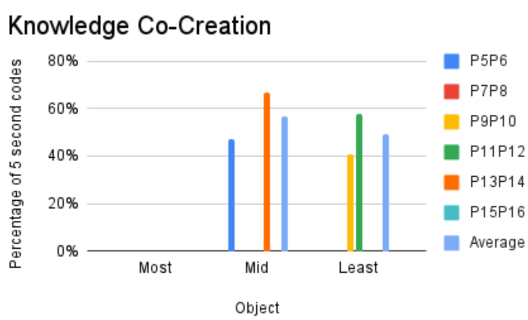
Code	Most		Mid		Least		Average		
	P7P8	P15P16	P5P6	P13P14	P9P10	P11P12	Most	Mid	Least
Coordination of people	0%	0%	0%	0%	0%	0%	0%	0%	0%
Coordination of materials	0%	4%	0%	0%	0%	0%	2%	0%	0%
Coordination of specific tasks	0%	12%	0%	0%	0%	0%	6%	0%	0%
Cooperation – Different steps	0%	0%	0%	0%	0%	0%	0%	0%	0%
Cooperation – Same step, diff. hardware	16%	0%	3%	0%	14%	16%	8%	2%	15%
Cooperation – Same step, same hardware	0%	8%	3%	11%	0%	0%	4%	7%	0%
Helping: Knowledge Co-Creation	0%	0%	47%	67%	41%	58%	0%	57%	50%
Helping: Knowledge Transfer	11%	12%	0%	4%	0%	0%	12%	2%	0%
Troubleshooting communication	0%	0%	0%	0%	0%	0%	0%	0%	0%
Communicating confusion over instructions	0%	4%	0%	0%	14%	0%	2%	0%	7%
Collaboration not possible	0%	0%	0%	0%	0%	0%	0%	0%	0%
Cooperation (overall)	16%	8%	6%	11%	14%	16%	12%	9%	15%
Coordination (overall)	0%	15%	0%	0%	0%	0%	8%	0%	0%
Communicating (overall)	0%	4%	0%	0%	14%	0%	2%	0%	7%
Helping (overall)	11%	12%	47%	70%	41%	58%	12%	59%	50%

Table 7.3: This table shows the percentage of data points in the specificity assembly task that were coded with each code.

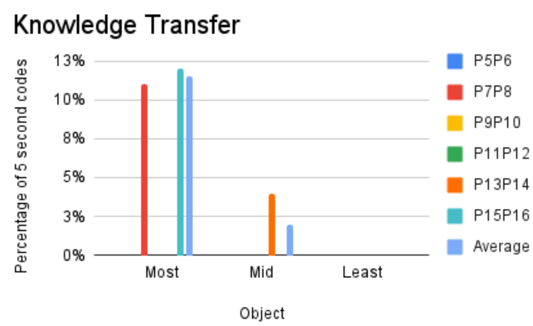
Helping: Knowledge Co-Creation During the assembly of the mid-specific and least specific devices, we observed knowledge co-creation. This occurred when there was uncertainty to create the knowledge, together, about where the ratchet and pawl should be mounted. Participant groups during both assemblies started with creating the knowledge of how the ratchet and pawl should function together and then figured out how to mount the two as a unit, either matching holes in the mid-specific case or drawing holes in the least specific case. We hypothesize this is what led to the similar levels of knowledge co-creation for the mid and least specific cases (shown in Figure 7.7a).

Participant groups in both the mid and least specific cases continued to do knowledge co-creation when testing their ratchet and pawl mechanisms after their first attempt at assembly. If there were errors in the assembly, they co-created the knowledge of how to fix the mechanism. Errors in assembly only occurred in the case of the mid-specific device. Participants tested their ratchet and pawl mechanism before drilling holes in the least specific case, catching errors before they occurred.

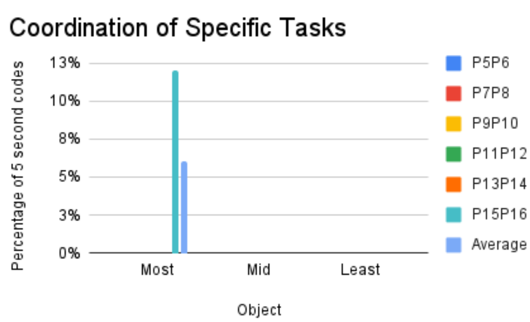
Coordination Though we hypothesized that specificity would have an impact on coordination both of people and of specific tasks, we did not observe any coordination of people during any



(a) The percentages of the knowledge co-creation of each group during their assembly. The mid and least specific devices show higher occurrences of knowledge co-creation as compared to the most specific device assembly.



(b) The percentages of the knowledge transfer of each group during their assembly. There is no large change across devices.



(c) The percentages of the coordination of specific tasks of each group during their assembly. There is no large change across devices.

Figure 7.7: The percentages of the codes we hypothesized would be affected by the specificity of the device.

build and only one instance of coordinating a specific task across all (Figure 7.7c). Participants all came into the task and began collaborating to build the object without the need to coordinate who would work on what. Additionally, as they were both working on solving the assembly problem, they did not coordinate what each person did. The only instance of coordinating a specific task was to coordinate the assembly of the clevis pins in the most specific device assembly.

Helping: Knowledge Transfer Though we hypothesized that specificity would have an impact on knowledge transfer, we did not observe a large difference in the occurrence of this code across devices (Figure 7.7b). Knowledge transfer happened in the most specific case when a participant was teaching the other how the clevis pins worked. This hardware was the same across both cases, but the instance of transfer was observed only in this case. Knowledge transfer specific to the least specific device occurred when one participant was pointing out the direction the clevis pin went in. He surmised this knowledge from the depression in the pawl and ratchet gear that the head of the clevis pin fit that feature of the hardware.

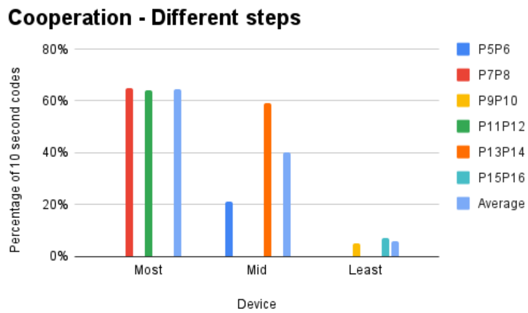
7.4.2 Parallelizability

Based on prior research [103] (Chapter 6), we hypothesized that the parallelizability of the device would affect how participants cooperated on different steps, coordinated, and helped each other. The percentages of coded data are shown in Table 7.4. We also found an unexpected increase in cooperation on the same step, but on different hardware, which influenced the cooperation codes overall.

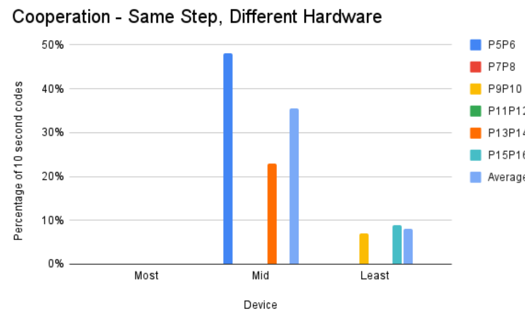
Cooperation - Different Steps We observed each group begin the task by splitting the assembly steps up after being told to assemble the device optimally and quickly. Instructions were laid out in three sections for all devices, and each participant attempted to tackle a different section of the device assembly. These were the few observations of cooperation on different steps seen in the least parallel device. However, participants soon realized they could not complete the steps on the least parallel device separately, and stopped cooperating on different steps, instead helping the other participant in their group with the first step.

For the most parallel device, participants ran into no issue with their original work split and built the device, cooperating on different steps for a majority of the assembly time. During the mid parallel device assembly, both groups started with the right and left corners of the device (Figure 7.3b). When the groups realized the buckets needed to be assembled into the corners, P13 and P14 left the task to a single person, and the other worked on the widgets, leading to increased cooperation on different steps. P5 and P6 did the task of bucket assembly together and then later built the widgets together (as it was the final remaining task), leading to less cooperation on different steps, but an increase in cooperation on the same step, different hardware.

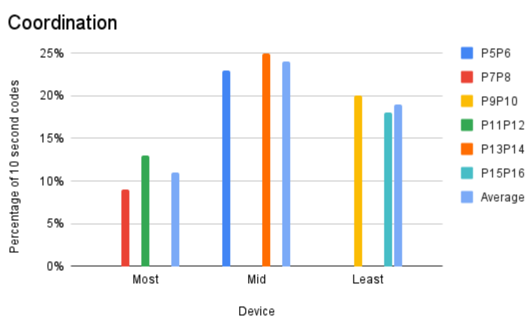
Cooperation - Same Step, Different Hardware Though we did not initially hypothesize a difference in cooperation on the same step, different hardware, we observed that the mid parallel assembly had more observations of this code than the least or most parallel devices. For the mid parallel device, cooperation on the same step, different hardware occurred in both groups when



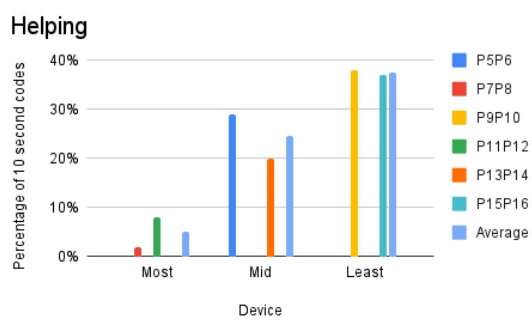
(a) The percentages of the "cooperation - different steps" code of each group during their assembly. The most parallel device results in the highest average percent. The mid parallel device is roughly mid-average percent. The least parallel has very little cooperation on different steps.



(b) The percentages of the "cooperation - same step, different hardware" code of each group during their assembly. The mid parallel device results in a high average percent. The most and least parallel devices have very little cooperation on different steps.



(c) The percentages of the coordination codes overall of each group during their assembly. Coordination is relatively similar across the different devices.



(d) The percentages of the helping codes overall of each group during their assembly. Helping behaviors increase in percentage as the device gets less parallel.

Figure 7.8: The percentages of the codes we hypothesized would be affected by the parallelizability of the device.

Code	Most		Mid		Least		Average		
	P7P8	P15P16	P5P6	P13P14	P9P10	P11P12	Most	Mid	Least
Coordination of people	9%	8%	4%	14%	7%	0%	9%	9%	4%
Coordination of materials	0%	3%	13%	11%	7%	15%	2%	12%	11%
Coordination of specific tasks	2%	5%	6%	9%	7%	4%	4%	8%	6%
Cooperation – Different steps	65%	64%	21%	59%	5%	7%	65%	40%	6%
Cooperation – Same step, diff. hardware	0%	0%	48%	23%	7%	9%	0%	36%	8%
Cooperation – Same step, same hardware	0%	0%	8%	0%	7%	12%	0%	4%	10%
Helping: Knowledge Co-Creation	0%	8%	19%	20%	31%	35%	4%	20%	33%
Helping: Knowledge Transfer	2%	0%	10%	0%	7%	1%	1%	5%	4%
Troubleshooting communication	0%	0%	0%	0%	0%	1%	0%	0%	1%
Communicating confusion over instructions	0%	0%	0%	0%	0%	0%	0%	0%	0%
Blocked collaboration	0%	0%	0%	5%	0%	0%	0%	3%	0%
Cooperation (overall)	65%	64%	77%	80%	20%	27%	65%	79%	24%
Coordination (overall)	9%	13%	23%	25%	20%	18%	11%	24%	19%
Communicating (overall)	0%	0%	0%	0%	0%	1%	0%	0%	1%
Helping (overall)	2%	8%	29%	20%	38%	37%	5%	25%	38%

Table 7.4: This table shows the percentage of data points in the parallelizability assembly task that were coded with each code.

finishing the task of building widgets. When there was only one task left (and thus no further way to parallelize the steps), the participants cooperated on the same step. However, P5 and P6 chose to cooperate on the same step, different hardware, when there was an option to continue parallelizing. Even though they were directed to assemble the device optimally and quickly, this did not always lead participants to parallelize when there was another option that also seemed to reduce assembly time.

Coordination We did not observe as strong an effect on coordination as cooperation; however, the most parallelizable device had slightly less coordination than the other two devices. We did observe more distinct instances of coordination happening in the mid and least parallel device assemblies as well (Figure 7.8).

We observed coordination of people and specific tasks happening at the beginning of the most parallel device's assembly. However, as time went on, there was not much need for the participants to continue coordinating as they were working on separate tasks. One group briefly confirmed with each other that P12 should start on the next step after finishing his first step.

During the mid parallel device assembly, participants coordinated to split initial work, then came back together when there was a dependency in assembling the buckets into the left and right corners. They further came back to coordinate when they moved onto the next tasks, seeing if there was a way to help each other or if they should split work independently. During the least parallel device assembly, participants again split initial tasks, but soon realized this was not possible and coordinated briefly to help one another on the first step. They then coordinated materials and specific tasks more throughout the tasks, and they determined how to assemble the device and which parts were for which step.

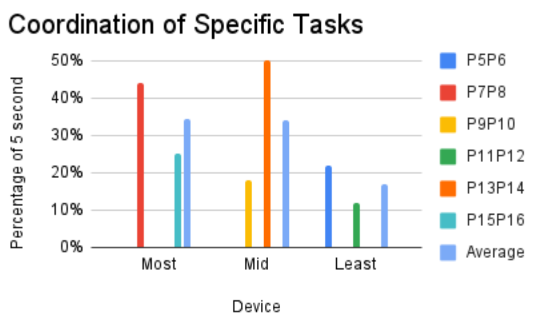
Helping We observed very little helping in the most parallel device assembly. Participants each had separate tasks and felt no need to ask for help, so they mostly worked on their own tasks. During the least parallel device assembly, participants often helped as they had nothing else to do during the assembly, and were both focused on the same task. During the mid parallel device assembly, P13 and P14 coordinated to build the left and right corners (one corner each) at the same time, so they could help each other along the way. They felt this would be the fastest way to assemble the device. This led to helping behaviors, both co-creating and transferring knowledge, during this task while they were cooperating on different steps. During the portion of their build when they worked on the frame and the widgets at the same time, they did not help as much. This suggests that the similarity of the parallelizable task plays a role in helping behaviors as well as parallelizability itself.

7.4.3 Physicality

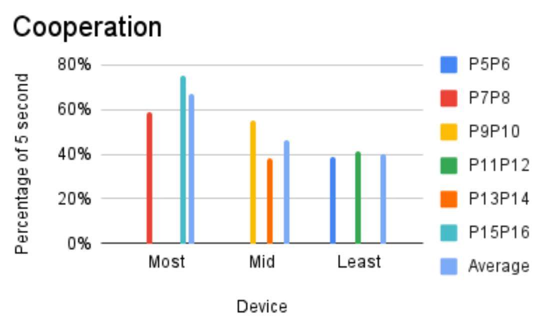
Based on prior research [103] (Chapter 6), we hypothesized that the physicality of the device would affect how participants cooperated on the same step and how participants coordinated on the specific task. The percentages of coded data are shown in Table 7.5. Notably, the stand was unused by P11 and P12 during their assembly of the least physical object. Additionally, they believe they had built the object wrong and corrected their perceived mistake, leading to longer build time and skewed percentages.

Coordination of Specific Tasks We observed a baseline of coordinating specific tasks throughout all assemblies, with a higher average percent during the most and mid physical device assemblies. The observed instances of coordination of the specific tasks were in managing the widgets and keeping them still while putting the two mounts together. In the most physical device assembly, a longer time was spent on this coordination, though the same percentage of time was spent due to a longer time being spent on achieving the task after coordinating. With the most physical device, we observed more strategies being suggested during the coordination of specific tasks than during the mid and least physical device assemblies. During the least physical device assembly, one group did not use the stand, which could have changed how they coordinated and skewed the results as well.

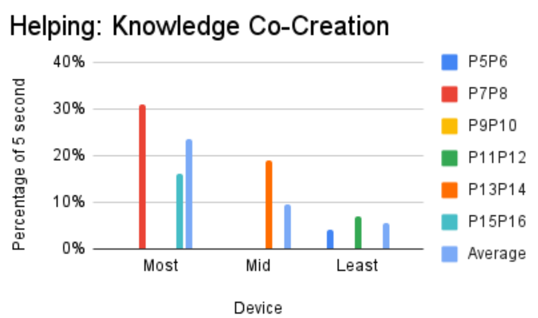
Cooperation We observed that the more physical a device is, the more groups cooperate on the assembly of that device. Across all devices, however, groups cooperated on building at least a third of the time. This could have been influenced by the physicality design, as even in the least physical case, there could be up to five unfixed parts to be managed at the same time. During the most physical device assembly, we observed that participants were more often using two hands each and holding multiple parts in a single hand. However, during the mid and least physical device assemblies, participants were more often using one hand and holding one part per hand.



(a) The percentages of the "coordination of specific tasks" code of each group during their assembly. The most and mid-physical devices have similar average coordination of specific tasks. The least physical device had less coordination of physical tasks.



(b) The percentages of the cooperation codes overall of each group during their assembly. The amount of cooperation decreased as the device decreased in physicality.



(c) The percentages of the knowledge co-creation codes of each group during their assembly. Knowledge co-creation is, on average, higher for the more physical devices.

Figure 7.9: The percentages of the codes we hypothesized would be affected by the physicality of the device.

Code	Most		Mid		Least		Average		
	P7P8	P15P16	P5P6	P13P14	P9P10	P11P12	Most	Mid	Least
Coordination of people	0%	13%	9%	13%	0%	0%	7%	11%	0%
Coordination of materials	0%	0%	0%	0%	0%	0%	0%	0%	0%
Coordination of specific tasks	44%	25%	18%	50%	22%	12%	35%	34%	17%
Cooperation – Different steps	0%	0%	0%	0%	0%	0%	0%	0%	0%
Cooperation – Same step, diff. hardware	56%	75%	55%	25%	39%	41%	66%	40%	40%
Cooperation – Same step, same hardware	38%	6%	0%	38%	0%	7%	22%	19%	4%
Helping: Knowledge Co-Creation	31%	16%	0%	19%	4%	7%	24%	10%	6%
Helping: Knowledge Transfer	0%	3%	0%	0%	0%	0%	2%	0%	0%
Troubleshooting communication	0%	0%	0%	0%	0%	0%	0%	0%	0%
Communicating confusion over instructions	6%	3%	0%	0%	0%	0%	5%	0%	0%
Blocked collaboration	0%	0%	0%	0%	0%	0%	0%	0%	0%
Cooperation (overall)	59%	75%	55%	38%	39%	41%	67%	47%	40%
Coordination (overall)	44%	28%	18%	50%	22%	12%	36%	34%	17%
Communicating (overall)	6%	3%	0%	0%	0%	0%	5%	0%	0%
Helping (overall)	31%	19%	0%	19%	4%	7%	25%	10%	6%

Table 7.5: This table shows the percentage of data points in the physicality assembly task that were coded with each code.

Knowledge Co-Creation Though we did not hypothesize that changing the number of unfixed objects in the devices would influence knowledge co-creation, we did see an increase in the code in the most and mid physical device assemblies. These instances of knowledge co-creation occurred when the participants were strategizing on how to approach the assembly and manage all the parts at the same time. We observed that in the most physical device assemblies, participants had to re-strategize after attempting a few assembly methods.

7.4.4 Difficulty

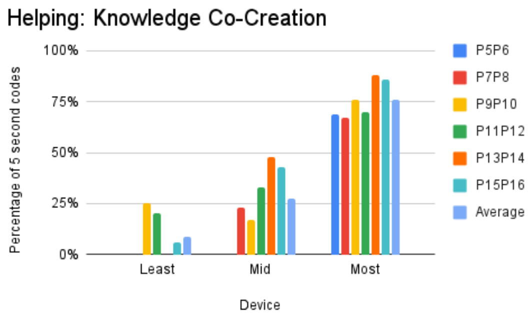
Based on prior research [103] (Chapter 6), we hypothesized that the difficulty of the device would affect the amount the participants coordinated on specific tasks, co-created knowledge, and transferred knowledge. The percentages of coded data are shown in Table 7.6.

Helping: Knowledge Co-Creation We observed that the most difficult device assembly had the most knowledge co-creation. During the least difficult device assembly (screw and nut fastening), both participants knew how to proceed and either checked in with their partner or simply began assembly. One group (P11 and P12) questioned whether the piece being screwed into had a specific orientation and resolved this uncertainty together, aligning more with the design principle of specificity than difficulty. This led to some knowledge co-creation, but on average, very little.

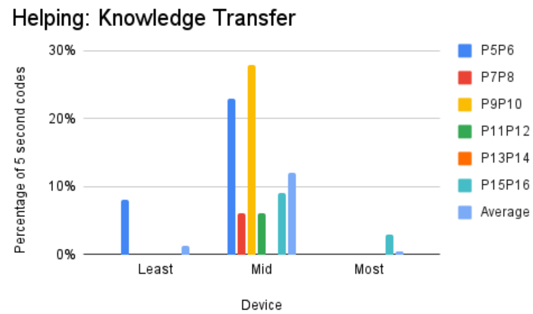
Code	Least						Mid					
	P5P6	P7P8	P9P10	P11P12	P13P14	P15P16	P5P6	P7P8	P9P10	P11P12	P13P14	P15P16
Coordination of people	0%	0%	0%	0%	22%	0%	0%	0%	0%	0%	0%	0%
Coordination of materials	0%	0%	0%	0%	0%	11%	0%	0%	0%	0%	0%	0%
Coordination of specific tasks	0%	11%	0%	13%	44%	17%	27%	29%	44%	28%	41%	20%
Cooperation - Different steps	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Cooperation - Same step, diff. hardware	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Cooperation - Same step, same hardware	46%	44%	0%	0%	44%	17%	0%	0%	6%	0%	17%	0%
Helping: Knowledge Co-Creation	0%	0%	25%	20%	0%	6%	0%	23%	17%	33%	48%	43%
Helping: Knowledge Transfer	8%	0%	0%	0%	0%	0%	23%	6%	28%	6%	0%	9%
Communicating troubleshooting errors	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Communicating confusion over instructions	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Collaboration not possible	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Cooperation (overall)	46%	44%	0%	0%	44%	17%	0%	0%	6%	0%	17%	0%
Coordination (overall)	0%	11%	0%	13%	56%	28%	27%	29%	44%	28%	41%	20%
Communicating (overall)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Helping (overall)	8%	0%	25%	20%	0%	6%	23%	29%	44%	39%	48%	51%

	Most						Averages		
	P5P6	P7P8	P9P10	P11P12	P13P14	P15P16	Least	Mid	Most
Coordination of people	0%	0%	0%	0%	13%	0%	4%	0%	2%
Coordination of materials	0%	4%	0%	0%	0%	0%	2%	0%	1%
Coordination of specific tasks	25%	46%	35%	23%	38%	29%	14%	32%	33%
Cooperation - Different steps	0%	0%	0%	0%	0%	0%	0%	0%	0%
Cooperation - Same step, diff. hardware	19%	17%	4%	7%	38%	5%	0%	0%	15%
Cooperation - Same step, same hardware	13%	0%	10%	0%	0%	14%	25%	4%	6%
Helping: Knowledge Co-Creation	69%	67%	76%	70%	88%	86%	9%	27%	76%
Helping: Knowledge Transfer	0%	0%	0%	0%	0%	6%	1%	12%	1%
Communicating troubleshooting errors	0%	0%	0%	0%	0%	0%	0%	0%	0%
Communicating confusion over instructions	0%	0%	10%	0%	0%	5%	0%	0%	3%
Collaboration not possible	0%	0%	0%	0%	0%	0%	0%	0%	0%
Cooperation (overall)	31%	17%	14%	7%	38%	19%	25%	4%	21%
Coordination (overall)	25%	50%	35%	23%	38%	29%	18%	32%	33%
Communicating (overall)	0%	0%	10%	0%	0%	5%	0%	0%	3%
Helping (overall)	69%	67%	76%	70%	88%	87%	10%	39%	76%

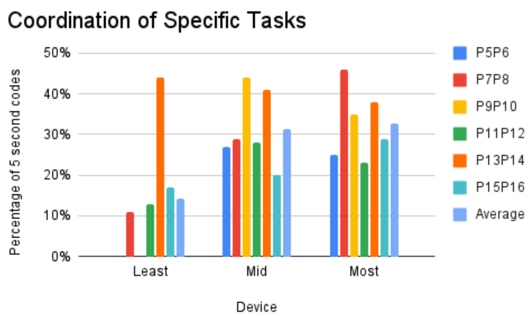
Table 7.6: This table shows the percentage of data points in the difficulty assembly task that were coded with each code.



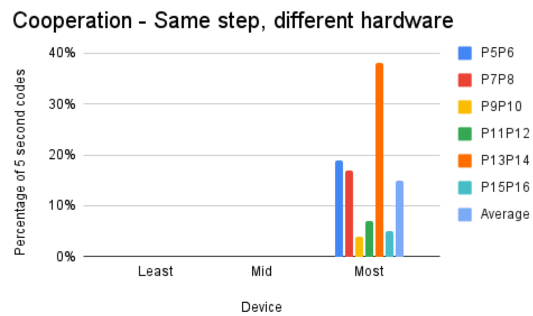
(a) The percentages of knowledge co-creation of each group during their assembly. As the devices got more difficult, the knowledge co-creation increased.



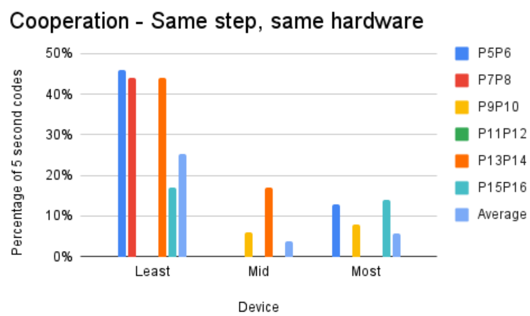
(b) The percentages of knowledge transfer codes of each group during their assembly. The most difficult device had the least knowledge transfer.



(c) The percentages of the "coordination of specific tasks" codes of each group during their assembly. Groups coordinated slightly more in the mid and most difficult device assemblies.



(d) The percentages of cooperation on the same step, different hardware codes of each group during their assembly. Groups only did this during the most difficult device assembly.



(e) The percentages of the "cooperation - same step, same hardware" codes of each group during their assembly. Groups did this more during the least difficult device assembly.

Figure 7.10: The percentages of the codes we hypothesized would be affected by the difficulty of the device.

During the assembly of the mid-difficult device (toggle bolt fastening), some participants (P5 and P6, P9 and P10) had prior knowledge of how toggle bolts worked and were able to proceed without co-creation of knowledge. All other groups, however, spent time collaboratively reasoning through the toggle's mechanism before implementing their solution. The implementation stage of the toggle bolt often took longer than the implementation of other fastening methods. This led to split levels of knowledge co-creation as it was necessary in some groups, and unnecessary in others. However, on average, about a quarter of the time was spent co-creating knowledge.

During the assembly of the most difficult device (shackle and clevis pin fastening), no participants entered with prior knowledge, and they spent substantially more time figuring out the mechanism. Most groups co-created knowledge to arrive at a solution, though strategies varied. P13 and P14 were able to solve it quickly, while P9 and P10, as well as P15 and P16, initially believed the clevis was a trick component and instead devised alternate attachment methods. A majority of the time was spent co-creating knowledge for the most difficult device across all groups.

Statistical tests support these observed trends. Comparing the least difficult and mid difficult devices, the sample size was too small for a Wilcoxon Signed Rank test, and a permutation test did not reach significance ($p = 0.125$), suggesting there was no significant increase in the knowledge co-creation for the mid difficult device assembly. Comparisons between the least difficult device and most difficult device ($p = 0.0313$) and between the mid difficult device and most difficult device ($p = 0.0313$) using the Wilcoxon Signed Rank test both yielded significant differences, indicating that the shackle and clevis fastening (most difficult) elicited reliably greater knowledge co-creation than the easier fastening tasks.

Helping: Knowledge Transfer Knowledge transfer was observed primarily in the mid-difficult device assembly. We observed P6 and P9 transferring knowledge of how toggle bolts work to their partner during the assembly. The other instances of knowledge transfer occurred when participants discovered something quicker than their partner and shared their knowledge. For example, P16 and P12 figured out that the toggle shouldn't be screwed all the way down on the bolt, to leave room for it to bend and fit through the mounting piece.

The few instances observed of knowledge transfer in the least and most difficult device assemblies were also instances of participants discovering something before the other. In the least difficult, one participant suggested a method of assembly first, before building. In the most difficult device assembly, P16 figured out how to open the shackle on his own and showed P15.

Statistical tests comparing the frequency of knowledge transfer between fastenings were marginal but informative. The difference between the least difficult and mid difficult devices (permutation test, $p = 0.0578$) and the mid difficult and most difficult devices ($p = 0.0627$) approached significance, suggesting that the toggle fastening elicited more knowledge transfer than the simple screw and nut or the complex shackle and clevis mechanism.

Coordination of Specific Tasks We did not observe strong effects of difficulty on coordination, though overall levels were slightly higher in the mid and most difficult devices. Statistical tests did not show significant differences, but trends suggested greater coordination in these conditions. The Wilcoxon Signed Rank test comparing the least and mid difficult devices approached significance (p=0.0938), as did the comparison between the least and most difficult devices (p=0.0625), while no difference was found between the mid and most difficult devices (p=1).

For the screw and nut fastening (the least difficult device), coordination was minimal and typically limited to participants stating their plan before carrying it out. For the toggle fastening (mid difficult device), we observed more coordination, as participants needed to sequence multiple steps: unscrewing the toggle, attaching piece A, partially screwing on the toggle, inserting the toggle through piece B, and then fastening the bolt. The number of steps in the sequence created more opportunities to coordinate specific tasks.

For the shackle and clevis fastening (the most difficult device), coordination occurred primarily around figuring out how the pieces fit together and how the shackle should fasten them. This often led to passing the hardware back and forth, with participants saying "here, you try" when stuck. The physical difficulty of inserting the shackle through the clevis also led to handoffs, as participants alternated attempts to complete the fastening.

Cooperation - Same Step, Different Hardware Though we did not hypothesize a difference in cooperation for different difficulties of devices, we saw cooperation on the same step, with different hardware only in the most difficult device assembly. This was due to the participants dividing the hardware as they attempted to problem-solve. All groups at some point had one participant working on how to open the shackle while the other worked on how to mount the pieces onto the clevis pin.

Cooperation - Same Step, Same Hardware Though we did not hypothesize a difference in cooperation for different difficulties of devices, we saw cooperation on the same step, same hardware increase for the least difficult device. Some groups chose to have one person hold the hardware together while the other screwed it together. Since this was not observed in all cases of the least difficult device assembly, we theorize that this was not necessary, but was done so that the participants felt as if they collaborated.

7.5 Discussion

In this section, we discuss our findings as suggestions for how the hardware design features specifically influence collaborative behaviors in an isolated lab setting with willing and cooperative volunteers. We also discuss further study possibilities briefly and summarize them in Section 7.6.1.

7.5.1 Specificity

Knowledge Co-Creation The mid and least specific devices were designed with many and infinite options for assembly, respectively. During the assembly of these devices, we observed participants employing the strategy of understanding the desired outcome and testing the assembly. Given the relatively similar amount of knowledge co-creation across the mid and least specific devices, it seems that the effective shift of problem-solving strategy from following the hardware design specifications for assembly to understanding the desired outcome and testing the assembly happens in the case of many options, and continues to be the case for infinite options in how to assemble the hardware. All participants tried to understand how to assemble the hardware by looking at it first. When the information was apparent (in the case of the most specific device), participants used this to build the device. When the information was not available from the hardware or instructions, participants shifted to collaborative problem-solving.

We hypothesized that specificity would affect knowledge co-creation in an inversely proportional manner. However, our results show that the effect may be more like an inverse sigmoid, possibly having a steeper cut-off point that represents a shift in problem-solving strategy. This would require further testing with more than three objects in order to fit a higher-dimensional curve to the results.

We also observed that in the case of the mid-specific object, participants were more willing to try test configurations of the device and adopt a trial-and-error strategy mixed in with problem-solving.In the case of the least specific device, participants found the cost of error to be high, and adopted a "measure twice, cut once" strategy while problem solving. These two strategies lead, in this specific case, to similar amounts of knowledge co-creation. This suggests that while different problem-solving techniques might be influenced by this perceived error risk, the effect on the amount of knowledge co-creation might be minimal. Further study with deeper analysis into the types of knowledge co-creation would be needed to determine if there are hardware design features that produce similar amounts but different qualities of knowledge co-creation.

Knowledge Transfer Previous work showed instances of knowledge transfer about non-specific hardware [103] (Chapter 6). However, we observed no instances of knowledge transfer about non-specific hardware. This suggests that specificity was not the only influencing factor in the previous research; there was another design feature that contributed to the opportunity to discuss the non-specific hardware. Further study is needed into how the design features are interrelated and if the combination of design features influences behaviors differently than each of the design features individually. Further study could investigate whether the combination of factors operates in an additive or multiplicative manner, or whether their interaction follows a qualitatively nonlinear pattern.

Coordination Again, previous work showed instances of coordination of people and specific tasks regarding non-specific hardware [103] (Chapter 6). However, we observed no instances of coordination of people and no instances of coordination of specific tasks regarding the changed hardware features across devices. This again suggests that specificity was not the only influencing factor in the previous research. Coordination was influenced by a combination of design features working together to provide the opportunity to coordinate people to do non-specific tasks or to figure out how to coordinate non-specific tasks.

7.5.2 Parallelizability

Overall, we observed a tradeoff between cooperation on different steps and helping behaviors during the assembly of differently parallelizable devices. The most parallel device allowed for the most cooperation, but decreased the communication of the group, and there were fewer helping behaviors and distinct instances of coordination. However, having a device with built-in dependencies at different points can increase the cooperation on the same step, the number of times a group coordinates, and the amount of helping behaviors. Helping behaviors can also be increased by devices that are parallel, but have similarities in the parallel tasks.

Cooperation When participants had a choice to cooperate on different steps or the same step, they did not always choose different steps to be optimal and time-efficient. This could indicate that participants saw value in close cooperative work, which resulted in more helping behaviors. This suggests value in different forms of cooperation through assembly (closely working together and more distantly working together). Hardware designed just for one or the other may not present the ideal learning experience or collaborative experience, as pure cooperation on different steps could lessen other forms of collaboration.

Cooperation was also seen in this case in the repetition of assembling widgets. When there were no tasks left, participants would cooperate on the same task and do so easily, as the task was repetitive. Further research could explore the interdependence of repetitiveness and parallelizability to determine the relationship they both have with collaborative behaviors.

Coordination Coordination was seen in all cases as participants tried to satisfy the instruction to build optimally and quickly. The instructions were laid out in a manner such that participants initially agreed to split tasks according to the task split of the instructions. However, the hardware being designed to have dependencies between steps, but still have some parallel tasks, created more coordination later in the process in the form of strategizing. Participants would discuss how to handle the dependency, then split back up. Entirely serial devices led to less of this strategizing and more coordination to help each other with the same task.

Helping Entirely parallel devices do not necessarily encourage helping behaviors; however, devices that have parallel tasks that are similar can encourage helping behaviors even when working on different tasks.

7.5.3 Physicality

Though we observed some differences in percentages, the results suggest that the amounts of physicality did not change enough to show large changes in the coordination and cooperation of the participants. In the least physical device, the number of moving parts could be chosen to be as high as five, which could have influenced the baseline of at least one-third of the time spent cooperating across all groups. Future research should test more extremes of physicality to see if the effects have an upper limit and lower limit to the amount of coordination of specific tasks, and whether cooperation is possible. Furthermore, the introduction of the stand was an optional piece of hardware and thus skewed the results of the least physical device. Future research could explore how optional pieces of hardware affect the assembly or how participants choose to use these optional pieces of hardware.

The results of the coordination of specific tasks did show a slight increase in percentage across the most and mid physical device assemblies. However, the kind of coordination being done was not captured in the codes. In the mid-physical device assembly, coordination of specific tasks was often done once, and then assembly was achieved with the first strategy. With more moving parts, participants discussed strategy over a longer portion of time, but spent longer testing these strategies and assembling the device. Further analysis methods are needed to ascertain the different types of coordination and what type of coordination could be most beneficial for education or other applications. Then further research is needed to ascertain how physicality could influence these supposed different kinds of coordination.

When the device had more moving parts (most physical), participants' hands were more engaged (often multiple hands and multiple parts per hand). Further research is needed to determine if this is a desirable educational outcome and how the physicality could possibly overwhelm students if too much simultaneous manipulation is needed. However, when there were more moving parts, participants co-created knowledge more to strategize how they would handle all the moving parts. When designing for collaboration, physicality in the form of simultaneous manipulation can lead to more knowledge co-creation.

7.5.4 Difficulty

Overall, we observed trends that support our initial hypotheses that knowledge co-creation would increase with difficulty and knowledge transfer would be highest in the mid difficulty case. However, we did not see an increase in coordination of specific tasks with difficulty, but did see cooperation shift with difficulty. Overall, our results approach statistical significance; however, our sample is still small and not diverse. Future research could include a larger participant pool. Additionally, further research could include more nuanced measurements, separating the brainstorming phases and the implementation phases to further compare the impact on knowledge co-creation and transfer.

Helping: Knowledge Co-Creation Task difficulty shaped not only the amount of knowledge co-creation but also the kinds of strategies participants employed. When participants encountered difficult fastening mechanisms, they frequently questioned the instructions themselves and searched for entirely different solutions that might still satisfy the stated goal of assembling the device. This highlights that difficulty was not simply a matter of longer problem-solving time, but a trigger for broader exploration of what constituted a valid solution. This was coupled with cases where only one participant could be hands-on at a time. In this case, participants looked to other objects in the device for answers. For example, while one participant struggled with the shackle, their partner examined the clevis pin and articulated how the two components must connect.

Difficulty also interacted with prior knowledge. When both participants lacked relevant prior knowledge, they were more likely to engage in joint reasoning, co-constructing an understanding of how the mechanism worked. In contrast, uneven prior knowledge often reduced co-creation, as the more knowledgeable participant took the lead while the partner primarily observed. These dynamics suggest that prior knowledge acts as a mediating factor in whether difficulty produces co-creation or limits it.

We also observed characteristic interaction patterns during difficult tasks. Often, one participant physically manipulated the hardware while the other observed and suggested possible approaches. Groups alternated roles, with participants either trading off attempts or contributing verbal suggestions to refine the strategy. This pattern of one actor and one observer created space for distributed reasoning, as participants could monitor progress, propose alternatives, and co-construct knowledge even without both being hands-on at the same time.

Taken together, these findings suggest that difficulty fosters knowledge co-creation not in a uniform way but through pathways shaped by prior knowledge and physical affordances (such as physicality or specificity). Future work should examine how these design principles are interdependent and in what manner to achieve the ideal amount of knowledge co-creation during assembly. Further research should also include more participants to verify if the statistical significance found holds with a more diverse population.

Helping: Knowledge Transfer We observed that difficulty influenced knowledge transfer primarily through uneven familiarity. When hardware was familiar to one participant but not the other, knowledge transfer occurred as the knowledgeable participant explained the mechanism. This often shifted the group dynamic into one person taking a leading role, with the partner in a more receptive role. In some cases, even when no participant entered with prior knowledge, one participant figured out the mechanism more quickly and transferred this understanding to their partner. These findings suggest that difficulty in the form of asymmetry of familiarity creates knowledge transfer. This is hard to measure before participants have the devices in front of them, so future work should investigate how these role shifts through knowledge transfer influence broader collaboration dynamics and whether they constrain or support mutual engagement in problem solving.

Coordination of Specific Tasks Our findings suggest that coordination was not strongly driven by task difficulty alone, though there were indications of slightly greater coordination with the mid and most difficult devices. The unobvious nature of the mid and most difficult tasks leads to more coordination on how to do the tasks. Coordination may also be influenced by other design principles; for example, specificity could influence the difficulty if there are more guiding features to create the answer. Future work should investigate these intersections to determine whether coordination arises from difficulty, specificity, or the interplay between design principles in shaping assembly strategies.

Cooperation Though we did not expect differences in cooperation, we did observe them. In the most difficult device assembly, participants cooperated on the same step, using different hardware to attempt to find the solution by observing both the clevis and the shackle. This may have been influenced by other design principles as well (such as physicality, giving the opportunity to split the hardware). However, the opportunity was also created by the lack of familiarity, drawing participants to try to find the solution simultaneously with their partner.

We also observed cooperation on the same step, same hardware, in the least difficult device assembly. This cooperation was not entirely necessary, as shown by the fact that some groups did not do it. However, groups choosing to cooperate in this instance are interesting, as it could potentially have been done to make the participants feel like they collaborated on the assembly, as they had nothing to discuss before. This could lead participants to feel as if their participation in the assembly was not needed, and if done in an educational setting, could lead to adverse effects on attitudes toward collaborative learning [75]. Further study is necessary to determine the effect of difficulty on participants' attitudes after assembly.

7.6 Conclusions

In this paper, we presented a study of paired collaborative assembly with 12 participants assembling one device each of devices designed for specificity, parallelizability, and physicality, and three devices designed for difficulty. We utilized our collaborative coding scheme (Appendix A) to analyze collaborative behaviors to determine how the change of design influenced these behaviors.

Our observations and analysis revealed that hardware design principles elicited collaborative actions. We studied four different design principles, finding that:.

DP2. Less **specificity**, or features to indicate or error-proof assembly, elicited knowledge co-creation.

DP3. More mental **difficulty** led to higher knowledge co-creation. Higher knowledge transfer was seen with mid-level difficulty, as one participant had prior knowledge and shared it with the other. More difficult hardware led to cooperation on different hardware, with pairs simultaneously working to physically find an answer. More difficulty also led to more coordination of the specific task, usually while co-creating knowledge.

DP4. **Parallelizability** in hardware led to more cooperation on different steps, but less helping behaviors. A somewhat parallelizable assembly led to more coordination.

DP5. More **physicality**, referring to more simultaneous manipulation needed, led to more coordination of specific tasks, more cooperation on the same step, and more knowledge co-creation.

7.6.1 Future Work

Through this study, I presented initial causal evidence of the influence of hardware design on collaborative behaviors. This study, however, has several limitations and paths for future work. One limitation of the study is the small sample size. Due to ordering effects found during initial testing, pairs could build only one of the three devices designed for three of the four design principles. This led to small samples for comparison and extremely low statistical power in the results. Future studies should include more participants in order to test the statistical significance of the results. Similarly, the population of the study is relatively uniform, with most participants being graduate students from engineering fields. Future studies should include more diverse populations. Furthermore, the participants all seemed open to collaboration without needing further prompting. Future work could include participants less inclined to collaborative behaviors to see how results differ.

This study was also limited to changing one design principle at a time. Future studies could vary a variety of design principles to investigate if the combination of factors operates in an additive or multiplicative manner, or whether their interaction follows a qualitatively nonlinear pattern. The interdependence of repetitiveness and parallelizability makes this combination of particular interest for study.

Future work could also generally explore the effects of these collaborative behaviors on attitudes and collaborative skills. None of the participants found the hardware to be frustrating in this study; however, the risk of difficult hardware is increasing frustration to a detrimental amount. Future studies should attempt to quantify this trade-off point. Future studies should also quantify the relationship between observed collaborative behaviors and the participants' collaborative skills and attitude toward collaboration. Initial attitudes collected in this study suggest participants held different opinions about collaboration than what we observed.

Future studies should also identify what the ideal collaborative actions are for learning during these tasks. Our study measured the time spent on knowledge co-creation; however, we did not measure the quality of the knowledge co-creation. Further studies could investigate the type of knowledge co-creation and how it affects learning. Furthermore, we categorize coordination into people, materials, and specific tasks as this was the breakdown seen through the iterative coding process described in Section 6.3.3. However, the impact of these forms of coordination on learning is unclear and requires further study and possible re-classification. Finally, grounded cognition theory and activity theory posit that having physical engagement is beneficial for cognition and learning. This should be empirically studied to see if cooperation, and potentially how many hands a participant is engaging, affects learning.

Chapter 8

Comparative RoboLoom Assembly Study

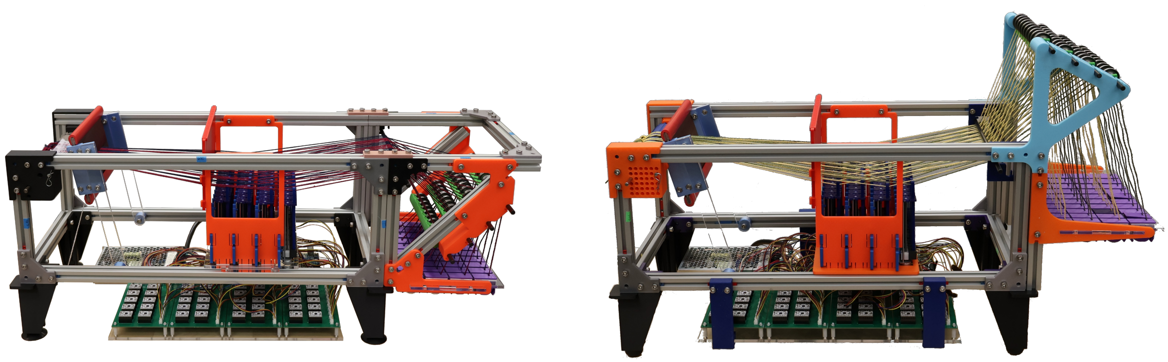


Figure 8.1: Two versions of RoboLoom: One designed for speed of assembly (Version A, left) and the other designed for collaborative assembly (Version B, right).

8.1 Introduction

Building on the findings from the earlier studies presented in this thesis, I now turn to testing the causal effects of hardware design on collaborative assembly in a larger assembly task. In previous work, I identified design features of RoboLoom and other devices that appeared to influence students' collaborative behaviors during assembly. These features included specificity, parallelizability, physicality, and difficulty. I then showed the causal relationships between designing for and against four of these design features. In this study, I sought to study this causality in the context of a classroom and a larger assembly task.

To do so, I designed a new version of RoboLoom that has less specificity and more parallelizability. We deployed the two versions of RoboLoom (one designed for speed of assembly [Version A], one designed for collaborative assembly [Version B]) in two undergraduate classrooms. A total of 16 students built RoboLoom Version A, and 10 students built RoboLoom Version B. I recorded their assembly, breaking down the data into five-minute time steps and analyzing their behaviors using the coding scheme developed in Section 6.3.3 and summarized in Appendix A. This approach isolates the specificity and parallelizability design principles, allowing for the comparison of the effect of specific hardware design choices on coordination, cooperation, and communication in a real-world setting. By examining how differences in design led to differences in collaborative practice, I aim to provide stronger evidence for the causal role of hardware design in shaping human-human collaboration.

8.2 Related Work

8.2.1 Collaboration

As in Section 2.6, we define collaboration as a process of two or more people collectively working towards a shared goal where the output of the group cannot be easily separated into individual contributions [21, 24, 65, 96]. We use the 3Cs model: Communication, Coordination, and Cooperation [25, 33] to break down collaborative behaviors more specifically. Coordination is often defined as the process of organizing the people, activities, and resources necessary to accomplish a shared goal, ensuring shared understanding about the state of each [22, 25, 33, 65, 96]. Conversely, cooperation is the process of individuals working on achieving specific tasks that contribute to a shared goal [8, 33, 65, 96]. Baker distinguishes between cooperation and collaboration in that "cooperation works on the level of tasks and actions, collaboration works on the plane of ideas, understanding, representations" [8]. Communicating, we break down and focus on the ideas of communication used by participants to help solve problems encountered in trying to complete the collaborative task. We break this further into helping with knowledge co-creation, the exchange of ideas back and forth between two or more people, and knowledge transfer, the exchange of knowledge from one person to another without reciprocal knowledge exchange.

8.2.2 Designing Loom Hardware for Collaborative Assembly

Based on our definition of collaboration, technology that supports collaboration must be designed for supporting communication, coordination, and cooperation [65]. The design principles put forth earlier in this thesis (Section 6.5) each theoretically contribute to creating a task that requires multiple actors to accomplish the goal. They also provide opportunities in which creating a shared context and having a shared organization become beneficial for the completion of the task as well. However, the prior work establishing these five design features does not show causation, as it was purely observational. Furthermore, close, but not direct comparisons were drawn as the hardware shared intended purposes across the looms, but was not similar. These design principles were tested causally earlier in this thesis (Chapter 7). However, these assembly tasks were small and did not study the effects of coordination and overall changes of collaboration in a larger setting across a longer assembly period.

This study builds directly on this prior work to systematically vary three portions of the RoboLoom design in specificity and parallelizability to observe the larger effects of these design changes across a large system. In this way, we study coordination as it becomes necessary in a larger task setting. We also study if changes to some hardware have an effect on how participants approach assembly tasks afterwards (i.e., will participants be more inclined to co-create knowledge after setting this principle because of a hardware design feature?).

8.3 RoboLoom Design Changes

In this chapter, we test two versions of RoboLoom. Version A of RoboLoom was designed for speed of assembly based on the designs presented in Chapter 4. Version B of the RoboLoom was designed to have less specificity and less parallelizability than Version A, but otherwise remain the same. Version B contains three design changes: warp beam mounting, creel, and frame.

8.3.1 Warp Beam Mounting

Version A of RoboLoom is designed with a warp beam bracket that has two sets of holes (Figure 8.2). To assemble, the student puts the warp beam shaft in the top hole, large enough to fit the shaft and the clevis pin for the pawl in the other hole on the top of the bracket. Assemblers have a choice of which pair of holes to choose, but the bottom set, the pawl will not catch, showing immediately that it's the wrong set of holes.

Version B of RoboLoom's warp beam mounting system has a bracket with a grid of 6x6 holes. All holes are the same size with no indication of which hole is used for the warp beam and which for the pawl. Once the bracket is mounted, the assemblers need to determine which hole to put the warp beam and pawl in. The specifications were given that the top of the warp beam be level with the back bar of the tensioning system. The lack of features indicating the mounting position of the warp beam and pawl makes it less specific than Version A.

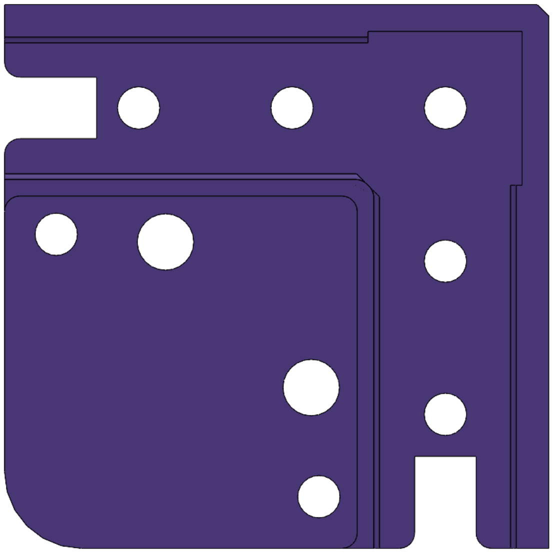
8.3.2 Creel

Version A of RoboLoom has the creel designed as a separate stand-alone frame that is attached to the loom after being assembled by two straight brackets. The two sides of the creel can be built in parallel to each other, then connected by the width beam of the creel. Shown in Figure 8.3.

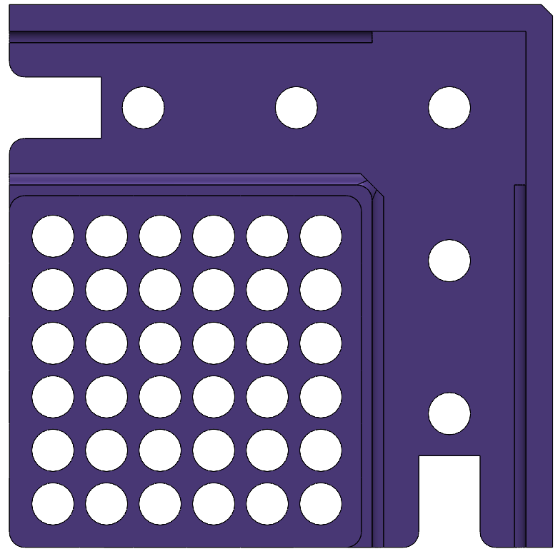
Version B of RoboLoom has a creel designed as two individual beams with 3D printed parts on them. Each of these beams can be assembled separately in parallel. However, for the creel to get structure, it must be fastened to the main frame. The bobbin holder and tension rods cannot be held in the creel without having built the frame of the creel onto the main loom frame. Because there is an ordering to how this version of the creel must be built, it is less parallelizable than Version A.

8.3.3 Frame

Version A of the RoboLoom has a frame in which the height beams and length beams are assembled on top of the Base Frame. The instructions indicate the order and proper arrangement of the 8020. There are no hardware features that specify this. Shown in Figure 8.4.

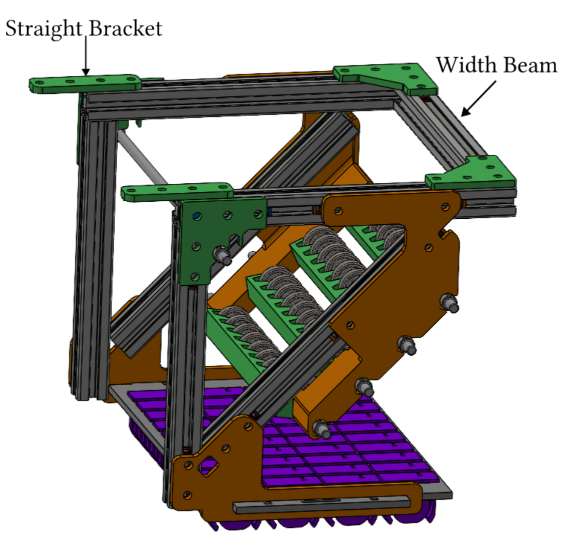


(a) The warp beam bracket design for RoboLoom Version A with only two options for mounting the warp beam and pawl.

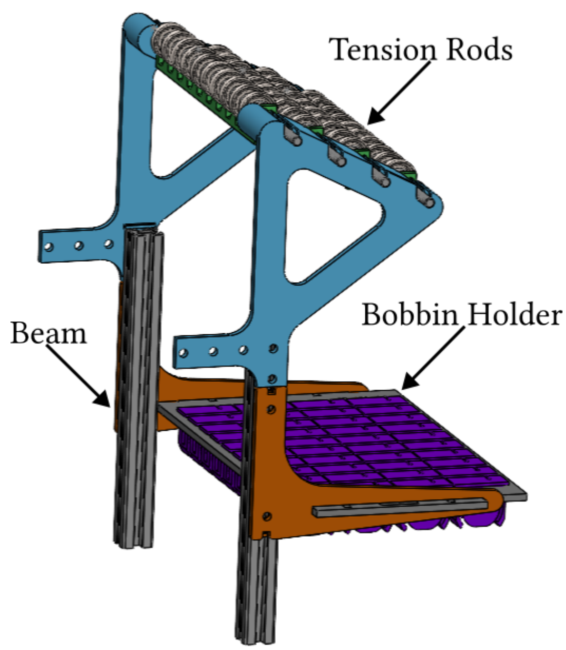


(b) The warp beam bracket design for RoboLoom Version B with many options for mounting the warp beam and pawl.

Figure 8.2: The warp beam bracket for RoboLoom Version A and B.

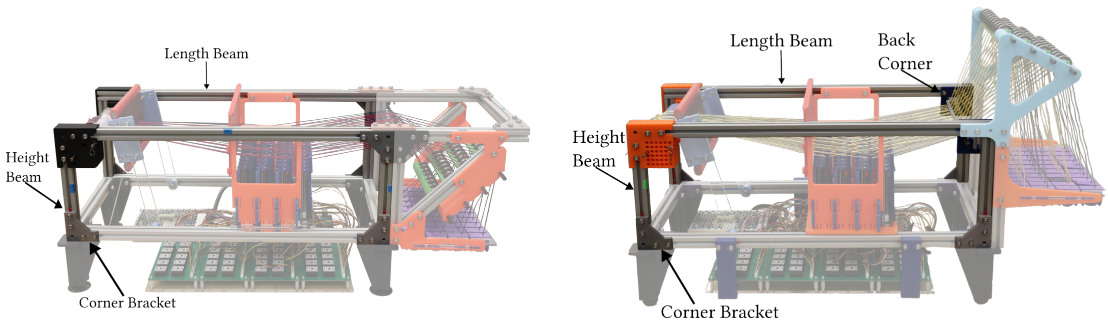


(a) The creel design for RoboLoom Version A.



(b) The creel design for RoboLoom Version B.

Figure 8.3: Creel designs for version A and version B of the RoboLoom.



(a) The frame design for RoboLoom Version A. (b) The frame design for RoboLoom Version B.

Figure 8.4: Frame designs for version A and version B of the RoboLoom.

In Version B, the frame hardware is relatively the same, with the back corner bracket changed to allow the creel to mount to the frame. However, the instructions were changed to not specify the way the aluminum extrusion in the frame should be arranged. The instructions only provide a final height and length of the frame.

8.4 Methods

To determine if the hardware design influenced collaborative behaviors, we deployed the two versions of RoboLoom in two different iterations of the same undergraduate course (based on the pilot course described in Chapter 5). In each iteration, students were recorded during their RoboLoom assembly. Students worked in interdisciplinary groups of 3 or 4. Groups spent between approximately 2.5 and 5 hours assembling their looms. The study was approved by Carnegie Mellon University's IRB.

8.4.1 Participants

Participants were recruited from the undergraduate class taught with RoboLoom and provided informed consent before the collection of their data. 16 students from Version A of the course chose to participate in our study. These participants were: one sophomore, five juniors, seven seniors, one fifth-year student, and two master's students. Participants came from diverse majors, including design (n=3), architecture (n=2), information systems (n=2), computer science (n=1), cognitive science (n=1), statistics and machine learning (n=1), electrical and computer engineering (n=1), mechanical engineering (n=2), computational design (n=1), economics (n=1), and human-computer interaction (n=1). Students were grouped to maximize interdisciplinary groups, as shown in Table 8.1.

10 students from Version B of the course chose to participate in our study. These participants were: one sophomore, three juniors, three seniors, one fifth-year student, and two master's students. Participants came from diverse majors, including environmental engineering (n=1), architecture (n=1), electrical and computer engineering (n=2), art (n=2), mechanical engineering

(n=1), drama (n=1), computational design (n=1), and design for interactions (n=1). Students were grouped to maximize interdisciplinary groups, as shown in Table 8.1.

RoboLoom Version A				RoboLoom Version B		
G1	G2	G3	G4	G5	G6	G7
S1 ● (Art)	S2 ● (Art)	S3 ●● (Art, Computer Science)	S7 ● (Computer Science)	S17 ● (Engineering)	S21 ● (Engineering)	S24 ● (Art)
S5 ● (Math)	S4 ● (Math)	S9 ●● (Computer Science, Engineering)	S10 ● (Computer Science)	S18 ● (Art)	S22 ● (Art)	S25 ● (Engineering)
S6 ● (Engineering)	S11 ●● (Psychology, Math)	S12 ● (Engineering)	S15 ● (Engineering)	S19 ● (Engineering)	S23 ●● (Computer Science, Art)	S26 ● (Art)
S8 ● (Art)	S16 ● (Art)	S14 ● (Engineering)	—	S20 ● (Art)	—	—

● Math ● Computer Science ● Art ● Psychology ● Engineering

Table 8.1: Group composition and student disciplinary backgrounds. Multiple dots indicate interdisciplinary students.

8.4.2 Data Sources and Analysis

We collected audio and video recordings of the assembly sessions and constructed a data log for each object the participant group constructed. Each data log was a descriptive transcription of each video of the loom version assembly, where actions and speech were captured in 5-minute segments. Table 8.2 shows the length of each group's build time and the version of the loom they built, as well as the number of 5-minute segments that comprise the data log for that RoboLoom Version build. Figure 8.5 gives a graphical summary of the coded data. Some portions of the Version A recordings are missing. For Group 1, the missing footage begins when participants were nearly finished with the assembly, with only the heddle bank sides and creel tension rods remaining to be installed. For Group 2, the missing segment includes the final steps of finishing the tension rods and attaching the creel to the base frame. Group 3's recording is complete, while Group 4's recording is missing the portion showing the installation of the beater. In our analysis, we use the coding scheme previously described in Section 6.3.3 to evaluate the collaborative actions based on the 3Cs framework.

RoboLoom Version A				RoboLoom Version B		
G1	G2	G3	G4	G5	G6	G7
2:35:50	2:53:59	2:28:25	3:32:23	2:34:25	3:42:20	5:06:59

Table 8.2: Completion times (hh:mm:ss) for each group using RoboLoom Version A and B.

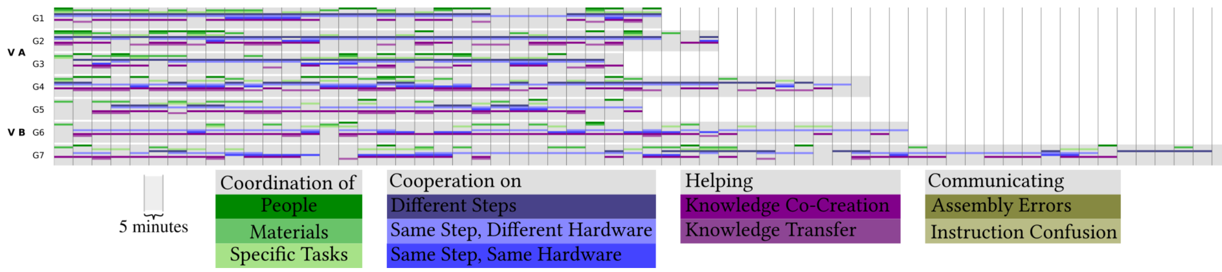


Figure 8.5: A graphical summary of the codes assigned to each 5-minute block of each loom construction session. "V A" and "V B" denote the RoboLoom Version A and Version B, respectively.

8.5 Findings

Here we show the results of the coding process for the overall loom assembly and the loom assembly steps for only those steps where the design was changed. For the specific steps where designs were changed, percentages are reported as the total number of time steps with a given code divided by the total number of time steps where at least one participant was working on that assembly step.

8.5.1 Overall RoboLoom Assembly

Based on prior research [103] (Chapter 6), we hypothesized that the changes in the RoboLoom version B, being less specific and less parallel, would lead to more communication to help co-create knowledge, less cooperation on different steps, and less coordination. The percentages of coded data are shown in Table 8.3.

We ran two-tailed t tests of unequal variances on our results at a significance level of 0.05. We found that 'coordination of people' was significantly lower in Version B (M = 7.3%, SD = 3.1%) than in Version A (M = 26.0%, SD = 2.2%), $t(3.46) = 9.03$, $p = .002$. 'Cooperation – Different steps' also decreased from Version A (M = 82.0%, SD = 19.5%) to Version B (M = 18.7%, SD = 16.2%), $t(4.87) = 4.69$, $p = .006$. Finally, we saw 'Helping: Knowledge Co-Creation' increased, though not significantly, from Version A (M = 54.2%, SD = 11.1%) to Version B (M = 72.3%, SD = 15.2%), $t(3.52) = -1.73$, $p = .168$.

These results suggest that the decrease in overall parallelizable hardware did lead to a decrease in coordination of people and cooperation on different steps. These results also suggest that the change in knowledge co-creation due to the less specific hardware was not significant overall. This could be due to the relative time taken for each of these tasks. The loom frame took longer to assemble than the warp beam.

8.5.2 Warp Beam Mounting - Specificity

Based on prior research [103] (Chapter 6), we hypothesized that the changes in the warp beam of RoboLoom version B being less specific would lead to more communication to help co-create

Code	RoboLoom Version A				RoboLoom Version B			Average	
	G1	G2	G3	G4	G5	G6	G7	V. A	V. B
Coordination of people	28%	26%	27%	23%	10%	4%	8%	26%	7%
Coordination of materials	41%	26%	30%	16%	29%	20%	19%	28%	23%
Coordination of specific tasks	50%	26%	40%	30%	23%	22%	24%	37%	23%
Cooperation – Different steps	88%	94%	93%	53%	29%	0%	27%	82%	19%
Cooperation – Same step, diff. hardware	100%	83%	70%	84%	87%	78%	60%	84%	75%
Cooperation – Same step, same hardware	16%	11%	20%	28%	10%	27%	24%	19%	20%
Helping: Knowledge Co-Creation	56%	54%	40%	67%	84%	78%	55%	54%	72%
Helping: Knowledge Transfer	13%	43%	30%	53%	32%	11%	10%	35%	18%
Troubleshooting communication	9%	6%	0%	9%	19%	7%	10%	6%	12%
Communicating confusion over instructions	0%	3%	0%	0%	0%	2%	0%	1%	1%
Blocked collaboration	3%	3%	0%	5%	3%	0%	0%	3%	1%
Cooperation (overall)	100%	94%	93%	95%	87%	93%	89%	96%	90%
Coordination (overall)	78%	51%	70%	56%	52%	38%	44%	64%	45%
Communicating (overall)	9%	9%	0%	9%	19%	9%	10%	7%	13%
Helping (overall)	66%	71%	60%	86%	87%	80%	58%	71%	75%

Table 8.3: This table shows the percentage of data points for the assembly of RoboLoom Version A and Version B that were coded with each code.

knowledge. The percentages of coded data are shown in Table 8.4.

We ran two-tailed t tests of unequal variances on our results at a significance level of 0.05. We found that 'Helping: Knowledge Co-Creation' increased significantly from Version A (M = 42.5%, SD = 25.3%) to Version B (M = 93.3%, SD = 11.5%), t(4.39) = -3.55, p = .02. Though this test reached significance, the power of the test remains very low as the number of time steps that participants were working on the warp beam were between two and five, yielding fairly large percentages.

We hypothesize that the relative changes in the codes of coordination on specific tasks, same step, different hardware, and same step, same hardware were because the warp beam assembly also changed between versions. In Version A, the warp beam was assembled during the frame assembly as a structural element of the loom due to how the warp beams were 3D printed. Because Version B was designed to have more trial and error, the warp beam was able to be placed again in the loom while the frame was already built. This could have led to the differences in the other codes seen in the data.

8.5.3 Creel - Parallelization

Based on prior research [103] (Chapter 6), we hypothesized that the changes in the creel design of RoboLoom version B, being less parallelizable, would lead to less coordination of people, less

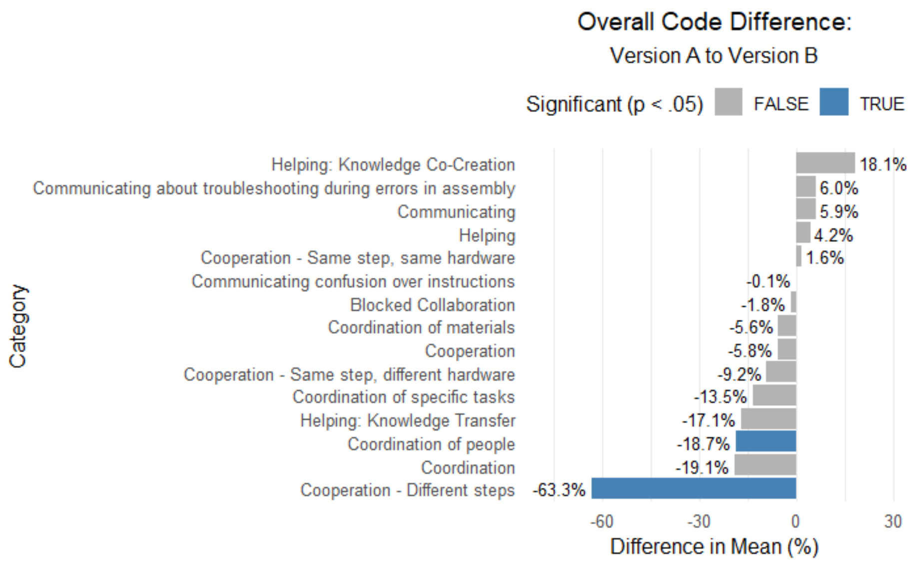


Figure 8.6: The percentage difference of the codes between building RoboLoom Version A and Version B. The codes shown in blue reached statistical significance at a p of 0.05.

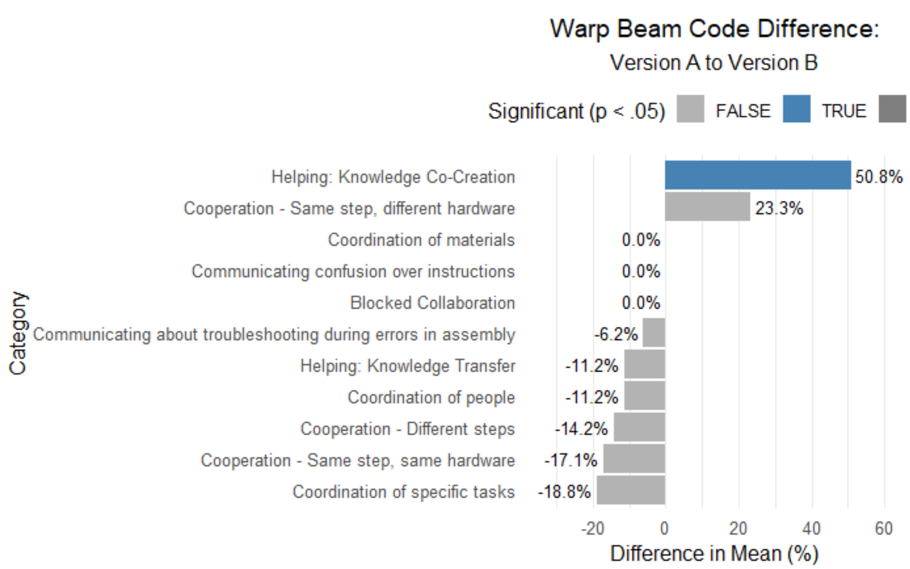


Figure 8.7: The percentage difference of the codes between building RoboLoom Version A and Version B for the steps involving building the warp beam. The codes shown in blue reached statistical significance at a p of 0.05.

Code	RoboLoom Version A				RoboLoom Version B			Average	
	G1	G2	G3	G4	G5	G6	G7	V. A	V. B
Coordination of people	0%	20%	0%	25%	0%	0%	0%	11%	0%
Coordination of materials	0%	0%	0%	0%	0%	0%	0%	0%	0%
Coordination of specific tasks	0%	0%	25%	50%	0%	0%	0%	19%	0%
Cooperation – Different steps	25%	60%	25%	0%	0%	0%	40%	28%	13%
Cooperation – Same step, diff. hardware	0%	0%	0%	0%	50%	0%	20%	0%	23%
Cooperation – Same step, same hardware	0%	20%	50%	25%	0%	0%	20%	24%	7%
Helping: Knowledge Co-Creation	50%	20%	25%	75%	100%	100%	80%	43%	93%
Helping: Knowledge Transfer	0%	20%	25%	0%	0%	0%	0%	11%	0%
Troubleshooting communication	25%	0%	0%	0%	0%	0%	0%	6%	0%
Communicating confusion over instructions	0%	0%	0%	0%	0%	0%	0%	0%	0%
Blocked collaboration	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 8.4: This table shows the percentage of data points for the assembly of RoboLoom Version A and Version B warp beam that were coded with each code.

cooperation on different steps, but more cooperation on the same step, different hardware. The percentages of coded data are shown in Table 8.5.

Code	RoboLoom Version A				RoboLoom Version B			Average	
	G1	G2	G3	G4	G5	G6	G7	V. A	V. B
Coordination of people	12%	13%	6%	4%	14%	14%	0%	9%	9%
Coordination of materials	6%	13%	0%	4%	0%	0%	0%	6%	0%
Coordination of specific tasks	29%	0%	0%	13%	14%	0%	6%	11%	7%
Cooperation – Different steps	71%	63%	100%	52%	0%	0%	35%	72%	12%
Cooperation – Same step, diff. hardware	71%	13%	0%	22%	0%	29%	35%	27%	21%
Cooperation – Same step, same hardware	18%	25%	0%	17%	0%	43%	0%	15%	14%
Helping: Knowledge Co-Creation	6%	63%	0%	35%	86%	57%	41%	26%	61%
Helping: Knowledge Transfer	6%	13%	0%	17%	29%	29%	0%	9%	19%
Troubleshooting communication	0%	0%	0%	4%	0%	0%	6%	1%	2%
Communicating confusion over instructions	0%	0%	0%	0%	0%	0%	0%	0%	0%
Blocked collaboration	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 8.5: This table shows the percentage of data points for the assembly of RoboLoom Version A and Version B creel that were coded with each code.

We ran two-tailed t tests of unequal variances on our results at a significance level of 0.05. We found that 'Cooperation – Different steps' decreased from Version A (M = 71.5%, SD =

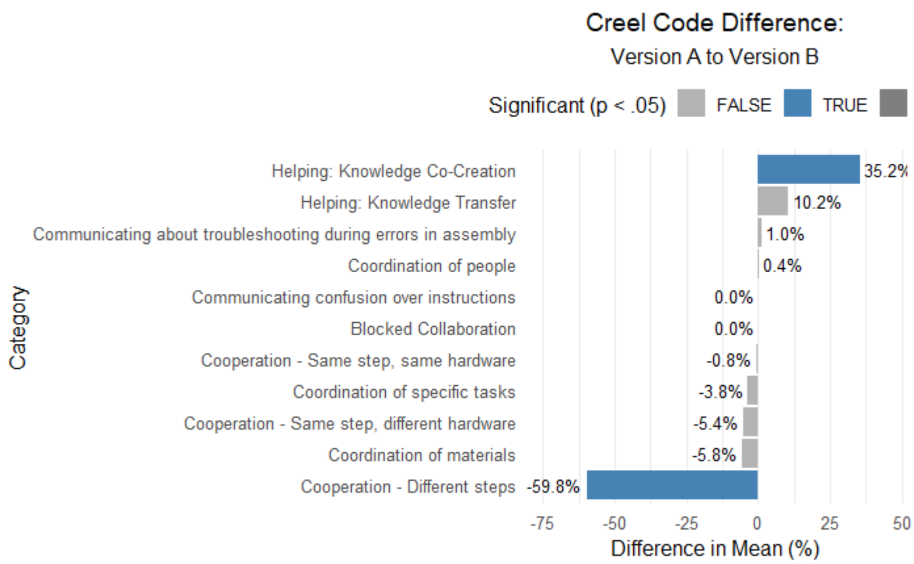


Figure 8.8: The percentage difference of the codes between building RoboLoom Version A and Version B for the steps involving building the warp beam. The codes shown in blue reached statistical significance at a p of 0.05.

20.5%) to Version B (M = 11.7%, SD = 20.2%), t(4.50) = 3.85, p = .015. Additionally, 'Helping: Knowledge Co-Creation' increased, though not significantly, from Version A (M = 26.0%, SD = 29.0%) to Version B (M = 61.3%, SD = 22.8%), t(4.94) = -1.80, p = .132.

We did not observe a difference in cooperation on the same step, different hardware between versions of the creel. This suggests that the cooperation on different steps was not replaced by participants trying to cooperate more on the same step.

8.5.4 Frame - Specificity

Based on prior research [103] (Chapter 6), we hypothesized that the changes in the frame building instructions of RoboLoom version B, being less specific, would lead to more communication to help co-create knowledge. The percentages of coded data are shown in Table 8.6.

We ran two-tailed t tests of unequal variances on our results at a significance level of 0.05. We found that 'Helping: Knowledge Co-Creation' increased, though not significantly, from Version A (M = 29.0%, SD = 27.4%) to Version B (M = 56.7%, SD = 12.3%), t(4.36) = -1.79, p = .142. Additionally, we saw 'Cooperation – Different steps' decreased, though not significantly, from Version A (M = 48.8%, SD = 27.4%) to Version B (M = 10.0%, SD = 17.3%), t(4.47) = 1.86, p = .128. We additionally saw near-significance testing with the code 'Communicating about troubleshooting during errors in assembly'. This code increased, nearly significantly, from Version A (M = 0.0%, SD = 0.0%) to Version B (M = 14.0%, SD = 6.0%), t(2.00) = -4.04, p = .056. The decreased specificity of the instructions and hardware for the frame led to participants building Version B of RoboLoom, leaving out the mounting pieces for subsystems of assembly later in the instructions. Though there were reminders to add these mounting parts, participants more often forgot them when there was no guiding video.

Code	RoboLoom Version A				RoboLoom Version B			Average	
	G1	G2	G3	G4	G5	G6	G7	V. A	V. B
Coordination of people	0%	38%	25%	0%	0%	0%	8%	16%	3%
Coordination of materials	0%	0%	13%	0%	10%	0%	0%	3%	3%
Coordination of specific tasks	43%	25%	13%	0%	0%	14%	25%	20%	13%
Cooperation – Different steps	57%	50%	88%	0%	30%	0%	0%	49%	10%
Cooperation – Same step, diff. hardware	57%	50%	13%	60%	30%	57%	17%	45%	35%
Cooperation – Same step, same hardware	29%	13%	38%	0%	0%	14%	25%	20%	13%
Helping: Knowledge Co-Creation	43%	13%	0%	60%	60%	43%	67%	29%	57%
Helping: Knowledge Transfer	0%	25%	13%	60%	10%	0%	8%	25%	6%
Troubleshooting communication	0%	0%	0%	0%	20%	14%	8%	0%	14%
Communicating confusion over instructions	0%	13%	0%	0%	0%	0%	0%	3%	0%
Blocked Collaboration	0%	13%	0%	0%	0%	0%	0%	3%	0%

Table 8.6: This table shows the percentage of data points for the assembly of RoboLoom Version A and Version B creel that were coded with each code.

8.6 Discussion

Our results suggest that modifying the specificity of RoboLoom resulted in changes to the knowledge co-creation of the participants. In RoboLoom Version B, the warp beam mount was made less specific. Participants co-created more knowledge when building this version than when building version A. We, however, did not observe statistically significantly more knowledge co-creation in the overall building. We hypothesize this is due to the relatively short amount of time spent on building the warp beam (10-30 minutes) as compared to the overall assembly (3-5 hours). Further study should be done to understand if the overall knowledge co-creation could change with more data or less specificity in the design elsewhere.

Our results also showed that participants cooperated on different steps more during the Version A build than the Version B build. This suggests that the more parallelizable hardware of Version A's creel resulted in more cooperation on different steps. There was also statistically significantly more coordination of people during Version A's build. This again suggests that the more parallelizable hardware of Version A's creel resulted in more coordination of people. This builds upon prior results, not showing coordination differences. This suggests that the size of the overall system and the number of members in a group could influence the coordination of people as well.

The results additionally show an unexpected increase in communication about troubleshooting. The less specific instructions lead to more errors and participants forgetting the mounting pieces during assembly. This then later led to communication about troubleshooting this error. Further study should be done to understand if this effect can reach statistical significance and what impact it has on groups' collaborative learning, frustration levels, or other attitudes.

Overall, the sample size for this study is relatively small, with only four and three groups

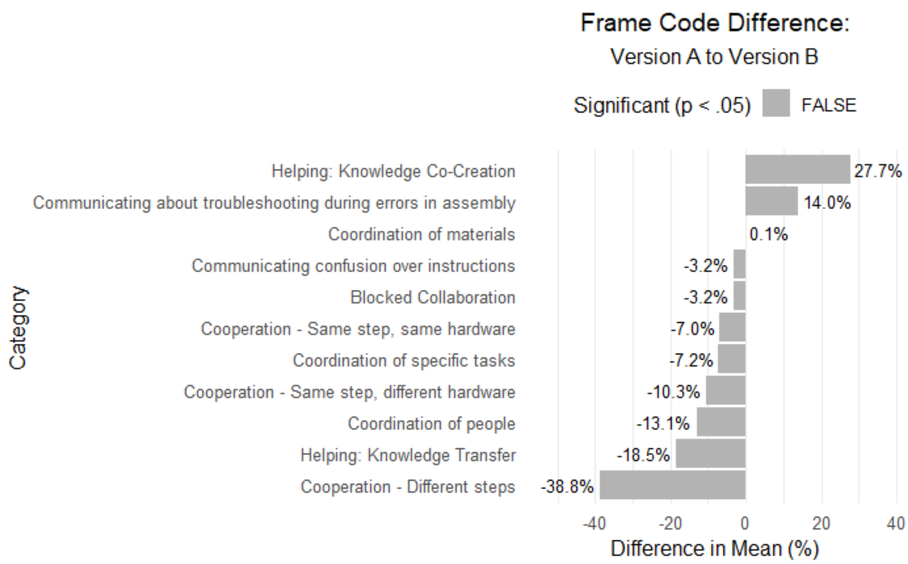


Figure 8.9: The percentage difference of the codes between building RoboLoom Version A and Version B for the steps involving building the frame. The codes shown in blue reached statistical significance at a p of 0.05.

for each version. The populations of participants, while diverse across majors, are not diverse in background or age. Further studies should be done to see what the effects of these changes to the RoboLoom hardware are on larger, more diverse populations.

Additionally, this study is limited to the observation of these collaborative behaviors. We observed that there was more knowledge co-creation, but we did not study the effect of these collaborative behaviors on groups' collaborative learning, attitudes towards collaboration, or collaborative skills. Further study should be done to understand the effect of the collaborative behaviors, so design recommendations can be made for beneficial collaborative assembly for engineering learning and collaborative skills.

8.7 Conclusions

In this Chapter, I presented a study done on the assemble of two versions of RoboLoom: Version A - designed for speed of assembly, and Version B - designed for collaborative assembly. We found that the less specific features of Version B led to more knowledge co-creation during assembly. Additionally we found the more parallelizable hardware of Version A led to more cooperation on different steps and more coordination of people.

Though promising, our results are limited. Future studies should focus on studying these effects on larger, more diverse populations. Additionally, future research should also study the effects of increased knowledge co-creation and increased cooperation on different steps and coordination of people on groups' collaborative learning and attitudes.

Chapter 9

Discussion

In this thesis, I have presented work that supports the existence of design principles that influence human-human collaboration during assembly tasks. In the following sections, I discuss these design principles and how they were shown to shape collaborative behaviors within the 3Cs framework.

Figure 9.1: A quick reference for the 3Cs framework.

9.1 Specificity

9.1.1 Coordination

Initial studies suggested that specificity may create the opportunity for participants to coordinate people and specific tasks. Single participants did ask for help when the hardware was not specific, and thus required more knowledge or problem-solving to assemble. However, we did not see this in all cases of non-specific hardware. This suggests there are other influences on when participants coordinate people, but confirms that it is possible to do so when hardware is non-specific.

Participants also were observed coordinating specific tasks when the hardware was non-specific. However, they did not always coordinate the specific tasks, especially in the controlled studies. This suggests that while the opportunity was made available by the hardware, there were other factors influencing the participant's final decision to coordinate or not.

Overall, while we see that it is possible to coordinate when hardware features are less specific, we do not always see this behavior from participants. This suggests that other factors (environmental, personal, etc.) may have more influence in determining whether participants take the opportunity to coordinate when hardware is non-specific.

9.1.2 Cooperation

Throughout our studies, we did not observe the specificity of the hardware to have a direct influence on the cooperation of the participants. This aligns with our hypotheses that the specificity of the hardware affects mostly the problem-solving aspects of assembly, thus not having a direct effect on the physical aspects of assembly.

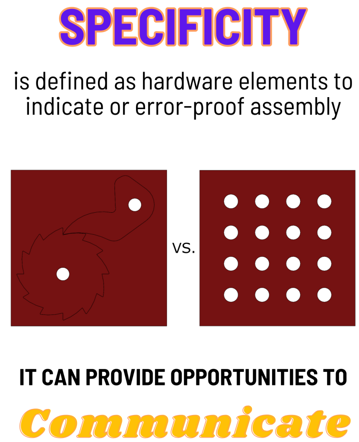


Figure 9.2: A quick reference for the Specificity design principle.

9.1.3 Communication

We found strong evidence that the specificity of the hardware shaped communication between participants. When specificity was reduced, we observed that participants co-created knowledge more. This aligns with our hypotheses that decreasing specificity would lead to reduced certainty in assembly, and that this uncertainty would lead participants to discuss and problem-solve more. The evidence across in situ testing (Chapter 8) and controlled studies (Chapter 7) is consistent, suggesting that specificity was a strong enough factor to influence participants' communication.

9.2 Parallelizability

We did observe that the parallelization of hardware, while providing the opportunity to split tasks, did not always act as a trigger for attempting to parallelize. We believe that parallelization is best used in moderate amounts, with forcing functions to continue collaboration when split apart (through the use of other design principles) and forcing functions to come back together (through the use of serialization).

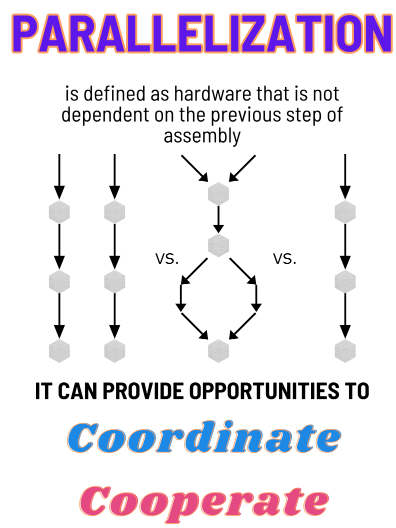


Figure 9.3: A quick reference for the Parallelizability design principle.

9.2.1 Coordination

Throughout our studies, when participants were attempting to parallelize tasks, we observed them coordinating people in order to do so. This always happened as a result of teams attempting to optimize assembly, even if they could not do so. This leads us to believe that while parallelization provides the opportunity for more successful coordination, it is not the trigger for teams to attempt coordination.

We also observed that in larger systems, there were more opportunities to coordinate people to parallelize than in the smaller lab study. This suggests again that the number of tasks was a larger influence on the coordination attempts than the parallelization of the hardware. Further studies could be done to understand the influence that parallelization has on successful coordination and participants' willingness to attempt coordination to parallelize again. For example, would a non-parallel system lead to failed coordination on the first attempt, and thus decrease the participants' likelihood of trying to coordinate again?

9.2.2 Cooperation

In all the studies we conducted, parallelization had a clear and strong influence on the cooperation of different steps. When the hardware did not have dependencies for assembly, participants were able to break into smaller groups and work on different tasks simultaneously.

We also observed a slight increase in other forms of cooperation when participants were not cooperating on different steps. This could be due to other factors in the environment, such as participants attempting to be quick in the assembly or being predisposed to attempting cooperation. However, the total amount of cooperation did change with the difference in parallelizable hardware. Further study is needed to understand if there are trade-offs between different types of cooperation and how hardware design elements can influence these trade-offs. Further study is also needed to understand what influences on learning and cognition come from different forms of cooperation.

9.2.3 Communication

We observed an inverse relationship between parallelization and communication when groups split up into individuals doing assembly tasks. When groups parallelized assembly, resulting in participants working individually on a task, there was a drop in communication related to assembly. Further study is needed to understand the effect that this has on learning or collaborative skill-building.

9.3 Physicality

We observed many factors of physicality that had different impacts on collaboration.

9.3.1 Coordination

Across our studies, we saw that an increase of parts requiring simultaneous manipulation led to an increase of coordination on specific tasks. When participants needed to manipulate items simultaneously, they often coordinated to do so, suggesting that physicality is a strong enough influencing factor to increase coordination.

9.3.2 Cooperation

Across our different studies, we did see that there was an increase in cooperation on the same step. In the more controlled study, the difference between cooperation on the same step but different hardware and the same step same hardware became harder to distinguish. The same hardware was defined as the same contiguous object, so once a part was attached to the overall object, it became part of the same hardware. Due to this, the distinction between the same hardware and different hardware is harder to extract. However, we still did see an increase in cooperation on the same step when hardware required more simultaneous manipulation.

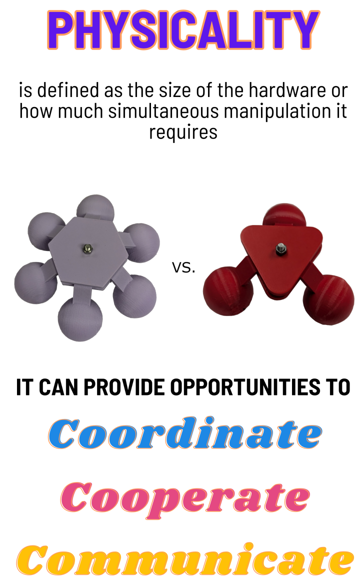


Figure 9.4: A quick reference for the Physicality design principle.

9.3.3 Communication

Throughout all of our different studies, we did see a slight increase in communication, either to co-create or to transfer knowledge. The instances of communication often overlap with coordination of specific tasks when participants were strategizing how to complete the task; at the same time, they were coordinating who would do what.

9.4 Difficulty

Difficulty can be defined as either mental or physical difficulty and has different impacts on collaboration, as described below.

Figure 9.5: A quick reference for the Difficulty design principle.

9.4.1 Coordination

Throughout our different studies, we did see a slight increase in coordinating specific tasks. However, these differences had high variance and did not reach statistical significance. These results suggest that the difficulty of the hardware creates an opportunity for coordination a specific tasks but does not have a strong influence over coordination. This could be due to other environmental factors having a higher impact. Further study is needed to determine the degree to which difficulty influences coordination.

9.4.2 Cooperation

We did not originally hypothesize that the difficulty of the hardware would influence cooperation. However, we did see slight trends in cooperation when the difficulty of the hardware varied. In the case of cooperation on the same step but different hardware, we witnessed participants simultaneously trying to problem solve by physical manipulation of different objects. We additionally saw a bit of a trend for cooperation on the same steps, same hardware, where participants would attempt to physically help when they were not contributing to problem-solving or mental assembly tasks. Further study is needed to determine if this hypothesized reasoning is the cause of this slight trend.

9.4.3 Communication

Across our various studies, we observed that the more difficult the hardware, the more participants communicated to co-create knowledge. When participants did not know how to complete a particular assembly task, they would communicate to problem-solve together. Further study is needed to determine if there are other causes for this increase, such as a predetermination for collaboration.

Across our studies, we also observed instances with difficult hardware where knowledge transfer occurred. In these instances, knowledge transfer would occur when one participant knew the solution to the problem and felt the need to share it with another. The difficulty of the hardware captured how familiar participants were and created a gap in knowledge between the participants. However, it is unclear if the difficulty caused the participants to share the knowledge with each other or if this was due to other environmental factors. Further study is needed to determine what factors encouraged participants to transfer the knowledge to each other in these cases.

9.5 Repetitiveness

Repetitiveness is simply defined as having repeated hardware elements and can influence collaboration opportunities as described below.

Figure 9.6: A quick reference for the Repetitiveness design principle.

9.5.1 Coordination

Across our studies, we saw that participants, when faced with repetitive tasks, would ask their teammates for help in completing these repetitive tasks, splitting the work. This gave the opportunity to coordinate people. As the participants worked, there were opportunities for trying to optimize assembly, and participants shared them, coordinating how they would complete their shared repetitive task. Further study is needed to understand the exact breakdown of repetitive tasks that trigger a participant to split the task without it becoming frustrating or tedious.

9.5.2 Cooperation

Across our studies, we observed an increase in cooperation
on the same step, different hardware, when tasks were repetitive, and participants chose to split the tasks. This happened in instances where there were as few as 4 repetitions of hardware. Further study is again needed to understand the exact breakdown of repetitive tasks that trigger participants to split the task without it becoming frustrating or tedious.

9.5.3 Communication

Throughout our studies, we observed participants communicating to co-create knowledge or transfer knowledge during repetitive tasks. This often coincided with the coordination of specific tasks. Participants would co-create knowledge as they figured out initially how to do the repetitive task together. Then, as participants discovered faster or better ways to do the repetitive task, they would transfer this knowledge to help their group assemble more optimally.

Chapter 10

Conclusions

This thesis aimed to explore three major questions:

RQ1. What features of a robotic system help facilitate collaborative assembly?

RQ2. What features of a robotic system help facilitate collaborative experiences in a classroom?

RQ3. What are the design principles for assemblable devices designed for human-human collaboration during assembly?

In this thesis, I have answered these three questions through a number of studies. RQ1 is answered through the study listed in Chapter 5 and Chapter 6. RQ2 is answered in the study in chapter 5. And RQ3 is answered in the studies in Chapter 6, Chapter 7, and Chapter 8.

RQ2 was initially explored in Chapter 5 through the pilot testing of RoboLoom in an undergraduate class designed to teach weaving, math, and engineering. Through our observations and analysis, we saw that students in the course were able to collaboratively explore engineering concepts through the engineering assembly. However, some students reported that they felt as if they were "just following instructions", leading us to believe more open-ended features would better support collaborative learning.

In Chapter 6, we answered RQ1 through the exploration of the assembly of two looms: RoboLoom and the Ashford Shaft loom. These kits were chosen as they're both designed for assembly, but for a different number of end users. Our observations and analysis revealed that different hardware designs facilitated distinct collaborative actions in the loom-building process. We identified five key categories of hardware features: repetitiveness, specificity, difficulty, parallelizability, and physicality.

DP1. **Repetition** of hardware elements provided opportunities for coordination of people and tasks, cooperation on the same step but different hardware, and communication to transfer knowledge.

DP2. **Specificity**, or features to indicate or error-proof assembly, influenced the opportunity for communication to co-create or transfer knowledge. Hardware that was highly "error-proof" reduced the need for assistance, whereas open-ended hardware often required external help from instructors, limiting peer-to-peer assistance. Mid-range specificity fostered more collaborative actions through co-creation or transfer of knowledge.

DP3. **Difficulty**, both mental and physical, mirrored the effects of specificity. An optimal level of difficulty encouraged communication to co-create or transfer knowledge and the coordination of people, materials, and tasks. However, tasks that were too easy or too challenging reduced such interactions.

DP4. **Parallelizability** in hardware created the opportunities for cooperative behaviors.

DP5. **Physicality**, both in size and the requirement of simultaneous manipulation, created opportunities for coordination of people and tasks, cooperation on the same hardware pieces, and communication to transfer knowledge. Smaller hardware was observed to prevent physical collaborative actions.

From this, we conducted two further studies to answer RQ3. In Chapter 7 we test the causality of changing these design principles on collaborative behaviors and found stronger evidence that:

DP2. **Specificity** inversely influenced the amount of knowledge co-creation. Less specificity led to more knowledge co-creation.

DP3. **Difficulty** proportionally influenced knowledge co-creation. Increasing difficulty increased the knowledge co-creation. Difficulty also inversely impacted cooperation on the same hardware and proportionally impacted cooperation on different hardware.

DP4. **Parallelizability** proportionally influenced the amount of cooperation on different steps and inversely influenced the amount of helping through knowledge transfer or co-creation. The more parallel an assembly, the more cooperation on different steps, but the less communication to help through knowledge transfer or co-creation.

DP5. **Physicality** in the form of simultaneous manipulation proportionally impacted cooperation on the same step and knowledge co-creation. The more simultaneous manipulation needed, the more participants cooperated on the same step. More simultaneous manipulation also led to knowledge co-creation to coordinate how the task would be completed.

We then compared two versions of RoboLoom, one designed to be more specific and parallelizable (Version A), the other less specific and mid-parallelizable (Version B). We found that Version B building had more communication with knowledge co-creation and less cooperation on different steps.

In this thesis, I contribute three major results:

1. **RoboLoom**, a novel robotic Jacquard loom kit designed for collaborative education, validated through a pilot course teaching mathematics, engineering, and weaving.
2. **A codebook** for identifying collaborative behaviors based on the 3Cs framework.
3. A set of **hardware design principles** for eliciting collaborative behaviors during human-human collaborative assembly.

Chapter 11

Future Work

11.1 Future Work

Throughout this thesis, I have listed specific future work to further the development of the design principles and understand more about the causality between hardware design and collaborative behaviors. In this chapter, I also discuss the connection of these design principles back to cognitive theories to further support a theory of why they shape collaboration. Additionally, I discuss the differences in how these design principles perform in the highly controlled study versus the classroom studies and how this contributes to a larger design framework.

These future directions (listed in Section 7.6.1, Section 8.7, and below) encompass further causal studies to reach statistical significance, applications of all design principles in real-world contexts, studies of the interactions of these design principles in a single system, and applications of these design principles to other areas (such as scientific exploration).

Beyond applications to other areas, I believe this work holds interesting insights into psychology research on collaboration itself. In this thesis, I have drawn inspiration from psychology research into cognitive theories and how they impact collaboration. I believe that further studies into collaborative actions through hardware can give us insights into how people collaborate.

Given the findings of our studies showing different strengths of influence of hardware, I believe there are other environmental or personal factors that also influence collaboration. Understanding all of these factors holistically might give us a better idea of when this framework is useful and when we should call on other design elements to influence collaboration.

I believe interdisciplinary study with psychology experts could yield insights that improve this framework and also our understanding of people's cognition. For example, studying the predisposition of participants to collaborate and understanding how hardware changes the collaboration they're willing to do. If a person is not willing to collaborate, can hardware design draw on personality traits that would overcome their unwillingness, such as curiosity? I believe the study of these principles is informed by interdisciplinary collaboration, and also can help inform the other fields it leans on for insights.

11.2 Hardware Design Principles and Cognitive Theories

In this thesis, I have analyzed collaborative actions through many cognitive lenses to understand the diverse ways in which people think while collaborating on assembly tasks. The hardware design features found in our three studies are informed by these cognitive theories, but do not yet fully capture all of them. Further study is needed to link the hardware design with these different forms of cognition. This would provide causal evidence of how hardware influences cognition. Such an understanding of how hardware interactions shape cognition would allow us to more accurately predict how people will behave, with applications reaching to HRI, and to more accurately understand how people learn, with applications reaching towards education.

11.2.1 Sociocultural Theory and Conversation Analysis

Both Sociocultural Theory and Conversation Analysis posit that social interactions have an influence on cognition [16, 42, 77, 104, 105]. These exchanges are captured in the communication codes of our developed codebook (Section 6.3.3).

Hypothesis Through these theories, I hypothesize that when there is more conversation, the participants are more likely to learn from the assembly as they are engaging the social aspect of cognition.

Conversation can be influenced by many of the hardware design principles. Specificity and difficulty give participants an opportunity to talk and fill gaps in their knowledge as they problem-solve the assembly. Hardware that brings participants physically together can also create the opportunity for conversation, such as physicality requiring simultaneous manipulation. Repetition can also give the opportunity for shared tasks, creating a shared topic of conversation. Conversely, parallelizability can split participants and take away their common tasks, taking away an impetus for conversation. However, planned serialization could force participants to come back together and converse again.

Future Work Further study is needed to determine if these hardware elements, creating the opportunity for conversation, have an impact on cognition about the assembly task and if that impact affects learning. Another interesting line of study around the social impacts of hardware design is with regard to repetitive hardware. Often in the case of repetitive hardware, once participants were comfortable completing the task, they began to socialize with their teammates. The repetitive hardware created an opportunity for a small formation of community. Further study could explore the impact of this socialization opportunity on the team dynamics going forward in a class.

11.2.2 Grounded Cognition Theory and Activity Theory

Both Grounded Cognition Theory and Activity Theory posit that actions, including physical manipulation, impact the cognition of the participant [9, 10, 11, 26, 27, 52, 67, 81, 82, 115].

Hypothesis I hypothesize that different hardware designs will cause different actions, and these actions will have different influences on cognition. For example, in Chapter 8, I designed the warp beam of RoboLoom to be less specific. This required a different set of interactions from the participants, selecting where they would mount the warp beam, testing their mounting position, and trying again. I hypothesize that this different set of physical manipulations would cause the participants to think differently about the warp beam assembly, and how this ties back to the tensioning mechanism of the loom and eventually the abstract concepts of tensioning mechanisms in general.

Future Work Further study is needed to determine if specific actions were more impactful on cognition due to the design of the hardware. This study should investigate if this change in specificity changed how participants cognate about the assembly, and if this change in cognition impacted their learning, and if they used these actions to ground abstract concepts.

Additionally, further study should be conducted to understand how the physicality of the activity impacts cognition. For example, does the size of the hardware impact the actions being taken enough that it changes how the participants think about them? Research in grounded cognition theory has shown differences in cognition when motor functions are employed, but there is not yet enough evidence to conclude if the kind of motor movement changes cognition [31, 37]. Thus, further study should examine if students remember concepts differently when presented with very small hardware (requiring fine motor movement), medium-sized hardware (requiring midline motor movements), and larger hardware (requiring gross motor movements).

11.2.3 Distributed Theory

Distributed Theory posits that cognition is held not only in individual minds, but in the processes and artifacts in the group [41, 44, 95, 104, 105, 113]. This is loosely reflected in the case of parallelization and repetition. In parallelizable tasks, the full knowledge of the assembly is not held in one individual's cognition, but is represented in the final stage through the assembled device. Similarly, this is seen in repetitive tasks as well. In some instances, a single item was assembled and left for another team member to copy at a later date in assembly. This shows the distribution of knowledge in the assembled artifacts in the system.

Hypothesis I hypothesize that storing and passing information through hardware will lead to more feelings of connectedness during group assembly than without any interaction between participants.

Future Work Future studies should investigate the impact that this distribution has on team dynamics and collaborative learning. A future study could achieve this through testing serial assembly of the same object between group members and studying the impact of these conditions on team dynamics and cognition. For example, in building the same Lego set, one after another, participants could be allowed to share no information between assembly sessions, share information only verbally between sessions, share information only through the physical Legos, or share information both through physical Legos and verbal communication. Varying the hardware design principles of the Lego set, like adding repetitive elements, could give insights into what information is seen as important or easy to share through physical objects.

11.2.4 Communities of Practice

Communities of Practice are defined as a group of people sharing a broader goal and communicating information to collectively get better at achieving that goal [53, 56, 105, 113, 115]. One hardware design principle that ties back into this is repetitiveness. When tasks were repetitive, participants formed a small-scale version of a community of practice. Participants shared the broader goal of finishing the assembly task and communicated information to collectively get better at achieving this goal. This opportunity for conversation could additionally be analyzed through conversation analysis or sociocultural theory.

Hypothesis I hypothesize that the small-scale communities of practice will increase group belonging and have a positive impact on the learning experience.

Future Work Future research should identify the impact on group dynamics and learning that these small-scale communities of practice have. A future study could be designed to understand the effects of the directness of a shared task on the community's dynamics and cognition. For example, a pair could be assigned two distinct tasks, such as washing dishes and folding clothes, in one condition. In the second condition, the pair could be assigned two related, yet distinct tasks, like washing dishes and drying dishes. In the third condition, the pair could be assigned the same repetitive task, like folding clothes. Varying how shared the task is, from independent to sharing a larger goal, to sharing motion-level actions, could give insights into the impact of these actions on cognition and group dynamics.

11.2.5 Coordination Mechanisms

Coordination Mechanisms is a theory that focuses on the analysis of the cognition of a group of people through the object that concertizes the results of their articulation work [100]. While the hardware design principles gave participants opportunities for coordination when present in our study comparing RoboLoom and the Ashford loom (Chapter 6), no elements of the studied systems explicitly concretized this work. Coordination tasks for these systems were relatively small and often short-term, task-oriented, thus limiting the potential need for any coordination mechanism.

Hypothesis I hypothesize that the concretization of this coordination work will influence how the participants think about the task. I speculate that the participants will learn more as the experience will have a larger and more widespread impact on their cognition.

Future Work Future studies could address this in longer-term collaborations, for example, weaving on and learning with RoboLoom in a class, or on larger assembly systems. I hypothesize there would be more opportunity for a coordination mechanism when weaving on RoboLoom (trading off between weaving, controlling the motors, and tensioning), and that other hardware design elements besides those studied could contribute to concretizing the coordination necessary to collaboratively weave on RoboLoom. Specifically, these physical coordination mechanisms would need to be configurable such that the group would be prompted to coordinate, then store the results of their discussion in these mechanisms.

In a different system, this could take the form of re-configurable control hardware. For example, a team of three attempting to control 12 degrees of freedom of a system with 6 re-mappable sliders could determine who controlled what degrees of freedom and store the information in the sliders by mapping them to that degree of freedom.

11.3 Design Framework

11.3.1 Refinement

In this thesis, I have established that there are hardware design elements of systems that affect collaborative actions. I have studied these causally in controlled environments and within larger systems in real-world environments. However, I have seen a shift in behavior between controlled lab studies and the in situ classroom studies. This suggests that the hardware design elements are not entirely distinct and can influence each other when manipulated in a larger system. I propose that the design features influence each other in larger systems in a way that requires further study. Such a study should explore if hardware varied along two axes of the design framework simultaneously impacts collaborative behaviors differently than varying one design principle at a time.

I also found that there were different levels of effect on collaborative behaviors for certain design principles, suggesting there could be a weaker influence on behaviors depending on different environmental or personal factors. This suggests a larger framework of collaborative influences that hardware design fits into. Further studies should establish these effects and determine the differences between hardware that provides opportunity for collaboration and hardware that necessitates collaboration. I hypothesize that many of the unexpected influences are a result of the hardware design's effect belonging somewhere on the scale from no influence over, to presenting an opportunity for, to necessitating collaboration.

In relation to environmental factors, there were interesting observed tradeoffs between different forms of cooperation. Further study is needed to understand if there is an inherent level of cooperation that participants strive to achieve and if this influences when they choose cooperative behaviors as much as hardware influences what cooperative behaviors they choose.

Finally, I observed an interesting trend in the influence of mental challenge over physical behaviors and physical challenge over mental behaviors. When hardware was mentally challenging, such as was the case with non-specific or difficult hardware, participants sometimes tried to physically problem-solve more (e.g., testing if hardware opened, assembled in certain ways, etc.). Conversely, when hardware became physically challenging, as was the case with physicality or difficulty, participants often turned to more mental solutions, like communication and coordination, to ease the problem. Further study on this effect of the physical/mental crossover is needed to understand how this can influence collaboration and learning.

11.3.2 Application

The main application of this framework throughout my thesis has been on RoboLoom as an explicit example of an assemblable engineering education kit. RoboLoom provides a good example application of these hardware design principles, as it is a physical device that was designed to be assemblable. The broader application of education is ideal for the application of this framework, as the primary goal of education is for students to learn. To improve learning, we can shape and understand cognition. As this framework was informed by cognitive theories, it has a theoretical basis to hypothesize that the hardware design principles will shape cognition. However, as previously discussed, empirical evidence of this is still needed. Engineering as a domain provides an ideal application of this design framework, as it can involve building or working with physical systems and is inherently interdisciplinary, thus it often requires collaboration.

Other applications of this design framework include areas where human teams interact with or through physical systems, not just in assembly. This could include applications like disaster robotics or scientific exploration. In these cases, human teams are using robotic tools (sometimes teams of robotic tools) to collaboratively explore areas and find information. Having a hardware design that is more cooperative and encourages communication might aid in the goals of these tasks. Further research should be done on how these design principles could apply to these systems and if this would be a beneficial use of the design framework, as hypothesized.

One drawback of this framework is that it prioritizes collaborative behaviors over other outcomes, such as time, cost, or other efficiency metrics. In the case of disaster response, this framework would have to be adjusted to consider and prioritize time as a factor. In the case of scientific exploration, the framework may need to be adjusted to consider cost as a factor. In situations such as manufacturing or service industries, this framework may not be appropriate at all, as cost and time are the main priorities of the application, and collaboration is rarely considered.

Appendix A

Collaboration Coding Scheme

This appendix details the 3Cs Collaboration Coding Scheme that we developed over the course of this thesis. We then combined observations during an assembly task with the 3Cs model of collaboration to specifically capture the nuance between the modes of collaboration (coordination, cooperation, and communication). From this, we defined four code categories, each with specific codes, grounded in both data and theory. The codes are meant to break down the 3Cs framework into specific, measurable behaviors during a collaborative assembly task. These codes are not mutually exclusive and often appear together as coordination, cooperation, and communication create collaboration as a whole. We also include a code for instances where collaboration was attempted, but not completed, due to hardware design. The codes are:

1. *Coordination* - participants are engaging in behavior to manage others, activities, or resources needed to assemble the loom (e.g. defining group roles, gathering loom materials, etc.)
   (a) *Coordination of People* - Participants are managing the distribution of labor (e.g. the group decides to have participant 1 do task A and participant 2 do task B).
   (b) *Coordination of Materials* - Participants are managing the materials for the task (e.g. participant 1 calls out materials while participant 2 finds them).
   (c) *Coordination of Specific Tasks* - Participants are managing how and when to accomplish their activities.
2. *Cooperation* - participants work separately on tasks that both contribute to the shared assembly goal (e.g. building separate parts of the loom concurrently)
   (a) *Cooperation - Different Steps* - Participants or participant groups work on two steps of assembly concurrently (e.g. building the loom frame and mounting the loom's motors).
   (b) *Cooperation - Same Step, Different Hardware* - Participants are working on the same step of assembly, but interacting with distinct pieces of the loom (e.g. participant 1 builds a shaft of the loom, and participant 2 builds another).
   (c) *Cooperation - Same Step, Same Hardware* - Participants are working on the same step of assembly and interacting with the same loom materials (e.g. participant 1 holds pieces together while participant 2 secures them).

3. *Communication* - Participants discussing ideas and confusion during the assembly process.
   (a) *Helping: Knowledge Co-Creation* - Participants are creating knowledge about the loom's assembly with each other. This requires contributions from all participants involved to build the knowledge together.
   (b) *Helping: Knowledge Transfer* - One participant transfers knowledge to another with the goal of aiding the second in the task.
   (c) *Communicating About Troubleshooting* - Participants talk through troubleshooting when there is an error in the assembly process (e.g. participant 1 determines a part is backward and the group discusses how to fix it).
   (d) *Communicating Instructions Confusion* - Participants communicate when the instructions are unclear (e.g. Participants determining which pieces the instructions refer to).
4. *Blocked Collaboration* - A participant attempted collaboration but did not succeed. This does not include participants who are off task.

Examples of the codes are shown in Table A.1.

Table A.1: Examples for each code from the study described in Chapter 6.

Code	Example Video Description
Coordination of People	P1 suggests splitting into teams of two stating the sections are individual. Group agrees. P1 suggests she start with the frame because she hasn't done it. P2 suggests she take heddles. P1 suggests P2 can work alone or with someone else because there are four sets of motor frames.
Coordination of Materials	The group coordinates passing the ruler around.
Coordination of Specific Tasks	P3 and P2 talk about the instructions and needing to make sure certain elements are certain ways. Some conversation on how to achieve this.
Cooperation - Different Steps	P5 works on the loom 8020, screwing things in. P6 preps the foot holders with screws.
Cooperation - Same Step, Different Hardware	P6 and P5 each putting in height beams on different ends of the RoboLoom.
Cooperation - Same Step, Same Hardware	P5 holds, P7 and P6 screw in.
Helping: Knowledge Co-Creation	P6, P5, and P8 talk about where the length beam on the RoboLoom goes. P6 suggests they watch the video. P8 and P5 do and talk it through. P5 and P6 talk about doing the cross beams first.
Helping: Knowledge Transfer	P4 watches the instructional videos with P1. P1 helps P4 understand the next steps.
Communicating About Troubleshooting	P8 looks at the bottom and notices one beam is shorter than the other. The group talks through this mistake and tries to figure out what they need to do to fix it.
Communicating Instructions Confusion	P7 asks P6 about the instructions. P6 explains the instructions to P7. P6 uses the materials to indicate her meaning.
Blocked Collaboration	P4 asks if P3 feels redundant. P3 says yes a bit. P3 doesn't have a hands-on task right now.

Appendix B

Loom Assemblies

B.1 RoboLoom

The assembly instructions for RoboLoom were presented in two forms: a written document and a series of videos. The written document can be found on the RoboLoom website at `https://sites.google.com/view/speerloom/assembly-guide`. A screenshot from one of the assembly videos with the example instruction text is shown in Figure B.1.

Figure B.1: A screenshot from the RoboLoom assembly instructions showing a step of assembly. The written text document for assembly lists this instruction as "Put the holders onto the bottom square pieces... Put a screw through the holes and each side of the foot holder and screw on a t-nut to each screw. Slide the 8020 into the holder."

RoboLoom (Figure B.2) is designed with three major subsystems: the main frame of the loom, the motors, and the tensioning system. Each of these subsystems was designed to be built separately for most of the assembly, then attached to the main frame as part of the final assembly of motors and the tensioning system. This design was chosen to allow for simultaneous work and decrease building time for students [102].

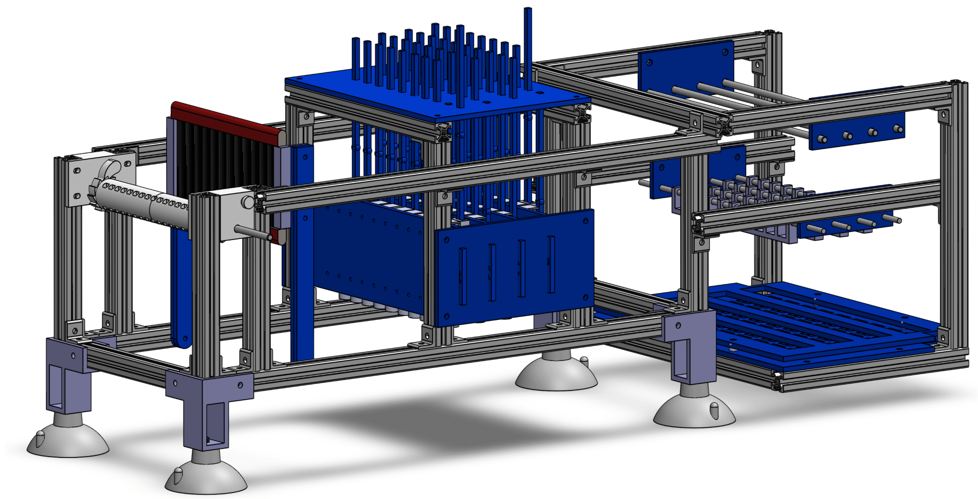


Figure B.2: RoboLoom. Designed with subsystems to be assembled in collaborative groups [102].

The main frame of RoboLoom is made from slotted aluminum extrusion and primarily held together through off-the-shelf corner brackets, t-nuts, and machine screws, except for a few custom 3D-printed brackets (Figure B.3a). The base of the main frame is assembled with custom 3D-printed foot holders that attach to the aluminum extrusion through machine screws and t-nuts. The foot holders are designed with an open path such that the aluminum extrusion can be placed in one of two orientations, as shown in Figure B.3d. However, the loom is designed such that only one of the orientations is correct. Once the base is built, the height beams are attached to RoboLoom using corner assemblies. Corner assemblies are corner brackets pre-assembled with screws and t-nuts as shown in Figure B.3c. These assemblies need to be carefully done such that the t-nut is facing the appropriate direction to slide into the slotted aluminum extrusion. T-nuts must also be carefully tightened such that they are fixed onto the screw, but not too tightly to slide into the slotted aluminum extrusion. Different lengths of aluminum extrusion are then used as the height beams to mount various components of the loom. To assemble, height beams are aligned with the corner assemblies and fixed down symmetrically on either side of the loom. Finally to ensure the frame is sturdy enough to withhold tension, a length beam is attached on each side of the loom to each of the four height beams of that side of the loom by aligning each of the corner assemblies on the height beams as the length beam is slid across the loom (shown in Figure B.3b).

The motor subsystem of RoboLoom houses the motors and heddles to actuate the yarns as shown in Figure B.4. There are 40 motors for each of the yarns in the loom. Each motor is part of a motor assembly that consists of the motor, the heddle (which holds the yarn), and a 3D-printed attachment mechanism shown in Figure B.4b. Once assembled, each motor assembly gets mounted onto a frame that holds 10 motors (shown in Figure B.4c). Each frame is then mounted to the loom simultaneously by sandwiching the frames between two slotted mounting plates attached directly to the main frame (Figure B.4d). Once in place, a heddle guide (laser cut from acrylic) is added to the loom, which requires aligning each of the 40 heddles with a corresponding hole in the heddle guide plate.

The tensioning system and warp beam of RoboLoom (Figure B.5) hold the warp yarns in the loom and ensure they are tensioned properly. These subsystems consist of the tensioning cage and the warp beam. The tensioning cage holds tensioning rods that have tensioning spacers with springs and disks in them. The rods must be mounted carefully by first aligning the rod through one side of the mount, moving the rod to the edge of the tensioning spacer, then aligning the rod with the other mount and sliding it through the second mount. The warp beam consists of a beam that is held in place with a ratchet gear and pawl, allowing the warp yarns tied down onto the beam to be rolled up in one direction, but not unrolled, so tension is held.

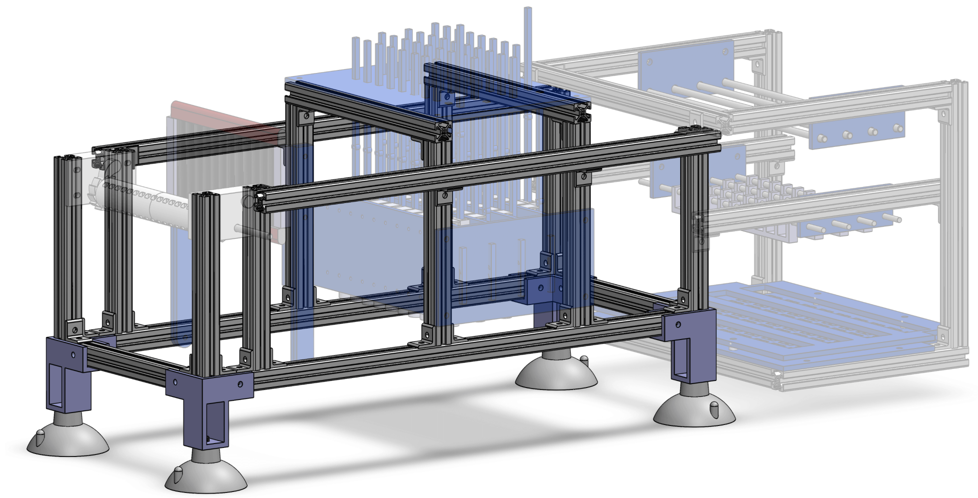
B.2 Ashford Loom

The Ashford Loom (Figure B.6) is designed for hobbyist weavers to build and weave on individually. The assembly instructions for the Ashford Loom were provided online for free by Ashford. They take the form of pictorial instructions with a few words explaining each step. An example step is shown in Figure B.7, and the manual can be found on Ashford's website at `https://www.ashford.co.nz/instructions/SS610.pdf`.

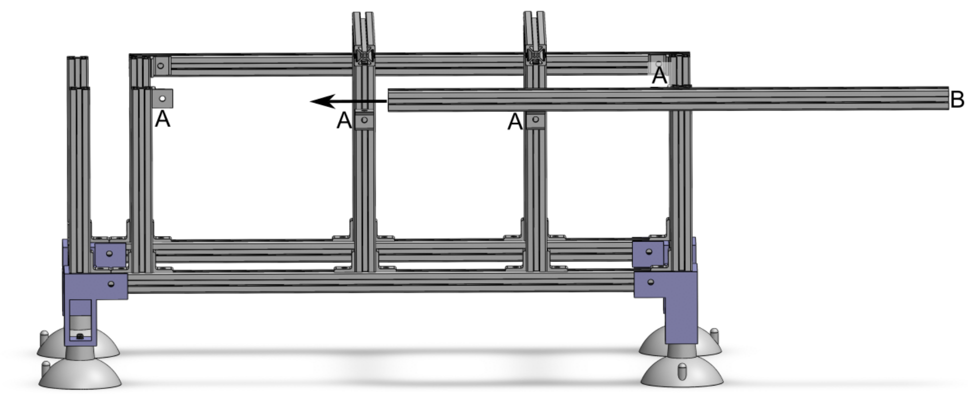
The loom is built from hardwood to ensure durability and sturdiness, and common hardware, to be accessible for building by hobbyists. It is structured with a main frame (Figure B.6c) that holds the other components of the loom as well as the warp yarns, holding the yarn under tension to allow for proper weaving.

Inside the main frame is the castle of the Ashford Loom (Figure B.6b), which is designed to hold the 16 shafts of the loom, which manipulate the yarns in the loom for weaving. The castle also has levers with ropes tied from them to the shafts to allow for manual actuation of each shaft (and consequently the yarns attached to that shaft). The shafts are built by inserting two metal bars into a piece of wood, threading on the heddles, then finishing the shaft with a final piece of wood, as shown in Figure B.7a. They are then placed in the castle and attached by threading a cord around the respective lever and securing the cord to the shaft. The shafts are then calibrated to ensure they are at the correct height for weaving. The instructions for this process are shown in Figure B.7b.

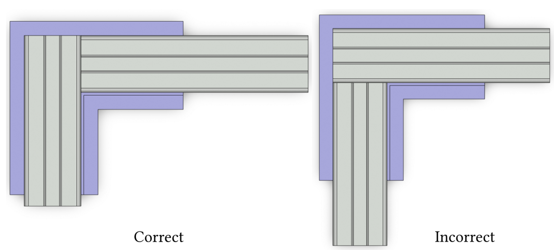
Finally, the Ashford Loom has a beater (Figure B.6d) to help weavers push back their cloth as they weave, which is suspended from the frame and the castle.



(a) RoboLoom with the main frame highlighted. The main frame consists of the aluminum extrusion used to provide the load-bearing structure to the loom, as well as off-the-shelf and custom brackets.



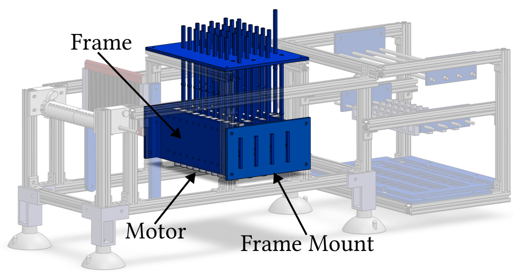
(b) The attachment of the length beam. At each of the attachment points (A), the corner assembly must be aligned as the length beam (B) is slid across the loom during assembly.



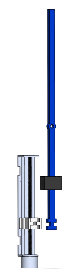
(c) A corner assembly which consists of an off-the-shelf corner bracket, machine screws, and t-nuts pre-assembled to make attachment easier.

(d) The possible configurations of aluminum extrusion in the foot holders.

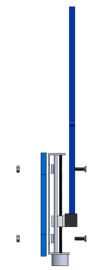
Figure B.3: Components of RoboLoom's main frame.



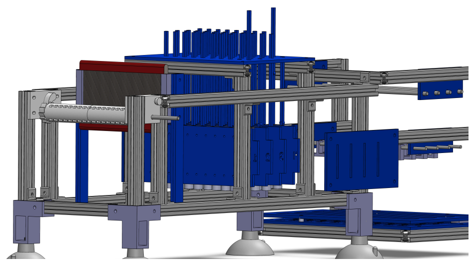
(a) RoboLoom with the motor subsystem highlighted. It consists of 10 motor assemblies mounted to each frame and the mounts for the motor frames.



(b) A motor assembly



(c) The motor attachment to the frame.



(d) The frame mount and how it attaches to the loom

Figure B.4: Components of RoboLoom's motor subassembly

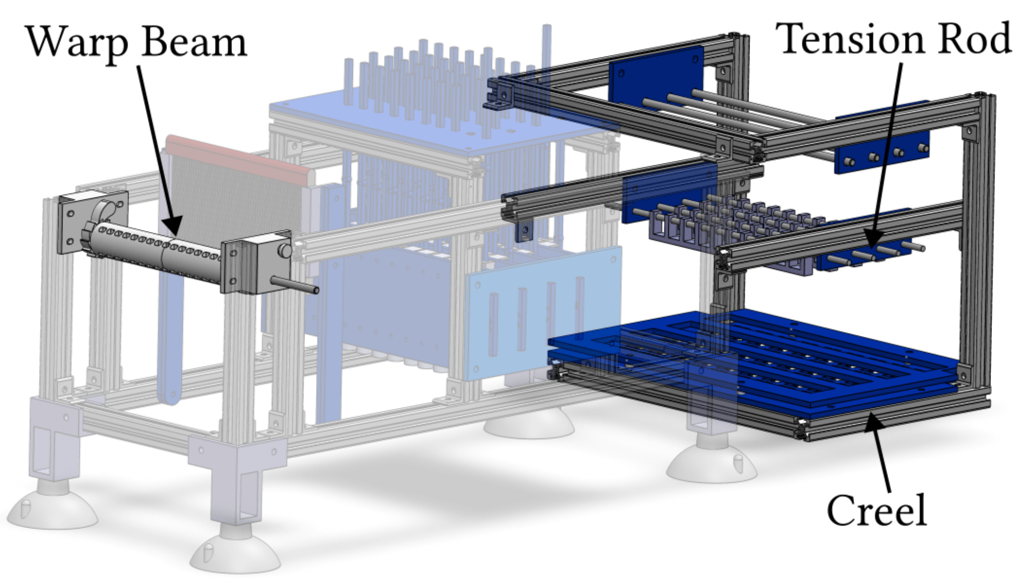
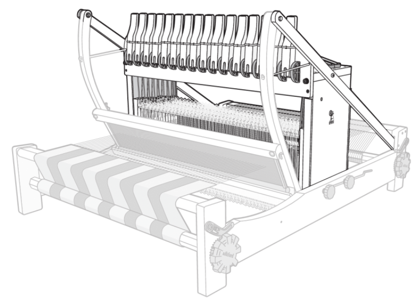
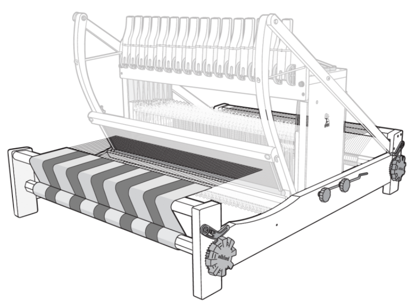


Figure B.5: RoboLoom with the warp beam and tension system highlighted.

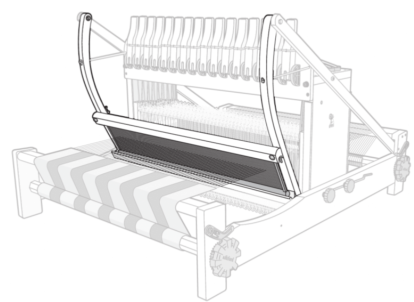
(a) The Ashford Loom [109].



(b) The castle of the Ashford Loom. It holds the shafts and levers to actuate the yarns in the loom.

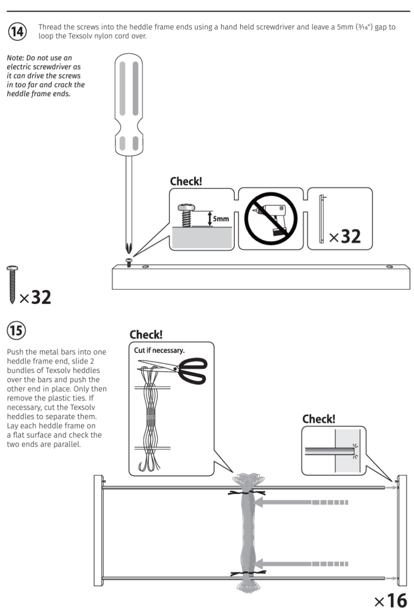


(c) The frame of the Ashford Loom. Made to hold the castle and tension on the yarns of the loom as cloth is woven.

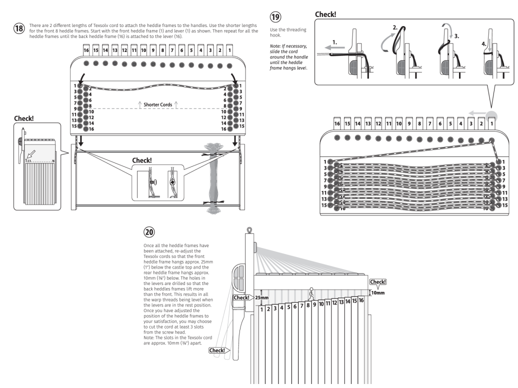


(d) The beater of the Ashford Loom, used to make cloth and suspended from the castle and frame.

Figure B.6: Images of the Ashford Loom. Images taken from the publicly available Ashford loom website at `https://www.ashford.co.nz/instructions/SS610.pdf` [109].



(a) Instructions for the assembly of shafts in the Ashford Loom. Each shaft is made as a box with two metal sides and two wooden sides.



(b) Instructions for the threading and calibration of the shafts into the Ashford loom. Each shaft is attached by a cord, then threaded through a handle, and then calibrated.

Figure B.7: Example instructions for the Ashford Loom assembly. Images from the Ashford web page, `https://www.ashford.co.nz/instructions/SS610.pdf` [109].

Appendix C

RoboLoom Course Surveys, Reflections, and Interviews

In the pilot course taught with RoboLoom, we used surveys to gather student background, reflection questions to gather students' thoughts on the assignments and work they were completing, and interviews to gather students' final thoughts on the course. The questions for each are in the following sections.

C.1 Surveys

To measure student background across different disciplines and their attitudes towards those disciplines, we administered a survey on math, engineering, arts and fiber crafts, and collaboration. The surveys are included in the sections below.

C.1.1 Math

We asked students about their experience with and attitudes towards math. We first asked "What comes to mind when you think of a mathematician?" and then "The last thing that I learned in Math was ___." Finally, we asked participants to respond the the following series of statements on a scale of: strongly agree, agree, neutral, disagree, strongly disagree.

1. I enjoyed learning the last thing that was taught to me in math class.
2. The last time I took a math class was so long ago, I don't even remember what I learned.
3. Math has been my worst subject.
4. I would consider a career that uses math.
5. Math is hard for me.
6. I am the type of student to do well in math.
7. I can handle most subjects well, but I cannot do a good job with math.
8. When choosing classes, I have tried to stay as far away from math as possible.

9. When choosing classes, I try to incorporate some math classes into my studies.
10. I am confident I could do advanced work in math.
11. I only take a math class when I have to.
12. I can get good grades in math.
13. I am good at math.
14. I am good at working with people who are good at math.
15. I find people who are good at math to be rigid.
16. I find people who are good at math to be unfriendly.
17. I find people who are good at math are creative.
18. People who are good at math are very hard to work with.
19. Being good at math is a skill you are born with.
20. Anyone can become good at math.
21. I think it is valuable to work with people who are good at math.
22. I can use what I have learned studying math to solve problems in engineering.
23. I can use what I have learned studying math to create art.

C.1.2 Engineering

We asked students about their experience with and attitudes towards engineering. We first asked “What do you think makes someone an engineer?” and ”I have written a program in ___”. Finally, we asked participants to respond the the following series of statements on a scale of: strongly agree, agree, neutral, disagree, strongly disagree.

1. I have experience with computer programming.
2. I would consider a career that uses computer programming.
3. Computer programming is hard for me.
4. I like to imagine creating new inventions.
5. If I learn engineering, then I can improve things that people use every day.
6. I am confident in myself when building and fixing things.
7. I am confident in myself when I learn something new.
8. I am interested in how machines work.
9. I am curious about how electronics work.
10. I have designed an electronics circuit before.
11. I know how electricity works.
12. I am confident I can help my peers in solving engineering problems.
13. I have assembled a robot before.
14. I have designed a robotic system before.

15. I believe I can work well in projects with engineers.
16. Engineers are very rigid.
17. Engineers are hard to work with.
18. Engineering requires creativity.
19. I believe I can be successful in a career in engineering.
20. I believe engineering is only for people who are good at math.
21. I consider myself an engineer.
22. Anyone can become a good engineer.
23. I think it is valuable to work with people who are good at engineering.
24. I can use what I have learned studying engineering to solve problems in math.
25. I can use what I have learned studying engineering to create art.

C.1.3 Arts and Fiber Crafts

We asked students about their experience with and attitudes towards engineering. We first asked "How would you define a fiber craft?", "What do you think makes someone an artist?", "I have experience with the following fiber crafts", and "I have experience with the following art forms". Finally, we asked participants to respond the the following series of statements on a scale of: strongly agree, agree, neutral, disagree, strongly disagree.

1. I have never used a loom before.
2. I would consider a career in the arts.
3. I consider myself an artist.
4. I like to imagine creating new art works.
5. I am confident in myself when I engage with the arts.
6. I believe engaging with the arts is valuable.
7. I expect to use creativity in my career.
8. I expect to use design skills in my career.
9. I am interested in working with physical materials.
10. I enjoy working with my hands.
11. I enjoy learning with my hands.
12. I am curious to learn a fiber craft.
13. I have designed a textile artifact before.
14. I believe people interested in the arts are not good at math.
15. I believe that fiber crafts do not involve computational principles.
16. I believe I can work well on projects with artists.
17. Being skilled in the arts is something you are born with.

18. Anyone can become creative.
19. I think it is valuable to work with artists.
20. I can use what I have learned studying art to solve problems in math.
21. I can use what I have learned studying art to solve problems in engineering.

C.1.4 Collaboration

We asked students about their experience with and attitudes towards engineering. We first asked "What comes to mind when you think about a group project?" and "Describe a time when you worked with someone from a different academic background than yours. How did it go?". Finally, we asked participants to respond the the following series of statements on a scale of: strongly agree, agree, neutral, disagree, strongly disagree.

1. When it comes to school work, I prefer to work in teams.
2. There are certain homework problems that I would rather do by myself.
3. Whenever I have a group project in school, there is always one person who doesn't do any work.
4. Whenever I do a team project, I find that I do more work than my teammates.
5. There are certain school group projects that can only be completed by groups of people.
6. There are certain school group projects that can only be completed by groups of people with diverse expertise.
7. I enjoy working with people from diverse academic backgrounds.
8. I value learning activities outside my academic interests and formal studies.
9. When solving a problem, I enjoy thinking about how different fields might approach the same problem in different ways.
10. Not all problems I will be faced with in my work can be solved without going beyond my own field's expertise.
11. I would like to learn many different disciplines.
12. I would like to specialize in one discipline very deeply.
13. In the future, I would like to become an expert in one discipline and collaborate with experts from other areas to solve complex problems.
14. Most complex problems can be solved by one person who learns about many different things.
15. I see connections between ideas in my academic interests and ideas in another student's academic interests, even if the fields seem very different.
16. I can use what I have learned in one field to solve problems in another field.
17. I often step back and reflect on what I am thinking to determine whether I might be missing something.
18. I frequently stop to think about where I might be going wrong or right with a problem

solution.

19. I can identify the kinds of knowledge that are distinctive to different fields of study.

C.2 Reflection Questions

We asked student to reflect on their experiences with the activities and assignments in the course to better understand what they took away from the exercises. Students were asked to reflect on building their RoboLooms through the following questions:

1. How did your team work together? Did you divide roles? Did you try to do everything together? Separately?
2. Reflect on how much of the work you did when building the loom. Do you feel you did the majority of the work? Or was it split evenly amongst teammates?
3. How much of the time were you actually hands-on with the loom or kit materials?
4. How engaged did you feel during the building process?
5. How much help did you feel you needed during building? Was the help provided by the course staff? From your teammates? Other teams?
6. What was the main challenge that you found when building the loom? Take a picture to include in your report! Discuss how you overcame it. Did your teammates help?
7. What was your biggest triumph when building the loom? Take a picture to commemorate it!

Students also had the experience of weaving and comparing different cloth structures. They were asked the following reflection questions for this activity:

1. Plain Weave
   (a) Do you think the patterns look different depending on the type of yarn or material you use? How?
   (b) What was your favorite yarn or material and why?
   (c) Take a picture of your teams plain weave to include in your report.
   (d) How did you set up your plain weave draft? Were there multiple ways to achieve this? Did your teammates have differing approaches/ideas?
2. Mystery Patterns
   (a) Could you figure out what the other teams' pattern was? Try to draw it!
   (b) What was your strategy? How did you work with your teammate to achieve your goal?
   (c) Do you feel confident that you were able to determine the other teams patterns from their weaving? Why or why not?

Students were also asked to create an 8-bit pixel-art symbol using matrix multiplication, then collaboratively combine their individual art pieces into one group symbol. They were asked the following reflections:

1. Are you happy with your group symbol? Do you feel like it represents all of your individual chosen symbols?
2. What was it like designing and weaving your own symbol? Were there parts that were particularly difficult?
3. How did your team work together? Do you feel like the work was split evenly? Do you feel anyone on the team took a lead for certain portions of the activity?
4. How do you feel you did with the activity? Were you comfortable/confident? Why or why not?
5. How much help did you feel you needed during the activity? Was the help provided by teammates/course staff/others?
6. How engaged were you in the activity? What are some reasons for your level of engagement?

Students additionally completed a final project and were asked to reflect individually on this project via the following questions:

1. Are you happy with your group project? Do you feel like it represents all of your individual ideas?
2. What was it like designing and weaving your own cultural symbol? Were there parts that were particularly difficult?
3. How did your team work together? Do you feel like the work was split evenly? Do you feel anyone on the team took a lead for certain portions of the activity?
4. How do you feel you did with the activity? Were you comfortable/confident? Why or why not?
5. How much help did you feel you needed during the project? Was the help provided by teammates/course staff/others?
6. How engaged were you in the project? What are some reasons for your level of engagement?
7. Are there any other personal comments that you would like the course staff to know about?

C.3 Interviews

We interviewed students at the end of the semester to better understand their experience in the class. We asked the following questions:

1. How did the process of building your loom affect your group dynamic if at all?
2. How did your group dynamic evolve over time?
   (a) Why do you think that was the case?
   (b) Did working with the loom affect your group dynamic?
   (c) Did the activities affect your group dynamic?
3. Did building your loom affect your engineering skills? If so, how?

4. Did building your loom affect your perception of your group's skills? If so, how?
5. Did using the loom during activities affect your engineering, math, or art skills? If so, how?
6. Did your perceptions of your own engineering, math, and art skills change during the activities? If so how?
7. Did your ideas of your groups engineering, math, and art skills change during the activities? If so, how?
8. Did you feel you learned anything about math, engineering or art throughout the class? If so, what and why?
9. Are there any interactions with your loom or your group that stand out to you?
10. Tell me about a time when your group worked well with the loom
11. Tell me about a time when your group worked poorly with the loom

Bibliography

[1] National Academies. *Education for Life and Work: Developing Transferable Knowledge Skills in the 21st Century*. National Academies Press, Washington, D.C., Dec 2012. ISBN 978-0-309-25649-0. doi: 10.17226/13398. URL http://www.nap.edu/catalog/13398. 1, 6.1

[2] National Academies. *Next Generation Science Standards. For States, By States.* National Academies Press, Washington, D.C., Aug 2013. ISBN 978-0-309-27227-8. doi: 10.17226/18290. URL http://www.nap.edu/catalog/18290. 1, 6.1

[3] National Academies. *A New Vision for Center-Based Engineering Research*. National Academies Press, Washington, D.C., Jul 2017. ISBN 978-0-309-45905-1. doi: 10.17226/24767. URL https://www.nap.edu/catalog/24767. 1, 6.1

[4] Ernest Afari and Myint Khine. Robotics as an educational tool: Impact of lego mindstorms. *International Journal of Information and Education Technology*, 7:437–442, January 2017. doi: 10.18178/ijiet.2017.7.6.908. 2.3

[5] Lea Albaugh, James McCann, Lining Yao, and Scott E. Hudson. Enabling personal computational handweaving with a low-cost jacquard loom. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. doi: 10.1145/3411764.3445750. URL https://doi.org/10.1145/3411764.3445750. 4.3, 4.6.1, 4.1, 4.6.2.1

[6] AliExpress. Stroke 80mm linear actuator, 2023. URL https://www.aliexpress.com/item/2255800220070257.html. 4.5.2

[7] Arduino. Arduino mega rev3, 2021. URL https://store-usa.arduino.cc/products/arduino-mega-2560-rev3. 4.5.2

[8] Michael J. Baker. Collaboration in collaborative learning. *Interaction Studies*, 16(3): 451–473, Jan 2015. ISSN 1572-0373, 1572-0381. doi: 10.1075/is.16.3.05bak. 2.6, 6.2, 7.2.1, 7.2.2, 8.2.1

[9] Lawrence W. Barsalou. Challenges and opportunities for grounding cognition. *Journal of Cognition*, 3(1):31. ISSN 2514-4820. doi: 10.5334/joc.116. 2.1.1, 11.2.2

[10] Lawrence W. Barsalou. Grounded cognition. *Annual Review of Psychology*, 59(1): 617–645, 2008. doi: 10.1146/annurev.psych.59.103006.093639. 2.1.1, 11.2.2

[11] Lawrence W. Barsalou. Grounded cognition: Past, present, and future. *Topics in Cognitive*

Science, 2(4):716–724, 2010. ISSN 1756-8765. doi: 10.1111/j.1756-8765.2010.01115.x. 2.1.1, 11.2.2

[12] B. K. Behera, Jiri Militky, Rajesh Mishra, and Dana Kremenakova. Modeling of woven fabrics geometry and properties. In Han-Yong Jeon, editor, *Woven Fabrics*, chapter 1, pages 1–33. IntechOpen, Rijeka, 2012. doi: 10.5772/38723. 4.2.2.2, 4.4.3.2

[13] Emily Breidenbach. Introduction to frame loom weaving, Sep 2022. URL `https://continuingstudies.saic.edu/introduction-frame-loom-weaving/1270-2409`. 5.2

[14] Eric Broudy. *The book of looms a history of the handloom from Ancient Times to the present*. Brandeis University Press, 2021. 4.1, 4.2

[15] D. Catlin and M. Blamires. *The principles of Educational Robotic Applications (ERA): a framework for understanding and developing educational robots and their activities*. The 12th EuroLogo conference, Paris, 2010. ISBN 978-80-89186-65-5. URL `https://repository.canterbury.ac.uk/item/86q3v/the-principles-of-educational-robotic-applications-era-a-framework-for-understanding-and-developing-educational-robots-and-their-activities`. 2.3

[16] Herbert H. Clark and Susan E. Brennan. *Grounding in communication*, page 127–149. American Psychological Association, Washington, DC, US, 1991. ISBN 978-1-55798-121-9. doi: 10.1037/10096-006. 2.1.1, 11.2.1

[17] Ulrike Cress, Jun Oshima, Carolyn Rosé, and Alyssa Friend Wise. *Foundations, Processes, Technologies, and Methods: An Overview of CSCL Through Its Handbook*, page 3–22. Springer International Publishing, Cham, 2021. ISBN 978-3-030-65291-3. doi: 10.1007/978-3-030-65291-3_1. URL `https://doi.org/10.1007/978-3-030-65291-3_1`. 2.2

[18] Himani Deshpande, Haruki Takahashi, and Jeeeun Kim. Escapeloom: Fabricating new affordances for hand weaving. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. doi: 10.1145/3411764.3445600. URL `https://doi.org/10.1145/3411764.3445600`. 4.2

[19] Kayla DesPortes and Betsy DiSalvo. Trials and tribulations of novices working with the arduino. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*, ICER '19, page 219–227, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361859. doi: 10.1145/3291279.3339427. URL `https://doi.org/10.1145/3291279.3339427`. 4.5.2

[20] Laura Devendorf, Sasha de Koninck, and Etta Sandry. An introduction to weave structure for hci: A how-to and reflection on modes of exchange. In *Designing Interactive Systems Conference*, DIS '22, page 629–642, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393584. doi: 10.1145/3532106.3534567. URL `https://doi.org/10.1145/3532106.3534567`. 4.2

[21] Pierre Dillenbourg, Michael J. Baker, Agnès Blaye, and Claire O'Malley. The evolution

of research on collaborative learning. In Spada, E., Reiman, and P., editors, *Learning in Humans and Machine: Towards an interdisciplinary learning science.*, pages 189–211. Elsevier, Oxford, 1995. URL https://telearn.hal.science/hal-00190626. 2.6, 7.2.1, 8.2.1

[22] Pierre Dillenbourg, Michael J. Baker, Agnès Blaye, and Claire O'Malley. *The evolution of research on collaborative learning*, page 189. Elsevier, Oxford, 1995. URL https://telearn.hal.science/hal-00190626. 2.6, 7.2.1, 8.2.1

[23] Frederick Thomas Peirce D.Sc., F.Inst.P., and F.T.I. 5—the geometry of cloth structure. *Journal of the Textile Institute Transactions*, 28(3):T45–T96, 1937. doi: 10.1080/19447023708658809. 4.1

[24] Mark Elliott. *Digital Library Of The Commons*. PhD thesis, Indiana University, 2007. URL https://dlc.dlib.indiana.edu/dlc/handle/10535/3574. 2.6, 7.2.1, 8.2.1

[25] Clarence A. Ellis, Simon J. Gibbs, and Gail Rein. Groupware: some issues and experiences. *Communications of the ACM*, 34(1):39–58, Jan 1991. ISSN 0001-0782. doi: 10.1145/99977.99987. 2.6, 3.2, 6.1, 7.2.1, 8.2.1

[26] Yrjö Engeström. *Activity theory and individual and social transformation*, page 19–38. Learning in Doing: Social, Cognitive and Computational Perspectives. Cambridge University Press, Cambridge, 1999. ISBN 978-0-521-43127-9. doi: 10.1017/CBO9780511812774.003. URL https://www.cambridge.org/core/books/perspectives-on-activity-theory/activity-theory-and-individual-and-social-transformation/D675EE5F0F5A2FB4A8C15F8A704517E4. 2.1.2, 11.2.2

[27] Yrjö Engeström. *Learning by Expanding: An Activity-Theoretical Approach to Developmental Research*. Cambridge University Press, Cambridge, 2 edition, 2014. ISBN 978-1-107-07442-2. doi: 10.1017/CBO9781139814744. URL https://www.cambridge.org/core/books/learning-by-expanding/6D0648C3DEDE20157B359E464AFDB8C1. 2.1.2, 11.2.2

[28] James Essinger. *Jacquard's web: How a hand-loom led to the birth of the information age*. Oxford University Press, 2010. 4.1

[29] Ylva Fernaeus, Martin Jonsson, and Jakob Tholander. Revisiting the jacquard loom: Threads of history and current patterns in hci. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 1593–1602, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450310154. doi: 10.1145/2207676.2208280. URL https://doi.org/10.1145/2207676.2208280. 4.2

[30] Paradise Fibers. Ashford katie table loom, 2023. URL https://www.paradisefibers.com/products/ashford-katie-table-loom. 4.6.1, 4.1, 4.6.2.1

[31] Lucia Foglia and Robert A. Wilson. Embodied cognition. *WIREs Cognitive Science*, 4(3): 319–325, 2013. ISSN 1939-5086. doi: 10.1002/wcs.1226. 11.2.2

[32] Mikhaila Friske, Shanel Wu, and Laura Devendorf. Adacad: Crafting software for smart textiles design. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300575. URL https://doi.org/10.1145/3290605.3300575. 4.2

[33] Hugo Fuks, Alberto Raposo, Marco Aurelio Gerosa, Mariano Pimentel, Denise Filippo, and Carlos Lucena. Inter- and intra-relationships between communication coordination and cooperation in the scope of the 3c collaboration model. In *2008 12th International Conference on Computer Supported Cooperative Work in Design*, page 148–153, Apr 2008. doi: 10.1109/CSCWD.2008.4536971. 2.6, 3.2, 6.1, 6.4.1, 7.2.1, 8.2.1

[34] Michael Gaffney. *Enhancing teachers' take-up of digital content: Factors and design principles in technology adoption*. Education Services Australia Ltd., 2010. 4.4.2

[35] Christian Giang, Alberto Piatti, and Francesco Mondada. Heuristics for the development and evaluation of educational robotics systems. *IEEE Transactions on Education*, 62(4): 278–287, November 2019. ISSN 1557-9638. doi: 10.1109/TE.2019.2912351. 2.3

[36] Ozdemir Gol and Andrew Nafalski. Collaborative learning in engineering education. *Global Journal of Engineering Education*, 11, 2007. 6.6

[37] Sandy L. Gonzalez, Veronica Alvarez, and Eliza L. Nelson. Do gross and fine motor skills differentially contribute to language outcomes? a systematic review. *Frontiers in Psychology*, 10, December 2019. ISSN 1664-1078. doi: 10.3389/fpsyg.2019.02670. URL https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2019.02670/full. 11.2.2

[38] Banko Grünbaum and Geoffrey C. Shephard. Satins and twills: An introduction to the geometry of fabrics. *Mathematics Magazine*, 53(3):139–161, 1980. doi: 10.1080/0025570X.1980.11976845. URL https://doi.org/10.1080/0025570X.1980.11976845. 4.2.2.1, 4.4.3.2, 4.5.3

[39] Ellen Harlizius-Klück. Weaving as binary art and the algebra of patterns. *TEXTILE*, 15 (2):176–197, 2017. doi: 10.1080/14759756.2017.1298239. URL https://doi.org/10.1080/14759756.2017.1298239. 4.1, 4.2.1, 4.4.3.2

[40] Giyoo Hatano and James V. Wertsch. Sociocultural approaches to cognitive development: The constitutions of culture in mind. *Human Development*, 44(2/3):77–83, 2001. ISSN 0018-716X. 2.1.1

[41] James Hollan, Edwin Hutchins, and David Kirsh. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction*, 7(2):174–196, June 2000. ISSN 1073-0516. doi: 10.1145/353485.353487. 2.1.2, 11.2.3

[42] Judith A. Howard and Daniel G. Renfrow. *Social Cognition*, page 259–281. Handbooks of Sociology and Social Research. Springer US, Boston, MA, 2006. ISBN 978-0-387-36921-1. doi: 10.1007/0-387-36921-X_11. URL https://doi.org/10.1007/0-387-36921-X_11. 2.1.1, 11.2.1

[43] Ken Hurst. *Engineering design principles*. John Wiley & Sons, Inc., 2004. 4.4.3.3

[44] Edwin Hutchins. Distributed cognition. *International Encyclopedia of the Social and Behavioral Sciences. Elsevier Science*, 138:1–10, 2000. 2.1.2, 11.2.3

[45] AVL Looms Inc. Jacq3g jacquard loom, 2023. URL https://avllooms.com/products/jacq3g?variant=19975805206586. 4.3, 4.6.1, 4.1, 4.6.2.1

[46] Casey Jones. Interdisciplinary approach - advantages, disadvantages, and the future benefits of interdisciplinary studies. *ESSAI*, 7, Apr 2010. URL https://dc.cod.edu/cgi/viewcontent.cgi?article=1121&context=essai. 4.1

[47] Lee Jones and Audrey Girouard. Learning with stitch samplers: Exploring stitch samplers as contextual instructions for e-textile tutorials. In *Designing Interactive Systems Conference*, DIS '22, page 949–965, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393584. doi: 10.1145/3532106.3533488. URL https://doi.org/10.1145/3532106.3533488. 4.2

[48] Victor Kaptelinin and Bonnie A. Nardi. *Acting with Technology: Activity Theory and Interaction Design (Acting with Technology)*. The MIT Press, September 2006. ISBN 978-0-262-11298-7. 2.1.2

[49] Hiroshi Kato, Toshio Mochizuki, Hideo Funaoi, and Hideyuki Suzuki. A principle for cscl design: Emergent division of labor. page 2652–2659. Association for the Advancement of Computing in Education (AACE), 2004. ISBN 978-1-880094-53-2. URL https://www.learntechlib.org/primary/p/12828/. 2.2.2

[50] P. A. Kirschner, R. L. Martens, and J. W. Strijbos. *CSCL in Higher Education?*, page 3–30. Springer Netherlands, Dordrecht, 2004. ISBN 978-1-4020-7921-4. doi: 10.1007/1-4020-7921-4_1. URL https://doi.org/10.1007/1-4020-7921-4_1. 2.2.2

[51] Paul Kirschner, Jan-Willem Strijbos, Karel Kreijns, and Pieter Jelle Beers. Designing electronic collaborative learning environments. *Educational Technology Research and Development*, 52(3):47–66, Sep 2004. ISSN 1556-6501. doi: 10.1007/BF02504675. 6.2

[52] Birgit Krogstie and John Krogstie. Using activity theory and community of practice to understand knowledge management processes of international enterprises with high diversity. In *FiIth Congress oI the International Society Ior Cultural Research and ctivity Theory (ISCR T)*, 2002. 2.1.2, 11.2.2

[53] Jean Lave and Etienne Wenger. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, September 1991. ISBN 9780511815355. doi: 10.1017/CBO9780511815355. URL https://www.cambridge.org/highereducation/books/situated-learning/6915ABD21C8E4619F750A4D4ACA616CD. 2.1.2, 11.2.4

[54] Victor R Lee and Heidee Vincent. An expansively-framed unplugged weaving sequence intended to bear computational fruit of the loom. In *Proceedings of FabLearn 2019*, pages 124–127. 2019. 5.1

[55] Pamela Liou. Home - doti: The dot matrix loom. URL http://dotilikestoweave.com/. 4.3

[56] Edward Lorenz. Models of cognition, the contextualisation of knowledge and organisational theory. *Journal of Management and Governance*, 5(3):307–330, September 2001. ISSN 1572-963X. doi: 10.1023/A:1014098928477. 2.1.2, 11.2.4

[57] Maria Luce Lupetti, Cristina Zaga, and Nazli Cila. Designerly ways of knowing in hri: Broadening the scope of design-oriented hri through the concept of intermediate-level knowledge. In *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '21, page 389–398, New York, NY, USA, March 2021. Association for Computing Machinery. ISBN 978-1-4503-8289-2. doi: 10.1145/3434073.3444668. URL `https://dl.acm.org/doi/10.1145/3434073.3444668`. 2.4

[58] Sandra May. Engineering design process, Jul 2017. URL `https://www.nasa.gov/audience/foreducators/best/edp.html`. 4.1, 4.2, 4.4.3.3, 5.4.1

[59] Emma Mercier, Molly H Goldstein, Preethi Baligar, and Robin Jephthah Rajarathinam. Collaborative learning in engineering education. In *International handbook of engineering education research*, pages 402–432. Routledge, 2023. 6.6

[60] Microchip. Mcp23017. URL `https://www.microchip.com/en-us/product/mcp23017`. 4.5.2

[61] Tassos A Mikropoulos and Ioanna Bellou. Educational robotics as mindtools. *Themes in Science and Technology Education*, 6(1):5–14, 2013. 2.3

[62] Syne Mitchell. *Inventive weaving on a little loom: Discover the full potential of the rigid-heddle loom*. Storey Publishing, 2015. 4.1, 4.4.1.1, 4.4.3.1, 4.4.3.2

[63] S. M. Mizanoor Rahman. Metrics and methods for evaluating learning outcomes and learner interactions in robotics-enabled stem education. In *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 2103–2108, 2020. doi: 10.1109/AIM43001.2020.9158900. 4.4.1

[64] Jennifer Moore, Joe Sanchez, and Alissa Tudor. Weaving a storytelling tapestry using computational thinking. In *IASL Annual Conference Proceedings*, 2021. 5.1

[65] Heloisa Moura, Dale Fahnstrom, Greg Prygrocki, and T. J. McLeish. Thinkeringspace: Designing for collaboration. *Visible Language*, 43(1):46–61, 2009. ISSN 0022-2224. 2.6, 6.2, 7.2.1, 7.2.2, 8.2.1, 8.2.2

[66] Ilan E Moyer. 24-bit friendship loom. URL `http://web.mit.edu/imoyer/www/portfolio/jacquard/index.html`. 4.3

[67] Bonnie A Nardi. Studying context: A comparison of activity theory, situated action models, and distributed cognition. 1995. 2.1.2, 11.2.2

[68] Geraldine Gonzaga Nerona. Effect of collaborative learning strategies on student achivemement in various engineering courses. *International journal of engineering education*, 1(2):114–121, 2019. 6.6

[69] Jerry Nevins. Weaving, 2004. URL `http://www.jnevins.com/weaving.htm`. 5.2

[70] Catherine Newell and Alan Bain. *Team-Based Collaboration in Higher Education Learning and Teaching: A Review of the Literature*. Springer, August 2018. ISBN

9789811318559. 6.1

[71] Jerry Nicholls. Month: July 2018, Jul 2018. URL https://r.jander.me.uk/index.php/2018/07/. 4.3

[72] Curtis W. Nielsen, David J. Bruemmer, Douglas A. Few, and David I. Gertman. Framing and evaluating human-robot interactions. In *Proceedings of Metrics for Human-Robot Interaction A workshop at the Third ACM/IEEE International Conference on Human-Robot Interaction (HRI08)*, pages 29–36, 2008. 4.4.1

[73] Donald A. Norman. *The Design of Everyday Things*. Basic Books, 2013. ISBN 978-0-465-07299-6. URL https://ia902800.us.archive.org/3/items/thedesignofeverydaythingsbydonnorman/The%20Design%20of%20Everyday%20Things%20by%20Don%20Norman.pdf. 1, 2

[74] Digital Weaving Norway. Tc2 loom: Digital weaving machine, 2023. URL https://digitalweaving.no/tc2-loom/. 4.3, 4.6.1, 4.1, 4.6.2.1

[75] Piia Näykki, Sanna Järvelä, Paul A. Kirschner, and Hanna Järvenoja. Socio-emotional conflict in collaborative learning—a process-oriented case study in a higher education context. *International Journal of Educational Research*, 68:1–14, January 2014. ISSN 0883-0355. doi: 10.1016/j.ijer.2014.07.001. 1, 6.3.3, 7.5.4

[76] Taiichi Ohno. *Toyota Production System: Beyond Large-Scale Production*. Productivity Press. ISBN 978-0-429-27301-8. doi: 10.4324/9780429273018. 6.5.2

[77] Irmak Olcaysoy Okten, Gordon B. Moskowitz. *Social Cognition*. Psychology Press, 2017. ISBN 978-1-315-17137-1. 2.1.1, 11.2.1

[78] Judith S. Olson and Gary M. Olson. *Computer-Supported Cooperative Work*, page 243–253. Elsevier, New York, January 2003. ISBN 978-0-12-227240-0. doi: 10.1016/B0-12-227240-4/00016-2. URL https://www.sciencedirect.com/science/article/pii/B0122272404000162. 2.2

[79] Daniëlle Ooms, Nick Voskuil, Kristina Andersen, and Hanna Ottilia Wallner. Ruta, a loom for making sense of industrial weaving. In *Companion Publication of the 2020 ACM Designing Interactive Systems Conference*, DIS' 20 Companion, page 337–340, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379878. doi: 10.1145/3393914.3395815. URL https://doi.org/10.1145/3393914.3395815. 4.3

[80] Joshua M. Pearce. Quantifying the value of open source hard-ware development. *Modern Economy*, 06(01):1–11, 2015. doi: 10.4236/me.2015.61001. 4.4.2

[81] Diane Pecher and Rolf A. Zwaan. *Grounding Cognition: The Role of Perception and Action in Memory, Language, and Thinking*. Cambridge University Press, January 2005. ISBN 978-0-521-83464-3. URL https://www.cambridge.org/us/academic/subjects/psychology/cognition/grounding-cognition-role-perception-and-action-memory-language-and-thinking,https://www.cambridge.org/us/academic/subjects/psychology/cognition. 2.1.1, 11.2.2

[82] Diane Pecher, Inge Boot, and Saskia Van Dantzig. *Chapter seven - Abstract Concepts: Sensory-Motor Grounding, Metaphors, and Beyond*, volume 54 of *Advances in Research and Theory*, page 217–248. Academic Press, January 2011. doi: 10.1016/B978-0-12-385527-5.00007-3. URL https://www.sciencedirect.com/science/article/pii/B9780123855275000073. 2.1.1, 11.2.2

[83] Reinhard Pekrun. *Emotions and Learning. Educational Practices Series-24*. July 2014. ERIC Number: ED560531. 1, 2.3, 6.3.3

[84] Kylie Peppler and Yasmin Kafai. *Transparency Reconsidered: Creative, Critical, and Connected Making*, pages 179–188. 01 2014. 6.5.2

[85] Kylie Peppler, Erica Halverson, and Yasmin B. Kafai. *Makeology makerspaces as Learning Environments (volume 1)*. Taylor and Francis, 2016. 4.1

[86] Kylie A. Peppler. *The sage encyclopedia of out-of-school learning*. SAGE reference, 2017. 4.1

[87] J. Pollitt. The geometry of cloth structure. *Journal of the Textile Institute Proceedings*, 40 (1):P11–P22, 1949. doi: 10.1080/19447014908664605. 4.1, 4.4.3.2

[88] Virginia I. Postrel. *The fabric of civilization how textiles made the world*. Basic Books, 2021. 4.1

[89] Luis P. Prieto. Classroom technology design guidelines: A focused survey. In *Proceedings of the Orchestrated Collaborative Classroom Workshop 2015 co-located with 11th International Conference on Computer Supported Collaborative Learning (CSCL 2015)*, pages 10–14, 2015. 4.4.1, 4.4.3

[90] Python. Python release python 3.8.0, 2023. URL https://www.python.org/downloads/release/python-380/. 4.5.3

[91] Nellie Annie Reed. *The book of school weaving*. Evans Bros., 1933. 5.2

[92] Abeera Rehmat, Hoda Ehsan, and Monica Cardella. K-2nd grade teachers' perceptions of computational thinking: Research findings and implications for integrating engineering and computational thinking in elementary education. In *2022 ASEE Annual Conference & Exposition*, 2022. 5.1

[93] Mike Robinson. *Design for unanticipated use.....*, page 187–202. Springer Netherlands, Dordrecht, 1993. ISBN 978-94-011-2094-4. doi: 10.1007/978-94-011-2094-4_13. URL https://doi.org/10.1007/978-94-011-2094-4_13. 2.2.1

[94] Tom Rodden. A survey of cscw systems. *Interacting with Computers*, 3(3):319–353, December 1991. ISSN 0953-5438. doi: 10.1016/0953-5438(91)90020-3. 2.2.1

[95] Yvonne Rogers and Judi Ellis. Distributed cognition: an alternative framework for analysing and explaining collaborative working. *Journal of information technology*, 9 (2):119–128, 1994. 2.1.2, 11.2.3

[96] Jeremy Roschelle and Stephanie D. Teasley. The construction of shared knowledge in collaborative problem solving. In Claire O'Malley, editor, *Computer Supported Collaborative Learning*, NATO ASI Series, page 69–97, Berlin, Heidelberg, 1995. Springer. ISBN 978-3-642-85098-1. doi: 10.1007/978-3-642-85098-1_5. 2.6, 7.2.1, 8.2.1

[97] Kurt Schaefer. Making a diy computer controlled loom, Jun 2019. URL https://retrotechjournal.com/2019/06/17/making-a-diy-computer-controlled-loom/. 4.3

[98] Brian Schmalz. Easy driver stepper motor driver. URL https://www.schmalzhaus.com/EasyDriver/. 4.5.2

[99] Kjeld Schmidt and Carla Simonee. Coordination mechanisms: Towards a conceptual foundation of cscw systems design. *Computer Supported Cooperative Work (CSCW)*, 5 (2):155–200, June 1996. ISSN 1573-7551. doi: 10.1007/BF00133655. 2.2.1

[100] Kjeld Schmidt and Carla Simonee. Coordination mechanisms: Towards a conceptual foundation of cscw systems design. 5:155–200, June 1996. ISSN 1573-7551. doi: 10.1007/BF00133655. 2.1.2, 11.2.5

[101] Albert Solberg. Systems engineering principles, Sep 2018. URL https://www.nasa.gov/consortium/SystemsEngineeringPrinciples. 4.4.3.3

[102] Samantha Speer, Ana P Garcia-Alonzo, Joey Huang, Nickolina Yankova, Carolyn Rosé, Kylie A Peppler, James Mccann, and Melisa Orta Martinez. Speerloom: An open-source loom kit for interdisciplinary engagement in math, engineering, and textiles. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606724. URL https://doi.org/10.1145/3586183.3606724. 6.1, 6.3.1, 6.3.3, 7.3.2.2, B.1, B.2

[103] Samantha Speer, Nickolina Yankova, Joey Huang, Carolyn Rosé, Kylie A Peppler, James McCann, and Melisa Orta Martinez. Designing looms as kits for collaborative assembly. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720376. doi: 10.1145/3746059.3747765. URL https://doi.org/10.1145/3746059.3747765. 7.3.2.2, 7.4.1, 7.4.2, 7.4.3, 7.4.4, 7.5.1, 7.5.1, 8.5.1, 8.5.2, 8.5.3, 8.5.4

[104] Gerry Stahl. Theories of cognition in cscw. In Susanne Bødker, Niels Olof Bouvin, Volker Wulf, Luigina Ciolfi, and Wayne Lutters, editors, *ECSCW 2011: Proceedings of the 12th European Conference on Computer Supported Cooperative Work, 24-28 September 2011, Aarhus Denmark*, page 193–212, London, 2011. Springer. ISBN 978-0-85729-913-0. doi: 10.1007/978-0-85729-913-0_11. 2.1.1, 2.1.2, 11.2.1, 11.2.3

[105] Gerry Stahl. *Theories of Collaborative Cognition: Foundations for CSCL and CSCW Together*, page 43–63. Computer-Supported Collaborative Learning Series. Springer US, Boston, MA, 2013. ISBN 978-1-4614-1740-8. doi: 10.1007/978-1-4614-1740-8_3. URL https://doi.org/10.1007/978-1-4614-1740-8_3. 2.1, 2.1.1, 2.1.2, 2.2, 6.3.3, 11.2.1, 11.2.3, 11.2.4

[106] Glenda S. Stump, Jonathan C. Hilpert, Jenefer Husman, Wen-ting Chung, and Wonsik Kim. Collaborative learning in engineering students: Gender and achievement. *Journal of Engineering Education*, 100(3):475–497, 2011. doi: https://doi.org/10.1002/j.2168-9830.2011.tb00023.x. URL https://onlinelibrary.wiley.com/doi/abs/

10.1002/j.2168-9830.2011.tb00023.x. 6.6

[107] Daniel D. Suthers. *Computer-Supported Collaborative Learning*, page 719–722. Springer US, Boston, MA, 2012. ISBN 978-1-4419-1428-6. doi: 10.1007/978-1-4419-1428-6_389. URL https://doi.org/10.1007/978-1-4419-1428-6_389. 2.2

[108] Karin Tengler, Oliver Kastner-Hauler, and Barbara Sabitzer. Identifying preliminary design principles for a robotics-based learning environment. In *2021 16th International Conference on Computer Science Education (ICCSE)*, page 771–776, August 2021. doi: 10.1109/ICCSE51940.2021.9569250. URL https://ieeexplore.ieee.org/abstract/document/9569250. 2.3

[109] Ashford Wheels and Looms. Sixteen shaft table loom, 2024. URL https://www.ashford.co.nz/product/sixteen-shaft-table-loom/. 6.1, B.6a, B.6, B.7

[110] Tony Wu, Shiho Fukuhara, Nicholas Gillian, Kishore Sundara-Rajan, and Ivan Poupyrev. Zebrasense: A double-sided textile touch sensor for smart clothing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 662–674, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375146. doi: 10.1145/3379337.3415886. URL https://doi.org/10.1145/3379337.3415886. 4.2

[111] Tony Wu, Shiho Fukuhara, Nicholas Gillian, Kishore Sundara-Rajan, and Ivan Poupyrev. Zebrasense: A double-sided textile touch sensor for smart clothing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 662–674, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375146. doi: 10.1145/3379337.3415886. URL https://doi.org/10.1145/3379337.3415886. 4.2

[112] James E. Young, JaYoung Sung, Amy Voida, Ehud Sharlin, Takeo Igarashi, Henrik I. Christensen, and Rebecca E. Grinter. Evaluating human-robot interaction. *International Journal of Social Robotics*, 3(1):53–67, Jan 2011. ISSN 1875-4805. doi: 10.1007/s12369-010-0081-8. URL https://doi.org/10.1007/s12369-010-0081-8. 4.4.1

[113] Manuel Zacklad. Communities of action: a cognitive and social approach to the design of cscw systems. In *Proceedings of the 2003 ACM International Conference on Supporting Group Work*, GROUP '03, page 190–197, New York, NY, USA, November 2003. Association for Computing Machinery. ISBN 978-1-58113-693-7. doi: 10.1145/958160.958190. URL https://dl.acm.org/doi/10.1145/958160.958190. 2.1.2, 11.2.3, 11.2.4

[114] Manuel Zackland. Communities of action: a cognitive and social approach to the design of cscw systems. In *Proceedings of the 2003 ACM International Conference on Supporting Group Work*, 2003. URL https://dl.acm.org/doi/abs/10.1145/958160.958190. 2.2.1

[115] Dimitris Zondiros. Communities of practice, activity theory, and distributed learning, 2008. 2.1.2, 11.2.2, 11.2.4