A Real-time Distributed Toolkit to Ease Children's Exploration of IoT

Torben Wallbaum
Flensburg University of Applied Sciences
Flensburg, Germany
torben.wallbaum@hs-flensburg.de

Swamy Ananthanarayan
Monash University
Melbourne, Australia
University of Oldenburg
Oldenburg, Germany
swamy.ananthanarayan@monash.edu

Andrii Matviienko
Technical University of Darmstadt
Darmstadt, Germany
matviienko@tk.tu-darmstadt.de

Susanne Boll
University of Oldenburg
Oldenburg, Germany
susanne.boll@uni-oldenburg.de

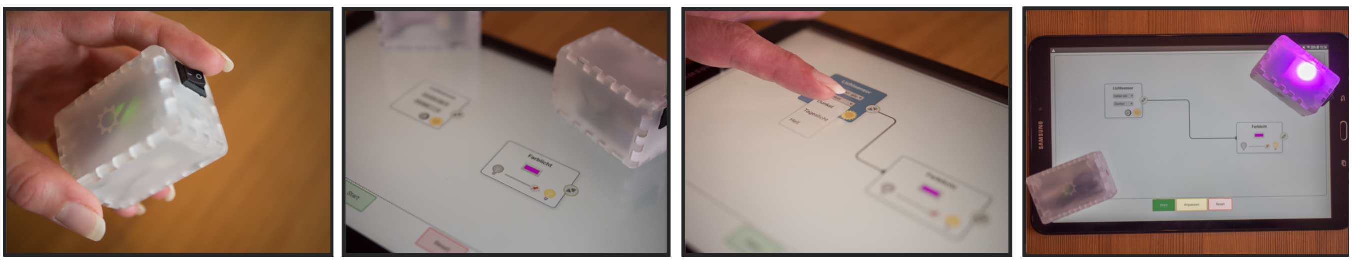

Figure 1: The IoT toolkit consists of distributed wireless modules that appear in a visual programming environment as soon as they are activated. They can be programmed in real-time and connecting sensor and feedback modules creates logical connections between them in the real-world.

ABSTRACT

Children are increasingly exposed to everyday objects with embedded computing and wireless capabilities. However, understanding how these devices collect data, communicate information to other devices, and interpret program instructions is not typically taught to children. Moreover, programming these devices still requires considerable knowledge particularly for primary school children. In this paper, we showcase a distributed toolkit of various sensors and output modules, each with wireless capability, that can independently or in concert work together. This is enabled by a programming environment with a real-time interpreter that can connect and update the state of the modules on-the-fly. We tested the system with 32 primary school children in an after-school study and found that the majority of children knew how to couple sensors to different output modalities to solve various devised scenarios. For some children, the toolkit was also used to build IoT games or fulfill personal tasks.

NordiCHI '20, October 25–29, 2020, Tallinn, Estonia

ACM ISBN 978-1-4503-7579-5/20/10…$15.00
https://doi.org/10.1145/3419249.3420179

CCS CONCEPTS

• **Human-centered computing** → **Interaction devices**; • **Software and its engineering** → *Visual languages*; • **Social and professional topics** → *Information science education.*

KEYWORDS

Internet of things; computational toolkit; children; education.

ACM Reference Format:

Torben Wallbaum, Swamy Ananthanarayan, Andrii Matviienko, and Susanne Boll. 2020. A Real-time Distributed Toolkit to Ease Children's Exploration of IoT. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society (NordiCHI '20), October 25–29, 2020, Tallinn, Estonia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3419249.3420179

1 INTRODUCTION

With the miniaturization of microcontrollers and wireless technologies, there is a growing portion of everyday objects such as lighting fixtures, thermostats, cameras, home appliances, and wearable technology that exchange data and work in cooperation. The Internet of Things (IoT) has changed how we interact with and experience technology. However, as intelligent objects evolve in functionality and complexity, most of us, as physicist and science writer Jeremy Bernstein expresses, "are increasingly surrounded by objects that we use daily but whose workings are a total mystery to us" [4].

This issue is particularly more poignant for younger children who are increasingly exposed to IoT technologies that work "like magic." Although, there has been a push to educate children on computational thinking utilizing physical computing toolkits [8, 34], tangible [3, 33, 38] and visual programming environments [11, 32], and wearable computing platforms [20, 31], there has been little in way of learning about how these devices interconnect and operate from an IoT perspective. Moreover, helping younger primary school children explore these concepts and design their own networked system still requires considerable effort and technical knowledge, because existing toolkits still follow a "one-microcontroller-to-many-peripherals approach that requires interfacing with a computer to write, compile, and download code" [21].

To address this issue, we designed a distributed toolkit of wireless modules where each module controls exactly one sensor or feedback modality. The modules can be programmed on-the-fly through a visual programming environment that sends instructions wirelessly to a real-time interpreter running on each module. Connecting the modules in the environment also links them in the real world and allows for different programs to be created. In Figure 1, we provide a simple scenario of how a child might use the toolkit to create an intruder alert system, where a light sensor placed by a door activates a remote RGB LED on a desk, perhaps as a notification of an annoying sibling. The child begins by turning on the light sensor module and the RGB LED module. This results in the modules appearing instantaneously on a tablet programming environment. Each of the modules can then be programmed in real-time via interface elements such as dropdowns, textboxes, and sliders. In this case, the child selects the light sensor to respond with limited light and the RGB LED to have a default color of purple. Drawing a connection between the two objects creates a natural if statement so that when the light sensor receives limited light (e.g., when the sibling is passing by), the RGB LED is activated to display the color purple.

In this paper, we offer two primary contributions:

- We present the design and implementation of a real-time, wireless, distributed toolkit aimed at lowering the barrier of entry to primary school children's exploration of IoT concepts.
- We discuss preliminary findings from an empirical evaluation with 32 elementary school children that offers a view into the types of activities children may engage in with this toolkit.

2 RELATED WORK

The aim of our work is to help primary school children experience IoT concepts in an engaging and non-intimidating way. Doing so may serve as a basis for learning more complex interconnected systems. In this section, we begin by examining IoT concepts and providing a simplified working definition of IoT for children. Next, we explore existing computational toolkits and physical computing platforms for children and highlight the hardware and software developments in this area. Lastly, we identify the gaps in these platforms our kit intends to fulfill.

2.1 IoT Background

The combination of hardware, middle-ware, and presentation that is representative of IoT typically refers to sensors or actuators with computing power and communication capabilities, glue software to enable the connections and data-exchange, and visualizations for interpreting the data [15]. Only recently, have educators made explicit the teaching concepts underlying IoT, including embedded programming, computer hardware, networking, and distributed systems [1]. Although, several courses have emerged from this discourse, they have all been at the university level [24]. Ideas such as embedded programming and distributed systems still remain abstract and technically advanced for younger audiences. It is unclear what IoT concepts are meaningful or what the best practices are for teaching these ideas to young children.

To lower the barrier of entry for primary school children, we employ a user-centric definition from Gubbi et al., where IoT is described as an "interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative applications" [15]. We adapted and simplified this definition in consideration of our target user group (children typically between 6-10 years of age) as, a) sensors that detect input from the world around them, b) feedback modalities that can follow rules and respond to the sensor data, c) software to transmit/receive data and manage connections between devices.

2.2 Toolkits and Platforms

There is already a broad body of work on computational toolkits and programming environments for children (see reviews [7, 22]). Although only a smaller subset of these are aimed at primary school children [39], it is a quickly growing area. Researchers have typically employed block-based interfaces such as Scratch Jr. [11] and KidSim [36], tangible physical manipulatives like KIBO [38], or hybrid environments [16] like Strawbies [17] in the design of these programming environments. Tangible toolkits such as Cubelets [33], and LittleBits [3] have been increasingly popular with younger children since they provide immediate sensory engagement [40], visibility and concreteness [5].

Additionally, physical computing platforms such as the LilyPad [8] and Flora (adafruit.com/flora), although still difficult for younger children, have helped broaden participation of underrepresented groups. To address some of the programming issues faced with these platforms, newer child-friendly kits such as the BBC micro:bit [34], Calliope (calliope.cc), and Kniwwelino (kniwwelino.lu), couple hardware with a block-based programming environment rather than the traditional text editor. While existing toolkits and platforms can certainly be used to explore IoT concepts (and have for K12 [35] and undergraduate students [25]), they are really designed to facilitate maker activities or teach programming. For younger primary school children, ConnectUs [26, 27], which consists of an interactive sensing cube embedded with a variety of sensors, light arrays, and Bluetooth technology, was designed with IoT in mind. Each cube can be wirelessly connected to other cubes and programmed via a block-based interface. Although, the system was described as conducive for learning IoT concepts [26, 27], it was never evaluated for that purpose, but rather as a toolkit in special needs classrooms [28]. Perhaps the work closest to our own is Sam Labs (samlabs.com), a commercial kit used by researchers to explore IoT tangibles for socio-emotional learning with primary school children [13]. They used paper-based conceptualization cards to scaffold children's pairing of various sensing and feedback modalities. Their designs were then programmed with the assistance of an adult.

2.3 Aspects of Modern Toolkits

The abundance of computational toolkits and platforms begets the question why these existing systems cannot be used to explore IoT concepts. As alluded to earlier, since these systems are designed primarily for maker activities and/or to teach programming, consequently, many of the boards follow a conventional embedded systems model, where a single microcontroller often controls multiple sensors and feedback modalities on the same board (e.g., BBC micro:bit, Calliope). Even with wireless capabilities integrated onboard, exploring IoT requires programming/debugging multiple boards via a block-based interface and then negotiating the communication between them. Since these programming environments are not typically supportive of multiple boards on the same screen, each board needs to be programmed separately. Moreover, many of the environments described in the previous subsection do not allow for instantaneous execution of code and still follow a traditional program-upload-debug cycle.

These aspects of modern toolkits can make exploring IoT concepts a technically fraught and time consuming experience for primary school children. In contrast, our approach follows a distributed model where sensor and output modules are wirelessly independent. Moreover, the programming environment supports multiple modules and evaluates program statements in real-time. Recent work by Cabrera et al. has shown that by enabling live programming, children (11-15 years of age) actually spend more time interacting directly with the physical device [9]. While there are certainly toolkits such as Talkoo [19] and Lego WeDo (education.lego.com/en-us/support/wedo-2) that implement visual flow-based programming, the individual hardware modules lack wireless connectivity and require either USB hubs or separate external wireless modules. Perhaps the only exception here is the previously discussed Sam Labs (samlabs.com) toolkit. It consists of Bluetooth enabled blocks that can be programmed wirelessly through a flow-based programming environment. The blocks require an initial pairing with the computer, but the programming environment follows a similar paradigm to our own.

Perhaps, the more important distinction between our toolkit and prior work, is the idea of "selective exposure" [7], defined by Blikstein as "aspects of technology that are either exposed or hidden" from children "depending on theoretical and pedagogical commitments" [7]. In our particular case, the inter-connectivity of modules, based on our working definition of IoT, is foregrounded to children rather than teaching programming or electronics. In the next section, we discuss how this design heuristic was used in the development of the toolkit.

3 IOT TOOLKIT

The toolkit (Figure 2) consists of cube shaped wireless modules that operate in a distributed fashion to ease children into exploring IoT concepts. The modules can be programmed from a visual programming environment installed on an Android tablet. The cubes are divided into input and feedback modules, where input modules abstract sensors and feedback modules serve as visualizations. Although the feedback modules can be used stand-alone (e.g., just having the serial LCD display your name), the input modules require feedback module(s) to visualize sensor data in some meaningful way. The input modules (Figure 2 left) consists of a distance, temperature and light sensor. The data from these modules can be visualized through a serial LCD display, motor, RGB LED, and speaker (Figure 2 middle).

At the heart of every module is an Wemos ESP8266 microcontroller with built-in WiFi. Each module is powered by a 500 mAh lithium ion battery that is controlled through a tactile switch located on the side of every module. Each module can also be charged via a micro-USB charging port. The circuitry is housed in a laser-cut semi-opaque Plexiglas case with a child-friendly icon to indicate the type of module. For example, the temperature module has a thermometer

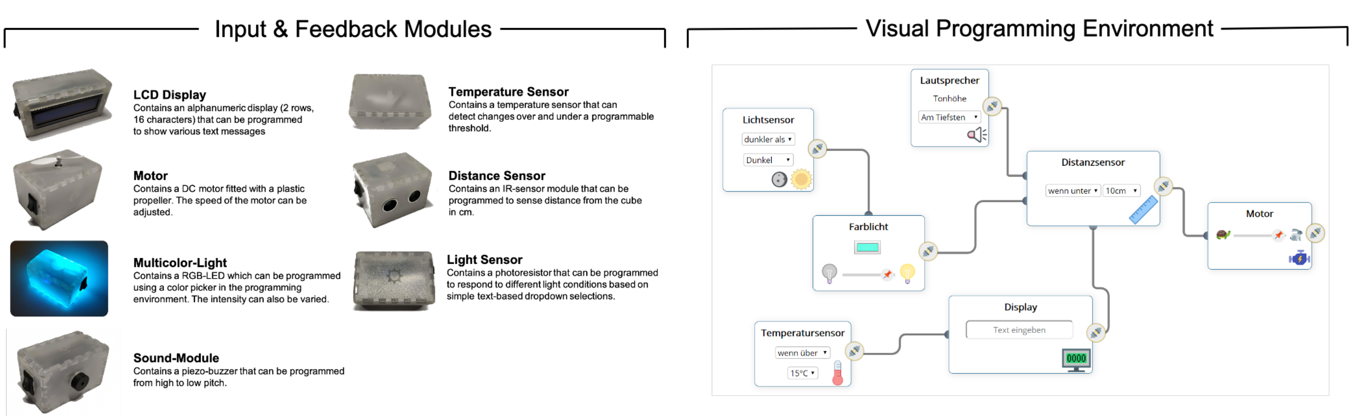

Figure 2: The toolkit consists of four feedback and three sensor blocks (left). A tablet-based visual programming environment can be used to program the modules and establish logical connections between them (right).

symbol etched on the surface. An exploded view of the circuitry for a sample module is shown in Figure 3. The majority of electronics is hidden and only the sensor or feedback element is "selectively exposed" to help children identify what they are programming. Thus, the actuator becomes the primary pedagogical focus of the interaction rather than the surrounding electronics which may be confusing to young learners. From a software perspective, modules

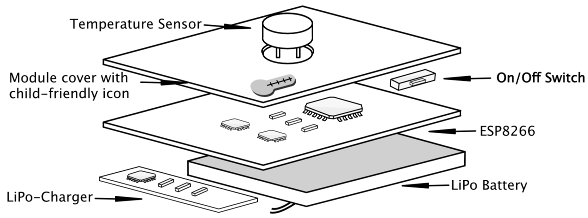

Figure 3: An exploded view of the temperature module

receive and send data/instructions over a shared WiFi network. Each module runs a real-time interpreter for executing instructions, which are formatted for communication as JSON messages. When a module is switched on, it advertises itself on the network and is immediately picked up by the visual programming environment on the tablet. Each module by default is running an infinite loop, so any instructions sent to the module will keep running indefinitely until it is modified. Typically, many programming environments for children require the child to drag-and-drop a "forever" block into which nested instructions can be placed. However, researchers have found that the concept of a forever loop was not understood by most primary school children [2], especially for polled actions like reading sensor values. As a result, we "selectively hid" this aspect from children in our design.

The visual programming environment consists of a blank canvas that is populated as modules get turned on. Each module (both input and feedback) has a virtual representation that specifies what kind of module it is along with instructions of how to program the module. Modules can be programmed using common user interface elements such as text fields, color pickers, sliders and drop-down menus. A change in state of the virtual module reflects instantaneously in the real-world module. We selectively chose to expose and evaluate syntax in real-time through GUI elements to help children focus on learning the modules quickly. This shifts the focus away from composing programs and downloading them via USB to each module.

Children can establish connections between the sensor and feedback modules by drawing virtual connections in the programming environment. A one-to-one connection between a sensor and a feedback module represents an if-statement. This can also be extended to one-to-many connections where input from one sensor can control multiple feedback modules. For example, when the light sensor receives no light, it can turn on both the RGB LED and the motor. The behavior of the LED and the motor depends on how they were initially programmed. Multiple sensor modules can also be coupled to create many-to-one or many-to-many relationships. When multiple input modules are utilized, then all the sensor modules have to be true (logical "AND" operation) to activate the feedback module(s).

Once the modules are programmed from the visual programming environment, the program is persistent (e.g., tablet can be turned off) and the modules can independently exchange messages with each other. Since our focus was to help children explore interconnections between devices, rather than teach them programming, we chose to expose the if-statement through virtual connection(s) instead of separate visual elements, as is typically the case with block-based programming environments.

4 TOOLKIT EVALUATION

The aim of our evaluation was to assess how younger primary school children (6-10 years old) would use our distributed toolkit to create IoT systems. Moreover, we wanted to capture children's reflections on the connections they made between the various modules.

To this end, we recruited 32 children (16 male and female) from a local primary school aged between seven and ten years old (2-4 grades, M = 8.53, SD = 0.84) for an in-school evaluation. The study was conducted in the school's "social room" where children typically went to read books, play games, and participate in informal craft activities. Of the 32 children, two 4^{th} grade children had prior experience with programming environments.

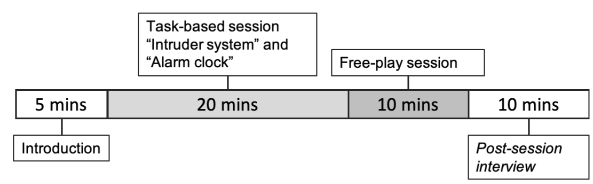

Figure 4: Each session consisted of a brief introduction, two IoT-related tasks, a free play period, and a post-session interview.

We began with a brief introduction of the toolkit and the programming environment to help children explore and familiarize themselves with the system. This was followed by two tasks and an unstructured play segment (Figure 4). For the first task, children had to create a morning alarm system and for the second, an intruder alert system. Children were free to solve the problems in whatever way they chose fit, especially since multiple modules could have been used to realize the solutions. For example, the intruder alert system can be realized through the light sensor as depicted in our sample use case in Figure 1, or through the distance sensor. Similarly, the speaker, motor or RGB LED could have been used to signal an alarm or alert. After children worked on the tasks, they were encouraged to freely explore the modules to express their own ideas (Figure 4). The study concluded with a brief semi-structured interview to help children showcase their creations and gather overall impressions.

To facilitate collaboration and communication during the study, we recommended children to form groups of two. As a result, we had a total of 16 groups, where each group was evaluated separately in the "social room." The study was conducted over the course of five days and it took approximately 45 minutes to evaluate each group. We obtained informed consent from the parents and children and the entire study was approved by our research institute. Children did not receive any compensation for their participation.

4.1 Data and Analysis

During the study, we conducted contextual inquiry while children were working with the toolkit as informed by Beyer and Holtzblatt [6]. We focused on contextual inquiry because we wanted children to take an active role in leading their session by demonstrating and talking about their tasks. Young children, typically have difficulty in describing what their technology needs and wants may be outside of the context they are working in [10]. We also maintained field notes, and video recorded children's screen interactions with the system. We conducted a qualitative analysis of the data using a two-step coding process beginning with descriptive coding stage to highlight relevant material, followed by a more in depth interpretive coding to identify specific patterns [23].

5 RESULTS

We were pleasantly surprised by how engaging our IoT toolkit was with the children. There was also an unanticipated playful aspect to our toolkit that made it quite popular. This was evidenced by our recruitment numbers. When we first started our evaluation, we intended to recruit 20 children (10 groups of 2). However, our initial set of participants enthusiastically advertised our work to their friends and twelve extra children decided to naturally participate. Moreover, during the week we conducted the study, children who had participated, kept returning to the "social room" to express further ideas for the toolkit. One 9-year-old boy offered to buy the kit from us with his €17 life savings.

5.1 Module Coupling

Of the 16 groups that took part in the evaluation, ten groups successfully finished at least one task. Eight of these ten groups were able to solve both the morning alarm and intruder alert tasks. Typical solutions for the morning alarm included coupling the light sensor with the RGB LED or speaker. A few groups used multiple feedback modules to make the alarm more salient. The intruder alert was the harder of the two tasks and was usually solved with the light or distance sensor coupled with the RGB LED or speaker. Two of the groups explored multiple feedback modules for the alert. For example, one group placed the light sensor by the door and used the RGB LED, speaker, and LCD serial display (with the text "ALARM") in conjunction to alert the user.

Of the six groups that had trouble completing both tasks, three understood how to couple sensors and feedback modules only after some guidance or during the free play portion of the evaluation. Even though, we expected age to play a factor here, we did not find it to be the case, as both groups of older and younger children missed the tasks. However, we did observe that children from the second grade had difficulty in understanding the input modules and with establishing connections between modules in contrast to third and fourth graders.

5.2 Understanding IoT

Instantaneous feedback played an important role in helping children better understand the toolkit. We observed a kind of sequential trial and error process where children started with exploring either the input or feedback modules separately and using the real-time feedback to debug any issues they faced. For three groups, this process stopped at just programming the various output modules. They however understood that the tablet was communicating and controlling the feedback modules remotely. In using just the LCD serial display, an 8-year-old boy had the following suggested use case: *"Display is good for home, so one can order food from the kitchen."*

For the other 13 groups, the exploration process continued with the connection of one input to one output. Figure 5 (2nd from left), shows a child trying to figure out the distance sensor by drawing a reference ruler on paper and bringing his hand closer to activate the coupled RGB LED module. By changing the distance sensor threshold values in the programming environment and having it reflect in real-time the child was able to better debug the system. The reference ruler in this case focuses on his understanding of measurements. The connection of one feedback module prompted the connection of others and this led children to try a daisy chaining approach (e.g., light sensor->RGB LED->LCD serial display). This was interpreted as first cube "talking to the second and the second to the third." However, since it did not have the desired effect, children eventually discovered that sensor modules can be connected to multiple outputs.

But perhaps more importantly, we found that children were able to connect the toolkit to smarthome scenarios. We already mentioned the potential use of the LCD serial display for ordering food at home, but children also found the distance sensor and RGB LED combination analogous to automated light systems. One child explained a possible use case, *"Light goes on when you enter a room. It's good for the house"*. Similarly, another 8-year-old girl, related how a similar distance sensor might be used in smart toilets: *"We have the same distance sensor in our toilet. If you come close to the toilet it comes on."*. Some groups further understood that blocks are interconnected and data is exchanged between them. The children reasoned that this should allow them to combine functionalities with each other.

5.3 Personal Games

During the free play segment of the evaluation, children appropriated the toolkit for their own tasks or invented new games. Perhaps the most popular aspect of our toolkit was the LCD serial display in part due to its expressivity. As one child commented, *"I like the display cube the most, because I can write everything that I want."*. Children used the module to display the names of their favorite football players, pets, or even give compliments to their group partner. They also used the LCD serial display to play a version of the "Who am I?" guessing game where the module (with name of animal or person) was held to the forehead instead of a card or piece of paper. Messaging was another popular activity with children; in this scenario one child would hide somewhere within the room with the serial display and the other child would write secret messages through the tablet.

Although these games were simple in nature utilizing one feedback module, there were seven groups that utilized both the input and output modules during free play. In many cases, it was used to better understand the workings of the toolkit. One of the groups

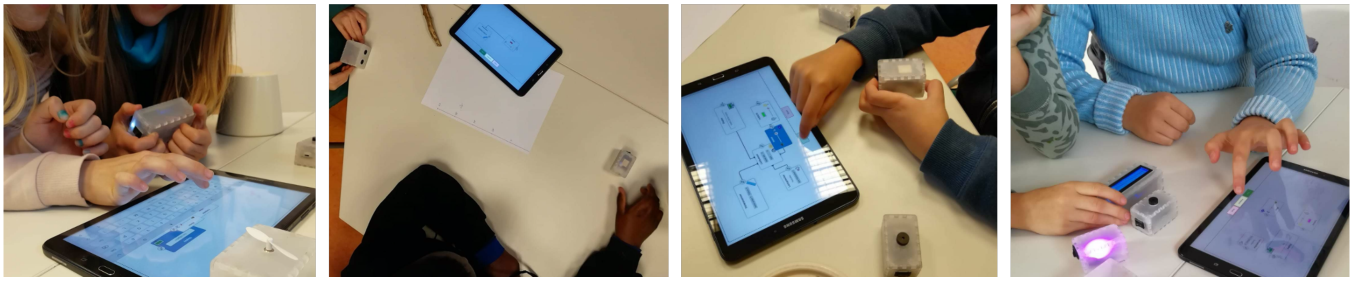

Figure 5: Different examples of toolkit use (from left to right): using the motor module with the distance sensor to make a nail polish dryer, understanding the distance sensor using a reference paper-based ruler, combining multiple modules for an alarm system, and creating a game by combining three feedback modules (LCD-text, sound-module and multi-color light).

(7 and 8 year old boys) which had trouble completing the original tasks, understood the toolkit only during free play. After experimenting with the modules, they used the temperature sensor along with the speaker to compare the ambient temperature inside versus outside. Another group, consisting of a 9-year-old boy and 8-year-old girl, devised a nail polish dryer (Figure 5 left) with the distance sensor, such that as the fingers got closer to the sensor, the motor turned on. One of the more complex examples, which used the logical "AND" operator, utilized the distance and light sensor with the LCD serial display. In this game, when the conditions for both sensor inputs were fulfilled (e.g., put one hand at the right distance and the other at the right height) the LCD display showed "You Won!". We had two other groups that used multiple input sensors and used the logical "AND" operation in this fashion. One of the groups explained it to us as, *"Both conditions have to be fulfilled to make an LED module on the whole time."*

5.4 Toolkit Pitfalls

We observed that children who had trouble completing the tasks had difficulty in understanding the sensors. The reasons for this were two fold: 1) some children had difficulty in identifying the icon symbols on the physical modules 2) there was some confusion that the sensors by themselves did not provide any feedback; they required a connection to be drawn in the visual programming environment to an output module. Consequently, some of the groups remained with exploring the feedback modules. This caused some children to connect the feedback modules together even though it did not accomplish anything programmatically.

One of the groups used two channel connections between each module block within the programming environment instead of just drawing one connection. They explained this to us as: *"This is so each module can communicate. If we only use one connection, only one of the modules can share information."* This can be seen as a problem with the way we visualize information flow within the application. However, it does indicate an understanding of communication between various distributed blocks that need to exchange information in order to work properly.

The toolkit was also limited in expressivity. With the RGB-LED module for example, a child could pick the brightness and color, but there was no way to blink patterns. Although our design was purposefully simplistic to teach children basic IoT concepts, participants in our study often requested enhancements or proposed newer modules. For example, one 7-year-old girl requested adding a color picker for sampling colors from the real world. Generally the use of the toolkit sparked creative ideas for further blocks that should be included in the kit. For example, one group asked for a camera block that would transfer pictures to a wireless "TV-Block" to see recorded pictures and videos. Although this was a good suggestion for an IoT application, most of the suggestions were for singular cubes based on personal interests. Some of these included a keyboard cube to enter text into the display cube, a magnet-cube, a math-cube to do calculations, a flying cube (inspired by the motor cube with the propeller), and a music-cube with different notes to compose a melody.

6 DISCUSSION

Our evaluation provided promising results in support of the distributed IoT toolkit for elementary school children. Over half the participants successfully completed at least one task and of the 6 groups that missed both tasks, 3 groups made logical module connections during the free play portion of the study. In this section we discuss the implications of our work particularly with respect to IoT thinking, the tradeoffs in our design, and the value of instantaneous feedback.

6.1 Towards IoT Thinking

The idea of interrelated computing devices that gather data and communicate wirelessly with each other may seem obvious to adults in the modern world. However, to many younger children the basic concepts of how this may be accomplished are far from transparent. In our study we found evidence of basic IoT thinking, where children linked sensors to feedback modalities through rule-based systems, not unlike what adults may do with real IoT systems. For example, an adult may program electronic blinds to automatically open when light is sensed outside. Similarly, children in our study utilized the temperature sensor to activate the speaker when they were outside. Moreover, we were encouraged that some groups were able to make the logical connection between the kit and various smarthome scenarios without our prompting. One group even highlighted the need for full-duplex communication between different modules. These examples showcase aspects of the definition of IoT (for children) proposed in subsection 2.1.

Although we had some measure of success with the toolkit, the sensor modules were sometimes simply hard to understand for a few children. This could have been because the tasks we designed were not representative of what children were actually interested in or would design on their own. For three groups, the free play portion proved to be more valuable in learning the toolkit than the actual tasks. A form of guided discovery, where the facilitator stimulates the learner in active inquiry and discovery [29], may be more appropriate with some children.

It may also have been that the lack of immediate feedback with the input modules, did not provide the necessary sensory engagement [40] for an easy entry point [37]. Another approach would have been to provide real-time sensor values as abstracted animated icons in the visual programming environment. For example, the data from the distance sensor can be depicted as an animated ruler which changes in size based on real-time interactions. This would lead to a more transparent interface one that is focused on revealing more of the inner workings of the system rather than a black box.

6.2 Between Transparency and Magic

A possible critique of our system is that it favors automation in place of transparency. By "selectively hiding" the electronics and programming syntax, it was much too easy to establish interconnected modules, resulting in less personal reflection. For the few children who had trouble with the kit, the feedback modules alone were compelling and engaging, preventing further exploration and problem solving. Moreover, the toolkit was considered a toy by many children, as evidenced by the free play portion of the study. This highlights a trade-off between making things work "like magic" and exposing all the details, an aspect of design that needs to be carefully considered in future implementations.

Indeed, as Gross and Eisenberg have pointed out, there is a "fine balance between eliminating needless complexity to make a more elegant design environment, and hiding important detail in the name of ease-of-use" [14]. In essence, the technological environment must encourage curiosity and mastery, and mastery almost always implies a struggle or a challenge. However, if the experience is too challenging then children may get frustrated. What is needed is a kind of "pleasant frustration" [12], where challenges feel hard but doable. This balance is indeed hard to strike with younger children.

6.3 Debugging by Design

The real-time nature of our toolkit was particularly salient in the sequential trial and error process children used in exploring the modules. Consequently, the instantaneous feedback, seemed to promote debugging naturally. Similar to other studies in this area, children were less concerned about saving their programs and focused more on interacting with the device [9]. We believe this was in part due to the absence of the traditional program-compile-upload-debug cycle. As recent research in this area has shown live programming seemed to prompt shorter more numerous interactions with the physical device, and supported a more incremental development process [9]. In recent years, researchers have highlighted the importance of debugging in the learning process for children and students. The idea of debugging as a kind of "productive failure" has been highlighted not only in computer science education [18] but also in physical computation [30]. Therefore, real-time feedback and programming may hold promise in promoting debugging activities in children, as a way to help them spend more time on tasks central to learning, rather than on the technicalities of making things work.

7 LIMITATIONS & FUTURE WORK

In our current implementation, the individual modules are somewhat bigger in size since we used off-the-shelf components. While this can be an advantage, especially for younger children, in handling the modules during play, it restricts the integration of the modules into crafts or body-worn creations. Future versions of the toolkit can be made smaller by designing our own custom PCBs that integrates the electronics on a single board. This would also help us in producing the new modules children requested more quickly (e.g., color picker, camera block, music cube).

From a software perspective, the current visual programming environment is somewhat limited in expression. Our goal was to provide a low barrier of entry for interconnecting modules and while we were successful in this aspect, it could certainly be improved. Supporting more logical operators would be a good starting point but also more explicit communication options between modules and external web services. A major part of modern IoT that is missing in our implementation is the integration of external web-services to provide additional contextual data such as weather information. This provides a more complete view of IoT and we imagine these services as "virtual building blocks" that can be integrated as easily as tangible blocks. However, more research needs to be conducted on how best to integrate these blocks in an understandable way to primary school children.

Lastly, our study lacks a formal evaluation of children's understanding of IoT. Based on our working definition, we used specific representative tasks and informal contextual interviews during the sessions to see how children would use our toolkit. In many ways, our work in IoT is at a very early stage. Although, we consider our work exploratory, future work will need to establish a more formal idea of what IoT concepts are applicable to younger children. These concepts and evaluation tools may need to be tailored based on age-related differences.

8 CONCLUSION

In this paper, we presented the design and implementation of a distributed toolkit to support children with exploring IoT concepts in a hands-on manner. We reported findings of our exploratory study with 32 school children investigating the types of activities children engage in with this toolkit. Our results suggest that children showed an understanding of basic IoT by linking devices using rule-based systems and started to connect scenarios with home automation systems. The real-time nature of our toolkit was particularly helpful in fostering debugging among children, particularly when exploring the modules. Given the rise of IoT technologies and the complexity created by interconnecting these devices and services, our "ready to use" approach focuses on exposing IoT concepts in an age-appropriate modern way.

ACKNOWLEDGMENTS

We would like to thank the primary school "Grundschule am Wiesengrund" for supporting us with our research and all the children who participated in our study.

REFERENCES

[1] Farha Ali. 2015. Teaching The Internet of Things Concepts. In *Proceedings of the WESE'15: Workshop on Embedded and Cyber-Physical Systems Education* (Amsterdam, Netherlands) *(WESE'15)*. ACM, New York, NY, USA, Article 10, 6 pages. https://doi.org/10.1145/2832920.2832930

[2] Swamy Ananthanarayan and Susanne Boll. 2020. Physical Computing for Children: Shifting the Pendulum Back to Papertian Ideals. *Interactions* 27, 3 (April 2020), 40–45. https://doi.org/10.1145/3386235

[3] Ayah Bdeir and Ted Ullrich. 2011. Electronics As Material: LittleBits. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction* (Funchal, Portugal) *(TEI '11)*. ACM, New York, NY, USA, 341–344. https://doi.org/10.1145/1935701.1935781

[4] J. Bernstein. 1993. *Cranks, Quarks, and the Cosmos: Writings on Science*. BasicBooks. https://books.google.de/books?id=i8LaAAAAMAAJ

[5] Marina Umaschi Bers and Michael S. Horn. 2010. *Tangible Programming in Early Childhood: Revisiting Developmental Assumptions Through New Technologies: Childhood in a Digital World*. Information Age Publishing.

[6] Hugh Beyer and Karen Holtzblatt. 1998. *Contextual Design: Defining Customer-centered Systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[7] Paulo Blikstein. 2013. Gears of Our Childhood: Constructionist Toolkits, Robotics, and Physical Computing, Past and Future. In *Proceedings of the 12th International Conference on Interaction Design and Children* (New York, New York, USA) *(IDC '13)*. ACM, New York, NY, USA, 173–182. https://doi.org/10.1145/2485760.2485786

[8] Leah Buechley and Benjamin Mako Hill. 2010. LilyPad in the Wild: How Hardware's Long Tail is Supporting New Engineering and Design Communities. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems* (Aarhus, Denmark) *(DIS '10)*. ACM, New York, NY, USA, 199–207. https://doi.org/10.1145/1858171.1858206

[9] Lautaro Cabrera, John H. Maloney, and David Weintrop. 2019. Programs in the Palm of Your Hand: How Live Programming Shapes Children's Interactions with Physical Computing Devices. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children* (Boise, ID, USA) *(IDC '19)*. Association for Computing Machinery, New York, NY, USA, 227–236. https://doi.org/10.1145/3311927.3323138

[10] Allison Druin, Ben Bederson, Angela Boltman, Adrian Miura, Debby Knotts-Callahan, and Mark Platt. 1998. *Children as Our Technology Design Partners*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 51–72.

[11] Louise P. Flannery, Brian Silverman, Elizabeth R. Kazakoff, Marina Umaschi Bers, Paula Bontá, and Mitchel Resnick. 2013. Designing ScratchJr: Support for Early Childhood Learning Through Computer Programming. In *Proceedings of the 12th International Conference on Interaction Design and Children* (New York, New York, USA) *(IDC '13)*. ACM, New York, NY, USA, 1–10. https://doi.org/10.1145/2485760.2485785

[12] James Paul Gee. 2005. Learning by Design: Good Video Games as Learning Machines. *E-Learning and Digital Media* 2, 1 (2005), 5–16. https://doi.org/10.2304/elea.2005.2.1.5

[13] Rosella Gennari, Alessandra Melonio, Mehdi Rizvi, and Andrea Bonani. 2017. Design of IoT Tangibles for Primary Schools: A Case Study. In *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter* (Cagliari, Italy) *(CHItaly '17)*. Association for Computing Machinery, New York, NY, USA, Article 26, 6 pages. https://doi.org/10.1145/3125571.3125591

[14] Mark D. Gross and Michael Eisenberg. 2007. Why Toys Shouldn'T Work "Like Magic": Children's Technology and the Values of Construction and Control. In *Proceedings of the The First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL '07)*. IEEE Computer Society, Washington, DC, USA, 25–32. https://doi.org/10.1109/DIGITEL.2007.55

[15] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems* 29, 7 (2013), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

[16] Michael S. Horn, R. Jordan Crouser, and Marina U. Bers. 2012. Tangible Interaction and Learning: The Case for a Hybrid Approach. *Personal Ubiquitous Comput.* 16, 4 (April 2012), 379–389. https://doi.org/10.1007/s00779-011-0404-2

[17] Felix Hu, Ariel Zekelman, Michael Horn, and Frances Judd. 2015. Strawbies: Explorations in Tangible Programming. In *Proceedings of the 14th International Conference on Interaction Design and Children* (Boston, Massachusetts) *(IDC '15)*. ACM, New York, NY, USA, 410–413. https://doi.org/10.1145/2771839.2771866

[18] Yasmin B. Kafai, David DeLiema, Deborah A. Fields, Gary Lewandowski, and Colleen Lewis. 2019. Rethinking Debugging As Productive Failure for CS Education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. ACM, New York, NY, USA, 169–170. https://doi.org/10.1145/3287324.3287333

[19] Eva-Sophie Katterfeldt, Mutlu Cukurova, Daniel Spikol, and David Cuartielles. 2018. Physical computing with plug-and-play toolkits:Key recommendations for collaborative learning implementations. *International Journal of Child-Computer Interaction* 17 (2018), 72 – 82. https://doi.org/10.1016/j.ijcci.2018.03.002

[20] Eva-Sophie Katterfeldt, Nadine Dittert, and Heidi Schelhowe. 2009. EduWear: Smart Textiles As Ways of Relating Computing Technology to Everyday Life. In *Proceedings of the 8th International Conference on Interaction Design and Children* (Como, Italy) *(IDC '09)*. ACM, New York, NY, USA, 9–17. https://doi.org/10.1145/1551788.1551791

[21] Majeed Kazemitabaar, Jason McPeak, Alexander Jiao, Liang He, Thomas Outing, and Jon E. Froehlich. 2017. MakerWear: A Tangible Approach to Interactive Wearable Creation for Children. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 133–145. https://doi.org/10.1145/3025453.3025887

[22] Caitlin Kelleher and Randy Pausch. 2005. Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. *ACM Comput. Surv.* 37, 2 (June 2005), 83–137. https://doi.org/10.1145/1089733.1089734

[23] N. King and C. Horrocks. 2010. *Interviews in Qualitative Research*. SAGE Publications.

[24] G. Kortuem, A. K. Bandara, N. Smith, M. Richards, and M. Petre. 2013. Educating the Internet-of-Things Generation. *Computer* 46, 2 (Feb 2013), 53–61. https://doi.org/10.1109/MC.2012.390

[25] Stan Kurkovsky and Chad Williams. 2017. Raspberry Pi As a Platform for the Internet of Things Projects: Experiences and Lessons. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education* (Bologna, Italy) *(ITiCSE '17)*. ACM, New York, NY, USA, 64–69. https://doi.org/10.1145/3059009.3059028

[26] Zuzanna Lechelt, Yvonne Rogers, Nicolai Marquardt, and Venus Shum. 2016. ConnectUs: A New Toolkit for Teaching About the Internet of Things. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. ACM, New York, NY, USA, 3711–3714. https://doi.org/10.1145/2851581.2890241

[27] Zuzanna Lechelt, Yvonne Rogers, Nicolai Marquardt, and Venus Shum. 2016. Democratizing Children's Engagement with the Internet of Things Through connectUs. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (Heidelberg, Germany) *(UbiComp '16)*. ACM, New York, NY, USA, 133–136. https://doi.org/10.1145/2968219.2971435

[28] Zuzanna Lechelt, Yvonne Rogers, Nicola Yuill, Lena Nagl, Grazia Ragone, and Nicolai Marquardt. 2018. Inclusive Computing in Special Needs Classrooms: Designing for All. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 517, 12 pages. https://doi.org/10.1145/3173574.3174091

[29] Detlev Leutner. 1993. Guided discovery learning with computer-based simulation games: Effects of adaptive and non-adaptive instructional support. *Learning and Instruction* 3, 2 (1993), 113 – 132. https://doi.org/10.1016/0959-4752(93)90011-N

[30] Debora Lui, Emma Anderson, Yasmin B. Kafai, and Gayithri Jayathirtha. 2017. Learning by Fixing and Designing Problems: A Reconstruction Kit for Debugging E-Textiles. In *Proceedings of the 7th Annual Conference on Creativity and Fabrication in Education* (Stanford, CA, USA) *(FabLearn '17)*. ACM, New York, NY, USA, Article 6, 8 pages. https://doi.org/10.1145/3141798.3141805

[31] Grace Ngai, Stephen C.F. Chan, Hong Va Leong, and Vincent T.Y. Ng. 2013. Designing I*CATch: A Multipurpose, Education-friendly Construction Kit for Physical and Wearable Computing. *Trans. Comput. Educ.* 13, 2, Article 7 (July 2013), 30 pages. https://doi.org/10.1145/2483710.2483712

[32] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. Scratch: Programming for All. *Commun. ACM* 52, 11 (Nov. 2009), 60–67. https://doi.org/10.1145/1592761.1592779

[33] Eric Schweikardt and Mark D. Gross. 2006. roBlocks: A Robotic Construction Kit for Mathematics and Science Education. In *Proceedings of the 8th International Conference on Multimodal Interfaces* (Banff, Alberta, Canada) *(ICMI '06)*. ACM, New York, NY, USA, 72–75. https://doi.org/10.1145/1180995.1181010

[34] Sue Sentance, Jane Waite, Steve Hodges, Emily MacLeod, and Lucy Yeomans. 2017. "Creating Cool Stuff": Pupils' Experience of the BBC Micro:Bit. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. ACM, New York, NY, USA, 531–536. https://doi.org/10.1145/3017680.3017749

[35] Bill Siever and Michael P. Rogers. 2019. Micro: Bit Magic: Engaging K-12, CS1/2, and Non-majors with IoT & Embedded. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. ACM, New York, NY, USA, 1237–1238. https://doi.org/10.1145/3287324.3287527

[36] David Canfield Smith, Allen Cypher, and Jim Spohrer. 1994. KidSim: Programming Agents Without a Programming Language. *Commun. ACM* 37, 7 (July 1994), 54–67. https://doi.org/10.1145/176789.176795

[37] Amanda Strawhacker and Marina U. Bers. 2015. "I want my robot to look for food": Comparing Kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces. *International Journal of Technology and Design Education* 25, 3 (01 Aug 2015), 293–319. https://doi.org/10.1007/s10798-014-9287-7

[38] Amanda Sullivan, Mollie Elkin, and Marina Umaschi Bers. 2015. KIBO Robot Demo: Engaging Young Children in Programming and Engineering. In *Proceedings of the 14th International Conference on Interaction Design and Children* (Boston, Massachusetts) *(IDC '15)*. ACM, New York, NY, USA, 418–421. https://doi.org/10.1145/2771839.2771868

[39] Junnan Yu and Ricarose Roque. 2018. A Survey of Computational Kits for Young Children. In *Proceedings of the 17th ACM Conference on Interaction Design and Children* (Trondheim, Norway) *(IDC '18)*. ACM, New York, NY, USA, 289–299. https://doi.org/10.1145/3202185.3202738

[40] Oren Zuckerman, Saeed Arida, and Mitchel Resnick. 2005. Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. ACM, New York, NY, USA, 859–868. https://doi.org/10.1145/1054972.1055093