From computational thinking to computational making

Conference Paper · September 2015

DOI: 10.1145/2800835.2800926

CITATIONS
26

READS
2,375

7 authors, including:

Thomas von Rekowski
Universität Siegen
25 PUBLICATIONS **289** CITATIONS
SEE PROFILE

Jennifer Rode
University College London
57 PUBLICATIONS **1,811** CITATIONS
SEE PROFILE

Jennifer Booker
Rutgers, The State University of New Jersey
4 PUBLICATIONS **155** CITATIONS
SEE PROFILE

Konstantin Aal
Universität Siegen
81 PUBLICATIONS **1,171** CITATIONS
SEE PROFILE

From Computational Thinking to Computational Making

Jennifer A. Rode*, Anne Weibert+, Andrea Marshall*, Konstantin Aal+,
Thomas von Rekowski+, Houda el Mimoni*, and Jennifer Booker*

*Drexel University
3141 Chestnut St. Philadelphia, PA, USA
jen@acm.org; am343, he52, grb25@drexel.edu

+University of Siegen
Kohlbettstr.15, 57072 Siegen, Germany
first.last@uni-siegen.de

ABSTRACT
Computational thinking is considered best practice for teaching computing and more broadly to solve problems and design systems, however as computing extends beyond the desktop (for instance increased integration of ubicomp technologies) so too must our educational methods. Exposure to ubicomp technologies is most accessible through the maker movement. With this in mind we argue we must move from computational thinking to computational making as an educational framework. Here we present a case study of children's making to support our vision for a broader conception of computational making.

Author Keywords
Maker Culture, Comp. Thinking, Computational Making

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI); K.3.2 Computers and education, literacy

INTRODUCTION
Traditional methods to teach computing are highly analytical and focused on solving mathematical problems, manipulating text, or using predefined objects to create simple games [15]. These methods have been critiqued for failing to engage diverse students [12, 16], or those who have less linear, creative or hands-on thinking styles [23]. Traditional methods focus more on getting to the right (and only) answer to the problem using a prescribed algorithm, whereas maker culture thrives in bricolage and recognizing the importance of the process not just the end result. Kafai and colleagues [27] have discussed the benefits of applying maker culture to computing education, to promote computational thinking. Here we will show that *computational making* provides better ways to attract a diverse range of students to computing fields.

UbiComp '15, September 07-11, 2015, Osaka, Japan
 ISBN 978-1-4503-3574-4/15/09…$15.00
DOI: http://dx.doi.org/10.1145/2750858.2804261

Marshall [37], Marshall and colleagues [38] consider the potential of tangible interfaces to support learning. While evidence to support this belief is limited, Marshall [37] calls for more work to demonstrate the benefit of using physical materials and provides guidance on the use of tangible interfaces for learning. Similarly, other studies [16, 28, 53] argued that E-textiles are good for teaching computing.

Kafai and colleagues [28] examined the use of e-textiles for introducing key computing concepts. The study endeavored to understand students' learning by analyzing the circuits and program code created by students in pre-Advanced Placement (AP) high school computer science classes (AP classes give American students college credit) as well as their e-textile creations and their views of computing. The findings recommended using crafts materials and activities such as e-textiles for designing introductory courses that can broaden participation in computing.

Buechley, et al. [16] conducted a user study with children who designed soft wearables using LilyPad Arduino. The results suggested that e-textile workshops were engaging; they facilitated the exploration of art (aesthetics) and fostered gender diversity since e-textiles succeeded in engaging girls in computer Science Education using arts.

Weibert, et al. [53] explored the appropriateness of e-textiles for teaching programming to mixed gender groups ages 8-12. The result of their study demonstrated that e-textiles have the potential of promoting both genders' computational literacy, thus disrupting binary gender roles that has been contracted by conventional "masculinist attitudes towards technology." Kafai and colleagues [27] argue that coding is a crucial skill for all children as it can be a way to encourage "computational thinking." They called for the conception of "computational participation" instead of computational thinking because "computational participation" moves beyond the individual to focus on wider social networks and a DIY culture of digital "making." Kafai et al [27] argue children nowadays do not code for the sake of coding, instead it is an intrinsic skill for participating in computational communities. Much of this work is situated in a literature around making which is now part of a global movement of makerspaces and makerfaires spanning the globe [4, 7, 26, 35].

Our paper makes a two-fold contribution. First, it provides a case study to supplement the work of Kafai, Buechley and

others supporting the efficacy of using e-textiles and LilyPad to teach computing and computational thinking. We present early findings, which suggest the potential of Ardublock, a Scratch-style [36] Arduino interface for teaching computational thinking. Finally, we show that additional skills are needed beyond computational thinking to be successful. We discuss these skills and present the set with a new concept of computational making.

LITERATURE REVIEW

Making

Making as the act of creating tangible artifacts has gained attention in research from various disciplines in recent years. It has been described as an activity that is apt to link the digital and the physical [44], recent works in the fields of CHI and CSCW have focused on its relation to computing [18, 39]. Playful in its nature, making has been described to include "cultural and material engagement, decisions around tool use, the leveraging of industrial infrastructures around materials and standards, and the crucial role of knowledge sharing and building new literacies" [48]. Other important characteristics of making are concerned with individual creativity [30, 32, 41], collaboration [57], and problem solving [34]. With this, the do-it-yourself approach to technology [48] that is vital for making has been described as a contribution to the building of sociotechnical identity [48, 52]. Our work aims to develop educational best practices for teaching making using our fieldwork as a starting point.

Computational Thinking

While the use of computational thinking (CT) as a skill can be traced back to the time when humans invented the computer, Papert, in the 80's, pioneered as he brought attention to computing in the context of education. Harel and Papert [25], Papert [40] posit that children develop procedural thinking through LOGO programming. The idea of CT was popularized by Wing's [55] influential article "Computational thinking". Wing [55] main argument was CT "represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use", caught the attention of a broad and varied range of the academic community. Cuny et al. [20] defined CT as "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can effectively be carried out by an information-processing agent". CT aims to train people to think like computer scientists when confronted with a problem. Wing [55] specifies six characteristics of CT: (1) It is about "conceptualizing, not programming". (2) "Fundamental, not rote skill" as every human being must be able use CT to better function in modern society. (3) "A way that humans, not computers, think" since humans are more brilliant and imaginative in solving problems. (4) "Complements and combines mathematical and engineering thinking" as its formal foundations. (5) "Ideas, not artifacts." Ideas can be used to solve problems and communicate with others and (6) "For everyone, everywhere" since CT is so integral to human endeavors not just computer science.

The literature on teaching computing and CT is rich as it has been subject to research for the last two decades. Nevertheless, CT for kindergarden-12th grade (ages 5-18 approx.) has just recently become a topic of significant research inquiry. Guzdial [24] highlighted the importance of research on CT: "Research in computing education will pave the way to make CT a 21st century literacy that we can share across the campus." By doing so, he entices researchers to draw on a variety of disciplines (education, sociology, and psychology…) and improve their understanding of how to teach computing better. Barr and Stephenson [11] argue that today's students live and work in a world that is heavily influenced by computing principles, thus it is not sufficient to wait until students are in college to introduce such concepts. In Europe, an awareness for a comprehensive and early introduction to computing and CT has settled and manifested in national curricula for primary and secondary schools (e.g. [13, 33, 51]).

Barr and Stephenson [11] put forth an operational definition of CT which demonstrates how computational thinking can be embedded in K-12 classrooms in the US. The definition was a results of a multi-phase project done by the Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) and it stipulated that CT is a problem-solving process that includes the following characteristics:

- Analyzing and organizing data logically
- Visualizing data though abstractions, modeling and simulations
- Formulating problems in a way a computer can help in solving them
- Identifying, analyzing, and implementing solutions effectively and efficiently
- Using algorithmic thinking to automate solutions
- Generalizing and applying this problem solving process to other kind of problems

Barr and Stephenson's operational definition of CT considers the desktops as the environment of learning and does not embrace other ubiquitous computing environments. Denning and Rosenbloom have laid the grounds for a comprehensive approach to computing that is embracing making [21] in their argument for computing as a fourth great domain of science. "Computing interacts not only with people and other living systems, but with the physical world", they say (p.28), thus making the case for a holistic approach to the subject, and a revisiting of "deep questions in computing" (p.29), finding new answers "broad enough to cover everything" – the physical world included. Our work aims to identify and include the skills needed to address this physical aspect in computing.

METHOD

We conducted a study where we employed e-textiles and the LilyPad Arduino to playfully introduce children to computational thinking. This was done in two subsequent steps. In a first activity, an introduction to LilyPad was given, and a Bunny Bright electronics kit was used to teach children basic knowledge about circuits. In the second activity, children made an interactive stuffed monster using LilyPad Arduino, based on a modified lesson from the book of Buechley & Qiu, et al. [17].

Research Setting

Our research took place in a computer club in a primary school in one of the large cities in the Ruhr Area in Germany. This club is part of a network of computer clubs called come_IN which are mostly located in schools in intercultural neighborhoods in Germany (e.g.: [45, 46]). The clubs are openly structured and have a low barrier to entry. They work to foster cross-cultural understanding and respect in the intercultural neighborhood by offering the chance to participate in computer-based project work for the young and adult neighborhood community [54].

The club in the focus of our study is based in a neighborhood that stands out in the city because of its high population density, large number of families, comparatively young ages of its inhabitants, and 57.7% of the population has immigrant backgrounds. There are also high unemployment rates, low wages, and obstacles that make access to higher education difficult. Neighborhood inhabitants, a local non-profit organization and neighborhood managers brought the computer club to life in 2009. The immigrant backgrounds present in the club mirror the diversity of the surrounding neighborhood, with children and adults stemming from Turkey, Albania, Macedonia, Tunisia and Morocco.

Data Collection and Analysis

The study used a participatory action research design [28], based on a reflexive, anthropological approach to ethnography [19, 43]. We were involved in the computer club as tutors, guiding the weekly club sessions and providing assistance in computer-based project work as required (the details of the projects to follow in the next section). All club sessions were documented in short session observation notes or "jottings" in ethnographic parlance [21], supplemented with photographs taken by the researchers. The session observations were extended to provide ethnographic field notes, following each session [21].

Our data were codes in three rounds of analysis, to produce a grounded model of computational making. We employed an abductive, qualitatively analysis across iterative coding cycles [22] to explore the skills required to perform various types of task, based on an emergent theory of computational thinking [55]. In the first round, we identified the types of skills demonstrated in each session, when possible mapping these to computational thinking skills of 1) Analyzing and logically organizing data, 2) Identifying, testing, & implementing possible solutions, 3) Data Modeling, Data Abstractions, and Simulations, 4) Formulating Problems Such that Computers May Assist, and 5) Automating Solutions via Algorithmic Thinking [56]. In the second round, we identified cognitive breakdowns, indicated by episodes where the children had difficulties. We compared these to deficiencies in computational thinking skills, and wrote memos describing successful vs. failed attempts of children to perform each computational thinking skill. By the start of the third round of analysis, it had become clear that the skills required for project success went beyond those of computational thinking. We identified five new core skills 1) Aesthetics 2) Creativity, 3) Constructing, 4) Visualizing Multiple Representations, and 5) Understanding Materials. These are the basis for our theory of computational making. Our analysis ended when we reached theoretical saturation – that is when no new skills were revealed by further analysis, across sessions and across subjects.

The Activities

All children who participated in our studies had at least several months of previous computer club experience. There they had acquired knowledge in the use of different hard- and software (including assembling and installing a computer), among them text and photo editing software (for the creation of a newspaper, post cards and photo calendars) as well as audio and video processing software (used for the creation of trickfilms and an audio play), and MIT's visual programming environment Scratch [30] (used for the creation of games and story-telling activities). With regard to programming knowledge, the children had playfully explored loops, functions, operators and variables. The club was their first contact with e-textile project work. We planned our interventions so that they would build upon one another in their level of difficulty, with the first 'Bunny Bright' activity being mainly concerned with understanding circuits and applying this knowledge in a creative individual design, and the second 'Monster Making' activity building up on this by expanding this newly acquired knowledge of circuits, and their use in the design and programming of an interactive toy figure. Due to the open access structure of our computer club, two children dropped out of the club after the first activity, but other than that, participation remained constant.

Introduction to LilyPad

Conducting the Bunny Bright activity as our first e-textiles project in the computer club, an introduction to LilyPad was given to the children in the club. It provided them with a basic overview on e-textile's potential, and included a hands-on exploratory session. This introductory session consisted of three tasks: 1) connecting an LED, making it blink in different colors and learning how to mix colors with light; 2) controlling the LED with a light sensor; 3)

controlling the three different colors by use of three different means (switch, light sensor, temperature sensor). Source code for the completion of these tasks was provided to the children, and they had to edit this code in a limited way by changing values, or uncommenting code blocks. For the completion of these basic tasks, the children worked together in groups of 2 to 3 children.

Bunny Bright

For this activity, a Bunny Bright electronics kit was used, that is designed to teach basic electrical knowledge to children ages 5 and up [6]. Instructed by the tutors in the computer club, the children soldered the LED, a reed switch, a resistor and a battery holder onto a circuit board. The switch on the completed board could then be activated by a magnet, and the LED lights up. The completed circuit board was then included into a stuffed, sewn animal. For our project activity, this was by default a stuffed rabbit containing the board with a carrot housing the magnet. When the carrot was touched to the rabbit's mouth, its belly lit up to show it was happy. Some children elected to create alternate forms (see Figure 1). Five girls and five boys, ages 8 to 10 participated in our study. The results of this study were previously analyzed in terms of gender [53], however here we analyze them in terms of taught concepts required to master the task.

Figure 1. Figure projects included a swan on a lake, a dragonfly and a monkey with a banana.

Monster Making

This activity was based on one of the chapters in Buechley, et al. [17] "Sew Electric". Children made a stuffed monster from pieces of felt (see Figure 2), connected electronics and then learned how to program it to be interactive. Each project contained a LilyPad Arduino microprocessor, a speaker, and an LED all connected by conductive thread. The children learned how to program the monster such that when its paws were held together it played music. When the paws were not touching one another the LED just blinked. The paws were made of conductive fabric so touching them together completed the circuit.

Figure 2. Some of the children's monsters.

While typically programming occurs with LilyPad in the default Arduino integrated development environment (IDE), we taught children using Ardublock [10] thus departing from the project in the Buechley book. Ardublock is a drag and drop add-on to the Arduino IDE, similar in its appearance to MIT's visual programming environment Scratch [36]. We pursued Ardublock as we felt it would make it easier to avoid typos and syntax errors compared to the native Arduino programming language (see Figure 3). While there are several different Scratch-style add-ons for Arduino, such as Codebender[1] or Minibloq[2] at the time of our study Ardublock was the only one that created code that would run natively on the device. While other add-ons run simulations that require the Arduino device – in our case the LilyPad – to be connected to the PC, Ardublock allows children to use their e-textile projects while not connected to a computer.

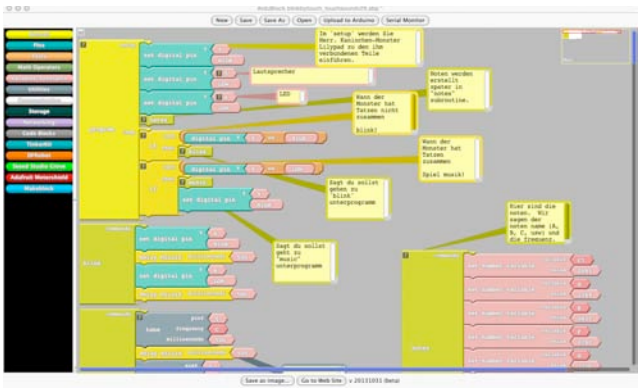

Figure 3. Ardublock with commented monster code.

The default "program" block contained the setup statements to initialize the digital pins, then the main loop which contained two *if* statements. They wrote two functions called by those *if* statements. The first played music. The second

[1] https://codebender.cc/

[2] http://minibloq.org/

caused the LED to blink. The main loop tested *if* the hands were together, the LED blinked. *If* the hands were not together, then the music function was called. Arduino plays music by specifying frequency, so middle C (C4) would be 261.63 Hz. The children had to specify frequency and duration for each note.

Eight children (five girls, three boys), ages 8-10, participated in our monster making study. We helped the children structure the code, talking them through writing and experimenting with different functions; typically with one function each session. For instance, one session focused on just making the monster's light turn on and off.

Figure 4. Testing the LED as a group.

COMPUTATIONAL THINKING

In the next section we will give examples of how over the course of the two activities, the children were able to explore and practice each aspect of computational thinking.

Analyzing and logically organizing data

This aspect of computational thinking only became relevant in the monster making activity. The first few sessions of the monster making activity were used to present and prepare the overall project. One of the authors showed her own completed monster to the children and demonstrated what the monster was capable of doing: if the hands of the monster are touching, a LED lights up and a melody plays. Seeing this, the children were eager to build their own monster to play their favorite music.

As an introduction to programming, the laptop and LilyPad of one of the tutors was used. The children sat around the laptop and observed the LED which was connected to the LilyPad. In a first step, it was explored how code had to be changed in order to make the LED light up in different colors (see Figure 4). Then the tutor showed the children how the code had to be altered to let the LED blink very fast. Playfully, children and tutor together explored how the code had to be changed to make the LED blink more quickly and in different colors. Organizing the code to alternate between different colors and different speeds was the final task, which demonstrated understanding of the written code and computational thinking.

Identifying, testing & implementing possible solutions

In the Bunny Bright activity, children were seen to experiment with various possible solutions for the placement of the magnet in the design of their choice. This was the case, e.g. when two girls had decided that they wanted their project to be a swan on a lake, and then decided to place the board in the sewn lake, and to include the magnet in the little white felt-swan.

In the introductory LilyPad session two different sensors were used to activate the LED: a heat sensor and a light sensor. The children used the debugger of the Arduino IDE to read out the values of the sensors. They realized the value was changing if, in the case of the light sensor, they held their hand over the sensor and it darkened; in the case of the heat sensor a source of heat was needed to change the value of the sensor. One boy used his breath to warm-up the sensor, while other children observed the debugger to choose the right threshold for the code to detect the temperature change. Afterwards they tested the value using their hands to dim the light sensor or cold air from outside for the heat sensor and adjusted the code after every evaluation. The children needed some help to start and understand the debugger; afterwards testing, adjusting and organizing code was no problem.

During the ongoing project, different problems occurred while the children tried to build their monster. The authors observed different strategies to deal with these challenges and overcome them.

One of the tutors drew a simple version of the circuit of the monster on a white board and explained which cables are connected to the power and which are used for the sensors. Afterwards the children were asked to extend the circuit layout and add the speaker. Many of them were afraid to say something wrong, other children were shy. The tutor started drawing another connection on the board, asking the children for instructions in doing so. One boy told him to connect it "*to the plus port of the power supply*". He obtained the marker from the tutor and drew the right connection on the board.

Another strategy was observed while a girl and a boy worked together on programming the LilyPad in the introductory session. She delegated the code writing and told him what he should write down, while he was responsible to handle the keyboard and mouse. After every written line of code, she compared her instructions with the screen to see if everything was correct. After they finished their code and tried to compile it, an error occurred (a missing semicolon at the end of a code line). Both of them used the instructions to identify the mistake and correct it. Other children were not patient enough to search the error on their own and asked for help; together with a tutor they successfully corrected the code.

The children tried to follow the instructions as exactly as possible. If they finished their part of the project and a problem occurred, they tried to identify and fix it. Other children helped them to overcome problems, if the tutor was not available (see Figure 5).

Figure 5. A student helps connect a sensor the right way.

These projects allowed children numerous opportunities to identify possible solutions by playfully engaging with project topics, which they tested through a combination of debugging and playfully experimenting with the code.

Data Modeling, Data Abstractions, and Simulations

In the LilyPad introduction, children playfully explored the concept of data abstraction when they were asked to make the LED blink and alter the frequency of the blinking. This was the case, e.g. when two girls tried random numbers in their code, thus creating a crazily wild blinking rhythm. Then they also set the number "1".

Girl 1: *"Hm. Does not work…"*
Girl 2: *"But why?"*

A tutor explains how "1" equals a millisecond and is not perceptible to the human eye. Then the girls were seen to systematically experiment by setting larger and larger numbers, exploring to find the threshold for visibility.

The monster making activity afforded the opportunity to create data abstractions. Writing music requires using the tone function and specifying the frequency (C4 would be 261.63 Hz), duration in milliseconds, and the number of the pin where the speaker was connected. The task required multiple data abstractions. First, the children would have to look at the sheet music and translate quarter, half or whole notes into seconds. Second, they would have to identify each note, and then look up each note on a chart to determine the appropriate frequency [2]. The children learned to do this but we got them to agree quickly it was tedious, and showed them how to write a little library function so that they could just write the note C or D. (They were only writing one octave of musical notes.) In this way the children were exposed to several forms of data abstractions.

Formulating Problems Such that Computers May Assist

On several occasions, the children had to ask questions in a way that the computer could help. Formulating the music in Hertz and Milliseconds is one such example.

Similarly, the children might think of the different components connected to the LilyPad by name, e.g. respectfully pronouncing the rather technical and complex sounding term "*FTDI breakout board*". But they learned quickly how the computer only remembers what pin it is sending or receiving data.

In the Bunny Bright activity the children were thinking in terms of their designed objects like swans sitting on a lake, or apes eating a banana – but on the technical level, this was a matter of circuits being complete (when the swan sat on the lake, or the ape's mouth touched the banana), or not. The children learned that they had to incorporate the binary condition of 'circuit-closed' or 'circuit-incomplete' into their respective e-textile designs.

In the Monster Making activity, the children might think of whether the monster has his hands together or apart, but the computer thinks in terms of pins and voltages (see Figure 6). In the case of our monster the hands are connected to digital pin 9. If the hands are together the circuit is complete, and thus the voltage will be high. In this way the monster project supported this aspect of computational thinking.

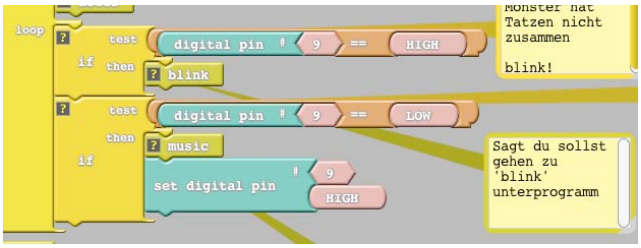

Figure 6. Basic loop code in Ardublock.

Automating Solutions via Algorithmic Thinking

Right from the start, when the monster making activity was introduced to the computer club and the children learned that they would learn to use the visual programming environment Ardublock over the course of the project, some boys in the club were eager to see hard, written programming code. Comparing a piece of written code to the graphical, colorful appearance of Ardublock, as well as Scratch, which they also knew, the children expressed their respect for the material:

Boy 1: *"Wow, this is tough!"*
Boy 2: *"Yeah, that's really complicated!"*

So the children learned that programming can have many different forms, but that there is an underlying logic which is similar. This was explored in one of the following sessions of the monster-making project. The children were

introduced to programming logic in a playful way: by playing a 'programming' game, where the group was divided in two teams – team A being the programmers, the other team B being the artifacts about to be programmed, the children explored how a real-world problem can be translated into basic programming logic. The task was for team A to have the other team B walk around in the room like elephants. Whenever two 'elephants' met, they were supposed to make a trumpeting elephant-like sound, and when a 'Stop' signal was given, the game would end. The only way team A was supposed to communicate this to team B was via pen and paper.

Initially, all boys in the group want to be the programmers. But as soon as the girls in the group discover that the task is also concerned with writing things up nicely with colored pens on paper, they are drawn to this task as well and in the end took over the programming task. In writing, the girls made sure that each of them had her turn in writing some parts of the tasks.

The 'programming' game lets the children experience several key characteristics of programming:

a) Precision is important – the children of team A struggle at first, to write the task in a way that leaves little or no room for misinterpretation.

b) They learn that not everyone in team B needs to know every detail of the overall task, and then discuss how to split it up. Some children, who already had gathered some programming experience with Scratch, proposed the idea to use different colors for writing up different parts of the task: red for 'start' and 'stop'; green for everything concerned with 'movement', and yellow for 'sound, speaking, or appearance'.

c) In carrying out the given written tasks that team A had prepared, team B then discovers time to be a key factor: how long is something supposed to happen? When should it start, when is it supposed to end? Also: do given tasks have an order? What happens if you mess with this order?

The children were exposed to automatic and algorithmic thinking first through this game, and then again as the engaged with programing concepts such as *loops* and *if* statements.

Generalizing & Applying CT

Over the course of the Monster Making project, it became apparent that most children had difficulty making sense of the abstract concept(s) of CT and related instructions, which presupposed certain basic concepts, such as minus being a synonym for negative when it comes to electric circuits etc. Often verbal instructions seemed to be ignored by the elementary school children; instead they were mimicking the teacher's hands-on demonstrations step-by-step. This way, the aimed-for results (e.g. a switchable circuit) were achieved but also mistakes such as sewing paper Arduino LilyPads on the cloth (as observed in a previous demonstration) were made.

On two occasions, activity games were used to convey CT concepts most effectively, thus laying the groundwork for future generalizing and application of CT concepts by the children. As described earlier, for the first game, participants impersonating programming objects were asked to instruct each other, experiencing CT related concepts firsthand. The second game was aimed at conveying the idea of note duration and pitch being represented by code values. By the help of animal metaphors, the teacher illustrated both concepts; a 'lion' representing a half note for instance, as the word 'lion' comprises two syllables (when said in German), a 'bear' representing a full note with only one syllable and so on. The participants' prior Scratch experience helped them to get started with the new Ardublock interface, as many design elements (e.g. colored code categories) resembled the familiar Scratch designs.

The two e-textile activities in sequence helped the children understand the concept of a circuit. Their initial understanding was from the Bunny Bright activity. In the Monster Making project, they remembered this and were able to build up on this initial understanding by creating an even more complex circuit that involved designing its layout and not solely soldering a predefined circuit shape. In this way they generalized their understanding of circuits in Bunny Bright to the Monster project.

COMPUTATIONAL MAKING

The findings from our two studies with the children indicate there are many important skills incorporated in e-textile activities. Our data shows that aesthetics, creativity, constructing, visualizing multiple representations, and understanding materials are five key factors. We will describe these in the following sections.

Aesthetics

Aesthetics is an integral component of computational making [29, 53, 54], and it is also a crucial part of computational thinking [30, 50]. There were instances during the Bunny Bright, as well as the Monster Making activity where the participants' desire for aesthetically pleasing artifacts lead them to refuse applying the right solutions into practice. This was the case, e.g. when one of the girls opted for a rather complicated layout of her monster circuit, because she did not want it to interfere with the face she had designed for her monster, or when another girl had to redesign her circuit because its conductive threads were crossing – she as well had followed her sense for an aesthetically pleasing appearance of her monster and failed to also take the practicability of her solution into consideration.

In some cases, a strong desire for an aesthetically perfect solution was even seen to hinder the unfolding of the children's creativity. This was the case in the Bunny Bright

activity, where a shy little girl, who was firm in her expression of what she considered to be aesthetical and "*pretty*", refused to come up with ideas of her own for the shape of her project, because she said it would never be as pretty as the sample Bunny that the tutor had shown.

Once confident with the overall project task, aesthetic decisions allowed the children to make autonomous judgments as they engaged in making; this lead to instances of interpretive flexibility [32] as they applied nonlinear processes to the act of constructing artifacts. Thus, the children's aesthetic choices also lead to emergent design agency [29], allowing them to expand their technical self-efficacy [42] and incorporate additional skills to computational thinking. Aesthetics selections became a part of the children's best practices. Creativity is another quality that can manifest concretely instead of remaining an abstraction with regard to computational making.

Creativity

Creativity is an important segment our participants applied to their Bunny Bright, as well as Monster Making activities. It is concerned with the individualization of the overall project task. Children were explicitly encouraged to incorporate something meaningful and unique from their daily lives into the project activity, e.g. by designing the monster in the colors of their favorite soccer club, or by decorating it in their favorite color. In that regard, we saw creativity as a problem solving tool as well as a tool for free expression – a form of skill building that allows for playful interpretive flexibility [31] in terms of decision making – that the children still had to explore. Even though children had no problem expressing what they liked and were a fan of, many had trouble at first to see how they would be able to turn this individual feature of theirs into a part of their monster project, thus reflecting their social values, and allowing them to build computational self-efficacy. When the children deviated from the proposed shapes of the projects and created something else, e.g. by making their monster be a bat or a black and yellow soccer monster, they demonstrated creativity was both an abstract problem solving mechanism and a concrete and tangible skill building reflected in the construction of the artifacts.

Looking at the two project activities as a sequence, we saw the children refine their creativity skills, in that they were able to employ it first – in the Bunny Bright project – for a free-form task, when they individualized the shape of their sewn felt figures. Then in the second project activity – the Monster Making – the children were able to explore creativity as a skill that can also help for the solution of a rather technical problem, in that case the aesthetically pleasing layout of a circuit on fabric.

Constructing

In order to engage in computational thinking as applied to making children had to be able to produce tangible objects. A variety of physical skills are required, among them sewing, soldering, using pliers, wire strippers, and other hand tools including scissors. In the Bunny Bright activity, the children exhibited difficulty with the dexterity required to manipulate hand tools due to lack of practice with such rudimentary tools such as scissors as well as more complex issues such as tools being sized incorrectly for children.

These same types of issues became apparent with regards to sewing. Children were unfamiliar with the handling of needle and thread, and some boys also had to overcome an initial unwillingness to deal with this task that they considered to be more "*for the girls*" though this attitude changed as they worked on the project.

While the sample project in the Bunny Bright, as well as the Monster Making activity used a buttonhole stitch to join the two pieces of fabric, about half of the children opted for a simple whip stitch in each activity (see Figure 7). These are the same basic stitch but the buttonhole pulls the needle through the loop before pulling tight. Despite the buttonhole stitches more even appearance and the children's concern for aesthetics the children found this stitch too hard. We believe this speaks to a difficulty manipulating tools and trouble visualizing in 3D as we will address next.

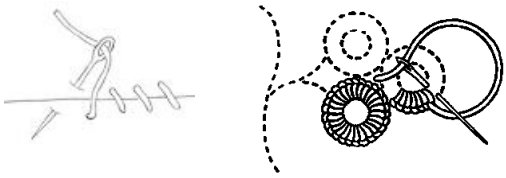

Figure 7. Whip stitch (left)[8] and buttonhole stitch (right)[1].

To avoid needles unthreading and increase resistance children worked with thread that was passed through the needle with both ends knotted together (contrary to typical sewing practice). Children had difficult tying tight clean knots. We showed them that the length of thread from your chin to your finger tip was a good working length, but children still had a hard time to estimate this, resulting in many tangles, knots, and ultimately shorted circuits.

Further, the children had difficulties attending to the front and backsides of their fabric to ensure nothing went awry. When cutting out shapes for the designs of their individual projects in the Bunny Bright activity, as well as cutting out shapes for eyes, nose and mouth of the monsters, the children would always initially layout these shapes on the felt that would resemble their position in a complete picture, and not position them close to each other in a way that would save felt from being unnecessarily cut.

When cutting out shapes from fabric, children in both activities were seen to have difficulties in making clean cuts. Many figures initially had fringy edges, which the children tried to repair by cutting off more fabric. In the end, the tutors had to intervene, telling the children that overlapping edges of the fabric could be cut off after the sewing and everything else was completed.

In the Monster Making activity, many circuits shorted or failed to light at all due to bad stitchery. This compounded debugging difficulties because one was unsure if problems were due to software, circuit design or construction (sewing). In these ways craft skills became vital to the making effort, and without them children's efforts to master the technology were impeded.

Visualizing Multiple Representations

The instructions in Buechley's "Sew Electric" [17] asked children to draw half the shape of their monster on a paper folded in half, so that once the paper was unfolded again, a perfectly symmetrical and evenly shaped monster was laid out. It appeared that the visualization of the complete monster shape, but only drawing half of it, was a challenge for many children. As a result, many of the initial monster layouts had a little dent in the middle of their heads, due to the children's inability to visualize half a circle.

In the Bunny Bright project the circuit was already predefined by the electronic kit which was used. The children didn't need to follow circuit diagrams or to mind not to cross the conductive thread. During the white board session it was clear that the children had to understand why they have to follow the 2D representations and especially watch out how to connect the sensors and the LilyPad.

In a first step, alligator clips were used to follow the instructions and create and test a working circuit with the LED, speaker and LilyPad. Afterwards printed pictures of the different parts were attached to the Monster to get an impression how the LED and sensor should be arranged without crossing the conductive thread connecting them with the LilyPad. The last step was to stitch everything on the fabric; problems occurred since the monster had an upper part and bottom part. While the LilyPad was fixed on the under part, the monster's paws with the conductive fabric were on the upper part. The children had problems to transfer the 2D circuit diagram into the real world 3D representation of the monster. A tutor helped to find a way how the conductive threads could be sewn without crossing each other.

Understanding Materials

The children sometimes experienced challenges that involved mapping the same functionality and the same outputs to different materials. For example, some of the children did not understand during the monster making activity that alligator clips and conductive thread both allowed electric currents to travel from one point to another. Many times we witnessed children struggling with the properties of various materials, such as paint versus wire. We also observed challenges that emerged surrounding knowledge of how different materials operate. For example, some of the children did not comprehend that conductive fabric affects how circuits could work, but they also mastered an understanding during the monster making activity of how conductive threads should not touch one another. Two of the children suggested that "it will break" and "it won't work afterwards", indicating they understood intrinsically that the circuits would short. Furthermore, some of the children were so accurate in their visual depictions of this concept that they clearly labeled the negative wire on their diagrams, indicating they remembered positive and negative threads should not cross (resulting in a shorted circuit). The children were also able to demonstrate independent mastery of these concepts, as we observed them teaching each other without our guidance.

DISCUSSION

There is significant support for computational thinking as core to best education practices in STEM, especially for computer science. Meanwhile STEAM oriented making which emphasizes art such as the LilyPad Arduino integrates arts and is credited with increasing diversity. Consequently, we have seen a call to move from STEM to STEAM and include the Arts [5, 14, 15] which has even been codified into US Law [3]. Making with e-textiles or through other creative activities is one way to address this [9, 27, 47, 49]. Further, as discussed by Kafai and colleagues [27], e-textiles provide a venue for discussing computational thinking. Here we have provided a case-by-case example of how introductory projects could be used to expose children to computational thinking skills. The Monster Making activity allowed all six aspects of computational thinking to be taught, and the Bunny Bright activity taught all aspects except programming.

Based on our success with physical game activities for conveying abstract or dry concepts to young learners, we recommend a gamification approach to help maintain attention among young learners while experiencing CT related concepts firsthand. Further we saw that e-textiles represent an attractive medium for teaching CT, but also assessed a need for addressing skills in a way that exceeds a regular teacher-led classroom environment. Children needed the opportunity to engage in bricolage to explore and internalize the CT concepts, and a directed learning style often resulted in children mimicking rather than understanding despite the teachers' best efforts. On the whole the project was successful in teaching CT.

Computational Thinking is clearly a core skill to making in that it speaks to individual creativity [30, 32], collaboration [57], and problem solving [34], but there is even more to it. Kafai and colleagues [27] have called to move towards computational participation, stressing the need to focus on applications, communities, remixing and tangibles. Our Bunny Bright and Monster Making projects had all of these attributes. The work done in the two projects supports the approach of Kafai et al. [27] and demonstrates how successful these types of projects can be in exposing children to computational thinking. Repeatedly however, our data showed additional skills are required for successful making, namely: aesthetics, creativity, constructing,

visualizing multiple representations, and understanding materials. While we do not insist this list is complete, we do contend that these skills are vital into addition to those of computational thinking. We call for these five skills, in addition to those of computational thinking to be taught for STEAM education. We have named this combined set of skills computational making.

STEAM skills are increasingly being acknowledged as vital [3]. However, as soon as we move from teaching computational thinking with a focus on the desktop and software, to ubicomp and the maker space, as we have shown different skills are required. Knowledge of software is still critical, but so is knowledge of electronics, engineering and craft skills like sewing, drawing or carving. Consequently, if we are to honor the calls for the importance of teaching STEAM skills the notion of computational thinking is too narrow. We call for a broadened notion of computational making as the starting point for future STEAM education.

CONCLUSION AND FUTURE WORK

We have shown how computational thinking can be taught by e-textiles, but more importantly how additional skills are needed. We broadened the concept to that of computational making. In the future we will conduct additional research with children making to verify our list of computational making skills is complete and accurate. Further, we need to work with the computer science community to develop best educational practices to teach these skills. Finally, while this paper suggests Ardublock is a potentially effective tool for teaching computational making, its use needs to be studied more thoroughly.

Here we have shown making using the Bunny Bright and Monster Making activities allowed children to practice a wide variety of computational thinking skills. However, a broader set of skills are required, and thus we make a call to better support the Arts in STEAM and benefit from the diversity that making allows. We need to expand beyond computational thinking in our teaching. Thus, we call for broadening the scope of educational best practice to include our five aspects of computational making. If we are getting computers off the desktop in order to achieve a computer for the 21st century we need to teach children the skills of computational making to go along with it.

ACKNOWLEDGMENTS

We would like to thank Anna Clapham, Tarika Chawla, Tamanna Chawla, Sven Draht, Jordan Jobs, Joschka Piscator, Akshay Sharma, and Volker Wulf. We are grateful for Susan Gasson's methodological guidance, and the children, teachers and parents who supported this research. This material is based upon work supported by the National Science Foundation under Grant No. 1253465.

REFERENCES

[1] *Buttonhole stitch*. 02-28-2015]; Available from: http://www.annatextiles.ch/vo_sti/voca3/voc3.htm.

[2] *Frequencies for equal-tempered scale, A4 = 440 Hz*. 02-27-2015]; Available from: http://www.phy.mtu.edu/~suits/notefreqs.html.

[3] *H. RES. 51 Expressing the sense of the House of Representatives that adding art and design into Federal programs that target the Science, Technology, Engineering, and Mathematics (STEM) fields encourages innovation and economic growth in the United State*.

[4] *Maker Faires Around the World*. July 4th, 2015]; Available from: http://makerfaire.com/map/.

[5] *Resources of STEM to STEAM initiative*. 02-27-2015]; Available from: http://stemtosteam.org/resources/.

[6] *Sparkfun Bunny Bright*. 02-28-2015]; Available from: http://www.sparkfun.com/tutorials/335.

[7] *Understanding Europe's Maker Movement*. July 4th, 2015]; Available from: https://webgate.ec.europa.eu/socialinnovationeurope/en/magazine/design-and-technology/articles-reports/understanding-europe%E2%80%99s-maker-movement.

[8] *Whip stitch*. 02-28-2015]; Available from: http://www.stitchpiecenpurl.com/whip-stitch.htm.

[9] Agrawal, H., R. Jain, P. Kumar, and P. Yammiyavar, *FabCode: visual programming environment for digital fabrication*, in *Proceedings of the 2014 conference on Interaction design and children*. 2014, ACM: Aarhus, Denmark. p. 353-356.

[10] Ardublock. *Ardublock a Graphical Programming Language for Arduino*. [cited 2015 02-27-2015]; Available from: http: //blog.ardublock.com/.

[11] Barr, V. and C. Stephenson, *Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?* ACM Inroads, 2011. **2**(1): p. 48-54.

[12] Benda, K., A. Bruckman, and M. Guzdial, *When life and learning do not fit: Challenges of workload and communication in introductory computer science online.* ACM Transactions on Computing Education (TOCE), 2012. **12**(4): p. 15.

[13] Berry, M., *Computing in the national curriculum.* A guide for primary teachers. Bedford: Computing at School, 2013.

[14] Bonamici, S. *Testimony on Capitol Hill. View CSPAN coverage of the US House Committee hearing on Science, Space and Technology*. 02-27-2015]; Available from: http://www.capitolhearings.org/.

[15] Boy, G.A., *From STEM to STEAM: toward a human-centred education, creativity & learning thinking*, in *Proceedings of the 31st European Conference on Cognitive Ergonomics*. 2013, ACM: Toulouse, France. p. 1-7.

[16] Buechley, L., M. Eisenberg, J. Catchen, and A. Crockett, *The LilyPad Arduino: using computational*

textiles to investigate engagement, aesthetics, and diversity in computer science education, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2008, ACM: Florence, Italy. p. 423-432.

[17] Buechley, L., K. Qiu, J. Goldfein, and S. de Boer, *Sew electric : a collection of DIY projects that combine fabric, electronics, and programming*. 2013.

[18] Buechley, L., D.K. Rosner, E. Paulos, and A. Williams. *DIY for CHI: methods, communities, and values of reuse and customization*. in *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. 2009. ACM.

[19] Clifford, J. and G.E. Marcus, *Writing culture: the poetics and politics of ethnography: a School of American Research advanced seminar*. 1986: Univ of California Press.

[20] Cuny, J., L. Snyder, and J.M. Wing, *Demystifying computational thinking for non-computer scientists.* Unpublished manuscript in progress, referenced in http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf, 2010.

[21] Denning, P.J. and P.S. Rosenbloom, *The profession of IT Computing: the fourth great domain of science.* Communications of the ACM, 2009. **52**(9): p. 27-29.

[22] Emerson, R., R. Fretz, and L. Shaw, *Writing ethnographic fieldwork*. 1995, Chicago, IL: University of Chicago Press.

[23] Grove, F., N. Jorgenson, B. Brummel, S. Sen, and R. Gamble, *Adapting rewards to encourage creativity.* Multi-Agent Systems for Education and Interactive Entertainment: Design, Use and Experience: Design, Use and Experience, 2010: p. 51.

[24] Guzdial, M., *Education Paving the way for computational thinking.* Communications of the ACM, 2008. **51**(8): p. 25-27.

[25] Harel, I. and S. Papert, *Software design as a learning environment.* Interactive learning environments, 1990. **1**(1): p. 1-32.

[26] Howard, C., A. Gerosa, M. Mejuto, and G. Giannella, *The Maker Movement: a new avenue for competition in the EU.* European View, 2014. **13**(2): p. 333-340.

[27] Kafai, Y., Q. Burke, and M. Resnick, *Connected code: Why children need to learn programming*. 2014: MIT Press.

[28] Kafai, Y., E. Lee, K. Searle, D. Fields, E. Kaplan, and D. Lui, *A Crafts-Oriented Approach to Computing in High School: Introducing Computational Concepts, Practices, and Perspectives with Electronic Textiles.* Trans. Comput. Educ., 2014. **14**(1): p. 1-20.

[29] Kafai, Y., K. Searle, C. Martinez, and B. Brayboy. *Ethnocomputing with electronic textiles: culturally responsive open design to broaden participation in computing in American indian youth and communities*. in *Proceedings of the 45th ACM technical symposium on Computer science education*. 2014. ACM.

[30] Kafai, Y.B., D.A. Fields, and K.A. Searle. *Everyday creativity in novice e-textile designs*. in *Proceedings of the 8th ACM conference on Creativity and cognition*. 2011. ACM.

[31] Kafai, Y.B., K.A. Peppler, Q. Burke, M. Moore, and D. Glosson. *Fröbel's forgotten gift: textile construction kits as pathways into play, design and computation*. in *Proceedings of the 9th International Conference on Interaction Design and Children*. 2010. ACM.

[32] Kafai, Y.B., K. Searle, E. Kaplan, D. Fields, E. Lee, and D. Lui. *Cupcake cushions, scooby doo shirts, and soft boomboxes: e-textiles in high school to promote computational concepts, practices, and perceptions*. in *Proceeding of the 44th ACM technical symposium on Computer science education*. 2013. ACM.

[33] Kultusministerkonferenz, *Aktivitäten der Länder zur Stärkung der mathematisch-naturwissenschaftlich-technischen Bildung*. 2011.

[34] Lewis, T., *Creativity in technology education: Providing children with glimpses of their inventive potential.* International Journal of Technology and Design Education, 2009. **19**(3): p. 255-268.

[35] Lindtner, S., G.D. Hertz, and P. Dourish, *Emerging sites of HCI innovation: hackerspaces, hardware startups & incubators*, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014, ACM: Toronto, Ontario, Canada. p. 439-448.

[36] Maloney, J., M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, *The scratch programming language and environment.* ACM Transactions on Computing Education (TOCE), 2010. **10**(4): p. 16.

[37] Marshall, P., *Do tangible interfaces enhance learning?*, in *Proceedings of the 1st international conference on Tangible and embedded interaction*. 2007, ACM: Baton Rouge, Louisiana. p. 163-170.

[38] Marshall, P., P.C.-H. Cheng, and R. Luckin, *Tangibles in the balance: a discovery learning task with physical or graphical materials*, in *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*. 2010, ACM: Cambridge, Massachusetts, USA. p. 153-160.

[39] Mellis, D., S. Follmer, B. Hartmann, L. Buechley, and M.D. Gross. *FAB at CHI: digital fabrication tools, design, and community*. in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 2013. ACM.

[40] Papert, S., *Mindstorms: Children, computers, and powerful ideas*. 1980: Basic Books, Inc.

[41] Peppler, K., D. Glosson, Y. Kafai, D. Fields, and K. Searle. *Articulating creativity in a new domain: expert insights from the field of e-textiles*. in *Proceedings of the 8th ACM conference on Creativity and cognition*. 2011. ACM.

[42] Rode, J.A., *An Ethnographic Examination of the Relationship of Gender & End-User Programming DISSERTATION*. 2008, Citeseer.
[43] Rode, J.A. *Reflexivity in digital anthropology*. in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011. ACM.
[44] Rosner, D.K. *Mediated crafts: digital practices around creative handwork*. in *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. 2010. ACM.
[45] Schubert, K., A. Weibert, and V. Wulf, *Locating computer clubs in multicultural neighborhoods: How collaborative project work fosters integration processes.* International Journal of Human-Computer Studies, 2011. **69**(10): p. 669-678.
[46] Stevens, G., M. Veith, and V. Wulf, *Bridging among ethnic communities by cross-cultural communities of practice*, in *Communities and Technologies 2005*. 2005, Springer. p. 377-396.
[47] Szczepa, A.F., #324, ski, C. Yost, N. Magden, E. Meaney, and C.I. Staples, *Opposites attract: computational and quantitative outreach through artistic expressions*, in *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*. 2013, ACM: San Diego, California, USA. p. 1-7.
[48] Tanenbaum, J.G., A.M. Williams, A. Desjardins, and K. Tanenbaum. *Democratizing technology: pleasure, utility and expressiveness in DIY and maker practice*. in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2013. ACM.
[49] Tempelton, R. *Testimony on Capitol Hill. View CSPAN coverage of the US House Committee hearing on Science, Space and Technology*. 02-27-2015]; Available from: http://www.capitolhearings.org/.
[50] Turkle, S., *Evocative objects: Things we think with*. 2011: MIT press.
[51] Undervisningsministeriet, *Informationsteknologi. LærePlanInformationsteknologi. Læreplan B og C niveau. København: Undervisningsministeriet B og C niveau. København: Undervisningsministeriet.* 2014.
[52] Wang, T. and J.J. Kaye. *Inventive leisure practices: understanding hacking communities as sites of sharing and innovation*. in *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. 2011. ACM.
[53] Weibert, A., A. Marshall, K. Aal, K. Schubert, and J. Rode, *Sewing interest in E-textiles: analyzing making from a gendered perspective*, in *Proceedings of the 2014 conference on Designing interactive systems*. 2014, ACM: Vancouver, BC, Canada. p. 15-24.
[54] Weibert, A. and V. Wulf. *All of a sudden we had this dialogue...: intercultural computer clubs' contribution to sustainable integration*. in *Proceedings of the 3rd international conference on Intercultural collaboration*. 2010. ACM.
[55] Wing, J.M., *Computational thinking.* Communications of the ACM, 2006. **49**(3): p. 33-35.
[56] Wing, J.M., *Computational thinking and thinking about computing.* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2008. **366**(1881): p. 3717-3725.
[57] Xie, L., A.N. Antle, and N. Motamedi. *Are tangibles more fun?: comparing children's enjoyment and engagement using physical, graphical and tangible user interfaces*. in *Proceedings of the 2nd international conference on Tangible and embedded interaction*. 2008. ACM.