The IoT Codex: A Book of Paper Engineering Techniques for Authoring and Composing Embedded Computing Applications

Kristin Williams* **Jessica Hammer*,†**
Scott E. Hudson*

November 2020
CMU-ISR-20-115C

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This paper was presented at PLATEAU 2020:
The 11th Annual Workshop on the Intersection of HCI and PL
November 2020, Co-located with SPLASH 2020

*Human Computer Interaction Institute, Carnegie Mellon University, USA
†Entertainment Technology Center, Carnegie Mellon University, USA

Keywords: Paper computing, Internet-of-things, IoT, Internet of Things, End user programming, Smart home, Tangible interfaces, Physical computing, Interactive Book, Craft, Stickers

Abstract

End user programming (EUP) for the smart home has become widely available in commercial systems through services like If This, Then That (IFTTT). This approach uses graphical user interfaces (GUIs) to encapsulate control flow constructs like if-then conditionals in easier-to-use abstractions like form filling. Yet, non-experts largely do not use them to author their own recipes, and instead, naively rely on experts to compose those they do use (often risking their own domestic privacy and security). To support non-experts with authoring and composing embedded computing applications for their home, this work adapts paper engineering techniques to create alternative abstractions and interaction techniques in the form of a book: the IoT Codex. This book contributes a suite of paper mechanisms to convey the affordances of an EUP language for domestic IoT. These mechanisms constrain what expressions can be composed in the language by using both shape and kinematics. At the same time, they encourage non-experts to tinker and experiment with physical artifacts to develop programming expertise using material properties such as adhesives, wax, gold leaf, and others common to scrapbooks. In doing so, the resulting IoT Codex contributes physical computing techniques to aid non-experts in selecting appropriate EUP elements and remixing them to realize their own IoT ideas.

The IoT Codex: A Book of Paper Engineering Techniques for Authoring and Composing Embedded Computing Applications

Kristin Williams
Human Computer Interaction Institute, Carnegie Mellon University, USA
http://www.cs.cmu.edu/~kmwillia
krismawil@cs.cmu.edu

Jessica Hammer
Human Computer Interaction Institute, Carnegie Mellon University, USA
Entertainment Technology Center, Carnegie Mellon University, USA
hammerj@andrew.cmu.edu

Scott E. Hudson
Human Computer Interaction Institute, Carnegie Mellon University, USA
scott.hudson@cs.cmu.edu

Abstract

End user programming (EUP) for the smart home has become widely available in commercial systems through services like If This, Then That (IFTTT). This approach uses graphical user interfaces (GUIs) to encapsulate control flow constructs like if-then conditionals in easier-to-use abstractions like form filling. Yet, non-experts largely do not use them to author their own recipes, and instead, naively rely on experts to compose those they do use (often risking their own domestic privacy and security). To support non-experts with authoring and composing embedded computing applications for their home, this work adapts paper engineering techniques to create alternative abstractions and interaction techniques in the form of a book: the IoT Codex. This book contributes a suite of paper mechanisms to convey the affordances of an EUP language for domestic IoT. These mechanisms constrain what expressions can be composed in the language by using both shape and kinematics. At the same time, they encourage non-experts to tinker and experiment with physical artifacts to develop programming expertise using material properties such as adhesives, wax, gold leaf, and others common to scrapbooks. In doing so, the resulting IoT Codex contributes physical computing techniques to aid non-experts in selecting appropriate EUP elements and remixing them to realize their own IoT ideas.

2012 ACM Subject Classification Human-centered computing → User interface programming; Human-centered computing → Interaction devices; Human-centered computing → Ubiquitous and mobile devices

Keywords and phrases Paper computing, Internet-of-things, IoT, Internet of Things, End user programming, Smart home, Tangible interfaces, Physical computing, Interactive Book, Craft, Stickers

Acknowledgements We would like to thank Rajitha Pulivarthy for her help with designing and fabricating some of the stickers presented here. This work was funded in part by National Science Foundation grant IIS-1718651.

1 Introduction

End user programming (EUP) for the smart home has become widely available in commercial systems through services like If This, Then That (IFTTT). These systems make EUP tools for home automation available to a larger audience at a lower price point than previously available [15, 16]. Prior work shows that the trigger-action programming model of systems

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:6

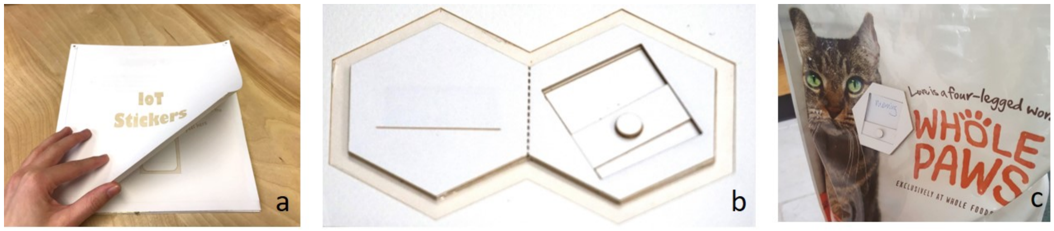

Figure 1 We introduce the IoT Codex: a book of inexpensive, battery-free sensors and interaction patterns to support linking everyday objects to software and web services using stickers (a). To use, a sticker is first peeled off from the book (b) and attached to a desired everyday object (c). Stickers can be customized during setup and composed with others to create more expressive applications using the composition space in the book's pages.

like IFTTT make programming Internet of Things devices available to non-experts [2, 10]. This approach uses graphical user interfaces (GUIs) encapsulate control flow constructs like if-then rules within user friendly abstractions like form filling [2, 10]. As a result of its widespread availability and preliminary successes, trigger-action programming has received substantial attention from researchers creating EUP systems for domestic IoT (*e.g.,* [9, 4, 1]).

Yet, trigger-action programming's expressive power likewise creates barriers for novice programmers. Choosing between event triggers (signalling change in a sensor reading at a point in time) and condition triggers (detecting states, context, or when criteria are met) introduced many problems [4, 7]. One study found that selection of triggers and actions had the lowest usability ratings of all measured dimensions [4]. This may be due to if-then rules ambiguously covering both events and conditions [7].

Interactive books offer promising alternative abstractions for trigger-action programming. Paper pop-up mechanisms housed in a book form factor can support users with manipulating and triggering sensors using tangible abstractions [11]. Similarly, sticker-based interactive books can support tinkering with circuits [12], peel and stick construction of circuits [5], and hand crafting remote messaging [3]. Sticker composition can introduce programming concepts like sequencing actions [6]. These books borrow techniques from GUIs that use shape to constrain permissible compositions in the language [6, 13]. For example, slots in a book constrain how physical cards can be inserted [14]. Paper tags, such as visual codes, enable dynamically mapping triggers to actions to co-evolve alongside familiar scrapbooking practices in a cognitively accessible form [8]. These paper mechanisms can thus support proprioceptive control over the material environment in a way that designs for agency in new technologies [8]. Paper mechanisms promise to facilitate a wider audience's exploration and expression of computational ideas by concretizing an EUP language's abstractions. By leveraging familiar paper engineering techniques, interactive books provide a suite of paper mechanisms that can expand participation in end user programming by attending to the material constraints that make some actions available while excluding others.

This work adapts paper engineering techniques to end user programming needs for domestic IoT. In doing so, it creates alternative abstractions and interaction techniques in the form of a book to support non-experts with authoring and composing embedded computer applications. Using off-the-shelf RFID tags and a long-range RFID reader, the book structures interactions around RFID augmented stickers using paper engineering techniques to encapsulate programming constructs like setting variables, reading data from a sensor, wrappers, if-then statements, and for loops. The book makes programming composition approachable to non-experts by using the codex form factor to scaffold introduction of these

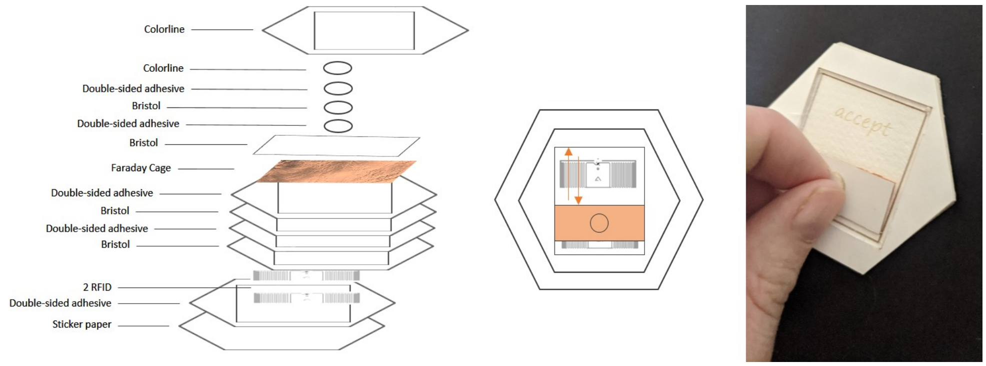

Figure 2 Above is an exploded view of the constuction of the toggle sticker, a diagram of how the slider mechanism selectively blocks one of the toggle's two tags, and lastly is a picture of a user moving the sticker's slider.

concepts and constrain how expressions can be composed.

2 The IoT Codex

Using RFID-based paper mechanisms, the IoT Codex integrates common sensors and web services with interactive stickers using a trigger-action architecture. This integration enables computational services to be tailored to everyday activities by attaching a sticker to the activity's associated object, setting parameters to be passed to the sticker's actions, and even composing the stickers together. Thus, authoring an embedded computing application for the home becomes as approachable to non-experts as attaching one of our interactive stickers to a desired object (Figure 1).

In line with other rule based systems, IoT stickers supports novice programmer composition by allowing differing rules—or, in this case, stickers—to be combined together. To facilitate combination, stickers are shaped in a way that suggests they can fit together using a hexagonal primitive (similar to the way puzzle piece metaphors work for other languages). For some stickers—dials and wrappers (described below)—stickers must be composed together in order for the mechanism to function.

2.1 Sticker Types

IoT Stickers use a Trigger-Action paradigm to support a programming structure approachable to novice programmers. Specifically, small if-then programs can cover a substantial amount of programming needs in an accessible form. We support these building blocks by using RFID stickers to scaffold construction of conditional statements. We do so by encapsulating one or more if-then rules in each Sticker. Manipulating the sticker's paper mechanism changes whether an RFID tag responds to a reader and serves as a trigger within our system for actions useful to the user. Stickers handle the mapping between triggers and actions to minimize trigger selection errors while still enabling user customization through sticker setup and composition. Unlike other trigger-action systems, stickers concretize conditionals in a tangible form.

Stickers bridge the low-level RFID reading and high-level software actions important for a user by using a unique ID to find and invoke the relevant actions for that sticker. This happens through the system's low level sensing of the tag, its interpretation of the tag's

Figure 3 Above is a view of each sticker type on its own page in the IoT Codex. From left to right: button, toggle, list, dial, and wrapper.

identity, gathering of the specific platform resources that will be used, and finally, invoking the desired action. For example, a Sticker could encapsulate the following rule: if one RFID associated with a toggle switch Sticker disappeared, while another appeared (indicating that the switch has shifted positions as in Figure 2), then a state change should be logged. This could achieve higher level use cases like tracking whether a pet cat had been fed (as depicted in Figure 1). 5 different basic sticker types are currently supported by the system: a button, a toggle, a list, a dial, and a wrapper (these are detailed more below).

The interactive book is subdivided into sections according to sticker type, with a progression from simple to more complex capabilities. In each section, there is an introductory page describing how the sticker works, followed by a few pages with example sticker applications. The last half of the book contains sections that use rooms and room-typical objects to support users with aligning their IoT ideas with the capabilities of the sticker system (see [17, 18] for the importance of rooms and objects as IoT cues).

Button Sticker. Button stickers are the most basic sticker type in our system. Buttons produce a single trigger and are invoked by covering the button sticker with the hand. A person activates the button sticker by peeling the sticker from the page which initiates the sticker setup process.

Toggle Sticker. The toggle switch sticker has two states. A trigger for the toggle is activated by moving the sliding cover to newly reveal the RFID tag beneath it while at the same time covering (and disabling) another tag.

List Sticker. The list sticker consists in 3 RFID tags in a row covered by pop-up style flaps. This supports a user in individually associating an action with each tag and allowing for greater complexity in the relationship between those tags if the user desires. Since more than one RFID tag can be activated at a time with the list sticker, more complex situations can be managed with this tag. For example, partially fulfilled tasks could be managed and tracked with the list sticker like a To Do list.

Dial Sticker. The dial sticker consists of up to 6 RFID tags fanned out in a circle and selectively activated when covered with a moving arm lined with foil. The dial sticker supports interactions that require a selection between a small set of values or states. This could include controlling the place of play in a video/audio file, manipulating rotation of a 3D object, or scrolling. Dial stickers are also supportive of cases where alternatives describe social relationships rather than the system's objects, such as identifying who in a set of people is responsible for a household chore.

Wrapper Sticker. The wrapper sticker consists of a larger shaped sticker that contains an area big enough for a button sticker. This supports the user in composing together wrappers and stickers to provide additional control and functionality over a sticker's actions. Wrappers can help with scheduling future behavior as with a timer, or creating a multiplicative effect such as increasing the magnitude of a counting action.

2.2 Sticker Setup

The setup process of each sticker varies according to its type and whether it is a pre-programmed sticker, pattern sticker, or blank sticker. The pre-programmed sticker is meant to cover cases where the sticker's functionality provides value by enabling an existing service to be linked to the physical world. The pattern sticker links sticker Triggers to specific Actions, but leaves customization open by supporting end user specification of variables or parameters. Blank stickers support choosing which supported pattern to associate with the sticker, and then customizing that pattern further as desired.

3 Conclusion

To support non-experts with authoring and composing embedded computing applications for their home, this work describes how paper engineering techniques and the book form factor can be used to create alternative abstractions and interaction techniques for trigger-action programming. This paper contributes a suite of paper mechanisms to convey the affordances of an EUP language for domestic IoT. These mechanisms constrain what expressions can be composed in the language by using both shape and kinematics. In doing so, the resulting IoT Codex contributes physical computing techniques to aid non-experts in selecting appropriate EUP elements and remixing them to realize their own IoT ideas.

References

1. Julia Brich, Marcel Walch, Michael Rietzler, Michael Weber, and Florian Schaub. Exploring end user programming needs in home automation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(2):11, 2017.
2. Anind K. Dey, Timothy Sohn, Sara Streng, and Justin Kodama. icap: Interactive prototyping of context aware applications. In *Proceedings of the 4th International Conference on Pervasive Computing*, pages 254–271, Dublin, Ireland, 2006.
3. Natalie Freed, Jie Qi, Adam Setapen, Cynthia Breazeal, Leah Buechley, and Hayes Raffle. Sticking together: handcrafting personalized communication interfaces. In *Proceedings of the 10th International Conference on Interaction Design and Children*, pages 238–241, 2011.
4. Giuseppe Ghiani, Marco Manca, Fabio Paternò, and Carmen Santoro. Personalization of context-dependent applications through trigger-action rules. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(2):14, 2017.
5. Steve Hodges, Nicolas Villar, Nicholas Chen, Tushar Chugh, Jie Qi, Diana Nowacka, and Yoshihiro Kawahara. Circuit stickers: peel-and-stick construction of interactive electronic prototypes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1743–1746. ACM, 2014.
6. Michael S Horn, Sarah AlSulaiman, and Jaime Koh. Translating roberto to omar: computational literacy, stickerbooks, and cultural forms. In *Proceedings of the 12th International Conference on Interaction Design and Children*, pages 120–127, 2013.
7. Justin Huang and Maya Cakmak. Supporting mental model accuracy in trigger-action programming. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 215–225, 2015.
8. Amanda Lazar, Caroline Edasis, and Anne Marie Piper. Supporting people with dementia in digital social sharing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2149–2162, 2017.
9. Sarah Mennicken and Elaine M. Huang. Hacking the natural habitat: an in-the-wild study of smart homes, their development, and the people who live in them. In *Proceedings of the 10th international conference on Pervasive Computing*, pages 143–160, Berlin, Germany, 2012. Springer.

10 John F Pane and Brad A Myers. Tabular and textual methods for selecting objects from a group. In *Proceeding 2000 IEEE International Symposium on Visual Languages*, pages 157–164. IEEE, 2000.

11 Jie Qi and Leah Buechley. Electronic popables: exploring paper-based computing through an interactive pop-up book. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pages 121–128. ACM, 2010.

12 Jie Qi and Leah Buechley. Sketching in circuits: designing and building electronics on paper. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1713–1722, 2014.

13 Jie Qi, Andrew" bunnie" Huang, and Joseph Paradiso. Crafting technology with circuit stickers. In *Proceedings of the 14th International Conference on Interaction Design and Children*, pages 438–441, 2015.

14 Cristina Sylla, Pedro Branco, Sérgio Gonçalves, Clara Coutinho, and Paulo Brito. T-books: merging traditional storybooks with electronics. In *Proceedings of the 11th International Conference on Interaction Design and Children*, pages 323–326, 2012.

15 Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L Littman. Practical trigger-action programming in the smart home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 803–812, 2014.

16 Blase Ur, Melwyn Pak Yong Ho, Stephen Brawner, Jiyun Lee, Sarah Mennicken, Noah Picard, Diane Schulze, and Michael L Littman. Trigger-action programming in the wild: An analysis of 200,000 ifttt recipes. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3227–3231. ACM, 2016.

17 Kristin Williams, Rajitha Pulivarthy, Scott E Hudson, and Jessica Hammer. Understanding family collaboration around lightweight modification of everyday objects in the home. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–24, 2019.

18 Kristin Williams, Rajitha Pulivarthy, Scott E Hudson, and Jessica Hammer. The upcycled home: Removing barriers to lightweight modification of the home's everyday objects. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.