ENTREPRENEURIAL THINKING IN THE DESIGN OF UBIQUITOUS COMPUTING

ALEMAYEHU (TEDDY) SEYED

Doctor of Philosophy

Department of Computer Science

Faculty of Science

University of Calgary

August 2019

ABSTRACT

From the smart assistant in a home providing the daily news, to the smart-glasses that notify you about your next meeting, Ubiquitous computing has arrived and is here to stay. However, despite our inherent dependence on ubiquitous technologies, a number of challenges still remain, such as how do we seamlessly interact with these environments using our everyday devices, to how do we provide them with context for interactions with ourselves and our data.

My dissertation work is concerned with (1) overcoming challenges in how Ubiquitous computing are designed and how we interact with them using our everyday devices, (2) if and how we can redesign these devices to better fit their context of use in these environments, and (3) how can we enable designers and novices to contribute to the field of uUbiquitous computing environments.

Moving beyond the research work for this dissertation, I also provide entrepreneurial reflections in each of the aforementioned areas, where I describe my journey and key lessons learned from working in a startup to co-founding multiple startups.

THE USAGE OF 'WE' AND 'I'

As both a researcher and an entrepreneur, collaboration is a fundamental aspect of creating impactful work. The key projects presented in this dissertation are projects that I led, and involved a number of collaborators. Without the support and contributions of my supervisors, collaborators and mentors, both on the research and entrepreneurial side, I would not have been able to thread this thesis together. As a result, I use the plural ‘we’ to describe collaborative efforts with my co-authors on the research side and co-founders on the entrepreneurial side. Additionally, for the entrepreneurial sections of this dissertation, ‘I’ is used in the context of my personal and reflective commentary as an entrepreneur and the journey that was undertaken.

PUBLICATIONS

In the duration of my time in the PhD, particularly on the research side, I have published or helped shape research projects in a large number of areas. The areas in which I have published or contributed to mirror those which are later discussed in this dissertation.

In the subsequent pages, I first provide the complete list of papers that were a part of the research side and journey in my dissertation, categorized by the aforementioned research areas. Not all of these papers and their materials have been used in this dissertation, but I have highlighted those which are. I have included some of these to provide context that while I may have focused on entrepreneurial tasks more heavily at points in my PhD, I was still contributing to several areas. Later, I follow up this with a visual summary of where each of the highlighted papers (and their associated materials) fit into the dissertation.

Exploring Ubiquitous Computing Environments

Teddy Seyed, Mario Costa Sousa, Frank Maurer, and Anthony Tang. 2013. SkyHunter: a multi-surface environment for supporting oil and gas exploration. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces (ITS '13).* ACM, New York, NY, USA, 15-22.

Apoorve Chokshi, Teddy Seyed, Francisco Marinho Rodrigues, and Frank Maurer. 2014. ePlan Multi-Surface: A Multi-Surface Environment for Emergency Response Planning Exercises. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops*

and Surfaces (ITS '14). ACM, New York, NY, USA, 219-228.

Teddy Seyed, Alaa Azazi, Edwin Chan, Yuxi Wang, and Frank Maurer. 2015. SoD-Toolkit: A Toolkit for Interactively Prototyping and Developing Multi-Sensor, Multi-Device Environments. In *Proceedings of the 2015 International Conference on Interactive Tabletops and Surfaces (ITS '15).* ACM, New York, NY, USA, 171-180.

Teddy Seyed, Frank Maurer (2014). The Geospatial Domain & Multi-Surface Environments. In Proceedings of the Open Geospatial Consortium Academic Summit 2014 (OGC 2014), Calgary, AB, Canada.

Novel Devices and Interactions

Teddy Seyed, Xing-Dong Yang, Anthony Tang, Saul Greenberg, Jiawei Gu, Bin Zhu, and Xiang Cao. "CipherCard: A Token-based Approach against Camera-based Shoulder Surfing Attacks on Common Touchscreen Devices." In *IFIP Conference on Human-Computer Interaction (INTERACT'15)*, pp. 436-454. Springer, Cham, 2015.

Teddy Seyed, Xing-Dong Yang, and Daniel Vogel. 2016. Doppio: A Reconfigurable Dual-Face Smartwatch for Tangible Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16).* ACM, New York, NY, USA, 4675-4686.

Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, Frank Maurer (2016). User Elicitation on Single-hand Microgestures. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '16), San Jose, CA, USA.
🏆 *Best Paper Honorable Mention*

Teddy Seyed, Xing-Dong Yang, and Daniel Vogel. 2017. A Modular Smartphone for Lending. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Tech-*

nology (UIST '17). ACM, New York, NY, USA, 205-215.

🏆 *Best Talk Winner*

Jun Gong, Xin Yang, Teddy Seyed, Josh Urban Davis, Xing-Dong Yang (2018). Indutivo: Tangible Input on Smartwatches using Inductive Sensing. In Proceedings of the ACM Symposium on User Interface Software & Technology (UIST'18), Berlin, Germany.

Jun Gong, Da-Yuan Huang, Teddy Seyed, Te Lin, Tao Hou, Xin Liu, Molin Yang, Boyu Yang, Yuhan Zhang, Xing-Dong Yang (2018). Jetto: Using Lateral Force Feedback for Smartwatch Interactions. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '18), Montreal, Canada.

Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang 'Anthony' Chen, Xiaojun Bi, Xing-Dong Yang (2018). WrisText: One-handed Text Entry on Smartwatch using Wrist Gesture. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '18), Montreal, Canada.

🏆 *Best Paper Honorable Mention*

Shan-Yuan Teng, Da-Yuan Huang, Chi Wang, Jun Gong, Teddy Seyed, Xing-Dong Yang, Bing-Yu Chen (2019). Aarnio: Passive Kinesthetic Force Output for Foreground Interactions on an Interactive Chair. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems(CHI '19), Glasgow, UK.

🏆 *Best Paper Honorable Mention*

Prototyping Tools for Novices

Teddy Seyed, Peli de Halleux, Michal Moskal, James Devine, Joe Finney, Steve Hodges, and Thomas Ball. 2019. MakerArcade: Using Gaming and Physical Computing for Playful Making, Learning, and Creativity. In *Extended Abstracts of the 2019 CHI Conference on Human*

Factors in Computing Systems (CHI EA '19). ACM, New York, NY, USA, Paper LBW0174, 6 pages.

Jo-Yu Lo, Da-Yuan Huang, Tzu-Sheng Kuo, Chen-Kuo Sun, Jun Gong, Teddy Seyed, Xing-Dong Yang, Bing-Yu Chen (2018). AutoFritz: Autocomplete for Prototyping Virtual Breadboard Circuits. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems(CHI '19), Glasgow, UK.

Teddy Seyed (2019). Technology Meets Fashion: Exploring Wearables, Fashion Tech and Haute Tech Couture. In CHI '19 Extended Abstracts on Human Factors in Computing Systems (CHI EA '19), (In Press), Glasgow, UK.

Teddy Seyed, Anthony Tang (2019). Mannequette: Understanding and Enabling Collaboration and Creativity on Avant-garde Fashion-Tech Runways. In *Proceedings of the 2019 Designing Interactive Systems Conference (DIS '19)*, ACM, New York, NY, USA.

Best Paper Honorable Mention

ENTREPRENEURIAL PREFACE

For a large majority of the chapters in this dissertation, a traditional academic writing style is used. However, for the entrepreneurial chapters, I deviate entirely away from this and write my entrepreneurial lessons, as reflections from the perspective of a PhD candidate. These sections have been written after I've become "wiser" so to speak. Pragmatically speaking, this means I am writing about successes and failures after the fact, in the context of the entirety of my journey through entrepreneurship thus far.

As an early preface to these sections, there are two specific aspects that need to be made clear in terms of how I present my entrepreneurial lessons and reflections:

1. Methodologically, I followed the "Reflective Rational Enquiry" process by Lawrence-Wilkes and Ashmore (2014). Generally, it forces one to self reflect in *multiple frames of reference* (in this case, I reflect as both a student as an entrepreneur), reflect on *actions taken* (e.g. why a certain decision was made and its consequences), as well as *socio-political contexts* (e.g. how do these actions affect society, or those around me, or the broader ecosystem in Calgary, Alberta and Canada).

2. A close friend of mine recently lent me the book "The Hard Thing About Hard Things: Building a Business When There Are No Easy Answers" by Ben Horowitz. The book encapsulated many of the things I didn't quite know or realize I needed to convey in my own reflections. The book is also extremely direct, and brutally honest. I've been inspired to write my reflections in a similar manner.

Ultimately, the journey, lessons and reflections that are conveyed throughout the entrepreneurial sections in this dissertation are my own, and based on my personal experience. I do not attempt to present a scientific formula or method, or necessarily contribute to the entrepreneurial literature. Instead, I describe my story that was an experiential learning process of over 6 years. They are meant to be as direct and to the point as possible. Throughout my academic journey as a doctoral student and candidate, the best feedback I received was what I *needed* to hear, not what I *wanted* to hear. Thus, these lessons and reflections follow a similar approach for those who wish to take a similar journey, student or otherwise.

ENTREPRENEURIAL VENTURES

On the entrepreneurial side of this dissertation, I participated in and co-founded several startups. In the subsequent pages, I provide a complete list of ventures I participated in during my PhD. A large portion of the materials in the entrepreneurial sections have appeared previously in these ventures. However, not all of these ventures and their materials have been used for this dissertation, but I have highlighted those which are. I then follow this up with a visual summary of where each of the highlighted ventures (and their associated materials) fit into the dissertation. Additionally, I have anonymized the local startup and its product, and as a result, it is not listed in the ventures below, but is instead mentioned in the visual summary.

Venture Name: Slate Scale Inc.

Role: Co-Founder

Product: Slate Scale - http://slatescale.strikingly.com

Venture Name: Transit Krowd Inc.

Role: Co-Founder

Product: TransitKrowd - http://municipalinnovators.ca/articles/calgary-hackathon

Venture Name: Marathoner Labs

Role: Co-Founder

Product: Emotica - http://www.youtube.com/watch?v=gHT3Xd53sPU

Venture Name: Wear Labs Inc.

Role: Co-Founder

Product: StitchKit.io -

http://www.wareable.com/fashion/make-fashion-stitchkit-electronics-765

Venture Name: The SOFIE Foundation

Role: Co-Founder

Product: StitchKit.io - http://www.thesofiefoundation.org

Thesis Chapters

Human-Computer Interaction (HCI) Research

Entrepreneurial Ventures

Part I

Exploring Ubiquitous Computing Environments

2. Multi-Surface Environment Toolkit

3. Case Study I: SkyHunter

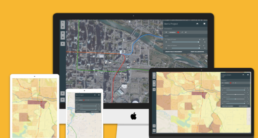
4. Case Study I: ePlan

Teddy Seyed, Alaa Azazi, Edwin Chan, Yuxi Wang, and Frank Maurer.2015. SoD-Toolkit:A Toolkit for Interactively Prototyping and Developing Multi-Sensor, Multi-Device En-vironments. In *Proceedings of the 2015 International Conference on Interactive Tabletops and Surfaces (ITS '15)*. ACM, New York, NY, USA,171-180.

Teddy Seyed, Mario Costa Sousa, Frank Maurer, and Anthony Tang.2013. SkyHunter: amulti-surface environment for supporting oil and gas exploration. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces (ITS '13)*. ACM, NewYork, NY, USA,15-22.

Apoorve Chokshi, Teddy Seyed, Francisco Marinho Rodrigues, and Frank Maurer.2014. ePlan Multi-Surface: A Multi-Surface Environment for Emergency Response PlanningExercises. *In Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, New York, NY, USA,219-228.

5. From SkyHunter to the Spatial Interactive Tabletop Software (SITS) product



Venture Name: Calgary-based Startup
Role:
Project Manager
Product Manager
Senior UI/UX Designer
Product: SITS (commercialization of work in Chapter 3)

Part II

Novel Devices and Interactions

6. Tangible + Dual-screen Smart-watch

7. Lendable smartphone

Teddy Seyed, Xing-Dong Yang, and Daniel Vogel.2016. Doppio: A Reconfigurable Dual-Face Smartwatch for Tangible Interaction. *In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4675-4686.

Teddy Seyed, Xing-Dong Yang, and Daniel Vogel.2017. A Modular Smartphone for Lending. *In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*.ACM, New York, NY, USA,205-215.

8. From Doppio to Emotica: Pivoting an idea

Venture Name: Marathoner Labs.
Role: Co-Founder
Product: Emotica

Part III

Prototyping Tools for Novices

9. Mannequette: A Fashion-Tech Tool

10. MakerArcade: A Prototyping Tool for Gaming

Teddy Seyed, Anthony Tang (2019). Mannequette: Understanding and Enabling Collab-oration and Creativity on Avant-garde Fashion-Tech Runways. In *Proceedings of the 2019 Designing Interactive Systems Conference (DIS19).* ACM, New York, NY, USA.

Teddy Seyed, Peli de Halleux, Michal Moskal, James Devine, Joe Finney, Steve Hodges,and Thomas Ball.2019. MakerArcade: Using Gaming and Physical Computing for Play-ful Making, Learning, and Creativity. In *Extended Abstracts of the2 019 CHI Conference onHuman Factors in Computing Systems (CHI EA '19).*ACM, New York, NY, USA, PaperLBW0174, 6pages.

11. StitchKit: A Fashion-Tech and STEAM Education Kit

Venture Name: Wear Labs Inc.
Role: Co-Founder
Product: StitchKit.io -http://www.stitchkit.io

ACKNOWLEDGEMENTS

While this dissertation has my name at the front, it is in fact a culmination of the work others have done around me to support, inspire, motivate, and continuously pushing me well past my own limits. I'd like to take the opportunity to thank some of them here.

- To Dr. Frank Maurer: There are no words that can be written for all that you have done. From my undergrad, to PhD, you have been not just an advisor, but someone I look up to, and I am truly honored to have been your PhD student. Because of your support and faith in my abilities, I learned not to fear things as much (especially public speaking!) and embrace the unknown. As a result, I was able to explore and do things in my PhD that others had not yet done, making it a life-changing experience. Ultimately, the researcher and entrepreneur I am today would not have been possible without you, and I hope one day that I can do for others, what you have done for me.
- To Dr. Tony Tang: Thank you for being an amazing co-supervisor. Because of our constant chats early in my PhD, and several towards the end, I became a very critical thinker in both HCI and research in general. Without the first paper we worked on way back in 2012, I know that I would not be as strong a researcher as I am today. I really appreciated the motivation you gave me, and the support for my crazy research ideas that may not have always made sense. I also learned a lot from you on how to supervise students, and I hope I can be the same for future students or youth, regardless of where I end up.

- To Dr. Xing Dong Yang: Thank you for coming into my life and introducing me to the crazy side of research. Watching you grow from a Post Doc in the iLab to the amazing professor and colleague you are now, has been inspirational. I will forever be thankful that you started me down a path of research that I didn't know I was capable of, being a supportive colleague, and more importantly a friend. You've had a lasting impact on my persona and career, and I look forward to continuing to work with you in the future and exploring the unknown!

- To Dr. Chad Saunders: Thank you for letting me take your introductory MBA Entrepreneurship class as a PhD student. I initially had no idea what to expect without a business background, but the class opened my eyes and showed me it was possible to combine research and entrepreneurship. Because of your class, I eventually became fearless on the business side, and it affected every other area of my life as a result. Thanks for also being on my supervisory committee, and providing insight and discussion over the years.

- To Abhishek Sharma: Thank you for being a truly great friend. We've been through a lot on the entrepreneurial side together, but it has been a true pleasure to watch you grow so much in your career, and in life. It has always been comforting to know that no matter where you go or what you do, that you will always be successful. I was really humbled to be able to attend your wedding, and I look forward to you and Tarishi growing together. One day in the future, we'll figure this entrepreneurial thing out and the rest will be history.

- To Chris Burns: It has been awesome to see you grow from the time we both finished MSc to now. I expected a lot, and you had surpassed all my expectations, and then some. Watching you grow and succeed was very always very motivational for me throughout my PhD, and I was also very proud of you because of how closely we worked together in our MSc. Your success always felt like mine, and for this PhD, its success is yours as well. Thanks for everything my friend.

- To Abraam Mikhail: A lot has changed since our undergrad and MSc! days I've finally finished the PhD, and you have an amazing wife and family. While we didn't get to meet up as much during the PhD as I would have liked, it was always great to know that I had a good friend around when needed. Thanks for always being a great friend through my time at the University of Calgary, and definitely beyond!

- To Theodore Hellmann: I've said it before, and I'll say it again. If you weren't my TA way back in my 2nd year of Computer Science undergrad, I would not be where I am today. Thanks for introducing me to the lab, guiding me in research, giving me feedback, getting me to think about grad school and everything else in between. I will always be glad to call you friend.

- To Tulio de Souza Alcantara and Larissa Ozeki: To the happiest couple I know, thank you for adding so much positivity and happiness during the PhD. It wasn't the easiest of journeys for me, but it was always wonderful having you around for crazy adventures and lots of laughter. I look forward to hanging out more, now that I am free!

- To my many teammates of hockey – Jamie, Jeremy, Chad, Gary and everyone else I played with in the leagues in Calgary – thank you for letting a random guy play hockey with you all who are so skilled in the game. I learned how to play on a team, the value of teamwork, and how to accept a role and become good at it. In the grand scheme of things, playing with you all really normalized things and allowed me to have a life outside of school, and more importantly, balance. I will always cherish the awesome games we played, the many blocked shots for my teammates, and hopefully I can continue my hockey career in the future...with a little less pain.

- To past ASE Lab members: Thank you for creating a wonderful environment that allowed me to grow as a person, learn and explore much on the technical side of research, and for having the trust in me to help you with your theses, and research projects. It has been amazing to watch you grow so much since your time in the lab, and I am proud to call many of you friends now.

- To past iLab members: Although I was never able to spend much time physically in the iLab due to all the work I was doing and constant travel, in the time I was able to be there, it was always enjoyable to have great research conversations and bounce ideas off each other. I hope to work with some of you in the future.

- To my entrepreneurial mentors – Evan Hu, Laurie Wallace, James Lochrie, Ray Muzyka: Thank you for being a critical and realistic group of mentors who played a pivotal role in my entrepreneurial journey thus far. Perhaps one of the best moments in my PhD was standing at the CDL stage and having you all say that my co-founder and I were more than capable of being successful, both technically and on the business side. It meant a lot for us because we didn't have traditional business backgrounds. Because of your feedback, I learned to look at things very critically, be confident in my own business skills, and it has ultimately had a transformative effect in my career. I am looking forward to meeting you all again one day in the future, but as a much more seasoned entrepreneur.

- To my Microsoft Research colleagues: Without a doubt, the projects that I worked on as part of my internships were the craziest part of my PhD. First off, I'd like to thank Tom Ball, Peli de Halleux and Michal Moskal for taking me in fresh and teaching

me so many things that I can't properly describe, and providing encouragement and feedback that I didn't know I needed. You are all amazing mentors, colleagues, and most importantly, great friends. I look forward to a lot of the fun times ahead working with you all. To Tom, thanks for answering my random email, and having the faith in me to do something that was crazy, and it was an absolute pleasure to pull it off successfully with you and the team. To Asta Roseway, thank you for showing me how to meaningfully combine art and technology, being so passionate about it, and also being an amazing mentor. The Game of Thrones chats were also pretty awesome! I will forever be indebted to you all for allowing me the opportunity to take an idea and move it forward in a way that I didn't know was possible. The faith and trust you have placed in me for the future won't be forgotten, and I intend on making you all proud. The best is yet to come!

I'd also like to thank Ani Babaian, Paul Johns and Gavin Jancke for their help and support in guiding an intern through unknown areas of research, and always being encouraging and so positive. It really meant a lot coming from you all, and helped motivate me during so many late nights in the office. I also can't forget a shout out to my colleagues and friends in the Hardware Lab (Sokunthea Neang, Lex Story, Todd Jurgensen, Teresa LaScala and Patrick Therien). Thanks for all the help and support with my projects, and encouraging me to explore even crazier ideas. You guys are all awesome, and just wait...there's more crazy projects coming in the future!

Last but not least, special thanks to the Microsoft Research Outreach Team (Tara Grumm, Gretchen Huizinga and others) as well as the Microsoft Research Video Crew (Wil Morrill, Ben Ericson, Jake Knapp) for all their help and support. Your support and belief in the ideas of everyone, and your willingness to share them with the world was very humbling to witness. I learned much from you all, and I can't wait to work

on what's next with you! It is my goal as a team, that we can win an Emmy or Oscar one day with the amazing storytelling that builds on the work we've all done so far!

To everyone else that I worked with, talked to in the hallways, or that came said hi to me at Microsoft Research: Thank you for creating an environment that let me prosper and grow in ways I didn't know I could. I came as a PhD intern, and left as something more than a PhD student. I look forward to joining you all in creating new and amazing things that are impactful and transcend research. I hope the world is ready!

- To James Devine: Thanks for being an awesome colleague and friend during our time together as interns. It was great bouncing so many ideas off you, grabbing fruit, and everything else. The late nights were also pretty fun, especially in that basement, haha! You are a brilliant researcher, and I can't wait to work with and learn from you in the future. Let's get this team back together and on the road again!

- To Rojin Vishkaie: Thanks for all your support over the years, and your skill and expertise in research, as well as your friendship have been truly cherished. It has been wonderful to see you grow into an amazing researcher over the past several years. I know the best is yet to come!

- To the University of Calgary, the Faculty of Science, and the Department of Computer Science: Thank you for allowing me the opportunity to explore the best version of myself. This PhD is a culmination of the years of work you have put into me, through scholarships, funding, entrepreneurial initiatives – such as the Hunter Hub, which gave me my entrepreneurial start – and so much internal support for the things I dreamed of or wanted to accomplish. I can honestly say, I don't think the type of work I did for the PhD could ever be done or attempted at any other institution in the world. Although I have completed my PhD, I will always be around to support the University of Calgary, no matter where I am in the world, or what position I hold. You have invested in me, and in return, I will succeed in my future endeavours.

Family is not an important thing.

It's *everything.*

— Michael J. Fox

To my parents Hussien and Mestawat,

Thank you for being amazing parents. I could not have done this without you.

To my brother, Ismael,

Thanks for being a great brother and an inspiration.

CONTENTS

LIST OF FIGURES

1

INTRODUCTION

"*Creativity is contagious. Pass it on.*"

Albert Einstein

In 1991, Weiser (1999) envisioned *Ubiquitous Computing*, which described computers appearing anywhere, at any time, and occurring in different formats beyond traditional desktops. Several decades later, with advancements in fields such as engineering and computer science, computing is now ubiquitous. Whether it's a meeting room in an office embedded with smart displays, a wearable device that tracks its wearers heartbeat, or a young child playing and learning from a smart toy, ubiquitous computing is now profoundly woven "*into the fabric of everyday life*".

Despite the large number of smart devices that are released each year, there are still a number of challenges that remain for ubiquitous computing. In Weiser's vision, there was a notion of devices and technology being "*indistinguishable*" in the environment, that is, to have the technology blend into the background. To accomplish this, the "*indistinguishable*" environment must be capable of rich interactions, be expressive, and contextually capable of interacting with users, their content and whatever form of devices they carry. Thus, the goal is to enable computing environments and devices that seamlessly interact with people and their environment, such that everyday tasks are augmented.

Adding a wireless component or a touch-screen to every device in an environment are two common approaches to achieving the goals of ubiquitous computing. However, as the Internet of Things (IoT) space has shown, not everything should be connected to the internet, aside from the many ethical, security and monetary implications. But doing this also excludes other types of devices which don't utilize screens (or need to), such as smart clothing with low-powered sensors integrated into them.

In this dissertation, I describe my research and entrepreneurial efforts to make ubiquitous computing more "*indistinguishable*". I do this by focusing on three areas: 1) enabling people to design, create and deploy interactive environments that use their everyday devices, 2) rethinking and redesigning these everyday devices themselves — and their associated tasks — to become "*indistinguishable*" and 3) providing tools for designers, novices and other non-technical users to create their own "*indistinguishable*" devices or environments.

1.1 DISSERTATION SCOPE

Overall, the research throughout this dissertation lies within the broad area of Human Computer Interaction (HCI). Within this context, I looked at ubiquitous computing environments and the everyday devices that operate within them (Figure 1).

Within the research space of ubiquitous computing environments, I restrict my explorations to real-world and domain-focused examinations, as the opportunities and lessons here have generally been under-explored.

Secondly, the space of devices that exist within ubiquitous computing environments is large and still growing, and coupled with my aim to tackle specific interaction challenges —

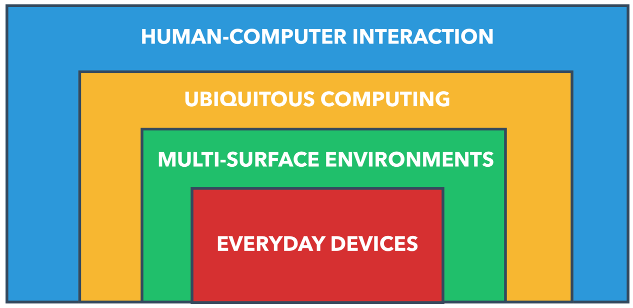


Figure 1: Research context.

such as limited physical constraints on devices — I restricted the research into rethinking and redesigning devices to focus on smartwatches and smartphones. I did not aim to study smartwatch or smartphone designs that I created in the wild, but instead present novel design concepts and interactions techniques that can inform the design of future devices, as well as their interactions with other devices and users in ubiquitous computing environments. Lastly, I restricted my research in tools for novices and non-technical users primarily to creative domains (e.g. fashion, gaming), as they could provide pragmatic lessons and contributions to ubiquitous computing as a whole.

For the entrepreneurial scope in this dissertation, I mirrored each research section with an entrepreneurial reflection and lessons learned. Each reflection focuses on commercialization activities, where I provide lessons learned strictly from my perspective as a student, using the "Reflective Rational Enquiry" process by Lawrence-Wilkes and Ashmore (2014).

1.2 DISSERTATION OVERVIEW

As this thesis blends both research and entrepreneurial explorations, it's structure is non-traditional and is organized based on both my research areas and parallel entrepreneurial journey within ubiquitous computing. This structure appears as a linear and sequential journey with distinct phases, however this is for the sake of literary convenience. Pragmatically, both my research and entrepreneurial explorations overlapped, sometimes occurred concurrently and ultimately the results and processes of one, affected the other. Figure 2 provides a visual overview of the interrelation between the sections and chapters of this dissertation its non-traditional blend.

In Section i, I define and describe research into a specific type of ubiquitous computing environment (Chapters 2, 3 and 4) — multi-surface environments — and I also reflect on early commercialization work in Chapter 5. Next, in Section ii, I present explorations into designing novel devices (Chapters 6 and 7) and then reflect on commercialization activities for one of these devices in Chapter 8. Section iii describes explorations into providing tools for non-technical users in creative domains such as fashion and gaming (Chapters 9 and 10), followed by a concurrent entrepreneurial reflection in Chapter 11. Finally, in Section iv, I provide a final summary of the entrepreneurial journey described in this dissertation (Chapter 12), followed by a summary of my key contributions and a description of future research and entrepreneurial opportunities in Chapter 13.

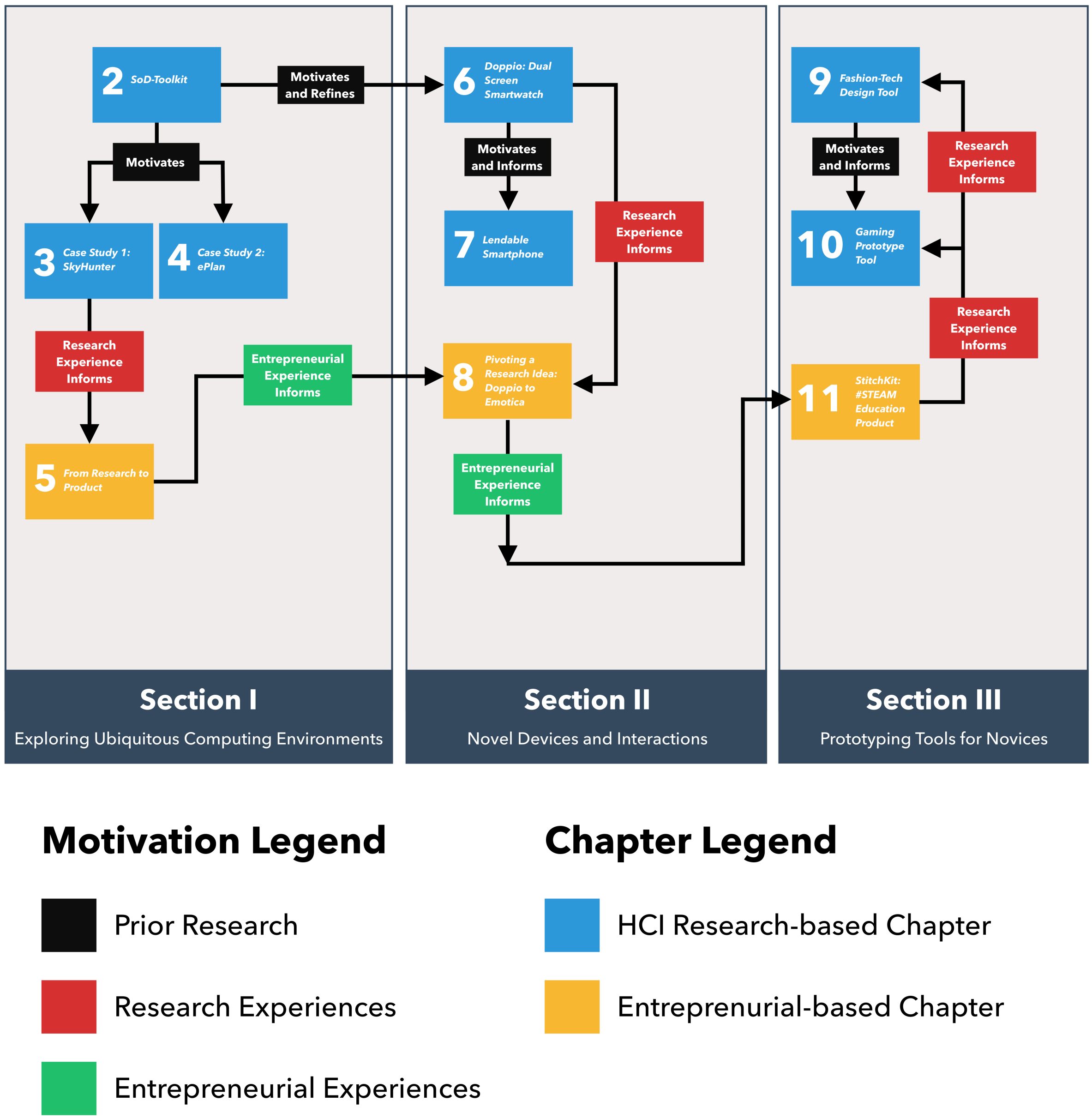


Figure 2: Overview of the research and entrepreneurial chapters, sections and their relation to each other.

Part I

EXPLORING UBIQUITOUS COMPUTING ENVIRONMENTS

The first part of this dissertation is comprised of research work related to a specific type of ubiquitous computing environment — Multi-Surface Environments (MSEs). I also provide a critical self-reflection of the beginnings of my entrepreneurial journey as part of the commercialization process for some of the aforementioned work. First, in Chapter 2, I describe a development tool that allows for novel explorations, prototyping and real-world deployments of MSEs. Next, in Chapter 3 and Chapter 4, I use two real-world case studies to explore several of the important interaction techniques and technologies in MSEs. Finally, in Chapter 5, I reflect upon how the work in Chapter 3 was commercialized and the key entrepreneurial lessons I learned.

2

SOCIETY OF DEVICES (SOD) TOOLKIT

As ubiquitous environments become increasingly commonplace with newer sensors and forms of computing devices (e.g. wearables, digital tabletops), researchers have continued to design and implement novel interaction possibilities. However, as the number of sensors and devices continues to rise, researchers still face numerous instrumentation, implementation and cost barriers before being able to take advantage of the additional capabilities. In this paper, we present the *SoD-Toolkit* —a toolkit that facilitates the exploration and development of multi-device interactions, applications and ubiquitous environments by using combinations of low-cost sensors to provide spatial awareness. The toolkit offers three main features. (1) A "plug and play" architecture for seamless multi-sensor integration, allowing for novel explorations and ad-hoc setups of ubiquitous environments. (2) Client libraries that integrate natively with several major device and UI platforms. (3) Unique tools that allow designers to prototype interactions and ubiquitous environments without a need for people, sensors, rooms or devices. We demonstrate and reflect on real-world case-studies from industry-based collaborations that influenced the design of our toolkit, as well as discuss advantages and limitations of our toolkit.

2.1 INTRODUCTION

The consumer space of computing technologies is experiencing a dramatic explosion of different size and form factors for devices (e.g. wearables, multi-touch wall displays). The capabilities of these devices can be expanded significantly when used collectively with other devices and sensors, effectively creating multi-device ubiquitous environments. Ubiquitous environments provide people with access to their information across many of their devices, with some devices being spatially aware of other devices in the environment. Spatial awareness in ubiquitous environments is directly linked to the sensors (either dedicated or device based) and further magnifies existing challenges as to how information and tasks can be performed effectively across different types of devices, within varying degrees of spatial awareness (Seyed et al. (2012)).

Research in ubiquitous environments primarily focuses on novel interaction forms between people and a set of devices (Weiser (1999)) (e.g. digital tabletops, mobile devices, smart watches), with proxemics being a common method of conceptualizing the interaction space (Ballendat et al. (2010); Greenberg et al. (2011)). As different form factors for devices becomes commonplace and the capabilities of sensors increase, novel forms of explorations between different devices (e.g. a Google Glass and a Smart Watch) is still fairly limited in the context of spatially-aware ubiquitous environments. Furthermore, exploration into real-world scenarios also faces limitations, much of which arises from the difficulty in building multi-device, spatially-aware environments, as many existing development kits are limited in sup-

***Note: The text in this chapter appears in the following publication:**
Teddy Seyed, Alaa Azazi, Edwin Chan, Yuxi Wang, and Frank Maurer. 2015. SoD-Toolkit: A Toolkit for Interactively Prototyping and Developing Multi-Sensor, Multi-Device Environments. In *Proceedings of the 2015 International Conference on Interactive Tabletops and Surfaces (ITS '15).* ACM, New York, NY, USA, 171-180.

port for different multi-sensor configurations, multi-device platforms and cross-connectivity and typically require complex software and hardware setups (Houben and Marquardt (2015)).

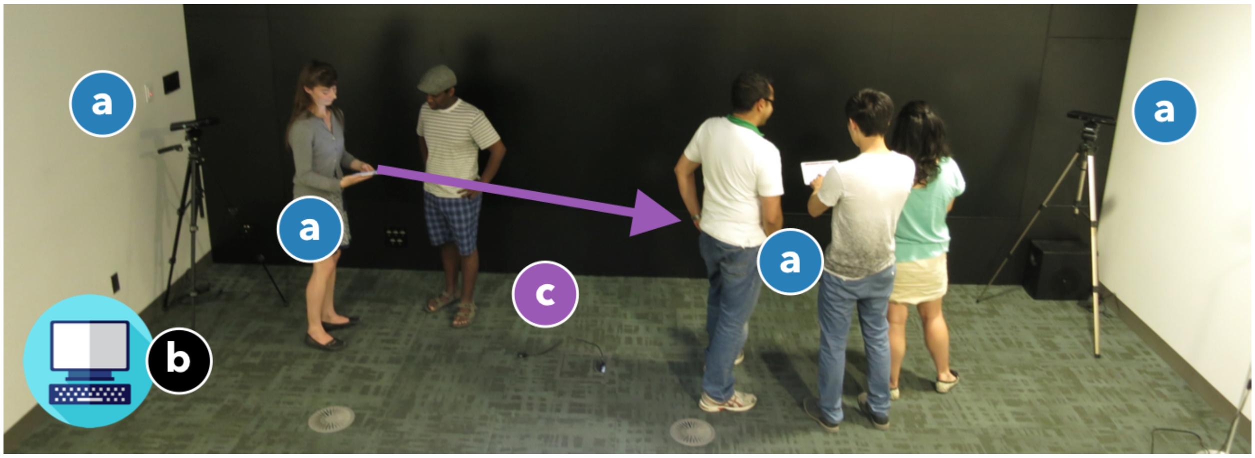


Figure 3: The *SoD-Toolkit* consists of (a) numerous devices and sensors that are supported by (b) software tools and components providing information that facilitates (c) spatial interactions between devices and the environment.

To bridge the gap and allow for richer explorations into spatially-aware ubiquitous environments, we introduce the Society of Devices (SoD) Toolkit (or *SoD-Toolkit*), a toolkit that facilitates exploring and developing multi-device applications and interactions in spatially-aware ubiquitous environments (Figure 3).

Overall, our primary research goal was *to allow for novel explorations of different types of multi-device, spatially-aware (through multi-sensor fusion) ubiquitous environments that can be augmented with a multitude of newer sensors and device platforms*. To address this goal, our toolkit abstracts sensor information from a multitude of sensors into a "plug and play" architecture, allowing researchers and developers to seamlessly fuse sensor information and utilize commercially available off the-shelf tracking technologies to provide spatial awareness in ubiquitous environments. Researchers and developers can also create additional modules for future sensors through the modular architecture of the toolkit. The toolkit also provides a number of client libraries that are built upon existing operating systems and platforms which include iOS, Android, Windows, as well as web-based technologies such as HTML5, Node.js and Javascript. This allows for a wide range of skill sets and experience, letting researchers and developers implement and design interactions and ubiquitous environments in languages they are comfortable with, reducing many common platform, language, technology and device barriers (Nebeling et al. (2014b)). In addition, the toolkit offers tools for researchers and developers to visualize and prototype interactions within varying levels of spatially-aware ubiquitous environments (due to limited hardware availability) or without the need of specialized hardware entirely. This reduces a significant hardware and cost barrier for researchers and developers, and can assist in more widespread research and application development for ubiquitous environments in both the research and consumer space.

The remainder of this paper, is organized as follows. We first review related work and then introduce the design of the *SoD-Toolkit*, including its key features, architecture and components.

Next, we describe two real-world domain-specific case studies that demonstrate the flexibility of the toolkit, how they influenced the design rationale of the toolkit and lessons we learned. This is followed by a discussion and reflection on the design and features of the toolkit compared to other approaches. Finally, we conclude this paper describing limitations and future work.

2.2 RELATED WORK

SoD-Toolkit is inspired by Weiser's vision of ubiquitous computing (Weiser (1999)) and built upon prior work in three areas of research: (1) Proxemics and Ambient Interactions, (2) Multi-Surface Interactions and (3) Application Programming Interfaces (APIs) and Toolkit designs.

2.2.1 *Proxemics and Ambient Interactions*

The research space of proxemic interactions is extremely rich and well explored. Greenberg et al. (2011) conceptualized proxemics in the context of ubiquitous environments with a number of investigations that focused on spatial relationships between users and objects, specifically applying five proxemic dimensions: orientation, distance, motion, identity and location. The intention of much of the research in this space is to "leverage people's natural understanding of their proxemic relationships to manage the entities that surround them" (Marquardt et al. (2011)). Applying these proxemic theories to sensors in ubiquitous environments was further explored by Ballendat et al. (2010), who used sensors to detect people and their devices, and better understand different types of interactions (i.e. implicit and explicit). Follow-up work by Marquardt and Greenberg (2012) examined the relationships between people in closer spaces through F-formations and micro-mobility and sociological constructs.

Marquardt et al. (2012a) also identified major challenges for proxemics, which included providing meaningful feedback, managing privacy and security and establishing connections between different types of devices. Establishing connections with different types of devices was the focus of early work in proxemics by Vogel and Balakrishnan (2004), who explored proxemics in relation to public ambient displays . They defined four discrete areas in front of devices, similar to Hall (1990) proxemic zones surrounding a person. In more recent years, research focused on devices such as whiteboards (Jakobsen et al. (2013); Schmidt et al. (2012)) and other forms of displays (Snibbe and Raffle (2009)).

While a majority of proxemics research has been heavily device-centric in relatively small or enclosed spaces, proxemics in larger more ambient spaces (and not around smaller displays) has been less researched. Active Badge by Want et al. (1992) is an early example of exploring larger scale proxemics . Their system uses a beacon sensor to track position of users in a work environment. UbER Badge by Laibowitz and Paradiso (2004) is another example, whose system uses proxemic badges. The badges are used to facilitate social interaction in various large meetings. In the context of a ubiquitous home, the EasyLiving project explored multiroom proxemics by combining a number of technologies to track people and their devices on a larger scale (Brumitt et al. (2000)). Research into proxemic interactions in larger and more ubiquitous spaces is directly tied to different types of tracking technologies, a large research area within ubiquitous environments itself. *SoD-Toolkit* builds upon many of the concepts in proxemics and ambient spaces with a goal of providing lightweight tools and libraries for developing and exploring interactions and applications in larger scale ubiquitous environments and domains, without researchers and developers being restricted in room size or choice of sensors.

2.2.2 *Multi-Device Interactions*

Multi-device interactions with a number of different device configurations is a rapidly growing research area, particularly with newer form factors of devices increasing in the consumer market (e.g. wearables). A major task in multidevice interactions involves the movement of information or content from one device to another (Seyed et al. (2012)). Much of the early work in multi-device interactions is based upon Rekimoto (1997)'s Pick and Drop, where pen input is synchronized across multiple computers, allowing a user direction manipulation of content between screens. Hinckley (2003) explored the notion of bumping tablets and stitching tablets together (via pen stroke across displays) as multi-device interactions for transferring content. Lucero et al. (2010) followed a similar approach, but instead used a pinching gesture across multiple mobile devices. In these types of multi-device interactions, input is typically synchronized to facilitate smooth interaction and information transfer (Chen et al. (2014b)).

Multi-device interactions can also be impacted by spatial awareness and proxemic relationships (Chen et al. (2014b)). Numerous sensors are employed to provide positioning and tracking of devices (and users) to take advantage of spatial relationships. Kortuem et al. (2005) used this approach when creating novel spatial widgets for user interfaces. Kray et al. (2008) used a digital tabletop as a center of mediation for proxemic relationships between mobile devices. More general work in these types of interactions were explored by Marquardt et al. (2012a) with the gradual engagement pattern, that maps device-to-device proximity as a function of different levels of information exchange. LightSpace by Wilson and Benko (2010), uses spatial awareness for interactions between and on physical surfaces.

Dividing information and interfaces across multiple devices results in interactions that are distributed (Chen et al. (2014b)). The iLand system by Streitz et al. (1999) is an early example of interactions distributed in a ubiquitous environment. Roomware is another example that inter-connects smart artifacts in a room, to augment both individual and collaborative tasks (Streitz et al. (2002)). More recently, interactions between newer forms of devices such as wearables has begun to appear in the research literature. Chen et al. (2014b) explored interaction techniques and gestures for distributed interactions between a watch and smartphone. In a similar fashion, Mayer and Sörös (2014) explored head-mounted display to interact with objects within view of a user. Overall however, distributed interaction techniques for wearables and newer forms of devices is still extremely under-explored (Chen et al. (2014b); Houben and Marquardt (2015); Wagner et al. (2013); von Zadow et al. (2014)).

Much of this work indicates the potential interaction techniques for ubiquitous environments, however a significant amount of implementation work is repeated for interactions that are synchronized, spatially-aware or distributed. *SoD-Toolkit* easily allows for the exploration of these types of interactions (or combinations thereof) and can facilitate researchers in exploring unconventional and yet-tobe explored multi-device spatially-aware interactions (e.g. between a head-mounted display and a smart watch).

2.2.3 *Application Programming Interfaces and Toolkits*

A significant amount of recent research has explored creating multi-device toolkits, primarily designed to overcome different aspects of the engineering challenges that come with building ubiquitous environments. Toolkits, such as Conductor (Hamilton and Wigdor (2014)) and Panelrama (Yang and Wigdor (2014)) and others (Chi and Li (2015); König et al. (2009); Schreiner et al. (2015)), focus specifically on multi-device application development through web-based interfaces. XDStudio also supports cross device application development, through the use of a GUI builder (Nebeling et al. (2014a)).

The Proximity Toolkit is the canonical example of a toolkit focused on larger ubiquitous spaces Marquardt et al. (2011). The toolkit gathers data from various tracking sensors to allow for sensor fusion, and provides an easily accessible API. It also allows for interactions to be observed via a visual tool. A major challenge with the toolkit however, is its reliance on high-end tracking systems (Nebeling et al. (2014a)) that are ideal for prototyping but not feasible in real-world deployments and its limited support for additional sensors. This inspired further work, such as XDKinect (Nebeling et al. (2014b)) which uses a single Kinect sensor to mediate interaction between different devices and also allows proxemic interaction and multi-modal input. In toolkits such as these, enabling multi-device interactions in ubiquitous environment requires knowledge about presence and position of devices, typically provided by a variety of sensors. Numerous approaches have been taken in the research literature, such as magnet based tracking (Huang et al. (2012)), radio tracking (Lucero et al. (2010))

and sensor fusion approaches (Greenberg et al. (2011)). Much of the work in prototyping ubiquitous environments also faces issues of cost (for sensors and devices) and room size, making it difficult for researchers to properly explore novel types of ubiquitous environments and multi-device interactions.

Implementing ubiquitous environments that are deployable in real-world environments still requires substantial engineering efforts for researchers and developers. It is still possible to build applications for different devices, with different sensors in ubiquitous environments, but the effort required prevents richer explorations Rädle et al. (2014). In contrast, SoD-Toolkit provides support for exploring multi-sensor, multidevice ubiquitous environments by providing (1) a "plug and play" architecture multi-sensor fusion, (2) native client libraries for major device, sensor and UI platforms and (3) tools that allow rapid prototyping without the need for people, sensors, rooms or devices.

2.3 SOD-TOOLKIT

In this section, we provide an overview of the *SoD-Toolkit* architectural components, its visualization and prototyping tool, as well as its multi-sensor fusion approach to create larger ubiquitous environments. To facilitate real-world deployments and novel explorations of ubiquitous environments, we primarily focused on off-the-shelf sensor hardware and common devices, as well as free and open source software. Our source code is also freely available to download as open source[1].

1 SoD-Toolkit-http://sodtoolkit.com

2.3.1 *Architecture*

The software architecture of *SoD-Toolkit* is composed of several components (Figure 4) but primarily, we discuss: (1) client libraries, (2) the locator service, and (3) the central server and communications module.

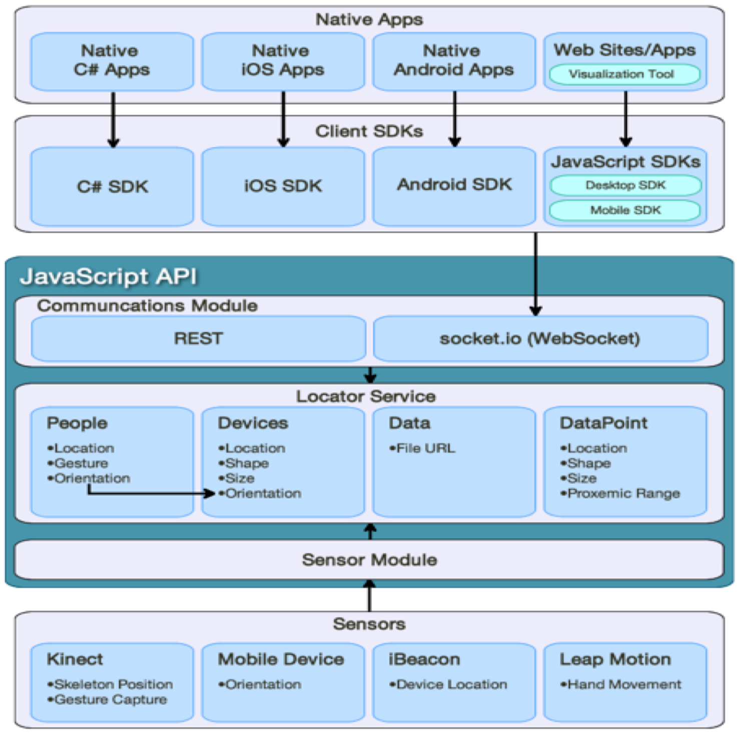


Figure 4: An overview of the architecture of the SoD-Toolkit.

2.3.1.1 *Client SDK Libraries*

The client SDK libraries have two primary functions: (1) they provide necessary information for spatial awareness in the environment, depending on the sensor or device and (2) provide a platform to be built upon for native application development. Within each client library, a sensor module captures and sends data to the locator service frequently (discussed in the next section) and the data captured varies, depending on device or sensor type. The libraries support:

1. **Various form factors of devices** such as digital tabletops, wall displays, smart watches, head mounted displays and mobile devices (regardless of implementation platform). If a device has built in sensors (e.g. accelerometers, gyroscopes), this information is sent over the network to the locator service, which is discussed in the next section. Due to the modular nature of the architecture, spatial information from a device is easily fused with other supported sensors in the environment. For example, we allow for the position and orientation of a device to be determined by fusing skeletal information of a user (from a Microsoft Kinect) with orientation of a device (from gyroscopes and accelerometers). For devices that are stationary (e.g. a wall display), we allow researchers and developers to set the physical location in space, through the visualization tools we provide, as we discuss later.

2. **JS web client** that is both platform and browser agnostic, deriving information from a device if it has sensors available. This client is similar to the platform specific device clients, however, this allows for support of webbased multi-sensor, multi-device ubiquitous environments, which is not common in many toolkits.

3. **The Microsoft Kinect** provides skeletal, position, identity and gestural information through its available skeletal stream. The Kinect client library supports a single Microsoft Kinect (version 1 or 2) and sends information over the network to the locator service at a rate of 30 skeleton frames per second. A single Kinect sensor has a tracking range from 1.2 to 4.5 meters, which creates difficulties when trying to create larger ubiquitous environments. We address this by allowing multiple Kinects to track users and use sensor fusion to expand the tracking area, create novel tracking areas, as

well as improve tracking accuracy. We discuss this sensor fusion in the Sensor Fusion approach section. Additionally, our Kinect client currently supports the “grab” and “release” gestures.

4. **The Leap Motion** provides detailed finger tracking of a user. Overall the sensor provides a more fine grained level of tracking that typically needs to be paired with a sensor that covers a larger area (e.g. Microsoft Kinect) to provide meaningful interactions. For example, if a user is further away from a Kinect and close to a wall display that isn’t touch enabled, but is connected to a Leap motion, the locator service interprets the position of the user and prioritizes sensor information from the Leap motion over potentially inconsistent Kinect skeletal data, particularly for skeleton and joint data.

5. **Apple’s iBeacon** which provide coarse grained positional information. The sensor provides position information of a device (either Android or iOS) or person (a person must have an iBeacon tag) in the form of close, near or far. While the sensor doesn’t provide as accurate tracking as a Kinect sensor, it can facilitate a larger ubiquitous space and server as a mediator, transitioning between high and low spatially-aware regions.

All clients for sensors and devices support native development platforms which include Windows (C#), iOS (Objective-C and Swift), Android (Java) and HTML5/JS, as well as support for popular IDEs (Microsoft’s Visual Studio, Android Studio and Apple’s Xcode), significantly reducing start-up effort for researchers and developers. For other programming languages, environments, developers are able to write wrappers over the libraries and APIs we provide. We also provide a set of example applications using different sensors and devices to illustrate code required to implement novel spatially-aware multi-device interactions (e.g. "flick" to device) or multi-sensor ubiquitous environment.

2.3.1.2 *Locator Service*

The Locator service is the hub that amalgamates spatial information from different sensors and devices, allowing the *SoD-Toolkit* to build multi-sensor spatially-aware ubiquitous environments. Information that is tracked includes different types of entities, such as sensors, devices (as well as their orientation) as well as users in the room. The locator service processes raw positional data from device and sensor clients that are distributed over the network, and transforms the data from a device-specific coordinate space into a locator service coordinate space, while also building higher level information about the state of the environment and the relationships between entities. Additionally, all entities that are tracked and processed by the locator service are in 3D.

As the locator service maintains position and distance between all entities, proxemic functions are readily available for researchers and developers to build upon. These functions include querying and filtering entities based on distance or within a certain distance range, and whether an entity is in the field of view of another entity, similar to Marquardt et al. (2011). We also allow researchers and developers to dynamically change proxemic properties of entities (e.g. location and orientation) through our visualization tool that we discuss in the next section.

The locator service uses an event-driven design, where clients subscribe to events that occur in the locator service, such as the proxemics previously discussed. For example, a wall display client can subscribe to an event that allows it to be aware of when a user approaches a certain range, similar to Marquardt et al. (2011) or when other entities are pointing in its direction. Lastly, the locator service also maintains information about data points, an entity unique to the *SoD-Toolkit*. The data point allows data to be assigned to specific physical locations in the room (through code or the visualization tool) and also supports proxemic properties already available to other entities. For example, a user can create virtual information spaces with different types of data in regions of a room or assign data to physical objects (e.g. a table) that can be interacted with via gestures or a device-based interactions.

2.3.1.3 *Central Server and Networking*

The central server is built into the locator service, and follows a client-server architecture. The server is implemented using the highly scalable and efficient Node.js platform and the underlying communication between the server and client libraries (on all platforms) use the Socket.IO networking module. These components are based upon the WebSocket protocol, which uses duplex bidirectional communication with significantly less overhead than other traditional methods based on HTTP. All information and data exchanged between the server and clients uses the standard JSON format.

Through the modular design of the networking and client library components, the *SoD-Toolkit* allows researchers and developers to easily extend support for future devices and sensors with minimal development effort. As we discuss in the upcoming case studies section, this was a direct result of collaboration with industry partners building real-world deployable ubiquitous environments.

2.3.2 *Visualization Tool*

The visualization tool for the SoD-Toolkit (Figure 5) has three primary functions, (1) to allow researchers and developers to monitor and understand entities – sensors, users, data points and devices – that are being tracked in the environment, as well as provide the state of fused sensors, (2) to allow for quickly prototyping ubiquitous environments without the need for hardware, people or room space and (3) maintain calibration data for multiple sensors (discussed in the next section).

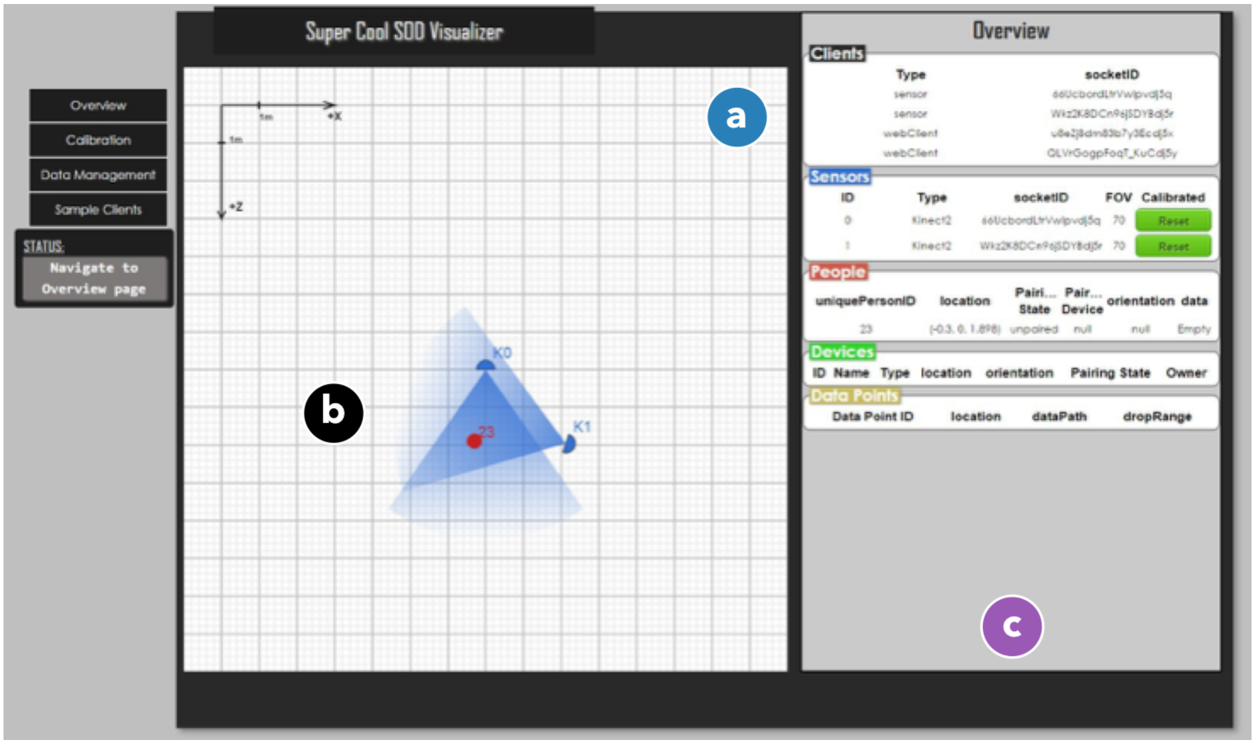


Figure 5: The visualization and prototyping tool for the *SoD-Toolkit*. (a) The tracked environment, (b) Entities tracked in the environment, (c) List of available entities and their current state, as well as clients currently connected in the environment.

In the visualizer tool, an overview list is provided, detailing the connected clients in the system, a list of available sensors in the environment, people and details on their location information, pairing state and if they currently hold any data, a list of devices and according details, and finally the location of data points and what data is contained in the data point. All entities that are visible in the tool, are user moveable components and easily configurable, allowing researchers and developers to physically remap sensors in the environment and dynamically change the layout of the environment with no additional code-required.

The tool also allows for quickly prototyping environments by allowing sample sensor and device clients to be created and connected into the system. This allows researchers and developers to conceptually build applications, environments and multi-device interactions independent of hardware and sensors and physical space. However, should they later decide to add in real-world sensors and hardware into their virtual environment, their built application or environment will already support the hardware. Furthermore, researchers and developers can also prototype mixed fidelity ubiquitous environments, allowing for a mixture of real and virtual sensors and devices (e.g. a virtual leap motion and a real-world Kinect sensor, with a fixed virtual wall display). Overall, the visualization tool is meant to be used throughout the development process – from development to deployment.

2.3.3 *Sensor-Fusion Approach*

As discussed earlier, one of our research goals was the novel exploration of ubiquitous environments with an expanded tracking space and multiple newer sensors. This goal requires sensor fusion techniques that are seamless in nature, as well as modular. The *SoD-Toolkit*, primarily relies on the Kinect (version 1 or 2) as the means for providing room-based tracking, as opposed to more expensive and harder to setup motion-tracking systems (Marquardt et al. (2012b)). Given the limitations of a single Kinect tracking system, we use a method of image comparison between multiple Kinect sensors to expand tracking area and improve accuracy.

Generally, the sensor fusion method for multiple Kinect sensors in *SoD-Toolkit* works as follows: (1) A common object (e.g. a water bottle) is placed in the common viewing and tracking area of two Kinects, (2) Two points from each sensor's view are selected (based on the common object) in the Calibration interface in the Visualization tool and (3) One Kinect sensor's vector (the user selected reference sensor) is translated to another Kinect sensor's vector through a custom algorithm and saved. Figure 6 illustrates this specific process from the perspective of the Visualizer tool. The translation of the Kinect data is also reflected in the skeletal tracking, as the Location service compares skeleton data with the list of people being tracked in the environment. The location service then ensures that a person being tracked by multiple Kinect sensors is presented only once by comparing its position with the relative position of tracked users. This process is for two sensors at a time, and for larger spaces with multiple Kinect sensors, the process is repeated for each additional Kinect sensor and the reference sensor.

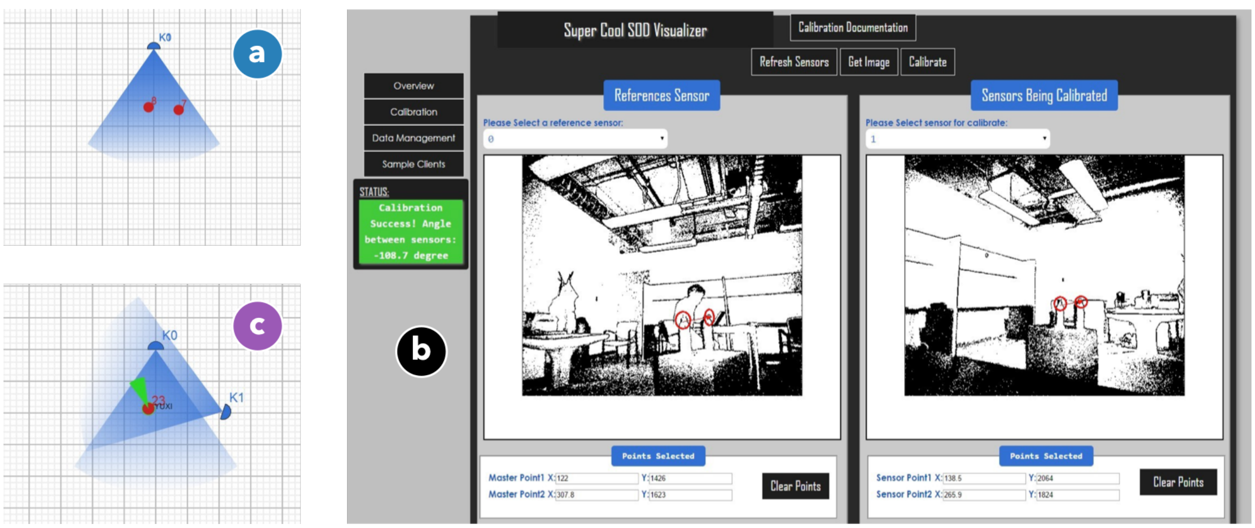


Figure 6: The visualization and prototyping tool for the *SoD-Toolkit*. (a) The tracked environment, (b) Entities tracked in the environment, (c) List of available entities and their current state, as well as clients currently connected in the environment.

Information from Kinect sensors and individual device orientations is also fused by the Location service to provide position and orientation of a device. This type of fusion, which we call a paired state —where a device and user are paired —then allows for multi-device interactions such as "flicking" and "pouring" (Seyed et al. (2012)) that require more accurate spatial-awareness. We also allow for sensor data from other sensors such as the iBeacon and

Leap motion to augment existing information that can be inconsistently provided by Kinect sensors in certain situations (e.g. a user is too far away from the sensor for fine grained finger tracking or is out of consistent tracking range to determine distance). In code, a researcher or developer can simply choose which sensor to use, either based on confidence levels or preference.

2.4 REAL-WORLD CASE STUDIES

To demonstrate the functionality and test the feasibility and applicability of the toolkit for real-world deployments of spatially-aware ubiquitous environments, we present two real-world industry case studies. The implementations we discuss were built in parallel with the toolkit itself, and shaped many of the decisions that led to the features and design of the toolkit. Both of the systems described in this section were built with different teams of developers and industry partners with whom we collaborated closely with. For each case study, we briefly describe the system and reflect on lessons learned from the domain that influenced the design of the toolkit.

2.4.1 *Oil and Gas Exploration*

The very first ubiquitous environment built in parallel with the *SoD-Toolkit* was (Seyed et al. (2013)). We worked closely with SkyHunter Exploration Ltd. who are located in Canada, and specialize in oil and gas exploration, and have proprietary technology that collects a variety of geo-spatial data. The data they collect is multidisciplinary, and ultimately increases the chances of discovering oil and gas significantly. Prior to building the ubiquitous environment, much of their collaboration with the stakeholders in the exploration and decision-making processes (i.e. geophysicists, geologists) was paper-based and ineffective due to large volumes of geo-spatial data, as well as the reliance on single user non-collaborative tools. Upon building the environment with spatial-awareness and various multi-device interactions, we received positive feedback in areas such as collaboration and interaction with geospatial data (Seyed et al. (2013)).

2.4.2 *Emergency Response*

Following the first industry case study for the toolkit and the adjustments made based on developer and industry feedback, we then continued to the emergency response domain. We collaborated with C4i Consultants, also located in Canada, who specialize in training software for military operations and emergency response. They provide a software tool ePlan, which is designed to simulate large scale emergencies, and train civic operators on how to respond with different types and scales of emergencies. Given the collaborative nature of emergency response planning environments and the number of individuals and devices that can be involved in the decision-making process, this was an ideal candidate for building a ubiquitous environment. We built a ubiquitous emergency response environment that built upon ePlan to drive simulations (Chokshi et al. (2014)). It allowed for larger groups of different stakeholders (e.g. fire, police, hazmat) to collaborate and communicate within a spatially-aware environment that contained a large wall display, tablets and a digital tabletop. Spatial interactions such as “flick” and “pour” allowed transfer of vital emergency information amongst the different stakeholders and their devices.

2.4.3 *Lessons Learned*

As we began to develop the toolkit in collaboration with our first industry partner, we relied exclusively on the Microsoft Kinect sensors for spatial-awareness and tracking in the environment. However, when developers deployed the system built with the toolkit and received feedback from system users about the general uncomfortableness of multiple highly visible tracking sensors in an environment (particularly in an office or meeting context), this facilitated our change in the toolkit from *single sensor rigidity* to *multi-sensor flexibility*. We also provided a set of common gestures from multi-device interactions (flick, pour, bump from Seyed et al. (2012)) for the developers to use for information transfer tasks, but our feedback from them indicated *the need for accessing raw information* to customize/ explore more detailed interactions, particularly as the skill on the development team was varied.

After the subsequent changes made in the toolkit based on feedback from developers in the first industry project, we then worked closely with a second industry partner in a different domain, emergency response. A primary challenge for developers of ubiquitous environments in this domain, is information is distributed across the room, usually in highly concentrated regions around specific personnel. At the time, the toolkit wasn't optimized enough to handle regions of coarse and fine grained tracking. This lead to the change of allowing for *sensor transition*, depending on the sensors available in the environment. The developers and industry partner also expressed interest in applying of *proxemic interactions to region specific data*, which led to the development of data points in the toolkit. Lastly, the developers and industry collaborators faced challenges in developing their ubiquitous environment due to limited access to highly confidential and private emergency response environments. To help facilitate developers, we found it important to enable them to *develop unencumbered by room access, devices or sensors*. This created features in the toolkit for easy prototyping, allowing developers to mock ubiquitous environments and multi-device interactions.

2.5 DISCUSSION

Implementing and designing larger ubiquitous environments, applications and multi-device interaction techniques is an extremely complicated task. We introduced the *SoD-Toolkit*that provides researchers and developers with a number of tools that support the design, prototyping and implementation of deployable multi-device, multisensor ubiquitous environments, applications and interactions. We allow researchers and developers to choose their own sensors and devices to build ubiquitous configurations (regardless of owning hardware) and use the tools provided to explore their ubiquitous designs. In this section, we compare *SoD-Toolkit*to other approaches using the thematic framework approach of Olsen (2007), similar to Houben and Marquardt (2015).

2.5.1 *Problem not previously solved*

A number of toolkits such as XDStudio by Nebeling et al. (2014a), Conductor by Hamilton and Wigdor (2014) and Panelrama by Yang and Wigdor (2014) reduced the barrier of entry for researchers and developers in creating and exploring ubiquitous environments, applications and interactions. SoDToolkit builds upon many of the concepts in these toolkits, and further extends the work with greater support for a diverse set of devices and sensor platforms, multi-sensor mappings, and the ability to explore novel multi-device interaction combinations. We also focused on providing a toolkit that is easily configurable and deployable in realworld contexts, a common criticism of toolkits for ubiquitous environments (Nebeling et al. (2014b); Rädle et al. (2014)). The toolkit also draws upon previous work in proxemic interactions and applications (such as Proximity Toolkit by Marquardt et al. (2011) and Grouptogether by Marquardt and Greenberg (2012)), generalizing ubiquitous approaches, and spatial awareness, allowing for the exploration and development of different facets of research in ubiquitous environments, such as multidevice interaction techniques and sensor fusion approaches.

Another limitation of several existing toolkits lies in the physical reliance on sensor and device hardware for prototyping and developing ubiquitous environments and multi-device interactions. We allow researchers and developers to explore multi-device, multi-sensor ubiquitous environments without the need for all hardware or sensor components, which becomes increasingly important and relevant as the underlying technology moves rapidly both in terms of cost and capability. Reducing this limitation allows for an expansion of research into ubiquitous environments.

2.5.2 *Generality*

To demonstrate the expressivity of the building block components of the SoD toolkit (Olsen (2007)), we discussed two industry case-studies that included interaction techniques and scenarios considered state of the art (Chi and Li (2015); Seyed et al. (2013)). We also highlighted the versatility and generality with these industry case studies, but through them recognized the limitations and strengths of the toolkit. A number of limitations of the toolkit, particularly in its focus for providing larger ubiquitous environments stem from the usage of the Kinect hardware. The Kinect does not provide the same level of extremely accurate tracking by large scale systems (e.g. Vicon) even with multiple Kinect sensors. This is further enhanced when coarser grained sensors (e.g. iBeacons) are used entirely. However, based on our collaborations with industry partners, we believe that for more real-world deployments of ubiquitous environments, this is a necessary trade-off. This provides an advantage, as hardware and sensor limitations are nullified by simply using existing hardware and sensor setups to develop novel and more futuristic multi-sensor, multi-device ubiquitous environments, applications and interactions. Furthermore, the toolkit can easily be extended to support newer sensor and hardware platforms as they emerge.

2.5.3 *Reducing Solution Viscosity*

Compared to other methods to develop real-world deployable multi-sensor, multi-device ubiquitous environments, applications and interactions, the *SoD-Toolkit* dramatically lowers the development viscosity (Seyed et al. (2013)) by providing a flexible architecture that allows for ”expressive leverage” (Houben and Marquardt (2015)). The design of the architecture also allows for the potential to transform several existing multi-device toolkits (e.g. XDStudio by Nebeling et al. (2014a)) into multi-sensor and spatially aware toolkits. Furthermore, as the toolkit abstracted and simplified many challenges in creating complex ubiquitous environments (as discussed previously), novice researchers and developers can focus on creating novel systems and interactions with minimal overhead, while those with more experience can freely modify different aspects of the toolkit, such as sensor-fusion approaches and sensor support.

2.5.4 *Empowering new design participants*

The lack of real-world deployable examples of larger multi-device ubiquitous environments (prototypes or otherwise) despite widely available sensors and devices, indicates a need for adequate toolkit support for researchers and developers. In the research literature, a number of elicitation studies have been performed to study interactions in ubiquitous environments, but very little follow-up work is typically performed, particularly from an implementation perspective. Additionally, multi-device explorations with newer forms of devices in ubiquitous environments (e.g. wearables) is still very minimal in the research literature, particularly in multi-sensor spatially-aware ubiquitous environments (Houben and Marquardt (2015)). *SoD-Toolkit* focuses on lowering the threshold for non-expert programmers, researchers and interaction designers and empower them to explore with new and existing sensor and device technologies for real-world deployments and contexts. To date, we have not performed an in-depth study with developers for the toolkit, but we have worked closely with developers and industry from the onset of implementation and design of the toolkit. This remains as future work, but we believe the toolkit is an important step in exploring novel larger scale ubiquitous environments with multiple sensors and multiple devices, as well as necessary for exploring novel multi-sensor, multi-device interactions.

2.5.5 *Power in combination*

Similar to prior toolkits, *SoD-Toolkit* combines hardware and sensor management, sensor fusion techniques, networking, disributed interfaces and prototyping into a single toolkit. As mentioned earlier, the architecture design allows for these components to be extremely decoupled, allowing for the addition or subtraction of different approaches. Additionally, as the *SoD-Toolkit* supports multiple platforms, it already integrates with several major UI frameworks (e.g. Objective C and Xcode), meaning developers can both prototype and build upon commercially available hardware.

2.5.6 *Can it scale up?*

In building multi-sensor, multi-device spatially-aware ubiquitous environments, particularly ones designed for deployment, the scalability of the sensors that provide spatial awareness for multi-device interactions becomes an issue. For toolkits such as Conductor by Hamilton and Wigdor (2014) and Panelrama by Yang and Wigdor (2014), the solutions are scalable as they do not rely on sensors and use scalable web-based technologies (e.g. HTML5). Alternatively, toolkits such as Proximity Toolkit by Marquardt et al. (2011) and XDKinect by Nebeling et al. (2014b), are either scalable but face challenges for real-world deployment due to complex setups or are limited in functionality due to limited sensor support. Undoubtedly, balancing spatial awareness and scalable solutions is difficult, especially when not using motion-tracking systems that aren't feasible for deployment (Nebeling et al. (2014b); Nebeling et al. (2014a)). The design of *SoD-Toolkit* supports multiple low-cost off the shelf sensors that can be used in concert, for a larger spatially-aware ubiquitous environment. Additionally, as *SoD-Toolkit* allows researchers to choose their sensors and devices, researchers and developers can choose their granularity of spatial-awareness in a ubiquitous environment.

2.6 CONCLUSION

The *SoD-Toolkit* offers a rich set of tools and software libraries for researchers and developers to prototype and develop deployable multi-device, multi-sensor interactions, applications and environments. We allow researchers and developers to easily “plug-and-play” popular off-the-shelf hardware (sensors and devices) to build their ubiquitous environments, while removing network and device management, sensor fusion and complex algorithmic challenges. The toolkit abstracts these challenges into an easily accessible API and events, integrating with several existing UI development software tools. As a result, the toolkit reduces complexity and lowers the barrier of entry for researchers and developers to design larger and more intricate multi-device ubiquitous environments. The toolkit also allows for future explorations of novel multi-device interaction techniques, multi-sensor designs and other unique ubiquitous environments that can be deployed in new and unexplored contexts and domains. Future work for the *SoD-Toolkit* includes integrating powerful device-centric sensors like the Google Tango, and gesture detection for different types of interactions in ubiquitous environments.

3

CASE STUDY I - SKYHUNTER: A MULTI-SURFACE ENVIRONMENT FOR SUPPORTING OIL AND GAS EXPLORATION

The process of oil and gas exploration and its result, the decision to drill for oil in a specific location, relies on a number of distinct but related domains. These domains require effective collaboration to come to a decision that is both cost effective and maintains the integrity of the environment. As we show in this paper, many of the existing technologies and practices that support the oil and gas exploration process overlook fundamental user issues such as collaboration, interaction and visualization. The work presented in this paper is based upon a design process that involved expert users from an oil and gas exploration firm in Calgary, Alberta, Canada. We briefly present knowledge of the domain and how it informed the design of SkyHunter, a prototype multi-surface environment to support oil and gas exploration. This paper highlights our current prototype and we conclude with a reflection on multi-surface interactions and environments in this domain.

***Note: The text in this chapter appears in the following publication:**
Teddy Seyed, Mario Costa Sousa, Frank Maurer, and Anthony Tang. 2013. SkyHunter: a multi-surface environment for supporting oil and gas exploration. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces (ITS '13).* ACM, New York, NY, USA, 15-22.

3.1 INTRODUCTION

Multi-Surface Environments are systems where interaction is divided over several different displays, which includes digital tabletops, wall displays, tablets and mobile phones (Seyed et al. (2012)). Because of the different sizes and capabilities of the displays in such an environment (e.g., resolution, mobility), they can support a wide range of different tasks and interactions. Key challenges for multi-surface environments still remain however. These challenges include finding what tasks can be accomplished in these environments and how collaboration can be made effective in these types of environments (Seyed et al. (2012)). Significant research has been done into different types of interactions, as well as collaboration for multi-display environments; however, very little work has gone into exploring multi-surface environments with real-world industrial partners. To move these potentially useful environments into the commercial space, as well as into the hands of industry, some benefits should first be displayed. As a result, we explored on the concepts of interaction and collaboration for multi-surface environments in the context of a specific domain, oil and gas exploration.

The oil and gas exploration process is both complex and multi-faceted. In a typical exploration project, several domains – geosciences, reservoir and production engineering, geophysics – must work together in a timely manner to achieve oil production goals as well as maintain the safety of the environment and personnel in the field. Multi- disciplinary teams are extremely common and their collaborative exchange is a necessity for the oil and gas exploration industry (Corbett et al. (2011)). The importance of the collaborative activities of a multi-disciplinary team is shown by their effect on operating costs of an oil and gas company, as well as the resulting activities from their decisions. These decisions require a meticulous process, strong collaboration and communication, as well as a common understanding of the exploration process (Sultanum et al. (2010)).

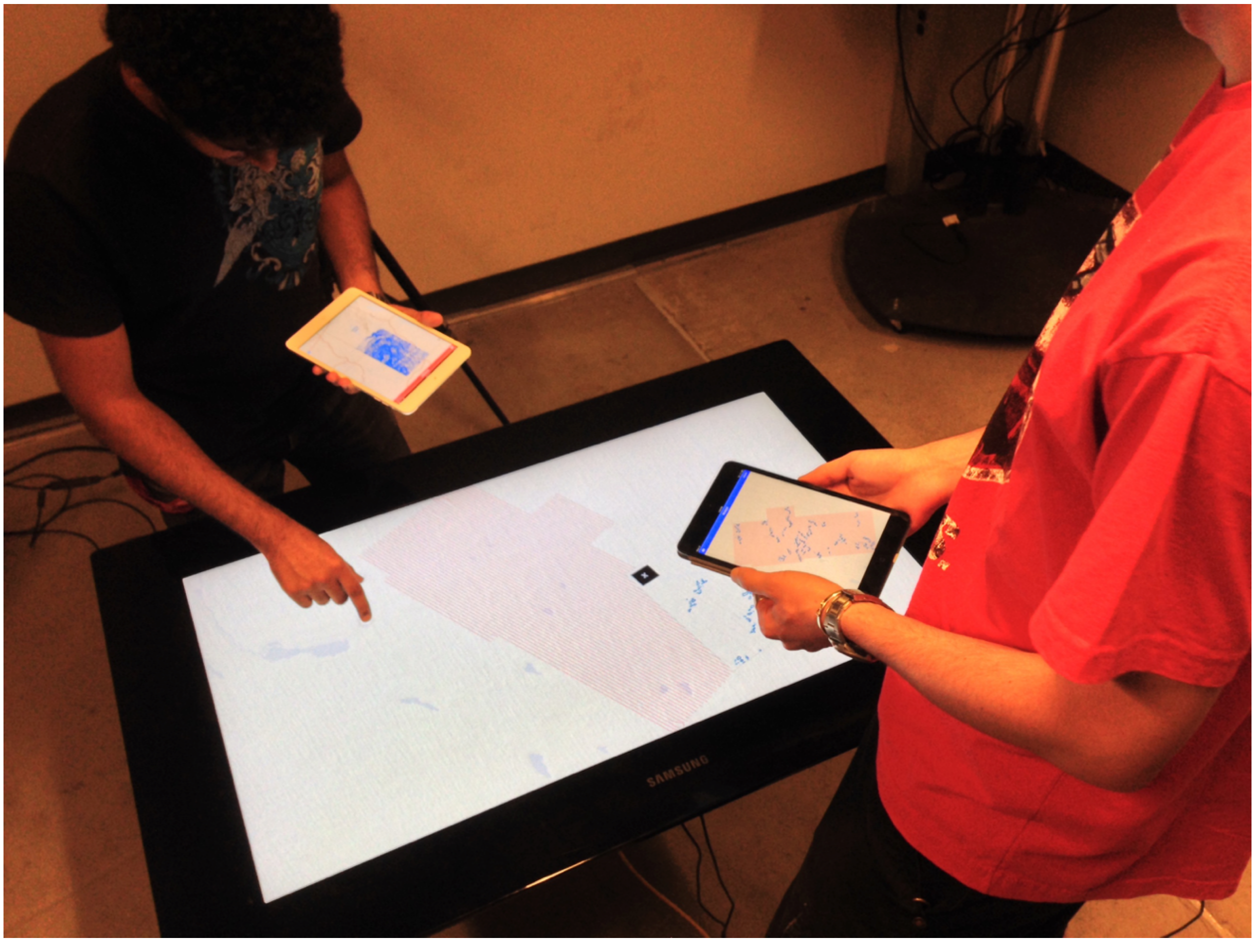


Figure 7: The SkyHunter multi-surface environment running with multiple iPads and a digital tabletop.

To address the research aspects and guide ourselves in the creation of a domain-specific interface for oil and gas exploration, we worked with Sky Hunter Exploration Ltd., who collect proprietary multi-disciplinary data, called microseep maps, which significantly increase the chances of finding oil and gas. We initially sought feedback about the domain and their current practices with the data, which included paper-mache mockups and paper maps and thus designed and developed a multi-surface environment (see Figure 7) to address not only this data but facilitate in the collaborative processes of oil and gas exploration. In this paper, we present our system designed with Sky Hunter Exploration Ltd. and share our reflections.

As a whole, the research space of multi-surface environments is very well explored. Significant research has been done in exploring the different ways in which the displays can be treated – continuously (Johanson et al. (2002); Rekimoto and Saitoh (1999)) or discretely (Prante et al. (2004); Streitz et al. (1999)) – as well as different interactions, such as flicking (Dachselt and Buchholz (2009)), or picking and dropping (Rekimoto (1997)), among others. Furthermore, the individual components of multi-surface environments, such as digital tabletops, have been shown to increase collaboration significantly and effectively (Scott et al. (2010)). A unique advantage of multi-surface environments consequently, is the benefits they can provide as a collaborative workspace. Examples of collaborative workspaces in the research literature include Collab by Stefik et al. (1987), which allowed groups of users to work together on desktop PCs and a large-scale wall display. Dynamo by Izadi et al. (2003) is another example that allowed users to move information to a shared wall display.

The sharing and connection of information is also highlighted by Streitz et al. (1999), who created an environment utilizing a digital tabletop, wall displays, and custom displays attached to chairs. This sort of interactive collaborative space is also shown in the iRoom project by Johanson et al. (2002), which allowed users to move content around different displays and devices.

Having different displays and devices in these collaborative multi-surface environments has been shown to lead to new discoveries or help support existing hypothesizes, particularly in the domain of astrophysics (Wigdor et al. (2009)). Furthermore, Tani et al. (1994) showed that displays in these environments can improve productivity significantly for spatial tasks. These collaborative spaces have also been explored in supporting the learning of abstract knowledge through both collaboration and interaction (Shaer et al. (2011);Shaer et al. (2010)).

For the oil and gas domain, the focus of this work, several different technologies have been explored to address collaboration. These include visualization rooms, haptic devices, as well as virtual reality (Bowering (1999)). A significant challenge presented by many of these interactions and technologies however, is they are limited to single user interaction. For oil and gas exploration, which is multidisciplinary, this is insufficient. In the context of multi-surface environments, there is very little research into oil and gas exploration. To our knowledge, this is one of the first prototypes to attempt to map the oil and gas exploration process to a multi-surface environment.

3.3 OIL AND GAS EXPLORATION

The process of oil and gas exploration and production involves many complex tasks, with multi-phase workflows and depends on a number of different variables from different groups of inter-related disciplines, such as geophysics, geology and engineering (Cosentino (2001)).

In a new exploration project, field measurements are used to gather different types of information about a potential location for drilling an oil well. Much of this measurement information strictly belonged to the aforementioned disciplines, which bring with them different perspectives and sometimes conflicting solution strategies (Sultanum et al. (2011)).

To support collaborative exchange in oil and gas exploration, a number of software tools are currently used. These tools interpret geological and geophysical data (among others) and result in typically 2D (and 3D) geospatial images and maps that are unique to the domains involved and often have different modalities and scales. This information is then used to facilitate in discussions to determine the best possible locations to drill for oil and gas.

However, these software tools and their output don't easily allow for collaboration. The processes are clumsy and there is a strong need for computational and visualization tools that properly integrate data from the numerous domains (Zamel et al. (2001)). Furthermore, the data in its current form isn't interactive for exploratory analysis and direct manipulation. This problem is further enhanced with the newer data collection techniques that exist in the oil and gas domain that result in data that require multi-disciplinary visualizations and analysis.

Sky Hunter Exploration Ltd., an oil and gas exploration company located in Calgary, Alberta, Canada, uses a proprietary measurement technique that detects charged particles leaking from the ground with a customized airplane (see Figure 8). The output of this measurement technique, after specialized interpolations, is a hydro-carbon map (see 9b). This map, when combined with data from the other disciplines in the exploration process, results in a significantly more informed decision for a drilling location.

The unique challenges of the data that Sky Hunter Exploration Ltd. present to the oil and gas exploration processes are presented in Figure 9. While a majority of the maps are in 2D (e.g. Figure 9a, 9b, 9c), the image planes can be at different depths or mixed. For example, in Figure 9d and Figure 9e, a hydrocarbon map is viewed in a top-down plane while the corresponding cross-slice is viewed from a side-plane, in a single mixed-plane image. For Sky Hunter Exploration Ltd., this data presents a conflict of modalities and is a significant barrier for its introduction into the decision making process for multi-disciplinary teams.

The workflow for a multi-disciplinary team using this unique data as described by Sky Hunter Exploration Ltd. is the following:

Step 1: The entire multi-disciplinary team surveys a prospective area for wells. Specifically, the *Land Man* – who has knowledge of the owners of any potential properties to drill oil

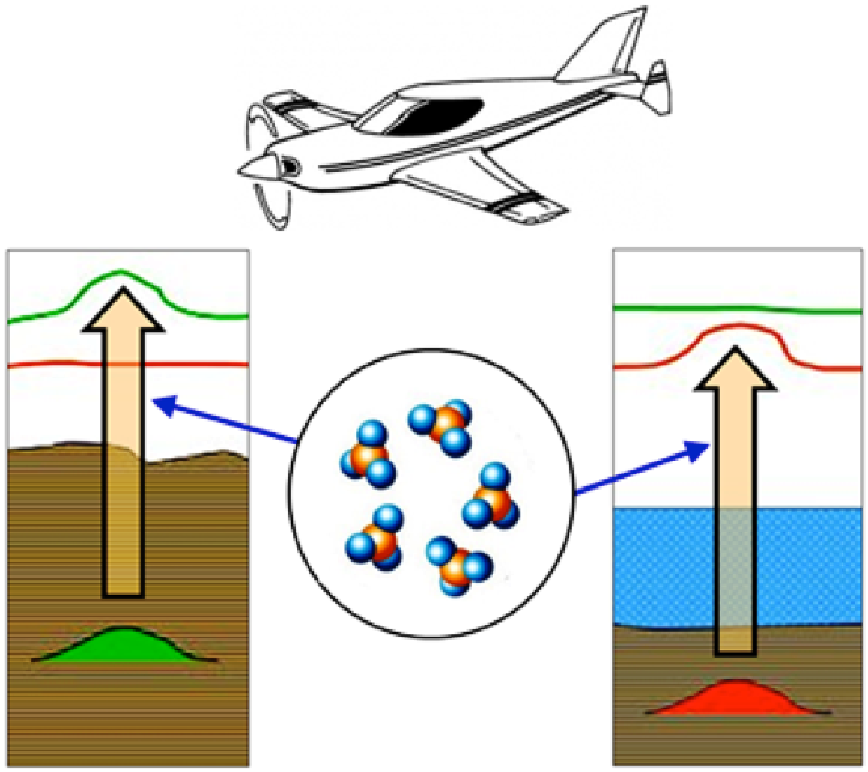


Figure 8: Sky Hunter Exploration Ltd. uses an airplane to fly to survey an area while using an air sampling device to record the intensities of hydrocarbons that leak from the ground.

upon – and the *Pilot* – who flies the customized airplane – are used to decide where to start an initial exploration.

Step 2: After the initial exploration by the *Pilot* has been completed, interpolation is performed on the measurements and the output, *hydro-carbon* paper maps are then used to highlight the results of the exploration. These maps are printed at different scales and different modalities for the various disciplines of the team and visually indicate zones of the highest concentration of charged particles when overlaid. Thus, this indicates the locations with the highest likelihood for successfully drilling a new oil well..

Step 3: The *Geophysicists* provide seismic information, while the *Geologists* provide sub-surface information, such as subsurface formations, all of which is also paper based. Com-

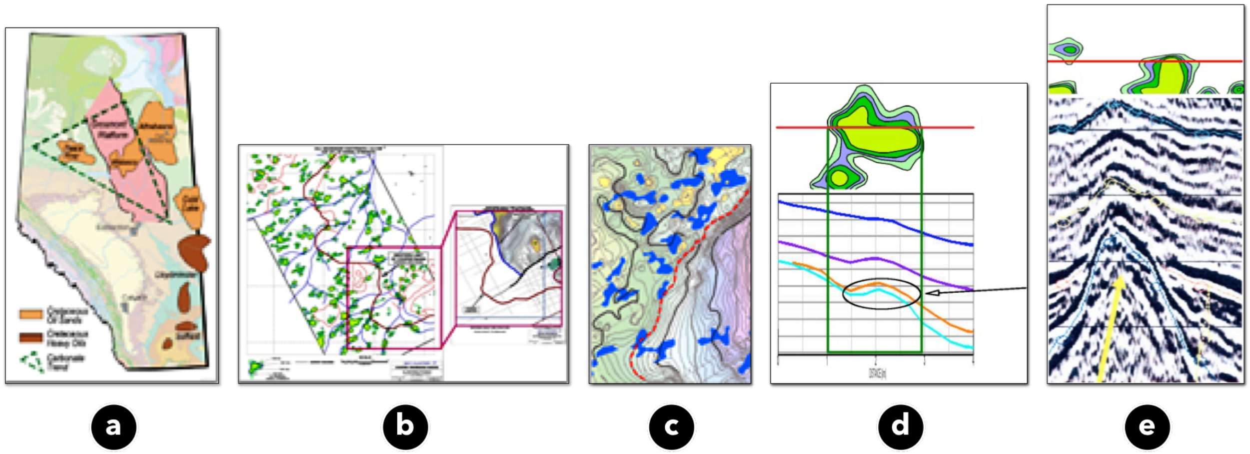


Figure 9: The various modalities and scales of data for Sky Hunter Exploration Ltd. (a) 2D map representing the province of Alberta and oil fields, from a top-down view (b, c) A hydrocarbon map including microseep footprints combined with a secondary image (both in top-down view) representing subsurface information underneath (d, e) Zoomed hydrocarbon map with microseep footprints, from a top-down view, with appropriately scaled subsurface information underneath (seismic data), from a side-view.

bined, these two disciplines provide visualizations and contextual knowledge for the entire multi-disciplinary team about the exploration environment; in addition, with the hydrocarbon maps, they provide a different visual perspective underneath the zones.

Step 4: The numerous engineering domains – reservoir, drilling and production – then use the static paper based data that has been combined thus far, to create "flow" models which may be used to determine the best location to drill an oil well, as well as strategies on how to drill a location.

Step 5: If a location has been determined, the production engineers then create an economically and environmentally viable production plan for drilling the location, which is then presented to a sponsoring company for the ultimate decision of whether to purchase the exploration area (if needed) and drill the location.

Overall, this workflow is highly collaborative and is tightly integrated with paper based processes and visualizations, which presents a number of unique challenges when building a multi-surface environment. This is described in the next section.

3.4 SKYHUNTER AND INFRASTRUCTURE

To support the workflow as described by Sky Hunter Exploration Ltd., we designed the *SkyHunter* multi-surface environment. This environment supports different visualizations of data for the multi- disciplinary team and also provides integration for numerous types of data that are available in the workflow described earlier. The *SkyHunter* multi-surface environment was designed in collaboration with domain experts from Sky Hunter Exploration Ltd., and this section summarizes the design considerations and its features.

3.4.1 *Design Considerations*

While working with Sky Hunter Exploration Ltd., we continually discussed and iterated over three main elements in the design, which are as follows:

1. Simplification of the different modalities of data and interacting with it.
2. Providing different domains access to their own private data and a means to share it with the team.
3. Bringing together all the data.

The first design element is related to the simplification of the interactions with the data in the workflow. As described by an expert from Sky Hunter Exploration Ltd., "*if I want* *to view seismic information flat, or in another orientation at the same time, I can simply click a button or another means that allows me to do this.*" This means, accommodating the different modalities (mixed 2D planes and 3D) with the different devices in the environment, as well as providing more opportunities for seamlessly interacting with the data, instead of "*continually printing out different scales of map data on lots of paper.*"

The second design element is of interest to Sky Hunter Exploration Ltd., as they described many of the disciplines to be "*somewhat sensitive of their own data in as they may have sensitive information they don't necessarily want to share (yet) or information they'd like to sell for profit later.*" Along these lines, providing the means for the different disciplines to interact with their own unique data in both a private and public manner, while preserving ownership, is extremely important.

The last design element, was also of importance, as it was described as "*a unique challenge with our data is that it requires a lot of other data to be properly understood. Being able to integrate seismic or subsurface maps in a system easily would make our process significantly easier.*" Integrating the different types of map information should be supported by the *Sky-Hunter* multi-surface environment, especially considering the number of domains and their data.

3.4.2 *Infrastructure*

As shown in Figure 10, SkyHunter is a multi-surface environment comprised of a number of components. To build this interactive environment, the MSE-API[1] framework was used. This framework, provides information such as device orientation and device location when utilized

1 MSE-API (SoD Toolkit) —- https://www.sodtoolkit.com

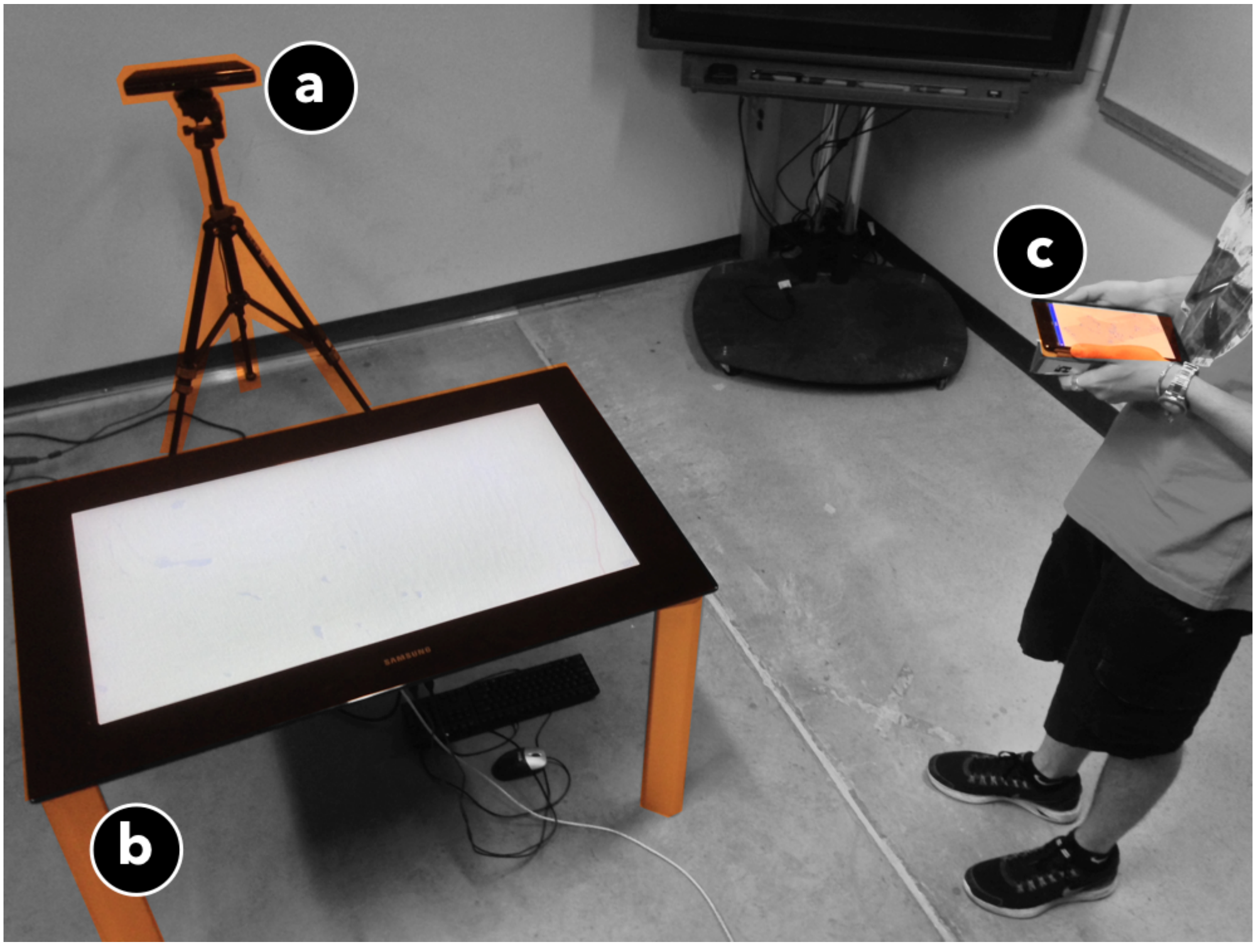


Figure 10: Overview of SkyHunter. (a) Microsoft Kinect used with MSE-API for providing tracking and orientation information (b,c) Tabletop and iPad running custom applications.

with the Microsoft Kinect[2] and easily allowed multi-surface interactions to be created for the *SkyHunter* multi-surface environment.

Specialized applications were also created for iPads[3] and the tabletop utilizing ESRI's ArcGIS API[4], which provides mapping capabilities and gestures such as pinching and zooming, as well as the automatic scaling of map data. All of the proprietary information provided by

2 Microsoft Kinect —http://www.xbox.com/en-CA/Kinect
3 Apple iPad - http://www.apple.com/ca/ipad/
4 ESRI ArcGIS —http://www.esri.com/software/arcgis

Sky Hunter Exploration Ltd, is stored on a backend ArcGIS server, providing an integrated solution for the different types of data for the different disciplines.

3.4.3 *Realizing the Design*

An important component in the multi-surface environment for maintaining collaboration present in the oil and gas exploration process is the Samsung SUR40 digital tabletop[5]. In the environment, it is used as the main hub of collaboration in the *SkyHunter* multi-surface environment. The software designed for the digital tabletop replaces much of the paper based interactions that are described in the exploration process, particularly for cases where maps of different scales and types need to be viewed and overlaid simultaneously. The tabletop also serves as the primary location where data from different sources and disciplines can integrated in the exploration process (particularly in Steps 2-3).

Much like the exploration workflow described earlier, where a domain expert provides a paper-based map for discussion, the tabletop initially contains no data until a domain expert provides it to the tabletop to begin the collaborative process. Unlike the paper based workflow however, a number of multi-surface interactions are provided to simplify privacy, interaction and sharing of this data.

To establish private domain data, individual iPads are used as a means to distinguish the multi-disciplinary roles in the application, and selecting an appropriate role on the startup of the iPad application allows for appropriate data to be displayed and interacted with. For instance, a *Landman* in the application is only able to view and share well data while a *Geophysicist* similarly will only be able to view and share subsurface formation data. This

5 Samsung SUR40 - www.samsunglfd.com/solution/sur40

allows for stricter control of data ownership and sharing, described as an important design consideration by Sky Hunter Exploration Ltd.

Interaction and sharing of data, as shown in the workflow, is critical to the collaborative process. Individually, both the tabletop and iPads provide pinching and zooming interactions for map data, which is useful for individual or group based interactions. However, to replace the paperbased interactions in the workflow, such as bringing a paper map to a central location or sharing a map with a specific domain expert, multi-surface interactions are available for users.

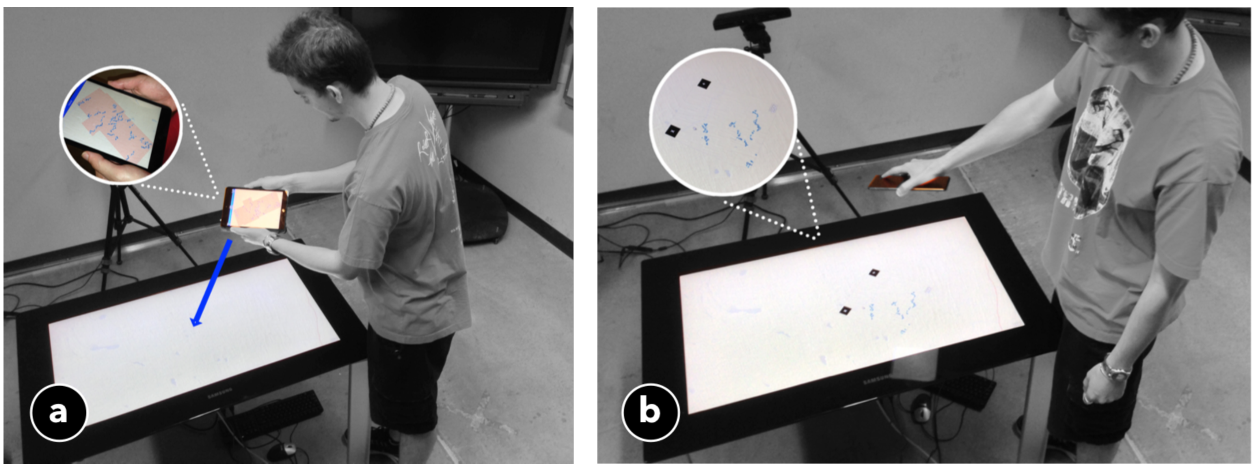


Figure 11: *Pouring* data. (a) The user performs a pouring interaction onto the tabletop after selecting subsurface data on the iPad (b) The subsurface data appears on the digital tabletop after the pouring interaction is completed.

The *pour* interaction is used to send selected map data from an iPad to the tabletop, in close proximity (see Figure 11). Similarly, a *flick* gesture (see Figure 12) is also used to send selected map data to the tabletop, but is not restricted by distance to the tabletop, unlike the *pour* interaction. Both these interactions are useful for Steps 2-4 in the workflow, where users provide data to collaborate and make decisions based upon combined data from the different disciplines. Additionally, the *Camera* gesture (see Figure 13) is used to capture the shared visible data on the tabletop on a user's iPad at the conclusion of the workflow.

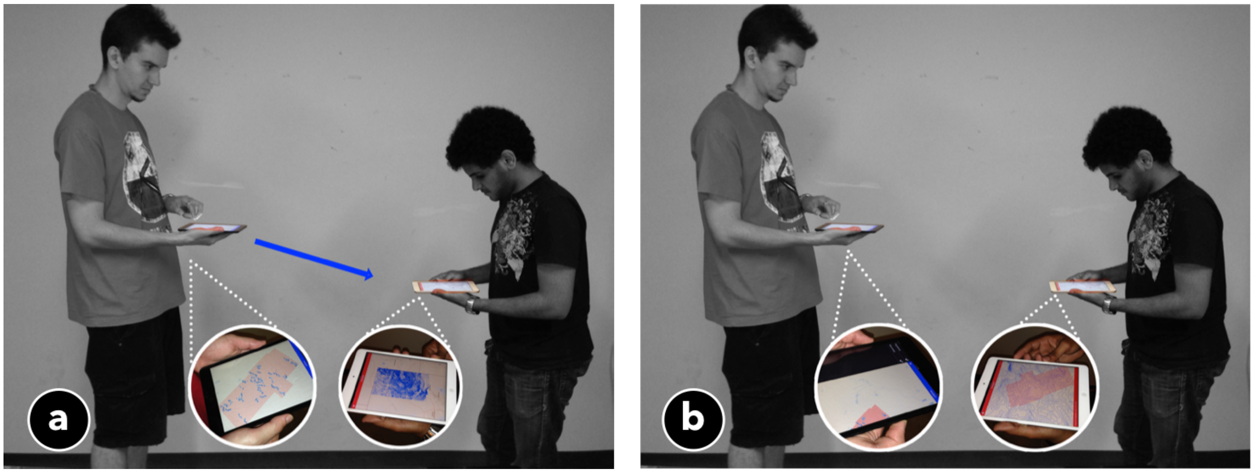


Figure 12: *Flicking* data. (a) The user selects geological formation data to send and flicks towards the other user. (b) The geological formation data appears on the targeted user's device.

As mentioned earlier, one of the biggest challenges of the Sky Hunter Exploration Ltd.'s data in the workflow is its different modalities and interacting with it, highlighted earlier in Figure 9. Due to the complicated nature of the data, its requirement of other data to be properly understood and unclear visualizations, there is a negative cascading effect on the workflow, particularly in Steps 2-3. To provide a better means of visualizing and interacting with seismic information (typically cross-slices) provided by *Geophysicists* and the *hydrocarbon* maps and other geographical information used in the workflow, a *slicing* interaction is used in SkyHunter. The *slicing* interaction allows the viewing of multi-modal information by combining information on the digital tabletop and the iPad. This is performed by placing the iPad down on the tabletop vertically (see Figure 14), resulting in seismic information being displayed on the iPad in the correct orientation, while a 2D map with various information is still visible on the tabletop. This specifically resolves the data representation issue presented in Figure 9c, where a top-down image can now be presented on the tabletop and the iPad can be used to display the side-view cross-slice.

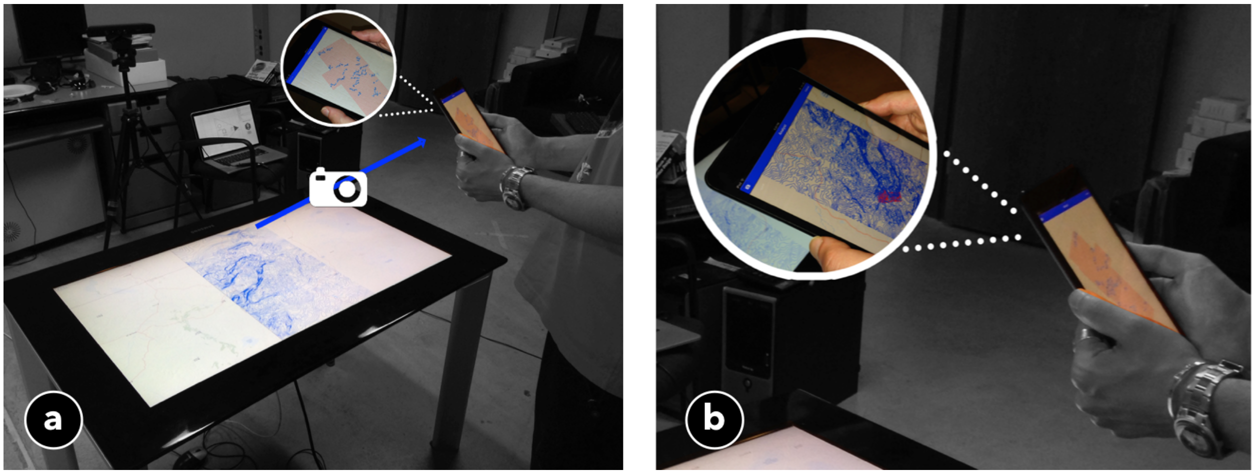


Figure 13: *Camera* interaction. (a) The user points the iPad at the tabletop and selects a button to capture an image of the data available on the tabletop, in this case, geological formations (b) The geological formation data appears on the user's iPad after selecting the data on the iPad to capture.

3.5 EARLY DESIGN CRITIQUE AND DISCUSSION

Getting feedback from users, especially in the multidisciplinary domain of oil and gas exploration is extremely critical. As we worked very closely with domain experts from Sky Hunter Exploration Ltd., we asked them to continually provide feedback through the various stages of our collaboration, including the prototype that is presented in this work. The goal of this extremely early feedback was to discuss the potential of multi-surface environments and applications to their domain as well as to brainstorm future development ideas. The feedback received is presented below in general themes.

Ease of Interaction: The feedback for being able to interact with all of the different data was extremely positive. Many comments were about how "*using pinch and zoom, exactly the same on the iPads and the large tabletop for the hydro carbon maps is useful. It's far better than printing out tons of maps for the same information*". It was also noted that the data was far easier to understand and manipulate now that it was easily accessible to the different

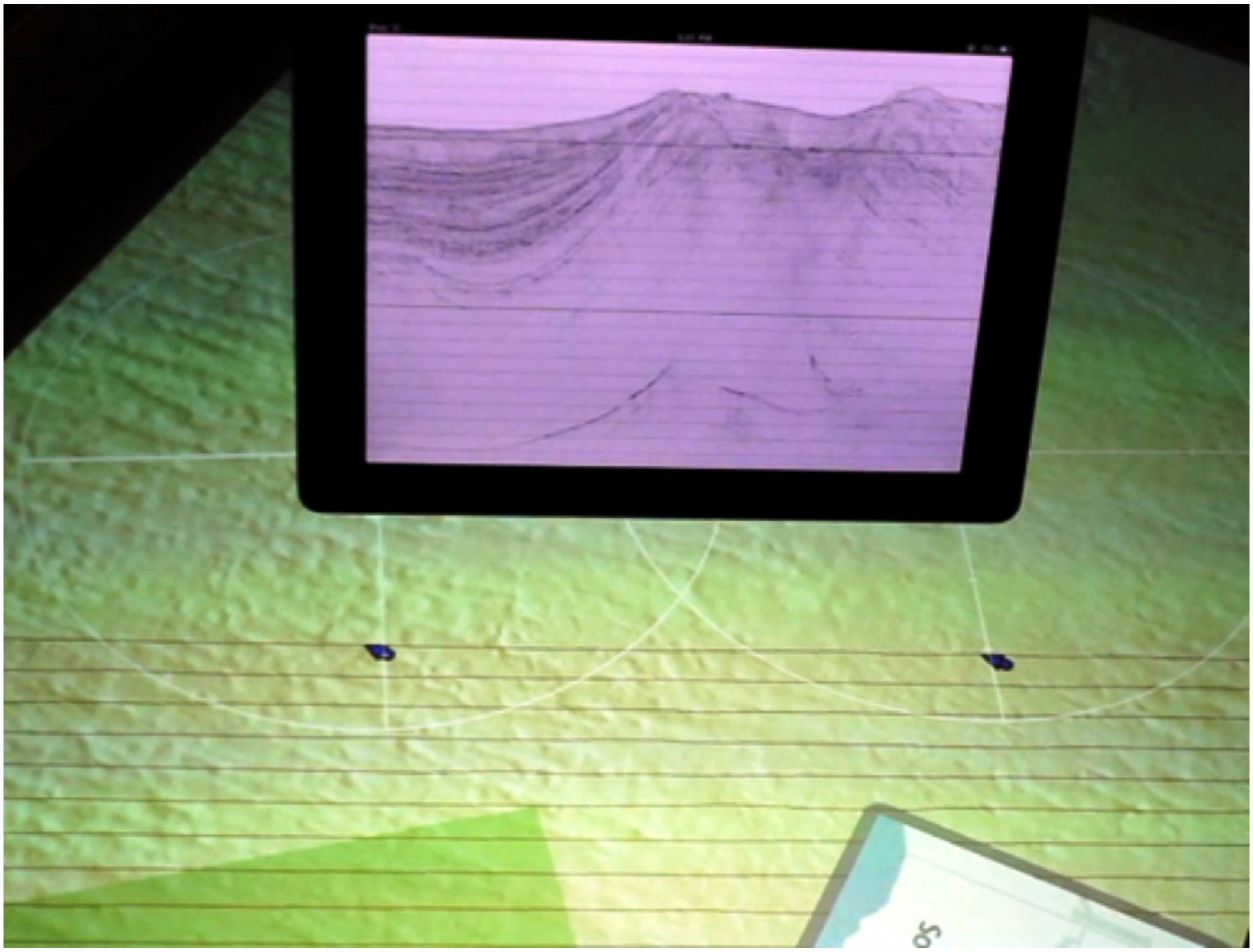


Figure 14: *Slicing* interaction, allowing a user to see seismic information underneath a selected microseep when an iPad is placed on the tabletop.

disciplines across the different devices in the environment. It was also suggested in a future version, that the different displays be synchronized to ensure everyone was on the same page in the collaboration, especially if they were in a remote location.

Collaboration: A marked improvement noted by Sky Hunter Exploration Ltd, was the increased collaboration that resulted from the digital tabletop and wall display in the environment. A geologist noted "*it was far easier to collaborate with my data now that everyone is centered on a tabletop. I can match up my data with the hydro-carbon maps easily and I can then get more data later if I need it*". The collaborative aspects of the tabletop for map based data was also highlighted, with the statement "*the tabletop is a great way to get everyone in the company together and on the same page. It's much better to have everyone gathered somewhere together, than sitting at their laptops and staring at a PowerPoint presentation*".

Visualization Enhancement: Given that the prior visualizations for Sky Hunter Exploration Ltd.'s data (and the decision workflow) was paper based or a paper-mache construct, placing it on different devices was an instant enhancement. The tabletop was also found to be useful for providing a big picture context of an area, as well as a means of sharing visual analysis of data or a specific area.

Multi-Surface Gestures and Interactions: The gestures implemented for *SkyHunter*, were based on previous work (Seyed et al. (2012)). We initially assumed that there would be interactions that would be close to a central location or further away (i.e. using a flick gesture to tabletop as a geologist could be working privately, away from tabletop). The immediate reaction to these interactions for this domain wasn't positive. The geologist of Sky Hunter Exploration Ltd. noted that "*while the interactions are cool, they aren't useful for this domain, as they might require additional training, or an IT guy who can impede the whole process and make everyone uncomfortable.*" Interestingly however, the camera gesture was well received and was found to be an easy way to take data away from a meeting, especially after everyone had agreed upon a drilling location.

Alternate Sources of Data: One of the most interesting aspects of the workflow of oil and gas exploration process, is the different types of data. A specific use case that Sky Hunter Exploration Ltd brainstormed was if different contractors entered the discussion with paper based maps, being able to overlay this data physically with the digital maps, would be extremely beneficial. This type of mixed modality interaction although unique, provides the perspective that not everything should be considered "digital" in this domain.

3.6 CONCLUSION

In this paper, we explored the use of multi-surface environments and focused on issues such as *collaboration*, *interaction* and *visualization* specifically, in the context of the oil and gas exploration domain. We approached the design of the multi-surface application by working with Sky Hunter Exploration Ltd. from initial brainstorming discussions to a working prototype and present a preliminary design critique and discussion about their utility for the domain. In the future, we plan to extend the prototype further and continue with a more complete evaluation of its system and its impact on the workflow of oil and gas exploration processes. We hope this initial work will trigger greater interest in applying multi-surface environments and applications to this domain, as we believe they can benefit greatly from these technologies, especially from an HCI perspective.

4

CASE STUDY II - EPLAN MULTI-SURFACE: A MULTI-SURFACE ENVIRONMENT FOR EMERGENCY RESPONSE PLANNING EXERCISES

Emergency response planning is a process that involves many different stakeholders who may communicate concurrently with different channels and exchange different information artefacts. The planning typically occurs in an emergency operations centre (EOC) and involves personnel both in the room and also in the field. The EOC provides an interesting context for examining the use of tablets, tabletops and large wall displays, and their role in facilitating information and communication exchange in an emergency response planning scenario. In collaboration with a military and emergency response simulation software company in Calgary, Alberta, Canada, we developed ePlan Multi-Surface, a multi-surface environment for communication and collaboration for emergency response planning exercises. In this paper, we describe the domain, how it informed our prototype, and insights on collaboration, interaction and information dissemination in multi-surface environments for EOCs.

***Note: The text in this chapter appears in the following publication:**
Apoorve Chokshi, Teddy Seyed, Francisco Marinho Rodrigues, and Frank Maurer. 2014. ePlan Multi-Surface: A Multi-Surface Environment for Emergency Response Planning Exercises. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14).* ACM, New York, NY, USA, 219-228.

4.1 INTRODUCTION

Large scale emergencies and disasters highlight the vulnerability of modern society to collapses of infrastructure that is crucial to daily life (e.g. roads, phone service, and electricity). A significant challenge with emergencies also arises from the different types that can occur, from unplanned events like natural disasters, train derailments, and chemical spills, to planned large-scale events like the Olympics and the World Cup.

Events such as the 2013 floods in Southern Alberta, as well as other recent major events and natural disasters have resulted in significant efforts by authorities worldwide to investigate how information and communication technologies (ICT) can both facilitate and improve upon existing emergency planning and response capabilities. These technologies are primarily found in emergency operations centre (EOC), where trained personnel need to make informed decisions in situations that can be both stressful and highly volatile while information is uncertain and incomplete.

For EOC personnel, a primary challenge is the number of information sources and amount of data that needs to be analyzed and continually monitored during an emergency. These sources include personnel in the field (e.g. firefighters, police, emergency medical services (EMS) or military) or third party sources (e.g. newspapers, television channels or citizens). As Walle and Turoff (2007) noted, "*accurate and timely information is as crucial as is rapid and coherent coordination among the responding organizations.*"

For each information source, there are different protocols before information can be exchanged effectively in the EOC and between personnel. The source of information determines how it can enter the EOC (i.e. via video, audio, or text). A traffic or incident camera could live-stream into the EOC, tweets could arrive via text, and information from ground-personnel may arrive via text messages, by phone or radio. EOC personnel need to ascertain the importance, authenticity, and accuracy of the information. Emergency personnel report through their chain-of-command, reporters (print, web, television and radio) may have their information fact-checked before broadcast, while citizens may live-tweet, post updates, or send emails as an emergency unfolds. When the EOC receives information updates from their personnel, reporters, or citizens, they may also need to monitor developing traffic congestion, incident cameras, and operational decisions that are continually made.

Multi-surface environments (MSE), are environments that contain multiple heterogeneous devices (e.g. tablets, wall displays, tabletops) that are spatially aware of each other, as well as the users in the environment (Seyed et al. (2012)). This provides an environment amenable to emergency response planning in multiple ways.

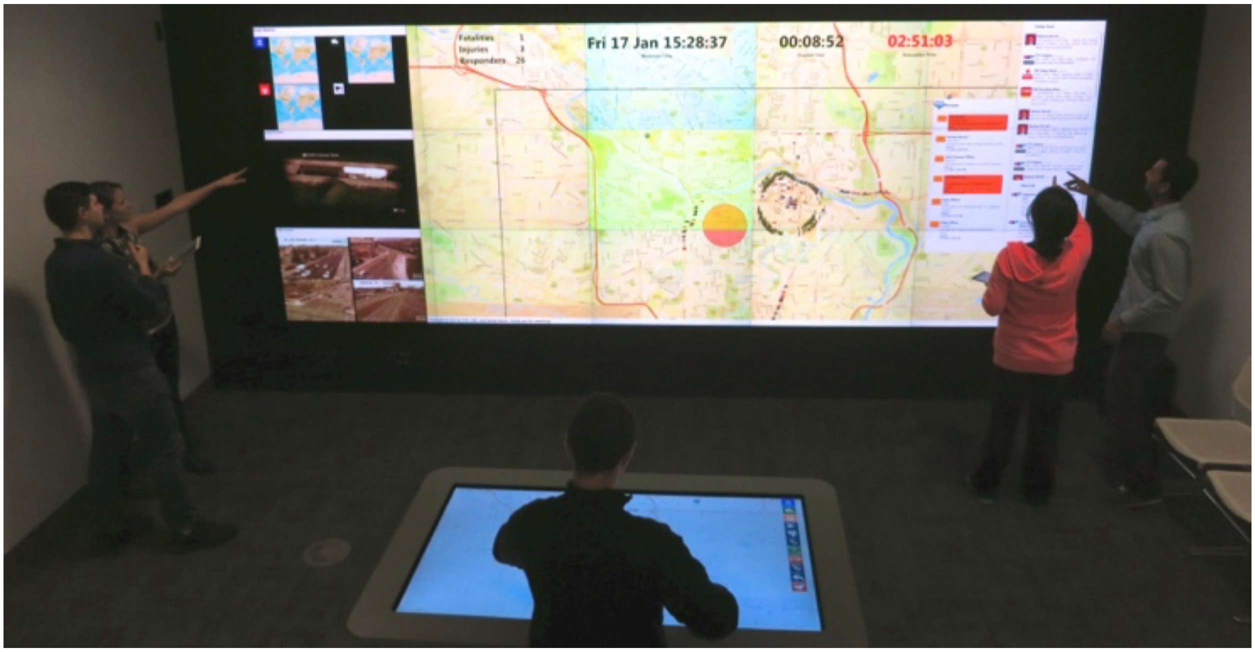


Figure 15: Users collaborating in the ePlan Multi-Surface environment.

First, by providing different areas for information triage (Figure 15), an MSE allows for different information sources to be viewed on different devices. Tablets can be used for personal, private workspaces from which personnel can privately communicate with colleagues.

Tabletops serve as a semi-public collaboration and cooperation area, and large wall displays serve as information radiators that publicly aggregate information from multiple sources.

Secondly, because MSEs are spatially aware (devices and people are tracked), it can support the sharing of information between these public and private devices through spatial interactions. For example, flicking or pouring information can occur such that it is context aware. A flick directed towards a wall without the wall display will not send information whereas a flick towards the wall display will render information on the device.

As Seyed et al. (2013) highlight, "*significant research has been done into different types of interactions, as well as collaboration for multi-display environments; however, very little work has gone into exploring multi-surface environments with real industrial partners.*" As part of the effort to validate these interactions in commercial environments, we explored collaboration and communication for multi-surface environments in the context of emergency response planning, the domain of our industry partner.

In this paper, we present ePlan Multi-Surface, a prototype for an EOC of the future that harnesses a multi-surface environment to enable tablets, tabletops, and a large wall display to interact with one another to create a connected communication environment. We also discuss the space of multi-surface environments and interactions in the context of EOCs and challenges presented during implementation.

4.2 BACKGROUND AND RELATED WORK

4.2.1 *Emergency Response Planning*

Emergency operations centres (EOCs) are central environments that allow for people from multiple organizations to gather for emergency planning exercises, during emergency events, or during the recovery phase after an emergency event. An EOC can be found in both public and private enterprises who need to conduct planning exercises on an on-going basis to ensure preparedness in emergency situations (Perry and Lindell (2003)). In these emergency-planning exercises (Figure 16), multidisciplinary teams of experts collaborate to define how they should prepare or respond to various scenarios (Tena et al. (2013)).

Figure 16: Images from tabletop emergency response planning exercises from (a) FEMA Operations Supervisor in planning session (Image courtesy FEMA/George Armstrong); (b) FEMA cross-border tabletop exercise (Image courtesy FEMA/Eilis Maynard); (c) US Navy command and control center afloat (Image courtesy US Navy/ Bobby Northnagle).

When emergency responders are training for the different scenarios, they conduct one of two general types of exercises: tabletop exercises and live exercises.

4.2.1.1 *Tabletop Exercises*

Tabletop exercises are based on the simulation of a realistic scenario and are either real-time or on an accelerated time. They can be run in a single room, or in a series of linked rooms that could simulate the division between responders who need to communicate and be co-ordinated. People involved in these exercises are expected to know the plan and they are invited to test how the plan works as the scenario unfolds. This type of exercise is particularly useful for validation purposes, particularly for exploring weaknesses in procedures (Gov.UK).

4.2.1.2 *Live Exercises*

Live exercises are a live rehearsal for implementing a plan, and can be particularly useful for testing logistics, communications, and physical capabilities. They are a useful training tool to help build experiential learning by having participants develop confidence in their skills and by providing experience on how it would be to use the plan's procedures in a real event (Gov.UK).

4.2.1.3 *Personnel and Artefacts*

In both real and simulated emergency situations, the EOC provides a key liaison role between municipal officials, external resources and policy makers. To help EOC staff coordinate emergency response with other key stakeholders and personnel, clearly defined principles are used, typically called the Incident Command System (of Edmonton (2019)).

Paper-based maps and documents are commonly used in both exercises (Figure 16) where they are referenced and annotated. Depending upon the situation, responders (e.g. police, EMS, and hazardous materials (HAZMAT)) communicate with people both inside and outside the room. It is vital that people inside the room have the most current information to form a common operating picture (COP) —a "*continuously updated overview of an incident compiled throughout an incident's life cycle from data shared between integrated systems for communication, information management, and intelligence and information sharing.*"[1] A COP helps to support decision-making, and to also to ensure that personnel in the field are working with current information.

1 FEMA- https://www.fema.gov/pdf/emergency/nrf/nrfglossary.pdf

4.2.2 *Tabletops and Surfaces*

Emergency response planning is comprised of many important tasks, from detecting and monitoring the emergency to the deployment of resources and communication management (Bortolaso et al. (2013); Gov.UK). Emergency response planning is also inherently a peripheral process [2]; critical information about an emergency can arrive from numerous sources (e.g. first responders, reporters, or online sources) and information processing and analysis are typically done in parallel with the primary emergency response-planning activity (Bakker et al. (2010)) frequently with interruptions (Edge and Blackwell (2009)).

In the HCI research literature, emergency response planning is a well-explored area, with several different technologies (e.g. tabletops (Bortolaso et al. (2013))) being used to assist in these tasks, as well as information management, collaboration, and efficiency (Hofstra et al. (2008)). However, common rules on interactions to improve collaboration are scarce as the interactions and interfaces are heavily impacted by the domain and the system's purpose. As highlighted by Bortolaso et al. (2013), co-locating people around a device does not mean that the collaboration will be improved: the trade-off between simplicity and functionality must be evaluated multiple times during the system's development.

uEmergency by Qin et al. (2012) is a forest fire simulation system running on a very large-scale interactive tabletop. This tabletop's dimensions (381cm x 203cm) allow several users to collaborate using the system concurrently while considering personal space (local and private workspace) and a global space (shared among all users and synchronized through a button). Users can interact with the system using a digital pen or touch gestures. It is possible to translate and resize a map using gestures with one and two fingers, respectively; to perform annotations dragging and dropping markers from a menu into the map; and changing the simulation's time point through a slider available on each personal workspace. Since all users are sharing on a physically large tabletop, collaboration is improved through visual cues from each user's actions.

Besides digital pen and touch gestures, physical tokens are also used in disaster planning systems on tabletops (Kobayashi et al. (2006)). They act as input, changing simulation parameters according to their physical position above the tabletop, and provide feedback through images projected on them. The manipulation of physical tokens to interact with emergency systems has reduced the learning curve of these systems.

While many of these systems utilize single tabletops or other devices and show a benefit in emergency response situations (Bortolaso et al. (2013)), they do not consider peripheral interaction scenarios where multiple users interact with each other on tablets, tabletops, and wall displays. Concurrently, these users are also analyzing and receiving different sources of information while conducting their emergency response planning exercises.

This provides an opportunity for the exploration in using multi-surface environments that contain multiple heterogeneous devices (e.g. tablets, wall displays, tabletops) and which permit a variety of different tasks and interactions (e.g. "flicking" to different screens) (Seyed et al. (2012)). This may be due to interface design, physical constraints such as orientation or screen size, or device constraints. The research space of multi-surface environments is well explored and significant research has been done in examining different ways in which interactions can take place (Dachselt and Buchholz (2009); Rekimoto (1997); Seyed et al. (2012)).

The collaborative nature of emergency response planning and the presence of multiple and heterogeneous devices in a room provides an opportunity for the study and experimentation of different types of gesture-based and peripheral interactions in the emergency response domain, described in this work by ePlan Multi-Surface.

4.3 EPLAN MULTI-SURFACE

We developed ePlan Multi-Surface, in collaboration with a military and emergency response simulation software company, C4i Consultants Inc.[2] (C4i), located in Calgary, Alberta, Canada. Not only did C4i provide feedback and ideas on features for the prototype, but they also provided the company's desktop simulation engine, ePlan, which we integrated with the prototype.

This section summarizes the design considerations for the prototype and its features.

4.3.1 *Design Considerations*

While working with C4i, we continually discussed and iterated on features that supported three main design elements. These features were inspired by features requested by C4i, or were features that we were looking to test in another domain outside of oil and gas (Seyed et al. (2013)).

4.3.1.1 *Privacy Levels in a Shared Environment*

We endeavoured to create spaces where personnel in the room from the police, EMS, and HAZMAT could make notes and control how and when they shared information with colleagues from the other services in the room. For example, we wanted the support the ability of the police to discuss and share information internally, make necessary revisions, and then share information with EMS and HAZMAT teams. Taken together, the needs of the personnel and the affordances of the devices, we looked to support the user's ability to keep information private on the devices before publicly disclosing it.

2 C4i- http://www.c4ic.com

4.3.1.2 *Information Integration*

There are multiple sources of information coming into an EOC from sources like the news media, Twitter[3], traffic cameras, incident cameras, personnel from the field, and from personnel in the room. In addition, through the simulation software, we had entities (buildings, people, and vehicles) that we were tracking on a large map to ensure we knew what was happening to the common operating picture (COP).

To reduce the information burden on personnel in the room, we sought a way to consolidate information from these sources so that users could focus on decision-making while knowing that the COP was updating in real-time. As we describe later, the large wall display acted as an information radiator where people could turn when seeking an update to the overall situation. This feature is in stark contrast to the tablets which displayed the local situation for each role independently.

4.3.1.3 *Inter-Device Communication*

Earlier we mentioned that communication was an important part of emergency response planning scenarios as there are multiple people in the room representing different organizations. When building this prototype, we were concerned with not only permitting the creation and dissemination of information between people in the same organization, but also people from different organizations.

Through the use of the MSE-API [4], we can support the sharing of information between the devices in the room through gestures (flick and pour) to reduce the users' burden in determining how to share information from their devices. Instead, users could focus on collaborating after they were able to establish a COP.

3 Twitter- http://www.twitter.com
4 MSE-API (or SoD - Toolkit) http://www.sodtoolkit.com

4.3.2 *Usage Scenario*

Throughout our iterative development process with C4i, we also grounded our development efforts with a mock incident created and validated by C4i with emergency response personnel. This incident is a train derailment in downtown Calgary, releasing a hazardous material.

4.3.2.1 *Step 1: Emergency Alert*

Issued At first, the head of the EOC (the 'chief'), the chief of the fire department in the case of the city of Calgary receives information in different mediums (text, email, and phone) from various sources (police, EMS, and HAZMAT) about an emergency event – the train derailment in the city's downtown. The chief then determines the type of emergency that is occurring and issues a local state of emergency alert to the city. While the chief is making his determinations, other EOC personal in the room are often interrupted with new information or are performing tasks simultaneously due to the evolving nature of the emergency.

4.3.2.2 *Step 2: Response Representatives Assemble*

After the alert has been issued, the relevant response personnel assemble in the EOC, and depending upon the severity of the event, these representatives may include members of the fire department, EMS, police, power companies, or the public works departments.

4.3.2.3 *Step 3: Emergency Response Plan Execution*

During the emergency response plan execution step, which lasts until the end of the emergency, numerous types of interactions occur. This session is the most critical component of an emergency response, as significant coordination and planning are done. This is where the chief would get the current status of the train derailment, set the evacuation radius for the spill, set roadblocks. Then, by triaging information from personnel in the room, the chief would share the updated COP with city officials and members of the public.

This step would be where, in the tabletop exercises mentioned earlier, the people involved in the exercise would be expected to know the plan to test how the plan works as the scenario unfolds. In the case of a live exercise, the personnel may test logistics, communications, and physical capabilities of the personnel. Executing the plan numerous times would help build experiential learning by having participants develop confidence in their skills. At the end of this step, a report is typically generated that summarizes the emergency and the contributions of the personnel involved.

4.4 THE PROTOTYPE

To fulfill *privacy levels*, we created applications that catered to the privacy affordances of the three devices. The iPad, as a handheld device, was the most private, the tabletop application, as it has a limited amount of space around its surface and orientation, operated semi-privately where users can only see the screen if they were around it, while the wall display was a fully public space visible to all participants in the room.

We constructed three iPad applications to support roles for the police, EMS, and HAZMAT whereby they could make notes and share information with other personnel in the same role. This enabled the police, for example, to make and share information with other police members, alter those plans, and then finally share the plans with people from EMS, HAZMAT, and the rest of the room.

To support *inter-device communication* (communication between the room's iPads, tabletop, and wall display), we sought an API that would provide tracking of both people and devices. Furthermore, the API had to use low-cost sensors that could cover a sizable room

(Azazi et al. (2013)) so that we could satisfy a requirement from our industry partner when showcasing the prototype to emergency management agencies in the area. We decided to use the MSE-API in its iteration that permitted the integration of multiple Kinects to cover a larger surface area as it provides gesture-initiated inter-device communication to track both people and devices in the room.

We supported *information integration* differently in all three applications:

The *iPads* are used for individual planning activities, and since we worked with both iPads and iPad minis, we built an application suitable for both dimensions [5]. The iPad applications integrated annotations and the entities from the ePlan simulation, and was able to send information to either the tabletop or wall display (Figure 17d).

The *tabletop* is used to integrate the role-specific (police, EMS, HAZMAT) plans into a comprehensive whole. Its information integration was based on its size, orientation, and location at the centre of the room. It was constructed to display ePlan's desktop simulation entities, and support the annotation on all three of the police, EMS, or HAZMAT layer. It was also able to receive information from the iPads, and send information to the wall display (Figure 17c).

The *wall display* is used to share factual information about the situation as well as the agreed upon plan. With both the largest screen and the most public device, the wall display showed different information sources simultaneously (Figure 17b). This figure shows how Twitter, ePlan's desktop simulation entities, live traffic cameras, news feeds, annotations, and messages were integrated. In addition, since it was not touch-enabled, it was a receiver of information from both the tabletop and iPads. As mentioned above, not only did the devices afford different privacy levels, but their screen sizes also afforded different information integration requirements.

5 iPad dimensions —http://www.apple.com/ipad/compare/

4.4.1 *Infrastructure*

As shown in Figure 17, *ePlan* is a multi-surface environment comprised of a number of components: A large, highresolution wall display (Figure 17b); digital tabletop (Figure 17c); multiple Microsoft Kinects[6], multiple iPads; and, a laptop with the ePlan desktop simulation software.

The MSE-API framework provides device location and orientation that results in a spatially aware environment, and it was used to provide inter-device communication and multi-surface interactions with the aid of multiple Microsoft Kinects. Custom applications were created for the iPads, tabletop, and high-resolution wall display.

C4i's ePlan simulation software, which is used during training exercises, stores information about entities (e.g. people, vehicles, and buildings), their location, and their routing information on a backend ArcGIS Server[7], allowing the tabletop, tablet, and wall display to receive the same information as the desktop software updates the simulation. In addition to the simulation information stored on an ArcGIS server, the police, EMS, and HAZMAT planning layers for the iPads, tabletop, and wall display were also stored on separate ESRI's ArcGIS layers.

4.4.1.1 *Tabletop*

We built the tabletop application with the idea that it is a collaborative space where one or more users gather to discuss information before an action is taken and shared with the room. Users can create annotations on one of the police, EMS, or HAZMAT layers before relaying that information to the field. At the same time, the tabletop merges these three layers onto

6 Microsoft Kinect —`http://www.xbox.com/en-CA/Kinect`
7 ArcGIS —`http://www.esri.com/en-us/arcgis/about-arcgis/overview`

one map so that the chief can see a future COP before sharing that information with the entire room through a touch-enabled gesture to send the information to the wall display.

4.4.1.2 *iPad*

The iPad applications were created to support three different roles – police, EMS, and HAZMAT. These three roles are operated by independent chains-of-command, and thus we created three independent layers on which annotations are created to support the collaboration amongst personnel from that organization. By separating these tools in role-specific user interfaces, we expect to reduce the cognitive burden for the people using the iPads so that they can focus on their own job and are not be burdened with filtering away UI options that are not relevant for them. Reducing the tool set can be important to reduce errors in high-stress situations (like emergencies). All three application provide some common functionality that permits the user to change the emergency response planning scenario update frequency, annotate shapes and lines, as well as toggling between a street or map view. Each iPad application operates on an independent, private ArcGIS layer so that police annotations and entities are not merged with either the HAZMAT or EMS layers.

4.4.1.3 *Wall Display*

This application consolidates information from eight different sources onto a 9600 x 3600 high-resolution wall display as seen in Figure 17b:

1. Shows the areas under review by the three iPad application;
2. Represents a camera showing a live-feed from the incident zone;
3. Represents the area where traffic cameras are live-streaming into the EOC;
4. A ticker showing news headlines from the area;

5. The map overview showing the entities (people, buildings, vehicles, etc.) from C4i's software along with annotations and other information shared from the tabletop and/or iPad applications;

6. Lists the messages that have been received by the EOC;

7. Live Twitter feed from people or organizations are being followed by the software; and,

8. More detailed information about news items that are scrolling through in (4).

Overall, the wall display is used as an information radiator to share factual information about the situation as well as an agreed upon plan.

4.4.2 *Multi-Surface Interactions*

Earlier we described the general steps that people working in an EOC might take when executing the train derailment emergency response plan. Next, we highlight interactions in our multi-surface prototype for that same usage scenario:

4.4.2.1 *Step 1: Emergency Alert Issued*

In the first stage, the chief is stationed by the tabletop and receives updates from personnel in the room through a one-finger flick (or pour gesture) that transfer information from their iPads to the tabletop where the chief can triage the information. Simultaneously, the wall display updates with information from local traffic cameras, news feeds, and Twitter.

4.4.2.2 *Step 2: Response Representatives Assemble*

Personnel from the police, EMS, and HAZMAT gather in the room with private iPads containing information about their people and vehicles in the field, possible routing information for ground personnel, and preliminary plans for their people. These EOC representatives may then share relevant information using flick or pour gestures at their discretion or use the information from other representatives in their own assessment for allocating their resources during the emergency.

4.4.2.3 *Step 3: Emergency Response Plan Execution*

In ePlan multi-surface, emergency response personnel are collaborating and consuming new information rapidly using iPads, while also simultaneously trying to keep track and manage the emergency through the wall display and digital tabletop. Representatives can share information from their iPads to the tabletop – where it can be triaged with the chief and other representatives – using a one-finger flick (or pour gesture).

If the same person wanted to share the information with the entire room, they could use a two-finger flick to share that piece of data with the large wall display. The wall display allows everyone in the room to see the information; the tabletop is used to assist in collaborative emergency response planning; or other iPads are used to facilitate communication between different representatives. While personnel are sharing information with others via multi-surface interactions, information from other sources (news feed, traffic cameras, and Twitter) are updating the wall display to ground on-the-floor discussions.

4.5 PROTOTYPE CRITIQUE AND DISCUSSION

Getting feedback from users in the emergency management planning domain is extremely critical. As we worked closely with domain experts from C4i Consultants Inc., we asked them to continually provide feedback through the various stages of our collaboration, including the prototype that is presented in this work. The goal of this feedback is to discuss the potential of multi-surface environments and applications to their domain as well as to brainstorm future research. The feedback received is presented below in general themes.

4.5.1 *Interactions and Gestures*

As with other applications that have been developed by our group in the oil and gas domain (Seyed et al. (2013)), we started from the gestures and interactions defined in prior work (Seyed et al. (2015)). Our group of users from C4i, however, gave mixed reviews to the gestures —point and flick, pour, and pull —that we implemented in the prototype. Point and flick along with pull gestures received positive reviews as "*natural gestures*", however the pour gesture whereby the user takes the iPad and rotates the screen so that the iPad screen faces the tabletop was thought to be cumbersome and "*not natural*".

Our users repeatedly referenced "*natural gestures*", focusing on the ease of use and learnability of the interactions. During emergency response, there are users who regularly work in the EOC, and there are others who will come to work there during the emergency (of Edmonton (2019)). As described during one of the interviews "*On a typical day, the EOC houses 25 staff. At the height of the recent Calgary flooding, Burrell (city of Calgary fire chief) estimated nearly 200 people were working in and around the ops room." In the case of the city of Calgary, irregular EOC personnel come from "the city's business units, along with agencies like Enmax (the local power company), ATCO (a local natural gas provider), Alberta Health Services, and industry groups*" (Herald). Our interviewees also mentioned that the gestures needed to be easy enough to learn simply by a user "*peeking over a colleague's shoulder*" and that during these highly-intensive events, users may not be able to remember the difference between a "*three-finger and four-finger swipe.*"

During our interviews, our C4i collaborators also repeatedly acted out gestures that incorporated both the surface of the tabletop and iPad, followed by an in-air movement. This combination of on-device gesture plus in-air gesture felt more natural when they wanted to share content from their device to other devices in the room.

4.5.2 *Information Overload*

In the city of Calgary, our collaborators noted that police collect video from cameras placed both in the police cruiser along with cameras on the officer's uniform (SUN et al. (2013); Aug 19 et al. (2013); Shingler (2014)), and with this, and social media sources of information (e.g. Twitter, Facebook, Instagram, Pinterest, Tumblr) that were not available a decade ago, the amount of information could become overwhelming. With this quantity of data, C4i mentioned that maintaining situational awareness is critical in environments like EOCs. Furthermore, they mentioned the need, with the quantity of information at their disposal to "*see [both] the forest and the trees.*"

Our interviewees also described how multi-surface environments and interactions could aid in information sharing where a user could swipe a picture, text, or video from their iPad or tabletop and have that information shared with a wider audience, especially when they compared this to the current situation where users "*go back to my desk to email that information to you.*" One subject remarked that the ease with which information sharing can take place using gestures should also reduce both the number of silos and the amount of information contained in those silos.

4.5.3 *Technology Reliability*

Throughout our interview sessions, a common concern was the Kinect tracking technology. When working, the three Kinects allowed for the coverage of the room, however, the tracking system would occasionally lose track of users and devices.

Our collaborators at C4i commented that a better system would allow for an AirDrop [8] -like or an "*AirDrop on steroids*" where the user can focus less on whether they are connected, and more on sharing content. Just like cellphones just work (with roaming charges) when people travel internationally, the tracking technology should be pervasive and reliable. They saw no issues in the possible privacy concerns with their suggestion to use on-board cameras on tablets, phones and computers to help track people and devices in the room.

4.5.4 *Areas to explore*

During our on-going collaboration with C4i, we kept track of items that we were able to implement for our current prototype, and those which could be explored during our continued collaboration. With that in mind, we generalized their feedback into four further themes:

4.5.4.1 *Rate of Data Growth*

We are seeing a tremendous growth in the amount of information available to users (Silver (2012)) and this growth is not likely to stop with the increased pervasiveness of cheap sensors that can provide directions to people while navigating New York City (Howarth). Our collaborators reaffirm this notion, seeing the increased use of metadata in military applications, and are sure that its use will follow into the civilian realm whereby emergency responders

8 AirDrop —http://support.apple.com/en-ca/HT203106

would be able to get historical and real-time information on cities, neighbourhoods, and buildings.

This additional information builds a more complete picture of the environment in which personnel in the field are entering. Furthermore, the people in the EOC will be able to simultaneously live-stream cameras from officers (Aug 19 et al. (2013)) and data on their location, foot speed, wind conditions, and fatigue level (Howarth). C4i indicated that our prototype would need to incorporate this information into information radiators or other applications without increasing the cognitive load on EOC operators.

4.5.4.2 *Using Multiple Sensors*

In this prototype, we explored a small set of surface-related gestures (swipe, pull, pour, and flick), however our collaborators see the opportunity to augment the prototype with voice-based commands and haptic feedback. One interviewer commented how Siri[9] can function in busy, noisy locations, and thus sees an opportunity to add voice-base commands to the gestures. Moreover, currently when a user attempts to send information to the wall display or tabletop, they get feedback on the screen indicating whether it succeeded or failed. Alternatively, C4i suggested that we provide haptic feedback indicating the success or failure of these gestures so that the user need not look at their screen as they move about.

4.5.4.3 *Security*

As with any technological system, security cannot be an afterthought, however, it was something that we did not explore due to time and complexity. C4i indicated that should we explore security, we must examine *information security*, *device security*, and *personal security*. C4i considered information security to be how we secure the information (picture,

9 Siri —http://www.apple.com/ca/siri/

video, or text); device security was related to whether a phone, tablet, or computer was secure enough to be authorized to be on the network); and, personal security was about whether the person should be in the room. Each of these areas would present an interesting path to explore in future work, particularly in the context of multisurface environments.

4.5.4.4 *Newness of technology*

A final comment made during our interview related to the new EOC that was opened by the Calgary Emergency Management Agency (CEMA) in 2012 (Herald). "*Though the city of Calgary's EOC was recently opened with new technology, the moment it opened the technology was actually old*". These are critical, highly stressful environments that require that systems work —failure is not an option. The technologies that we developed (and used) during the prototype do not fit the reliability criteria of an EOC yet, however it can be used to draw requirements and commentary from EOC users who are working with older technology.

4.6 CONCLUSION

In this paper, we presented our prototype, *ePlan MultiSurface*, a multi-surface environment for emergency response planning exercises that was designed with domain experts from C4i Consultants Inc. We discussed challenges such as collaboration, interaction and information dissemination in multi-surface environments for EOCs. We also presented a preliminary prototype critique and discussion about its utility for the domain.

Our future work involves extending the prototype and continuing with a more complete evaluation of the system and its impact for emergency response planning exercises. We believe that the emergency response domain is an ideal candidate for further exploration of multi-surface interactions and technologies in real-life applications as this domain stresses systems differently than applications in other domains.

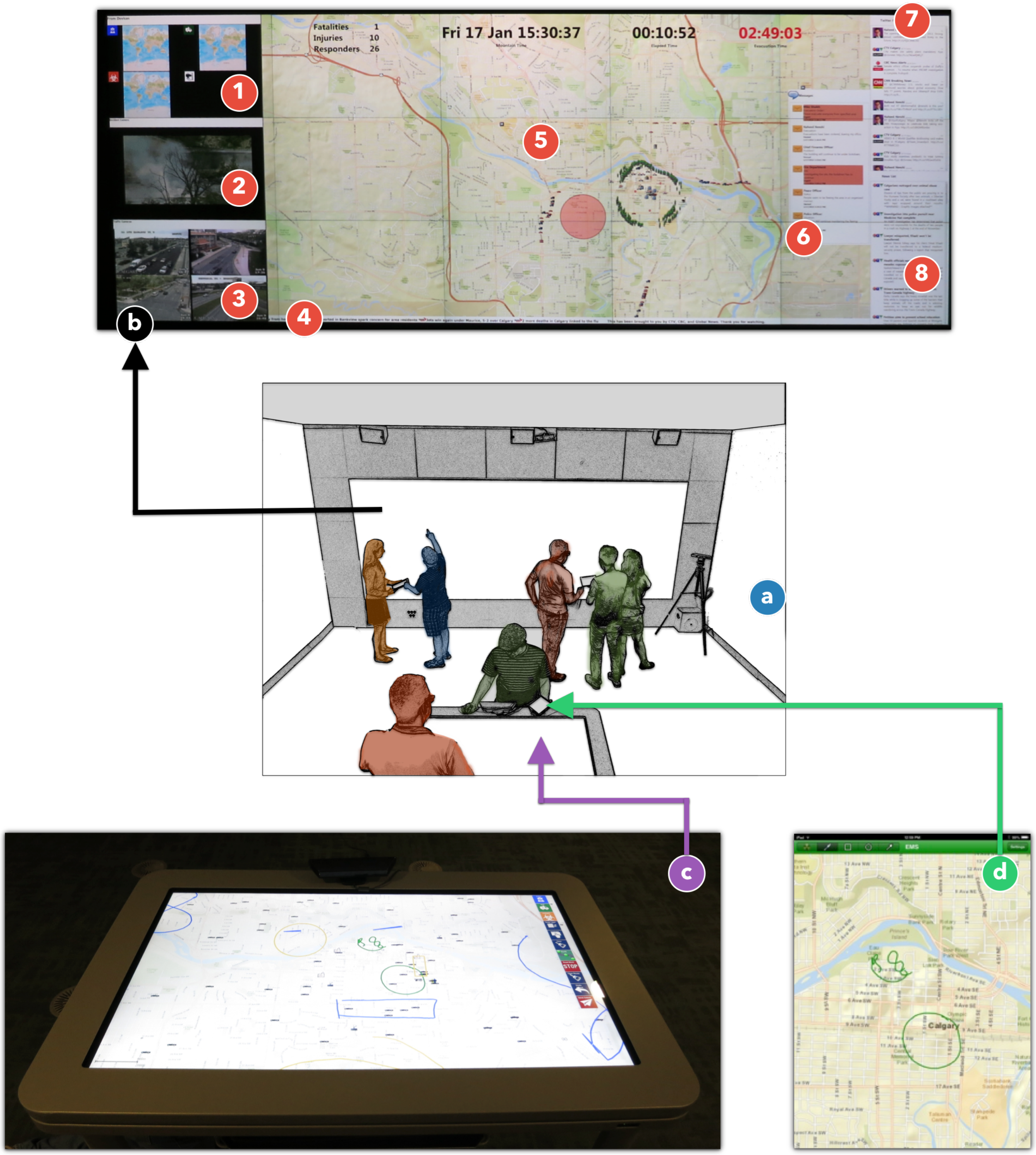


Figure 17: An overview of ePlan MultiSurface. (a) Highlighting the ePlan Multi-Surface environment, with different roles collaborating in an emergency scenario (green represents EMS, red represents fire, blue represents police and orange represents HAZMAT). (b) The wall display application and it's different components. (c) The tabletop application. (d) An iPad running in the EMS mode.

5

BEYOND SKYHUNTER: REFLECTING ON COMMERCIALIZATION

"If you don't know your full-throttle history, the whole story of how you came to where you are, it's kind of hard to put things together."

Nipsey Hussle

My entrepreneurial journey began at a small startup in Calgary, Alberta, Canada, whose aim was to commercialize academic research. One of the first projects it attempted to commercialize, was the work from Chapter 3. My role in this startup was to assist and lead in its commercialization efforts, primarily because I was one of the Project Leads when it first started in a research lab. In the research lab, the project was called 'SkyHunter', but at the startup, it became known as Spatial Interactive Tabletop Software (SITS)[1].

5.1 PRODUCT OVERVIEW

One of the key research lessons from SkyHunter (Chapter 3) and ePlan (Chapter 4), was that there was value in building a product focused on collaboration with geospatial data (i.e.

1 **Note: The original product name has been changed for the sake of anonymity.**

maps, map layers, etc) and to enable people to do this with different types of devices, which included tablets, smart phones, digital tabletops and wall displays.

SITS built upon geospatial collaboration from the prior research by making a digital tabletop (i.e. The Microsoft Surface[2]) a key requirement, and it also added additional features such as the ability to integrate data from other mapping or GIS sources (e.g. ESRI's ArcGIS[3]), multi-platform support (iOS, Android, Windows). The SITS tabletop app served as the main hub of collaboration with data, as tabletops had already been shown to be effective tools for collaboration in prior research (Scott et al. (2003)).

5.2 ROLES AND RESPONSIBILITIES

As mentioned earlier, my role in the startup was to lead the SITS product commercially. This meant wearing multiple hats — as one frequently does in a startup — and they were the following:

1. Project Manager
2. Product Manager
3. Senior User Interface (UI) Designer
4. Senior User Experience (UX) Designer

This may read as though it is several roles on paper, but pragmatically it was already what I was doing in a research lab, but it was now in the context of a startup. It is in

2 Microsoft Surface - http://www.samsung.com/us/business/support/owners/product/40-samsung-sur40-for-microsoft-surface-sur40/
3 ESRI ArcGIS - https://www.esri.com/en-us/arcgis/about-arcgis/overview

this transition where I began to learn *what not to do when trying to build a product as an entrepreneur*.

5.3 ENTREPRENEURIAL LESSONS LEARNED

In the following sections, I begin to reflect on and describe the first set of entrepreneurial lessons I learned through SITS, which I have placed into general categories. I would like to note (again), that these are specific to my experiences and observations, as are those which follow in the remainder of this dissertation.

5.3.1 *Lesson 1: Evolving Research Into Product*

One of the first lessons I learned when transitioning from a prototype in a research lab, to creating a product, was that the goals of each are fundamentally different. A research prototype is a proof of concept, meant to convey an idea or to explore something novel or interesting. But what is novel or "cool" on the research side, is sometimes not practical, or even necessary, for a commercial product.

As a product, SITS initially applied many of the research lessons from Part i, specifically how to use multiple devices to collaborate with geospatial data (i.e. maps and layers). This was reflected in what SITS ultimately became, an iOS (or Windows) app which would remote into an app running on a digital tabletop. Gone were the "cool" interactions such as *flick* , *pour* or camera from the research side (Chapter 3), and the Microsoft Kinect or other sensors. Gone was the complex setup for the environment, and in its place was a single interface to connect and add geospatial information from ESRI ArcGIS databases.

For the most part, SITS was driven by customer requirements (which I will touch on a bit more later), which was another key reason why it was substantially simplified from the SkyHunter version. As a Product Manager, I also learned that cost-wise, a research solution wasn't always going to be the best solution, as having a product rely on multiple types of hardware could not scale effectively — which was one of the goals of the *SoD Toolkit* (Chapter 2). The key lesson here, was that things that work great and are impactful on the research side, don't necessarily translate well, both economically and from a practical standpoint for products. What is more important is identifying what is research and what is product, and adjusting accordingly.

5.3.2 *Lesson 2: The Challenge with Commercialization*

Commercialization is a word I now consider misleading, especially in the context of research. As I mentioned in *Lesson 1: Evolving Research Into Product (5.3.1)*, it is critical to understand what is research and what is product.

One of the biggest benefits of commercialization, is the grants available which fall under this category. Commercialization grants from both the local, provincial and national government are critical in funding the transition from a research project to product, especially for SITS. It also led to additional developments in features, enabled the startup to hire additional resources, purchase needed equipment and build more of an organizational structure (more on that later).

But there is something to be said about *less is more* in the context of startups, particularly as it related to SITS. To frame this point, I will introduce and utilize the "Ice Cream Parlour" analogy below:

Let's say you're thinking about starting a brand new ice cream shop (for the sake of this analogy, let's call it Hefe's Ice Cream Parlour), whose key differentiator from other ice cream shops is to sell ice cream with flavours that nobody has ever seen or thought of before. This shop is going to be your first attempt in selling ice cream flavours to the public and to the world. When you first want to open your shop, you're given grant to commercialize some of the flavours you've worked on in your own kitchen for a while, hire people to explore and make new flavours for you and you even rent a moderately sized retail space for your shop. After some time however, things are not going great with customers, as a few customers want to taste but nobody is really buying your ice cream like you thought they would. Despite this, you're still doing OK because the commercialization money you're continually applying for and receiving is helping to expand your shop from ice cream to other types of pastries and snacks (which you think will help the overall business), expand your retail shop space, and even hire a marketing and sales team to help promote the shop itself. But still, nobody comes to buy your ice cream, and it makes you wonder if all the funding worth it in the end, and if anyone truly saw value in your ice cream to begin with?

In the analogy above, there are several fundamental questions that arise: Was the ice cream shop really helped or hindered by having so many grants? What would have happened if it had less grants, and instead focused on obtaining customers? What if it didn't add more ice cream flavours, and other snacks, and just focused on a few flavours of ice cream? What if it stopped selling ice cream and closed down?

While there are several different answers to the questions above, I believe the key question can be fundamentally reduced to: *Is less more for startups?* My personal answer to this was learned through SITS. For SITS, much like the ice cream shop, the grants were beneficial in that it allowed it to survive and continue, but also like the Ice Cream Parlour, it meant doing several different things. For SITS, several different ideas around geospatial collaboration were implemented or explored to keep it funded, as funding was conditional to continued work in the product, such as:

1. Becoming a geospatial data explorer.
2. Researching how to become a remote monitoring and control system for drones and other types of autonomous vehicles.
3. Exploring how the tabletop in SITS can be used for public engagement, and urban planning in cities.
4. Becoming a tool for forestry planning.

Notice that these ideas are wildly different than what SITS came from — the oil and gas domain and its collaboration with geospatial data. Some of these ideas are likely even valid, but by attempting to explore and build all of these different things, SITS became ineffective at all of them simultaneously. Given that SITS primarily relied on a tabletop (at the time), this was a failure, but a proper exploration could have occurred with additional commercialization funds. One of the main reasons for much of this exploration and continued grants for commercialization was because SITS was not successful in attracting customers (more on that later). This meant it was in a perpetual state of receiving funding successfully, but pragmatically going nowhere as a product.

Ultimately, it was in SITS I learned that *relying solely on commercialization grants and having too much of it)* means you are not truly running a startup, but instead (likely) propping up an idea when the market and customers have already spoken for you (or haven't). The less a start-up relies on commercialization grants (or transitions away from it), the more opportunity it has to either grow or die quickly (if needed).

5.3.3 *Lesson 3: Building an Organization Quickly vs. Effectively*

In order to create a great product in a startup, having a proper team organization and structure is critical. In the research labs I was a part of, team dynamic and organization was always very informal. For the most part, the more senior you were, the more research experience you had, making it easier to teach those with lesser experience. One of the things I have truly enjoyed about research labs to date, and likely one of the best aspects of doing a PhD, is the informal and organic nature of the environment which encourages teaching, learning, collaboration and mentorship, among several other things. However, it is challenging to maintain the spirit of the environment which encouraged the research idea when transitioning into a startup environment that is aiming to commercialize the idea.

For SITS, the organization (and environment) around it was built very quickly due to the numerous commercialization grants it received. In a very short time frame, there was a team of developers, multiple managers, an advisory board, and other typical things you would expect a product in a startup to have. The challenge became however, was the organization around SITS itself was built so rapidly, that it did not realize it was being ineffective, which also led to some of the challenges previously mentioned in *Lesson 2: The Challenge with Commercialization (5.3.2)*. However, let's return to the ”Ice Cream Parlour” analogy to frame the lesson in this section further:

You've just been hired to work at Hefe's Ice Cream Parlour. You have limited experience in making ice cream flavours, but you have also never sold ice cream in a shop either. When you first start, you have a small team, who are fairly junior and have some experience making ice cream, but like you, have also never sold ice cream. However, in a very short order, Hefe's Ice Cream Parlour receives an influx of grants and there is an expansion in your team. It then *quickly hires additional management and other resources to guide how the ice cream is being made, and who it should be made for. Some of the people hired for guidance or management are well liked by your team, but curiously these hires have no experience selling ice cream, but instead have experience making and selling other things like toys. Over time, as you and your team become more and more experienced in making the ice cream, there is a realization that making and selling toys is not the same thing as making and selling ice cream, and conveying this to the owners is difficult. Ultimately this affects the environment in which the ice cream is being made.*

As mentioned earlier, the team and organization around SITS was built quickly. Much like the Ice Cream Parlour, there was a hiring of relatively junior people (including myself) and others who had limited to no experience with building software properly (I discuss this in more detail in the next section), but instead had plenty of experience on the business side. While this was very useful when SITS first started, it later became difficult to focus SITS on a specific domain or area because there was so much input and opinions from several different sources within the startup. And unlike a research environment, where much of this was collaborative, it could not be for a startup because there needs to be a formal structure in place to guide a product. Therein lies the challenge, how does one build an organization with the spirit of a research lab, but the structure of a company, or even balance between them? Does one even need such a spirit in a startup?

While I do not (yet) have a good answer for this, the lesson I learned with SITS, is build a small and focused team that has the relevant experience first, then focus on building the product. In that order. As Horowitz (2014) stated: "*...sometimes an organization doesn't need a solution; it just needs clarity.*"

5.3.4 *Lesson 4: Research Experience vs. Industry Experience*

Donald Rumsfeld once famously said:

"*...there are **known knowns**; there are things we know we know. We also know there are **known unknowns**; that is to say we know there are some things we do not know. But there are also **unknown unknowns** – the ones we don't know we don't know.*"

This quote perfectly encapsulates what a graduate student can truly know and learn about industry without having been in it before, which in my case, had not happened yet. As a student, you may be extremely experienced in developing research prototypes **(the known knowns)**, but you will have limited to no experience in building a proper piece of software, even when accounting for internships. Pragmatically, this meant that I knew there were things I hadn't needed to encounter yet, such as properly deploying a system in a production environment, architecting a scalable system, or designing a proper data security scheme (**the known unknowns**).

In the beginning, SITS effectively had none of what was necessary for it to be a proper software product, and it was a more technically refined version of the SkyHunter prototype built the research lab. This was because much of the team was junior and didn't have the proper experience necessary in building a product (myself included). The first version was very simple, and only worked with local geospatial data, and did not have authentication schemes because there was simply no expertise available to build it properly, let alone supervise it.

As the Project Manager for SITS, this was frustrating because I knew that we could not really turn it into a product without the proper technical assistance, which eventually was hired. But this emphasized one important aspect for product development: the importance of a strong technical team. A strong product development team needs to have a mix of interns, new grads and experienced technical people who are able to both supervise and mentor. This was one of the critical failings of the early days of SITS.

Over time, as we added additional resources (e.g. an experienced software architect) and a far more experienced Project Manager, SITS did improve, and using the commercialization grants, we began to encounter the **unknown unknowns**. For example, one of the previously mentioned ideas explored with the grant money for SITS was remote monitoring and control system for drones. Practically, it seemed like a good idea until we found ourselves in the middle of a remote field doing a product demo. We took a large touch screen with us to this remote location, where conditions were cold enough a tent was required. Even in this tent, the conditions prevented the touch screen from working correctly, but what was more important was what we learned about the product idea itself. We learned that there were a number of regulatory issues around line of sight for the drone, which is important when attempting to operating it remotely. If the app relies on a large touchscreen, which can't function in cold-weather conditions and only functions inside a tent with limited line of sight, then it is simply not feasible. None of this we knew beforehand (the **unknown unknowns**).

As I reflect back now, I realize that I didn't really know very much when I started, but it was that realization that led me down the path of entrepreneurship, in an attempt to address many of the things I knew I didn't know. As a student, it is extremely critical to acknowledge the limitations of your knowledge, and then address the shortcomings.

For me, this is still ongoing, but brings me to a larger point: If you are running a startup yourself and you are hiring students, you also need to realize there is a limitation in what students know and don't know, and have a proper development and technical leadership structure in place to build a proper product. You simply cannot rely on newly minted students, or inexperienced ones, to build a functioning product, as they don't know how — at least not yet.

5.3.5 *Lesson 5: Product Management, Do You Have It?*

Early in my PhD, I was fortunate enough to take MBA courses at the University of Calgary. This was part of a pilot program by the University and the newly formed Hunter Hub that was designed to introduce non-MBA students to MBA-level entrepreneurial courses. I used this opportunity to explore and learn what the business side of entrepreneurship entailed. What I learned, at a very high level, was the following:

1. HCI research and entrepreneurship are very similar, in that both require some level of *research* (either for an idea or for a product), some level of *validation* (either a research study or some form of market validation) and ultimately some *outcome* (a prototype with a research paper or an alpha version of something alongside a pitch).

2. Business plans and market research was not an area I had been exposed to until the MBA courses, and they were very vital in getting a broader picture of the business side.

3. Combining business skills with technical skills is going to be the way forward, with or without a PhD or an MBA.

4. I had a lot to learn about the business side.

Pragmatically however, I took these courses so I could better understand how to contribute to the SITS product from the business side, rather than just the research and technical perspectives as I was doing in research labs already. At the time, I felt taking MBA courses would help, and they did, especially when it came to market research.

You are probably wondering, why is all of the above information relevant? Well, a product (and by extension, a startup) cannot succeed if it ignores the feedback it receives, what market research is showing, or what its competitors are doing. This was one of the critical mistakes with SITS.

The MBA courses were excellent in understanding how to do product research, and I quickly learned (along with others) that there was lots of competition, and maybe that the product itself wasn't feasible. Bringing some of this research into SITS, it led me to strongly consider how to approach product management, and being to attempt to answer several difficult questions about the product. While I could provide an extensive list of these questions, the following were key for me as the Product Manager of SITS:

1. What is your product?
2. What is the pricing model (or business model) of the product ?
3. What are your customers (really) saying about your product?

It was in attempting to answer these questions, that I realized the challenges SITS faced from a Product Management perspective. The inability (and in some cases, unwillingness) to answer them was a byproduct of some of the lessons I described earlier, but also due in part to several common pitfalls that startups face. Below, I address some of these pitfalls through the lens of the prior questions.

1. What is your product?

   In the Product Overview section, I provided my perspective on what SITS was. One of the main challenges with SITS however, was that there was no unified vision as to what the product was or could even accomplish. Some of this was previously mentioned in *Lesson 2: The Challenge with Commercialization (5.3.2)*, where I described a number of ideas being explored as to what it could accomplish. Over time, I learned that when answering this question, a startup needs to (continually) ensure that everyone is clear what the product is. And this definition should be clarified and refined through market research, regardless of its outcome.

2. What is the pricing model (or business model) of the product?

   Until I had taken the MBA courses, I was not entirely familiar with pricing or business models of products, let alone those that involved software. Additionally, one of the challenges with translating research into a product for a startup, is that it is difficult to determine if there is even fit in the market for the product itself (also known as *product market fit*). As SITS was often in a prototype stage, it was difficult to determine this.

   One other area important aspect in answering this question, is to examine what other competitors are doing for their business, and determine what is best for your product. With SITS, it was difficult to determine a proper pricing model given the lack of focus on what the product should be, and what market it was for. This was further magnified with the lack of immediate need to create one when the product was thriving on commercialization grant money.

3. What are your customers (really) saying about your product?

   Getting feedback (generally) on an idea is easy. The challenge is getting critical feedback that sometimes things you need to hear but don't want to hear.

   Getting feedback on an idea is generally easy. You can always ask your friends, family, co-workers and others. What is not easy, is getting critical and constructive feedback. Part of this is because sometimes it is difficult for others to tell you what you need to hear (e.g. an idea is bad, isn't feasible, or doesn't make sense) and instead they may choose to tell you what you want to hear (e.g. it's a great idea, keep working at it, etc.). The best feedback, and the feedback you should seek, is the critical type.

As I came from the research side, getting this type of critical feedback was common, and something I expected to see on the entrepreneurial side, but this was not always the case. And what was even more difficult, is how feedback is utilized in the context of a startup.

Answering this question for SITS was a challenging experience, because it meant there needed to be some critical thought put into the product itself and even some of the research I had done behind it. For example, we received a lot of really positive feedback about the tabletop experience, which was often mistake as positive feedback for the product. This type of feedback is excellent within a research context, but not necessarily for a product. This is because we needed feedback about the product, and not the hardware it was running on (i.e. the digital tabletop). Whether it is a digital tabletop, augmented reality, robotics or any other up and coming technology, your product is not the hardware (or technology) that it runs on, but rather what it does for customers.

SITS also received critical (and sometimes negative) feedback around whether it was actually better than a simple desktop running ESRI's ArcGIS software, the most common mapping platform at the time. This critical feedback was often difficult to process for the startup, and as a result, it was taken selectively rather than as a whole. From a pragmatic perspective, this is an understandable reaction, as the startup had invested significant time (and commercialization grants) in building SITS, and hearing negative feedback is difficult. But it is learning from this type of feedback that a startup can begin to understand what your customers are actually saying about your product.

Product Management is by no means a trivial job, and through SITS I learned how difficult it could be, and the challenges that need to overcome both externally and internally. I initially thought it would be similar to the processes I was familiar with in research labs, and research in general. But in a research lab, it is (generally) good to be optimistic, but in the product and startup side, I learned it was critically important to be realistic instead.

5.3.6 *Lesson 6: No Sales, No Marketing, No Future*

Marketing is a critical task for a startup, which ideally leads to sales and a (hopefully) sustainable future for a product. If the marketing is non-existent or insufficient, then ultimately the product will not be able to succeed. In *Entrepreneurial Lesson 3: Building an Organization Quickly vs. Effectively*, I described that one of the biggest challenges for SITS was that the startup around it was built too quickly. This also translated into how marketing and sales were done for the product.

Much like earlier, I will use the 'Ice Cream Parlour' analogy to illustrate the point of this section further:

Hefe's Ice Cream Parlour decides to focus on selling Strawberry ice cream for the next 6 months. You and your team begin creating several different types of flavours, all of which are being used by the newly hired Marketing and Sales team, who are well experienced in selling Strawberry ice cream. After several months, the Marketing and Sales team has not had significant success in attracting or selling customers the Strawberry flavours of ice cream, and they ask for Blueberry, Mango, Pineapple and other flavours of ice cream, with the rationale that having more flavours of ice cream is necessary to market and sell. Over time, you realize that the model of how the non-Strawberry ice cream is being sold isn't well understood by the

Marketing and Sales team, nor the startup overall.

From the start, SITS was focused on the oil and gas domain, which was the area initially targeted for marketing and sales, and thus what the startup was built around. But as described earlier, the multiple pivots meant that marketing and sales couldn't be done in an effective manner because the marketing and sales for each pivot needed to be different. Additionally, there was a lack of experience within the startup in many of the aforementioned areas. The most common mistake made in this context, was assuming we (that is the startup) were the customer or knew enough about these areas to make appropriate and informed decisions for SITS.

Certainly, there is *no silver-bullet solution for anything*, let alone for collaborating with geospatial data. But attempting to market SITS in this manner, meant that sales was difficult and it ultimately led to a failed product. As I describe in Lesson 3: Building an Organization Quickly vs. Effectively, a small and focused team is critical. By extension, the same is for the Sales and Marketing team. Ultimately, if there are no sales, a product in any startup has no future, and thus it is important to hire the appropriate team with as relevant experience as possible.

5.4 FINAL REFLECTIONS

Transitioning from a research mindset into a startup and nurturing the entrepreneurial mindset through SITS was a necessary experience. As I have demonstrated in some of the lessons already (some of which may seem obvious in hindsight), they needed to be learned and experienced in order to move forward. As a student, it was one thing to learn some of the material in my MBA and other academic courses (e.g. design or marketing), but an entirely different matter to be able to apply the material effectively in a startup. Fundamentally, it was the lessons in SITS and its startup that shaped much of my critical thinking, some of which I later applied in my own entrepreneurial ventures.

Part II

NOVEL DEVICES AND INTERACTIONS

In the second part of this dissertation, I describe work exploring new types of interactions and devices, as well as the beginning of my own explorations into starting new entrepreneurial ventures and attempting to commercialize my own research project. First, in Chapter 6, I describe a novel re-configurable dual-display smart-watch that uses tangible techniques for interactions. This is followed up in Chapter 7, where I present a concept modular smart-phone designed for lending. Finally, in Chapter 8 I reflect upon a second journey of commercialization (this time as a co-founder) with the research work from Chapter 6.

6

DOPPIO: A RECONFIGURABLE DUAL-FACE SMARTWATCH FOR TANGIBLE INTERACTION

Doppio is a reconfigurable smartwatch with two touch sensitive display faces. The orientation of the top relative to the base and how the top is attached to the base, creates a very large interaction space. We define and enumerate possible configurations, transitions, and manipulations in this space. Using a passive prototype, we conduct an exploratory study to probe how people might use this style of smartwatch interaction. With an instrumented prototype, we conduct a controlled experiment to evaluate the transition times between configurations and subjective preferences. We use the combined results of these two studies to generate a set of characteristics and design considerations for applying this interaction space to smartwatch applications. These considerations are illustrated with a proof-of-concept hardware prototype demonstrating how Doppio interactions can be used for notifications, private viewing, task switching, temporary information access, application launching, application modes, input, and sharing the top.

***Note: The text in this chapter appears in the following publication:**
Teddy Seyed, Xing-Dong Yang, and Daniel Vogel. 2016. Doppio: A Reconfigurable Dual-Face Smartwatch for Tangible Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16).* ACM, New York, NY, USA, 4675-4686.

6.1 INTRODUCTION

Smartwatches may be a convenient way to access information, but interaction is challenging due to the small form factor of the screen area. A small display size means only a limited amount of information can be shown. Proposed solutions to this problem include switching to a smartphone or tablet for some usage scenarios (Chen et al. (2014b)), extending the display by projecting onto the forearm (Olberding et al. (2013)), or placing touch displays around the entire watch band (Lyons et al. (2012)). Additionally, since most watches use direct touch on the display, the small display combined with fat fingers is problematic (Siek et al. (2005)). Using off-screen touch input (Harrison and Hudson (2009); Laput et al. (2014); Perrault et al. (2013)) or physical knobs and buttons, solves occlusion but limits the input space, while using voice input is conspicuous and slow (Van Buskirk and LaLomia (1995)). Researchers have responded with more expressive sensing to enable gestural interaction around the watch (Pasquero et al. (2011); Rekimoto (2001)) or by manipulating the entire watch face, similar to a joystick (Xiao et al. (2014)).

We significantly extend the idea of manipulating a watch as input by adding a second face that can be attached, oriented, and manipulated around a base watch face. This not only doubles the display area and can keep fingers off the display, but also creates a tangible input language with manipulations like stacking, peeking, hinging, adjacency, distance, and indirect input (see examples in Figure 18). We call this concept "Doppio", meaning double. It can be thought of as combining and tailoring aspects of Codex (Hinckley et al. (2009)), Paddle (Ramakers et al. (2014)), and Siftables (Merrill et al. (2007)) into a smartwatch form factor.

The primary contributions of our work are: 1) the concept of a reconfigurable dual-face smartwatch that enables tangible interaction and a larger display area; 2) the results of

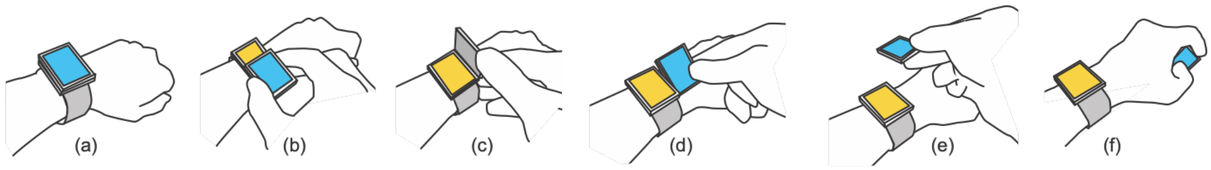


Figure 18: Illustration of Doppio interaction concept with tangible input examples: (a) top stacked on base for default use; (b) slide down to peek at information; (c) hinge to view a background app; (d) adjacent with hinge and pivot manipulation; (e) detached, distant top for sharing; (f) placing top in hand for indirect input. See also the accompanying video figure.

an exploratory study with a passive prototype probing how people might use this design space; 3) the results of a controlled experiment with an instrumented prototype to evaluate transition times between configurations and subjective preferences for different configurations and transitions; and 4) a proof-of-concept hardware prototype demonstrating how this style of interaction can be used.

6.2 RELATED WORK

After surveying smartwatch interaction techniques, we describe inspirational dual-faced conventional watches and relate our work to tangible multi-display devices.

6.2.1 *Enhancing Smartwatch Input*

Since touch is arguably the most common input method for smartwatches, mitigating the fat-finger problem and increasing the touch input space are well explored areas. For example, to accurately type on very small keyboard keys, Zoomboard (Oney et al. (2013)) uses multiple taps to zoom in on a specific key, while Swipeboard (Chen et al. (2014a)) uses consecutive directional swipes to select a character. TouchSense (Huang et al. (2014)) is an example of increasing the touch input space, in this case by exploiting differences in finger contact area when touching at different angles. Other approaches move touch input off of the display to reduce finger occlusion, sometimes also increasing the input space. For example, Ashbrook et al. (2008) investigate touch on a round bezel, Blasko and Feiner (2004) propose stroke gestures around the watch face guided by tactile patterns, Oakley and Lee (2014) explore capacitive touch sensing around the outside of the watch case, Watchit (Perrault et al. (2013)) enables a touch interaction language on the watch band, Laput et al. (2014) demonstrate touch sensitive buttons on the skin near the watch, Blasko et al. (2006) prototype watch interaction using a retractable string, Abracadabra (Harrison and Hudson (2009)) and Gesture Watch (Kim et al. (2007)) detect finger movements in the air near the watch, and Gesturewrist (Rekimoto (2001)) detects posture of the hand using an instrumented wrist band. Expanding touch input space is not our focus, but in principle, all these ideas could be combined with Doppio's tangible interaction. Two examples more closely related to our focus are Pasquero et al. (2011) haptic wristwatch that combines input by twisting the watch frame with finger gestures and touch, and Xiao et al. (2014) method of twisting, tilting, and pushing the entire watch face when attached to the band on a joystick-like mount. These examples demonstrate interactions that go beyond form and appearance of typical smart watch. Doppio also supports similar interactions and many more through its detachable top face design.

6.2.2 *Extending the Smartwatch Display*

Duet (Chen et al. (2014b)) expands the conventional approach of pairing a smartwatch with a smartphone into a cross-device and contextually-linked interaction language and WatchConnect (Houben and Marquardt (2015)) is a toolkit with a similar goal with wall-sized displays. More relevant are techniques that do not rely on any external device or display. Xu and Lyons (2015) study minimal smartwatch displays and Lenovo demonstrated a watch with a second display viewable by holding it to your eye (noa (a)), but a more typical approach is adding displays near the wrist. AugmentedForearm (Olberding et al. (2013)) proposes using the entire forearm as a watch-like display, which is demonstrated with a prototype constructed from four smartwatches. Facet (Lyons et al. (2012)) is a watch-like bracelet made of multiple touchscreens and DisplaySkin (Burstyn et al. (2015)) has a similar goal using a single flexible display wrapped around the wrist. Although these approaches significantly expand the display size, they also move away from the standard look and feel of a watch. The Doppio concept builds multiple small displays worn near the wrist, but adds the dimension of reconfigurable displays.

6.2.3 *Novel wristwatch designs*

Conventional watches are both utilitarian and aesthetic objects (Ashbrook et al. (2008); Xu and Lyons (2015)), so we were inspired by how conventional wristwatches position and manipulate two watch faces. The *Titanium Two Face* (noa (d)) has two adjacent watch faces to display two time zones. The *Piaget Altiplano Double Jeu* (noa (b)) has a hinged top face that opens to reveal a second dial on the watch base and the *Porsche Design P'6520 Heritage Compass* (noa (2011)) uses a similar hinged top face that opens to reveal a compass underneath. The *Jaeger-LeCoultre Reverso* (noa (c)) and *Ritmo Mvndo Persepolis* (Kansa (2012)) place the two faces on either side of the watch case, such that it can be flipped to switch between two dials. *Halda's Space Discovery* (**?**) comes with two faces, analog and digital, that can be swapped into a single housing on the band. The Doppio interaction language has a similar adjacent configuration with two faces side-by-side, a hinge action to access a secondary face, and a top face that can be detached.

6.2.4 *Multi-display Tangible Devices*

Multiple connected displays can also expand the interaction space. The Codex (Hinckley et al. (2009)) tablet has two hinged displays and uses hinge angle and orientation to changes between eleven modes including a detached mode by removing a display from the case for sharing. PaperFold (Gomes and Vertegaal (2015)) extends Codex to three hinged displays, but with permanently connected displays. Paddle (Ramakers et al. (2014)) has mechanical connections to fold into seven different shapes forming a tangible interaction language. The device requires two-hands, but introduces a “peeking” interaction expanded in our work. Siftables (Merrill et al. (2007)) demonstrate how multiple small displays can use orientation, proximity, and side-by-side adjacency for tangible interaction. The displays are not physically connected and are designed to be used on a table.

The Doppio concept combines and extends these ideas: hinging from Codex; peeking from Paddle; and orientation and adjacency from Siftables. However, Doppio is based around a smaller form factor and is specifically designed for a smartwatch context with tangible constraints such as an anchored display and one-handed manipulation. In spite of these constraints, a larger interaction space is created exploiting the reconfigurable nature of two displays that can be mechanically detached and joined in many ways.

6.3 DEVICE DESIGN OPTIONS

A number of different options influence the form factor and design of a Doppio device. As a convention, we refer to the two display faces in a Doppio smartwatch as the "top" and the "base." The base is securely attached to the band like a standard smartwatch, but the top can be attached to the base face, or any of its sides. While attached, the top can be slid, rotated, tilted or detached completely. When attached to the face of the base (called a "stacked" configuration), the combined top and base resemble a standard smartwatch. The following general requirements inform design choices for the top and base, attachment mechanisms, and sensors.

6.3.1 *Top and Base Shapes*

The shape of the top and base faces influences interaction affordances and capabilities. If both faces are rectilinear (e.g. square) or have large facets (e.g. hexagonal), the top may be stably attached to the sides of the base and different attachment points are well-defined. If both faces are curvilinear (e.g. circle), attaching the top to the base is less stable and less well-defined, but the top can be continuously rolled around the base. If the faces have an aspect ratio other than one (e.g. rectangular) there can be a landscape versus portrait distinction, and rotating the top while stacked on the base creates "off-axis" orientations that can be differentiated visually and by touch. We use rectangular faces in our prototypes and illustrations.

6.3.2 *Mechanism to Attach Top to Base*

Enabling high degree-of-freedom movements for the top when attached to the base, while also enabling the top to be detached, is challenging. We experimented with mechanical methods using LEGO and low-fidelity mockups and found ball joints mounted on a turntable a reasonable subset of desired movements. However, mechanical joints are difficult to fit into a small device and the movement constrains are limiting. We use magnets placed at key locations. These enable a variety of physical attachment locations that can be "broken" to manipulate or detach the top. Magnets can be easily integrated and allow large movement flexibility. The magnetic force also has an exploitable quality as the top is pulled towards some locations and repelled from others. However, a magnetic connection is less strong and the top could fall. A hybrid approach is tethering the top to the base using an elastic or retractable string. This would enable string-based interaction (Blasko et al. (2006); Pohl et al. (2013)), but would also be technically challenging to integrate in such a small form factor.

6.3.3 *Sensors*

Both faces have a digital display and touch screen, but other sensors are needed to detect orientation, attachment location, and position of the top relative to the base. Magnetic sensors can track position Chen et al. (2013) and attachment location, but precision is reduced due to attachment magnets. IMUs and magnetometers can track relative orientation of the top for rotating, hinging and pivoting. Proximity, light, force, or capacitive sensors can detect when and how the top is attached to the base. We use capacitive and IMU sensing.

6.4 INPUT SPACE

There are numerous ways to attach the top to the base and different ways for the top to be in a detached state. This creates a set of 124 *configurations* with many possible *transitions* between them. Additional *manipulations* can vary a configuration or transition, e.g. rotating, tilting, sliding, or moving the top relative to the base and touching the display of the base or top. In this section, we define and enumerate possibilities and define a concise terminology and syntax used throughout the remainder of the paper. We have not yet evaluated cognitive demands or semantics associated with this interaction vocabulary. Our aim is to show the range of possible interactions. In later sections, we observe how this input space could be used, evaluate times and preference for key transitions, and demonstrate how this input space can be applied to real applications.

6.4.1 *Configurations*

The configuration is determined by how the top is attached (or not attached) to the base and the orientation and position of the top relative to the base. The number of variations for each configuration type is noted in parenthesis and Figure 19 illustrates example configurations.

Closed (4) — The top is face down, attached to the face of the base such that no display is visible. The top can be rotated around the normal vector of the base, with the top orientation expressed as a compass direction relative to the base (N, S, E, W). As a convention, the relative orientation of the top is given in square brackets after the configuration name, e.g. "*Closed[N]*", "*Closed[E]*", etc. (Figure 19 a).

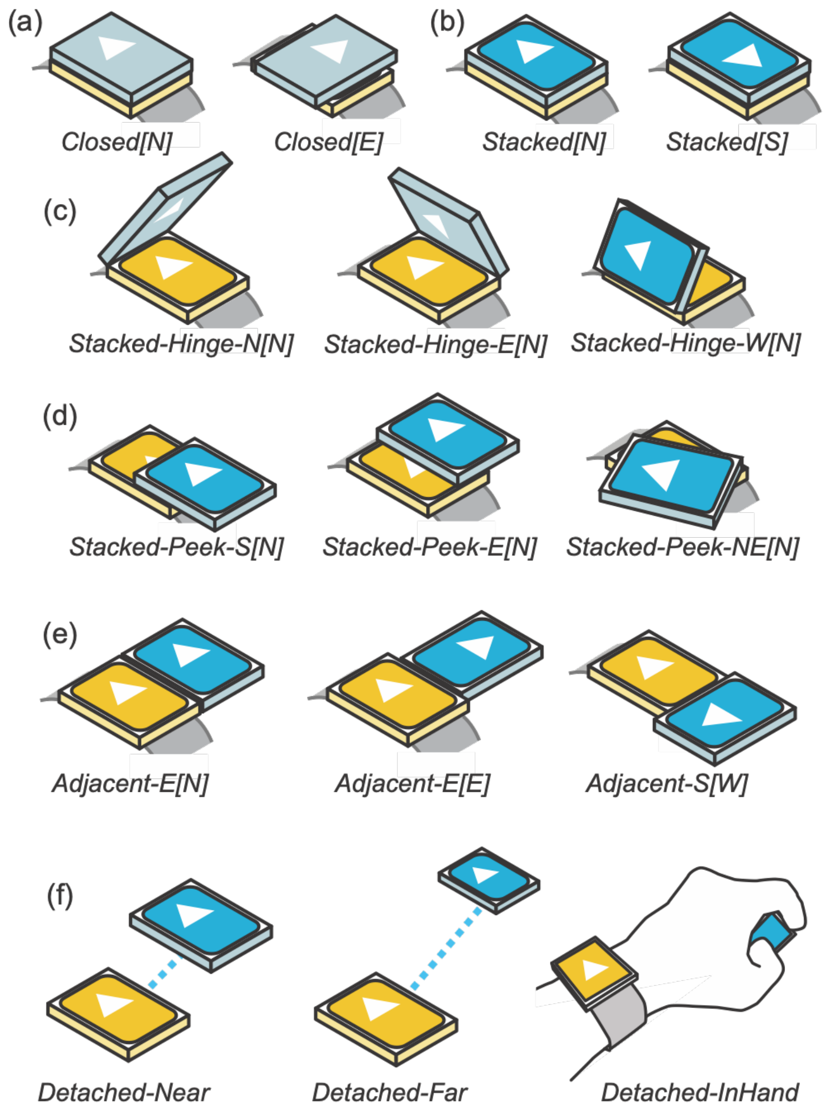


Figure 19: Example Configurations.

Stacked (4) — The top is face up, attached to the top of the base such that only the top display is visible. The relative orientation of the top is the same as Closed, e.g. "*Stacked[N]*", "*Stacked[E]*", etc. (Figure 19 b).

Stacked-Hinge (16) and *Closed-Hinge* (16) — One side of the top is attached to the base, resembling a hinged door or partially opened book, revealing the base face beneath. The relative orientation of the top is the same as Closed. The side of the base where the top is attached is given as compass direction (relative to base) after the configuration name, e.g. "*Stacked-Hinge-N[N]*", etc. (Figure 19 c).

Stacked-Peek (32) and *Closed-Peek* (32) — The top is attached to the base face, but partially slid in a direction so that a portion of the base face is revealed. The relative orientation of the top is the same as Closed, given as the pre-peek orientation. The direction that top has slid is given as compass direction (relative to base) after the configuration name, e.g. "*Stacked-Peek-S[N]*". Off axis peeks with 45° rotation are possible for peeking at a corner of the base, e.g. "*Stacked-Peek-SW[N]*". (Figure 19 d).

Adjacent (16) — The top and base are side-by-side, planar or near planar. The side of the base to where the top is attached is given as a compass direction (relative to base) after configuration name, e.g. "*Adjacent-E[N]*". The orientation of the top is the same as *Closed* (Figure 19 e).

Detached (4) — When the top and base are detached, the top can be near the base ("*Detached-Near*") or far from the base ("*Detached-Far*"). The exact threshold for near and far is determined by hardware constraints, but the guideline is that the top is near when the user is holding the top. Two special detached cases are when the top is held in the watch hand, "*Detached-InHand*", or when mounted on the watch band, "*Detached-Band* (Figure 19 f).

6.4.2 *Transitions*

Different physical actions are employed when transitioning between configurations, such as sliding the top on the base, sliding the top off of the base, attaching and detaching the top from the base, and rotating the top on the base of the face, its side, or edges (see Figure 20 for examples).

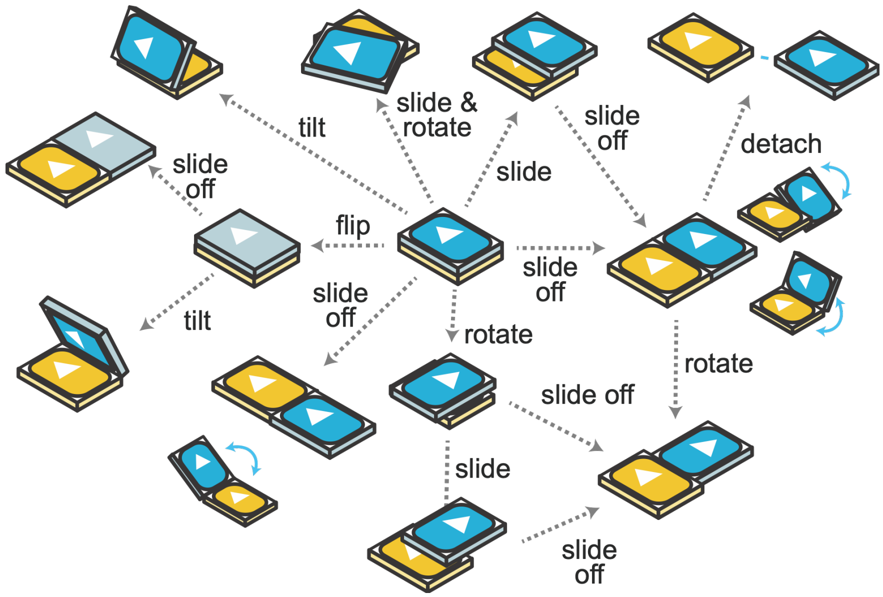


Figure 20: Example Transitions.

Transitions often pass through intermediate configurations. Consider transitioning from *Stacked[N]* to *Adjacent-E[E]*: one path is to slide from *Stacked[N]* to *Adjacent-E[N]* then a momentary detach, rotate, and attach the top to arrive at *Adjacent-E[E]*; another path is a rotation from *Stacked[N]* to *Stacked[E]*, then a slide to *Adjacent-E[E]*. Intermediate configurations can be filtered with a threshold, but the transition path can also form part of the interaction language.

6.4.3 *Manipulations*

The top can also be manipulated within the current configuration (see Figure 21 for examples).

Rotation — In *Adjacent* configurations, the top can *pivot* around the normal vector of the attached side or *hinge* around the attached side. The top can also pivot slightly around the normal vector of the face without leaving a specific Stacked or Closed configuration.

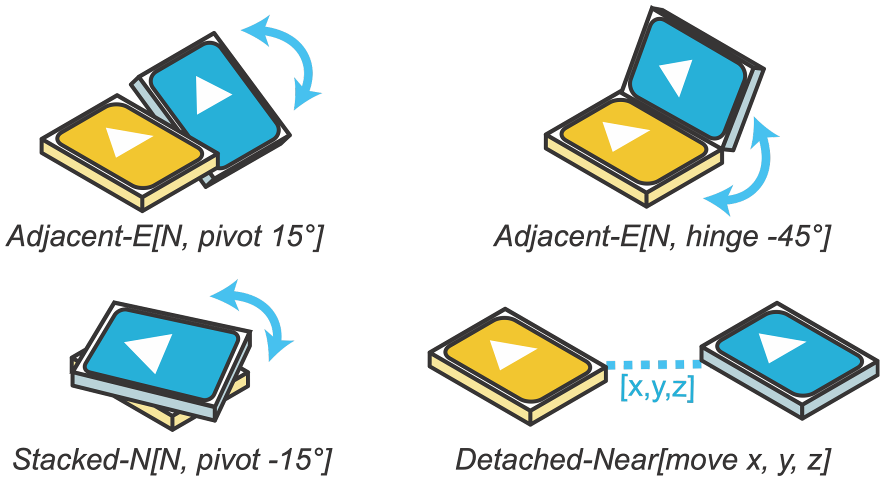


Figure 21: Example Manipulations.

The concise notation to specify these manipulations adds the axis (*hinge* or *pivot*) and angle between the square brackets (e.g. "*Adjacent-E[N, pivot 15°]*", "*Adjacent-E[N, hinge -45°]*", or "S*tacked-N[N, pivot -15°]*"). Note that rotating beyond a threshold will transition to another configuration, for example pivoting more than 45° in *Stacked[N]* transitions to *Stacked[E]*.

Translation — In *Detached-Near*, the position of the top relative to the base could be tracked. This could act as input for the base, or navigate information in the top.

Touch — Manipulations can also overload the meaning of a transition. For example, transitioning from *Stacked-Peek-S[N]* to *Stacked[N]* could be done while simultaneously touching the top face to change system behavior.

6.5 EXPLORATORY STUDY

To understand how people might use this input space, we conducted an exploratory study with six participants familiar with interaction design and wearable technologies. Our goal was to discover usage themes rather than thoroughly explore specific interactions. To help ground the discussion and provide the participants with a way to demonstrate their ideas, we supplied a simple cardboard prototype (Figure 22). The prototype used magnets to connect the top and base together such that the top was free to be moved and detached.

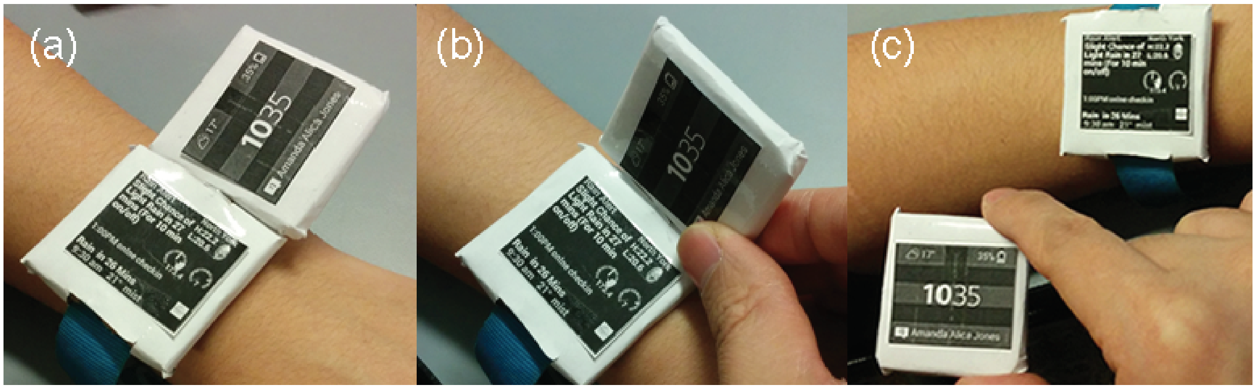


Figure 22: Paper prototype: (a) top attached to the side; (b) top hinged open; (c) top detached from the base.

6.5.1 *Protocol*

Participants were shown how to form different configurations and then encouraged to discuss situations in which they might use such a dual-face watch. To prompt thoughts and ideas, we provided pictures of common mobile applications (e.g. clock, calendar, weather). The study was video recorded so actions and behaviors could be examined. Comments were transcribed for thematic grouping.

6.5.2 *Results*

Overall, participants were positive about the form factor. Several commented on the usefulness of multiple displays, for example "*...with a regular smartwatch, I don't feel like reading emails on it, but a dual-screen watch adds more space and makes me feel that reading emails* *becomes possible*" [P3], and "*I like the form factor of the dual-screen watch and the fact that I can reconfigure the position of the top screen*" [P5]. Eight usage themes emerged:

E1 - Attaching the top to one side of the base can form a single unified display and interaction space, or each display can function separately for different tasks.

E2 - Some types of tangible input, like sliding the top a short distance, lets you provide input without covering the display with a finger.

E3 - The physical nature of tangible input seems like it would enable muscle memory for eyes-free interaction.

E4 - Attaching the top to the left or right side is more suitable for touch input since the top can rest on the arm.

E5 - When attached to the left or right side of the base, the maneuverability of the top can be impeded by the arm.

E6 - Once detached, the top could be handed to another person for sharing content like photos. However, a shared top should restrict access to personal information.

E7 - While attached to the base, the top can be hinged up and the face oriented away like a small public display, leaving the base as a private display.

E8 - When the top is hinged up and attached to the top of the base, it acts like a privacy shield.

6.6 CONTROLLED EXPERIMENT

From our exploratory study, we observed ergonomics and movement difficulty influenced input space usability. The goal of our controlled study is to measure how these issues affect transitions between key configurations by measuring time and subjective preference. The results inform interaction design considerations discussed later.

6.6.1 *Participants*

We recruited 12 participants (3 female), ages 18 to 45. All were right handed, and 2 currently owned a smartwatch.

6.6.2 *Apparatus*

An instrumented Doppio prototype was used to log transition times between configurations (Figure 23). The top and base are each 42 × 36 × 7 mm (similar size as the *Apple Watch*). Capacitive sensing is used to detect configurations. Four steel electrodes are attached to the base such that each electrode covers the side with a small portion wrapped around to the face (Figure 23b). Portions of the top are made conductive by completely wrapping the sides and face with copper tape, but leaving only eight smaller conductive areas on the back. The four base electrodes are wired to an Arduino to measure capacitive strength. Neodymium magnets are embedded into the top to increase connection reliability and provide a feeling of physical connection. A thumbpiece (Figure 23c) with a single electrode is used to accurately detect when the top was held in the hand.

The pattern of conductive areas on the base and top enable robust detection of groups of configurations, but not specific configurations. It can detect when the top is in a *Stacked* or *Closed* configuration, but not which one or the orientation of the top. Likewise, it can detect a *Stacked-Peek* configuration, but not the peek direction. It can also distinguish between *Adjacent-N*, *Adjacent-E*, *Adjacent-S*, and *Adjacent-W*, but not which side of the top is attached to the base. However, detecting when a configuration in a group is entered and exited provides the necessary data.

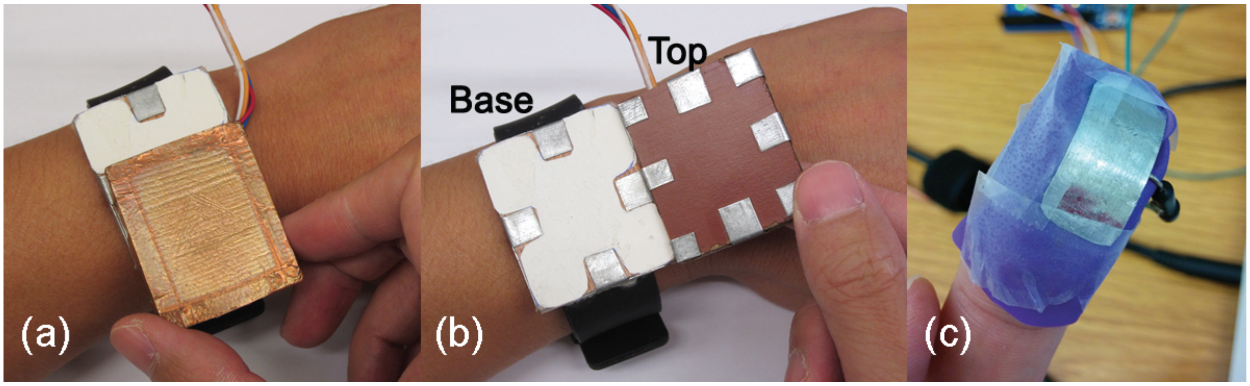


Figure 23: Prototype for timing transitions: (b) base electrode placement and top conductive pattern; (c) thumb piece.

A C# application prompted participants to perform transitions according to the experiment design. By prompting participants to transition from a certain start configuration to a certain end configuration, we could record specific transition times. During the experiment, we monitored their compliance and provided a redo function (accessible to both experimenter and participant) when a wrong configuration was accidentally transitioned to. The C# application logged events from the Arduino synchronized to the prompted transition. In many cases, the transition time is simply the duration between two detected configurations, but there are cases where we ignore an assumed intermediate configuration. For example, when rotating the top to transition from *Stacked[N]* to *Stacked[S]*, we ignore the intermediate con figuration since it must be *Stacked[W]* or *Stacked[E]*.

6.6.3 *Task*

Participants were required to transition from a specific starting configuration to a specific ending configuration. The start and end configurations were shown as graphical illustrations (like Figures 1-4). Illustrations and instructions were shown together on a notebook computer placed such that participants could see the watch and visual instructions simultaneously. All tasks were performed while standing.

The participant positioned the top in the starting configuration before a trial started, and transitioned to the end configuration as quickly and accurately as possible. When a trial was completed, a sound was played and the visuals and instructions were updated to reflect the next trial. Task completion time is the time from when the start configuration was exited until the ending configuration was entered.

An error occurred when the device did not detect the correct ending configuration, if the experimenter observed a wrong configuration or a detection failure, or if the participant self-reported a mistake or a technical problem. In all cases, an error tone was played and the trial was repeated.

6.6.4 *Design and Protocol*

From the input space, we evaluated 37 TRANSITIONS spanning 7 transition FAMILIES (listed in Figure 24). Many tested transitions are equivalent to non-tested ones by symmetry (e.g. the tested transition from *Stacked[N]* to *Stacked[S]* is equivalent to *Stacked[S]* to *Stacked[N]*). *Detached-Far*, *Stacked-Hinges*, and manipulations were not included due to ambiguity in the ending configuration position. We included *Stacked-Hinge* in a preference questionnaire.

The experiment design used 1 full practice block and 2 measurement blocks. Each block presented all TRANSITIONS grouped by FAMILY, with FAMILY ordered by increasing complexity of movement: *Stacked Rotations*, *Closed*, *Detaching*, *Straight Peeks*, *Corner Peeks*, *Stacked to Adjacent*, *Adjacent Rotations*. This aided learning compared to a counter-balanced design, and order effects are minimized because all families are presented in each block.

Within each FAMILY, 3 repetitions of related transitions were presented in randomized order (e.g. the 3 related transitions in *Stacked Rotations* from *Stacked[N]* to *Stacked[E]*, *Stacked[S]*, and *Stacked[W]*, see arrows in Figure 24). Participants took breaks as needed. In summary: 2 blocks × 37 TRANSITIONS (grouped by 7 FAMILIES) × 3 repetitions.

After the timed portion of the study, participants completed a questionnaire asking for subjective preference for all transitions in the timed portion with the addition of 4 *Stacked-Hinge* transitions. Ratings were from 1 to 7 using a continuous numeric scale (1 least preferred, 7 most preferred with decimal ratings like 3.5 permitted). Including questionnaire, the experiment averaged 40 minutes.

6.6.5 *Data Processing and Statistical Analysis*

Outlying trials with times more than three standard deviations from the mean were removed (34 trials, 1.3%). Repeated-measures ANOVA and pairwise t-tests with Holm correction were used for all measures. Trials are aggregated by participant and the factors under analysis. To correct for non-normally distributed time data, all statistical tests are performed on log transformed times.

6.6.6 *Results*

6.6.6.1 *Time by Transition Family*

There is a significant effect for FAMILY on time ($F_{6,66} = 15.21$, $p < .001$, $\eta^2 = .33$). Post hoc tests show: *Adjacent Rotations* (860 ms, SEM 14) are slower than all other families except Detaching (795 ms, SEM 29) and *Closing* (584 ms, SEM 37) (all $p < .05$); *Detaching* is slower than *Stacked Rotations* (572 ms, SEM 21), *Stacked to Adjacent* (541 ms, SEM 11), *Corner Peeks* (499 ms, SEM 17), and *Straight Peeks* (383 ms, SEM 8) (all $p < .01$); and *Straight Peeks* are faster than *Stacked Rotations* and *Stacked* to *Adjacent*. *Adjacent Rotations* and *Detaching* incur more movement overhead, but all family transition times are reasonable suggesting this style of tangible interaction is feasible.

6.6.6.2 *Time by Transition*

Figure 24 illustrates times for all tested transitions. We analyze each FAMILY with more than 1 transition separately. For *Straight Peek* and *Corner Peek* families, there are no significant effects of TRANSITION on time. *Peeking* is already among the fastest families, and this indicates there is little difference in time for different peek directions.

For the *Stacked Rotation* family, there is a significant effect of TRANSITION on time ($F_{2,22} = 23.748$, $p < .001$, $\eta^2 = .19$). Post hoc tests show *Stacked[N]* to *Stacked[S]* (755 ms, SEM 46) was slower than other transitions (mean 482 ms). Turning 180° is almost twice as slow as 90°: our observations suggest this is due to the majority of participants using a two-part turning strategy for *Stacked[N]* to *Stacked[S]* where the top is turned by 90 in two operations. Some participants discovered a faster strategy by grasping the top from the North side and turning it 180° in one operation.

For the *Stacked* to *Adjacent* family, there is a significant effect of TRANSITION on time ($F_{7,77} = 7.1413$, $p < .001$, $\eta^2 = .20$). Post hoc tests show *Stacked[N]* to *Adjacent-N[N]* and *Stacked[N]* to *Adjacent-S[N]* are both slower than *Stacked[N]* to *Adjacent-E[N]*, *Stacked[N]* to *Adjacent-W[N]*, *Stacked[W]* to *Adjacent-W[W]*, and *Stacked[W]* to *Adjacent-E[W]*. The general pattern is that attaching the top to the East or West sides is faster than North or South. We believe the advantage may be attributed to the way participants rest their hand holding the top on the other arm to make alignment of the top with the side of base easier.

For the *Adjacent Rotations* family, there is a significant effect of TRANSITION on time ($F_{11,121} = 8.911$, $p < .001$, $\eta^2 = .15$). Post hoc tests show: *Adjacent-E[N]* to *Adjacent-E[S]* is slower than *Adjacent-E[N]* to *Adjacent-E[E]*; *AdjacentN[N]* to *Adjacent-N[S]* is slower than *Adjacent-N[N]* to *Adjacent-N[E]*; *Adjacent-S[N]* to *Adjacent-S[S]* is slower than *Adjacent-N[N]* to *Adjacent-N[W]*; *Adjacent-W[N]* to *Adjacent-W[S]* is slower than *Adjacent-E[N]* to *Adjacent-E[E]* and *Adjacent-N[N]* to *Adjacent-N[E]*. Similar to the results for *Stacked Rotations*, 180° rotations incur overhead. In addition, there were some differences in 90° rotations: *Adjacent-W[N]* to *Adjacent-W[W]* is slower than *Adjacent-E[N]* to *Adjacent-E[E]*, *Adjacent-N[N]* to *Adjacent-N[E]*, and *Adjacent-N[N]* to *Adjacent-N[W]*. This provides some statistical evidence supporting the general trend visible in Figure 24: *Adjacent-W* rotations are more costly.

6.6.6.3 *Subjective Preference*

Figure 24 illustrates mean preferences for all tested transitions. We analyze each FAMILY with more 1 transition separately. For the most part, the pattern of preference corresponds to the pattern of times, and the significant differences follow a similar trend. For *Straight* and *Corner Peeks*, and *Stacked to Adjacent* there are no significant TRANSITION on preference effect. For the *Stacked Rotation* family, there is a significant effect of TRANSITION on preference (F2,22 = 17.406, $p < .0001$). Post hoc tests show *Stacked[N]* to *Stacked[S]* is less preferred (all $p < .01$), the same pattern as time. For the *Adjacent Rotations* family, there is a significant effect of TRANSITION on preference ($F_{11,121} = 6.721$, $p < .0001$). Post hoc tests found 15 pairwise differences, all involving *Adjacent-W[N]* to *Adjacent-W[E]*, *Adjacent-W[N]* to *Adjacent-W[S]*, and *Adjacent-W[N]* to *Adjacent-W[W]*. In each case these 3 transitions were less preferred than the others (all $p < .05$). This supports and extends the partial trend in significant differences of time.

The *Stacked Hinge* family was not included in the timing portion, but transitions in the family have a significant effect of TRANSITION on preference ($F_{3,33} = 15.793$, $p < .0001$). Post hoc tests revealed a strong preference order: *Stacked-Hinge-W[N]* is least preferred, then *Stacked-HingeS[N]*, and then *Stacked-Hinge-E[N]* and *Stacked-HingeN[N]* which are both most preferred.

6.6.7 *Discussion*

The combined time and preference results from the controlled study indicate that the majority of transitions are reasonable or cumbersome. Peeks are fastest (less than 400ms on average) and all variations received similar preference. *Adjacent Rotation* is the slowest family, but within an acceptable 860ms on average and the majority of its variations are rated neutral or marginally positive. Our results do identify transitions that should be used sparingly or avoided: all Adjacent Rotations on the west side and perhaps *Stacked[N]* to *Stacked-Hinge-W[N]* based on preference. Transitions requiring 180° top rotations were somewhat slower in all families and less preferred in many cases. We believe this is partly due to a commonly used inefficient rotation movement strategy, but regardless, transitions with more than 90° top rotations should be used with caution.

6.7 DESIGN CONSIDERATIONS

Based on device options, input space, and results of the formative and controlled studies, we generated a set of design considerations for functional characteristics and mappings for

Doppio interaction. Related formative study themes and controlled study results noted when applicable.

6.7.1 *Functional Characteristics*

Characteristics are opportunities, affordances, and constraints that influence how the input space is used.

One-Handed - Interaction is performed with the non-watch hand, but interesting counter-examples are when the top is held by the watch hand or operated by both hands. For example, the top may be held with the watch hand so thumb touches provide indirect input to the base (theme E6).

Stability – The magnets create a spring-like resistance favouring certain positions and orientations in attached configurations. During transitions, the top snaps into most configurations, and manipulations can be reset by snapping back to a preferred position and orientation (theme E3).

Shape - Given the rectangular shapes, *Stacked[E]* and *Stacked[W]* feel temporary. A wide *Adjacent* configuration is well suited to displaying larger information spaces (E1).

Touch Constraints - Touch input is not always possible or comfortable with the base and top. For example, the base face is impossible to touch in *Stacked*, possible to touch in *Stacked-Hinge*, and easy to touch in some *Adjacent* and *Detached* configurations. When *Adjacent*, the top is easiest to touch in *Adjacent-W* or *Adjacent-E* given arm support (E4).

Ergonomics - Configurations are not equal in terms of comfort and usability (E5). For example, compared to *Adjacent-E*, transitioning to *Adjacent-W* is awkward due to handover-hand action and occlusion. Configurations that orient the top away from the user (like *Stacked-Hinge-W[N]*) make the top awkward to view or manipulate.

6.7.2 *Functional Mappings*

Design characteristics create semantic mapping opportunities, general techniques, and useful qualities. Interaction techniques and scenarios encompassing many of these qualities are demonstrated in a following section.

Multitasking - *Stacked-Hinge* configurations can be used to monitor or switch applications (e.g. in *Stacked[N]*, the top shows the time, *Stacked-Hinge-E[N]* reveals weather app on the base, and a tap swaps weather to the top). Relative orientation is another way to multi-task (e.g. *Stacked[N]* shows time, *Stacked[S]* shows fitness app). The controlled study found that 180° top rotations are slower and less preferred, but this may be due to the type of grasp.

Extra Information - Fast and unstable *Peeks* are suited to temporarily viewing small bits of information like notifications or system status. They also increase privacy (e.g. generic notification icon on top, Peek to view content). *Corner Peeks* and *Side Peeks* can view extra information for the current app depending on the amount of information.

Slide Off for More - When a small bit of content is visible in *Peek*, the top can be slid off the base until it springs into an *Adjacent* configuration to view full information.

Sharing - When detached, the top can be momentarily given to someone else to play a game, or view content (e.g. a photo) (E6). A manipulation (e.g. pivoting) during the transition from *Detached-Near* to *Detached-Far* can indicate how much control the user has over the content (e.g. view current photo only, view the whole album). Manipulations on the base can also remotely control top content.

Modes - Application modes can be selected according to the *Adjacent* configuration (e.g. weather by city vs weather map) and *Adjacent Rotations* can refine the current mode further (show forecast or satellite map).

Tuning - Discretizing the hinging and/or pivoting rotation space can be used to change a viewing mode of an application from an *Adjacent* configuration (e.g. small hinge rotations to toggle a map between streets, satellite, and both).

Reduced Occlusion - In *Adjacent* or *Detached* configurations, touch input on the top can control an interface shown on the base eliminating occlusion (E2). For long term single-handed navigation, the top can be held in the watch hand and used like a thumb-activated touch pad to navigate content on the base (e.g. pan and zoom a map).

Silence and Reset - The *Closed* configuration can be used to silence notifications or if held temporarily, as a way to reset an application state (transition from *Adjacent* to *Closed*, then back to *Adjacent* to reset fitness activity tracking).

Private and Public - Hinge configurations have built in affordances to suggest public sharing (e.g. *Stacked-Hinge-N[N]*) or private viewing (*Closed-Hinge-N[N]*) (E7, E8).

6.8 DOPPIO DEVICE PROTOTYPE

As a full proof-of-concept, we built a high-fidelity hardware prototype with two high resolution touch displays able to detect all primary Doppio configurations, transitions, and manipulations. Here we provide device technical details, and demonstrate the interaction vocabulary in the following section. The prototype uses two Sony SWR50 smartwatches with factory wristbands removed. Each SWR50 is placed in a custom 3D printed housing (8.75 $\times$ 6.75 $\times$ 2 mm) to form the top and base (Figure 25). The prototype is larger and thicker than we envision for a real device, but it is effective for validating and demonstrating the concept.

The base housing has four embedded capacitive sensors linked via conductive tape to the top face frame and wired to an off-board Arduino. The top housing is wrapped in conductive tape and has four magnets embedded at the corners to snap into *Stacked* and *Adjacent* configurations. The pattern of triggered capacitive sensors enables detection of *Stacked* and *Adjacent* configurations.

Each SWR50 has a 1200 MHz processor with a 1.6” (320×280px) capacitive touchscreen and orientation and inertial motion sensors. The relative orientation of the top to the base for detecting Hinge and Pivot movements is determined using the built-in accelerometer and gyroscope of each SWR50. The proximity of the top to the base for Detached configurations is measured with the built-in magnetometer on the base SWR50 (similar to the method used by Abracadabra (Harrison and Hudson (2009))). 3D position tracking is possible via a pair of magnetometers (Chen et al. (2013)), but not currently implemented.

Each SWR50 runs custom Android Wear software that passes sensor state to a server via WiFi. Combined with the Arduino’s capacitive state, the server determines the current configuration, transition, or manipulation and updates the application display on both SWR50s also through WiFi.

6.9 INTERACTION DEMONSTRATIONS

We explore the characteristics and mappings in the Design Considerations section by applying them to specific system and application functions using the Doppio device prototype (see also the accompanying video).

6.9.1 *Global Functions*

Doppio interactions can be used for global functions like managing notifications, system status, and applications.

Managing Notifications - The transient nature of *Peeks* and the hidden base display in a *Stacked* configuration are combined to manage notifications. When a notification occurs, sliding the top down to *Stacked-Peek-S[N]* previews notification content on the base without altering the top display (Figure 26a). Sliding back to *Stacked* dismisses that notification, but sliding back while touching the top display saves it. While previewing a notification in a *Peek*, sliding off the base to an *Adjacent* configuration opens the associated application on both displays (Figure 26c). For more privacy, a notification preview can transition to a *Stacked-Hinge-N[N]* to shield the base display with top (Figure 26b).

Managing System Information and Settings - Instead of using space on the top display for system information, *Corner Peeks* can easily reveal this information on the base display. For example, pivoting the top slightly counter-clockwise from *Stacked* to *Stacked-Peek-SW[N]* reveals the bottom right corner of the base where the battery level can be shown. System settings could also be managed in this manner. For example, pivoting *Stacked* clockwise to *Stacked-Peek-SE[N]* could view the number of notifications in the bottom-left base corner, but if the same pivoting motion is done and the top allowed to immediately spring back this could silence notifications (Figure 26d).

Multitasking - Launching and switching between apps is challenging on a smartwatch. Transitioning to different *Stacked* and *Stacked-Hinge* configurations is a tangible way to switch between apps. For example, two foreground apps (like weather and fitness) can be mapped to *Stacked[N]* and *Stacked[S]* so that a half-turn of the top switches between them (Figure 27a-c). Alternatively, *Hinging* the top can access background apps like calendar on the base, and the background app can be swapped to the top by tapping the base display (Figure 27d-f). One background app can be dedicated to full system information and settings, accessed with a less preferred *Hinge* like *Stacked-Hinge-S[N]*.

Launching Applications - Apps can be launched by turning the top one-quarter turn to *Stacked[W]* or *Stacked[E]* to view icons for two groups of frequently accessed apps (Figure 28a,b). Tapping an icon launches the app. Additional apps can be viewed and launched by sliding the top off the base to an *Adjacent-S* configuration (Figure 28c).

6.9.2 *Applications*

We designed and built four applications to show how different design characteristics and mappings can be applied.

Weather - The default view of the weather app shows the current temperature at the current location (Figure 29a). A top *Corner Peek* views *extra information* like the humidity. We reserve top *Corner Peeks* for app information and bottom corners for system information. Additional extra information is revealed with a *Stacked-Peek-E[N]* to show an hourly breakdown of the weather forecast. Sliding the top to *Adjacent-S[N]* changes the mode to a four-day forecast and *Adjacent-N[N]* changes the mode to a weather forecast of four cities (Figure 29b). In this configuration, discrete input is provided with a hinge manipulation *Adjacent-N[N,hinge]* to show more cities (Figure 29c). Rotating the top from *Adjacent-N[N]* to *Adjacent-N[E]* changes the sub mode to an hourly forecast for individual cities.

Map – The wide *shape* when *Adjacent* is ideal for viewing a map spanning both displays. Pivoting the top can zoom for *continuous input* while avoiding *occlusion*. Alternatively, panning or pinching touch input can be performed on one display leaving an occlusion free view of the map on the other (Figure 30a). In situations where a hand is unavailable (e.g. carrying a bag), the top can be used *one-handed* by the hand wearing the watch for indirect input (Figure 30b).

Photos - Handing the top to another person is an example of *sharing* in a photo app. The top can be detached from the base and shared with others on two levels. The default is *Locked* where the top displays the current photo when the top was detached and disables all interaction (Figure 31a). Alternatively, if the top touch screen is held while detaching, the top is semi-locked enabling the other person to interact, but only with the photo app (Figure 31b).

Media Player - Media player control can be accessed without a graphical user interface detracting from media display. Similar to Xiao et al. (2014), video playback can be controlled by physically manipulating the top. For example, *Stacked-Peek-E[N]* and *Stacked-Peek-W[N]* to seek forward and backward (Figure 32) and *Stacked-Peek-N[N]* and *Stacked-Peek-S[N]* to increase or decrease volume.

6.10 LIMITATIONS AND FUTURE WORK

We discuss limitations of our work and suggest future research for exploring similar interaction spaces.

Device Size and Thickness - The size and thickness of Doppio can be further reduced with additional engineering efforts. The smartwatches in our current implementation can be replaced by mini touchscreen LCDs with custom designed PCBs and batteries. We believe that with today's technology, it is possible to create a dual-face smartwatch similar in size to a regular smartwatch. We also see alternative design options that can improve the Doppio form factor. For example, a better battery size strategy can be used. The base can have a larger battery than the top, as it needs power in smaller periods of time in the detached stage. The top can also be powered by the base when attached, allowing it to be made even smaller to reduce size and weight.

Battery Life - Batteries drain quickly in our current implementation due to frequent WiFi data communication between the two watch faces and because both displays are always on. Battery capacity can be increased, but power consumption can also be reduced. For example, switching to a low-power protocol like Bluetooth LE and automatically turning off the base when covered by the top.

Shape of the Watch - While we explore the notion of dualscreen smartwatches in a rectangular shape, a round watch face may provide a unique affordance for new types of interactions. For example, a round face allows users to perform a number of unique continuous input methods, such as pivoting or rotating the top along the curved edge of the base. Future research will investigate different shape options and interactions enabled by different shapes.

Alternative Designs - Doppio can further benefit from an additional screen on the back face of the top. This may introduce new interactions that are not possible using two screens. Future research should focus on exploring new interactions enabled by a double-sided top face, as well as methods to overcome the technical and engineering challenges introduced by an additional screen in a small watch form factor. Tethering the top using a string provides potential for new interactions, warranting further research.

Evaluation - Doppio warrants careful evaluation in the field. For example, understanding usability in real-world usage scenarios to identify issues not revealed in our controlled experiment and examining the benefits and limitations of sharing the top in different social settings.

6.11 CONCLUSION

Our work introduces the concept of a reconfigurable dual-face smartwatch, designed to address the limited interaction space in a small smartwatch form factor. Doppio's two display faces, including a detachable top, can form 124 different configurations with associated transitions and manipulations. Through an exploratory study, we investigated how users could practically utilize the Doppio concept. We also conducted a controlled experiment using an instrumented prototype to measure the transition time between the different configurations. The results of both studies and subjective preferences provided insight into a set of practical characteristics and design considerations to be applied to Doppio applications. Finally, to demonstrate the wide variety of novel Doppio interactions, we built a proof-of-concept hardware prototype, demonstrating how Doppio's interactions can be used for common watch scenarios and applications. As research in smartwatches and wearable devices continues to increase, we believe our work can inspire new ideas and designs for the future wearable devices.

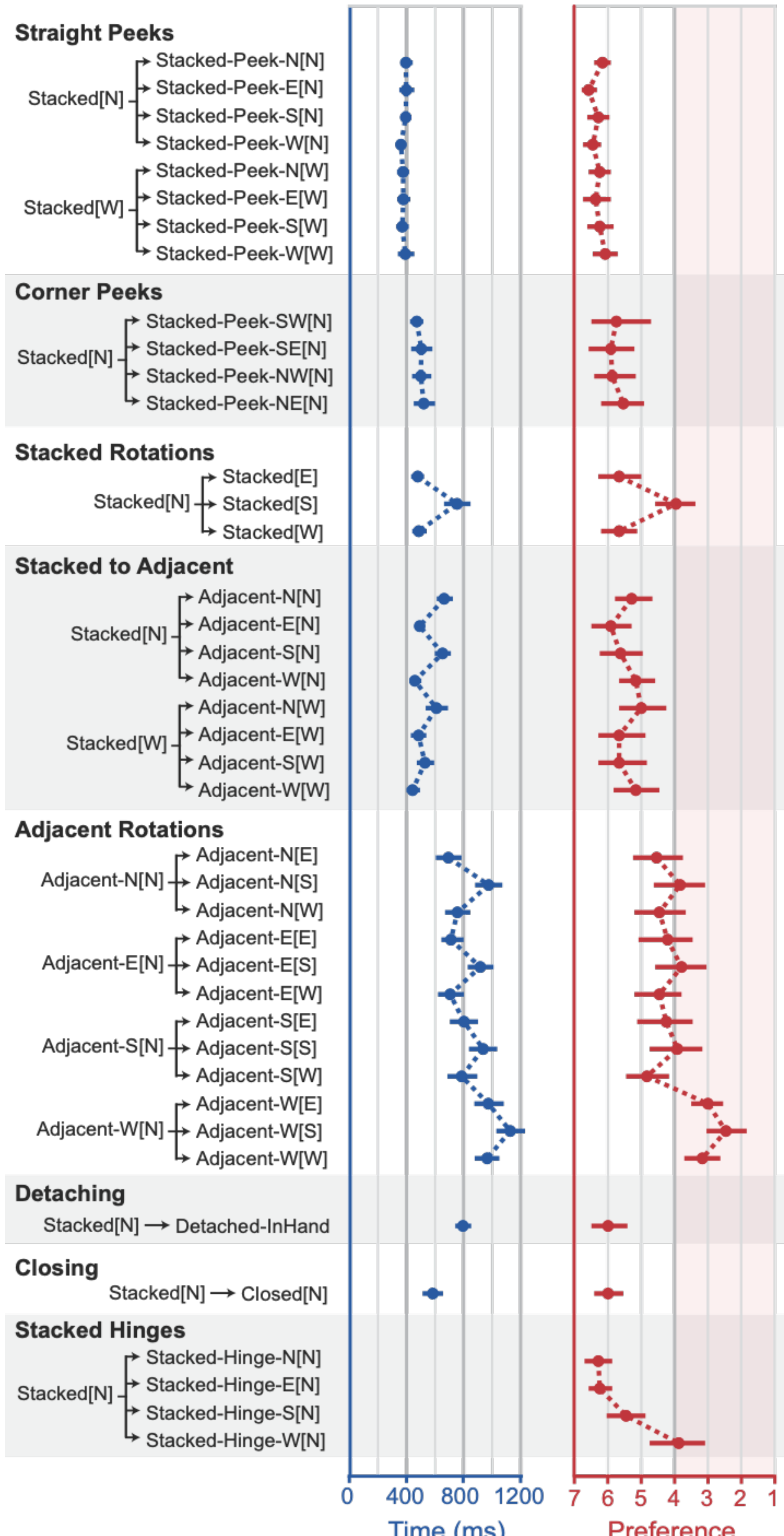


Figure 24: Time and Preference by Transition. Preference axes flipped to show correspondence with Time. Stacked Hinge times not tested. Error bars are 95% confidence interval.

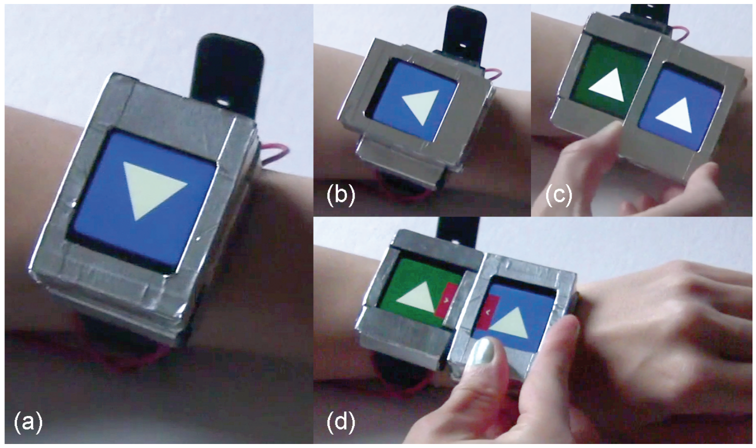


Figure 25: High Fidelity Prototype: (a) top stacked; (b) top rotated 90°; (c) top peek off base; (d) top attached to side of base.

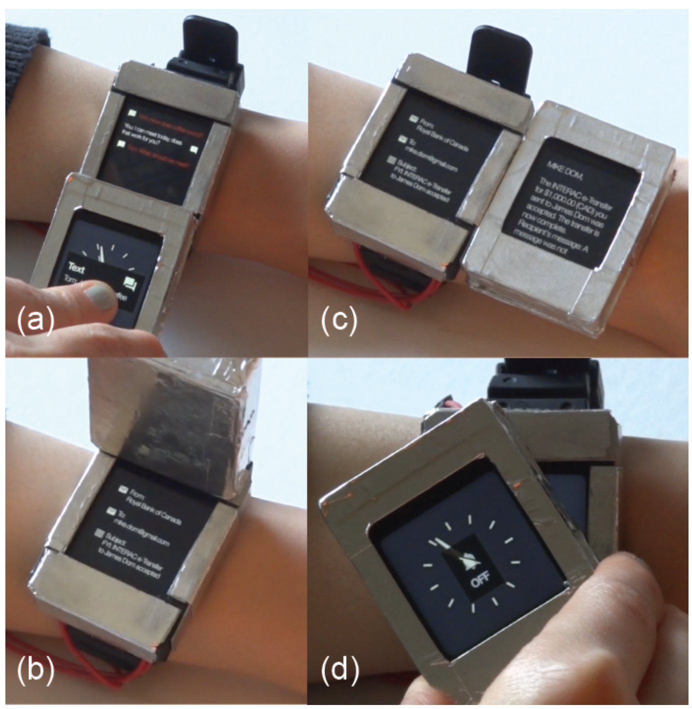


Figure 26: Notifications: (a) preview notification content; (b) viewing a notification more privately; (c) opening associated notification app using adjacent; (d) disabling notifications.

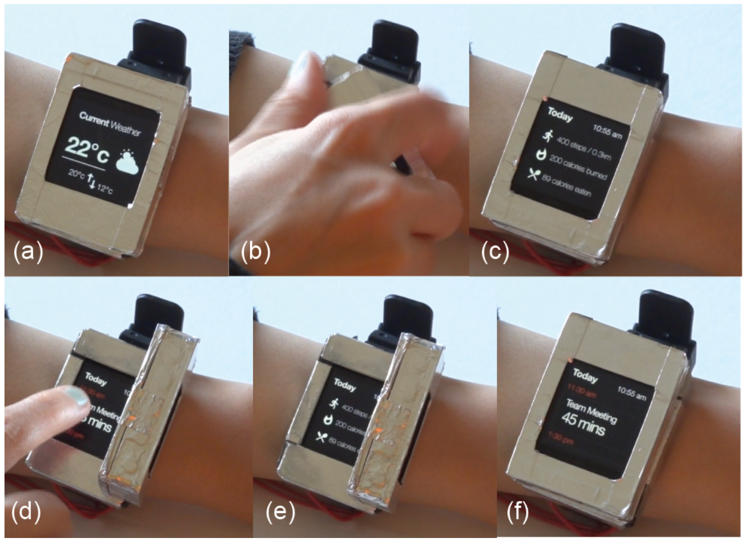


Figure 27: Multitasking: (a) foreground app; (b,c) turning to switch foreground app; (d) stacked hinge to view background apps; (e,f) swap background app to foreground with tap.

Figure 28: Launching apps: (a, b) frequent app icons in left and right groups; (c) viewing more app icons in group.

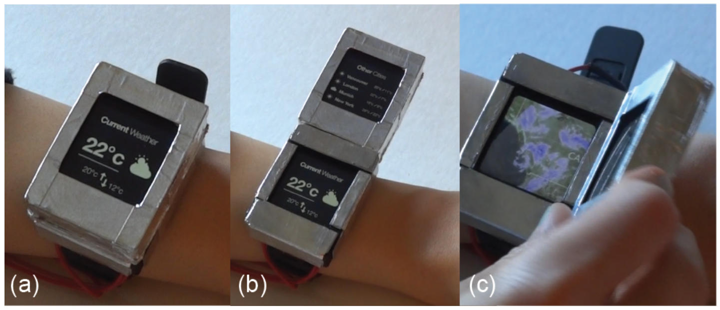


Figure 29: Weather app: (a) Stacked shows local conditions; (b) Adjacent shows city forecasts; (c) hinging alters map.

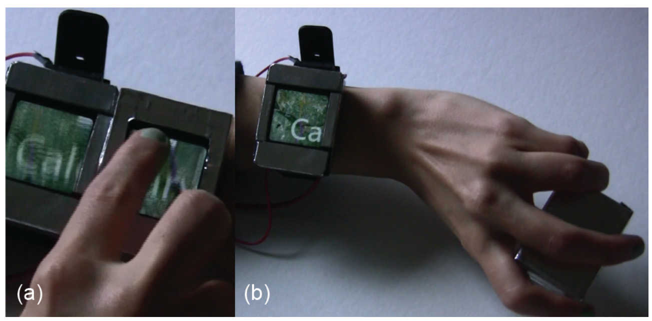


Figure 30: Map app: (a) Occlusion free map navigation on one Doppio screen; (b) One-handed interaction on Doppio.

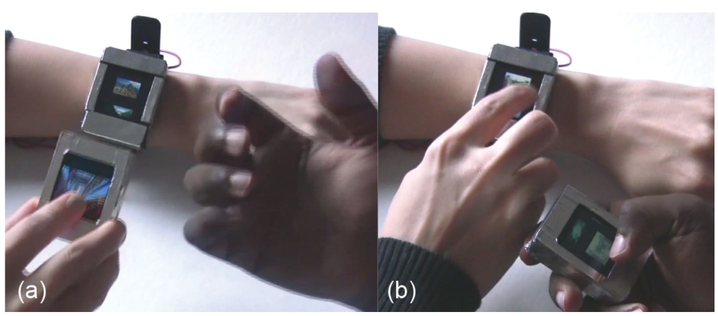


Figure 31: Photo app: sharing photos by sharing the top.

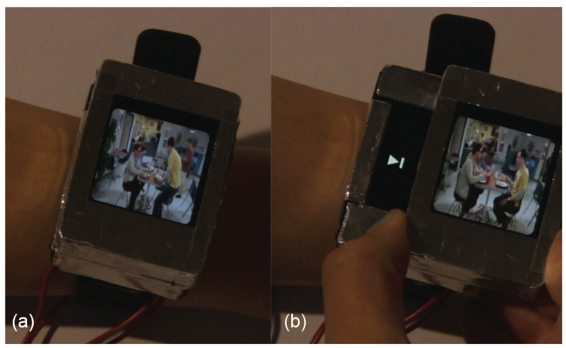


Figure 32: Media player app: (a) Pushing the top right to fast forward the video; (b) Pushing the top up/down for volume.

7

A MODULAR SMARTPHONE FOR LENDING

We motivate, design, and prototype a modular smartphone designed to make temporary device lending trustworthy and convenient. The concept is that the phone can be separated into pieces, so a child, friend, or even stranger can begin an access-controlled interaction with one piece, while the owner retains another piece to continue their tasks and monitor activity. This is grounded in a survey capturing attitudes towards device lending, and an exploratory study probing how people might lend pieces of different kinds of modular smartphones. Design considerations are generated for a hardware form factor and software interface to support different lending scenarios. A functional prototype combining three smartphones into a single modular device is described and used to demonstrate a lending interaction design. A usability test validates the concept using the prototype.

***Note: The text in this chapter appears in the following publication:**
Teddy Seyed, Xing-Dong Yang, and Daniel Vogel. 2017. A Modular Smartphone for Lending. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17).* ACM, New York, NY, USA, 205-215.
🏆 *Best Talk Winner*

7.1 INTRODUCTION

People already lend their smartphones to people they trust (Karlson et al. (2009); Liu et al. (2009)). Common reasons include entertaining a child with a game or video, letting a colleague make a phone call to avoid international roaming, or asking a passenger to update a GPS destination while driving (Bødker and Christiansen (2012)). The trouble is that it is difficult, or impossible, to control what content or functionality is shared while lending, and personal information can be accessed or destroyed intentionally or accidentally (Liu et al. (2009)). This is why the related notions of *trust* and *convenience* are primary factors when lending a device (Brush and Inkpen (2007); Matthews et al. (2016)).

Trust between the device owner (the *lender*) and the person borrowing the device (the *lendee*) strongly influences device lending (Matthews et al. (2016)). For example, smartphone lenders will typically refuse to lend to people they do not know well, like strangers, even if the lending need is important, short, and harmless. At the other extreme, smartphone owners typically lend their device to close relatives or friends, regardless of the actual level of trust. Since an implicit level of trust is communicated by a choice to lend, or not to lend, owners feel social pressure to lend (Hang et al. (2012)). Issues of trust are magnified when the lender restricts lending duration (Brush and Inkpen (2007)) or they are not in close proximity of their device (Hang et al. (2012)). For a lendee, borrowing a device also has risks, like forgetting to sign out of accounts or deleting personal information before returning the device (Hang et al. (2012)).

One method to increase trust is to use login profiles for lendees, such as a guest account (Liu et al. (2009)). However, setting up and configuring multiple profiles, as well as the process of signing out of the owner's account to sign into a guest account, takes time and mental effort. Primarily due to this inconvenience and effort, guest profiles are not used often or ignored entirely (Brush and Inkpen (2007)). Researchers have suggested methods to configure access control at the moment of lending (Hayashi et al. (2012); Liu et al. (2009)), or context-sensing access control (Seifert et al. (2009)), but these still require some manual interaction. More importantly, with current lending methods the owner must temporarily give up their phone. This is inconvenient because their task is interrupted and they may miss important notifications.

We believe issues of trust and convenience can be addressed using a modular smartphone designed for lending. Modular smartphones have been proposed to customize or upgrade functionality. We extend this idea to a smartphone that has modules that can be easily detached and lent. For example, a small piece can be lent to a friend to make a phone call, or a medium piece lent to a child to play a game (Figure 33). Lending starts and ends with physical detachment and attachment of an access-controlled piece, the owner is not inconvenienced because they retain their phone, and our lendingspecific interactions enable lendee monitoring and customizing access during the lending session.

The concept and implementation is motivated and guided by research investigating the closely related topic of sharing devices, an online survey about trust and convenience when lending current smartphones or future modular ones, and a lab study to evaluate different modular form factors and elucidate requirements for the device and lending interactions.

Based on this formative work, we built a functional hardware and software prototype to demonstrate how a lendable device could work and validate the design in a small usability study. Our contributions are: (1) the concept of a modular smartphone designed for trustworthy and convenient device lending; (2) the results of our formative work, where we first motivate and validate the concept and then explore modular form factors for lending; and (3) an interaction design for device lending with a modular smartphone realized in a functional proof-of-concept hardware prototype.

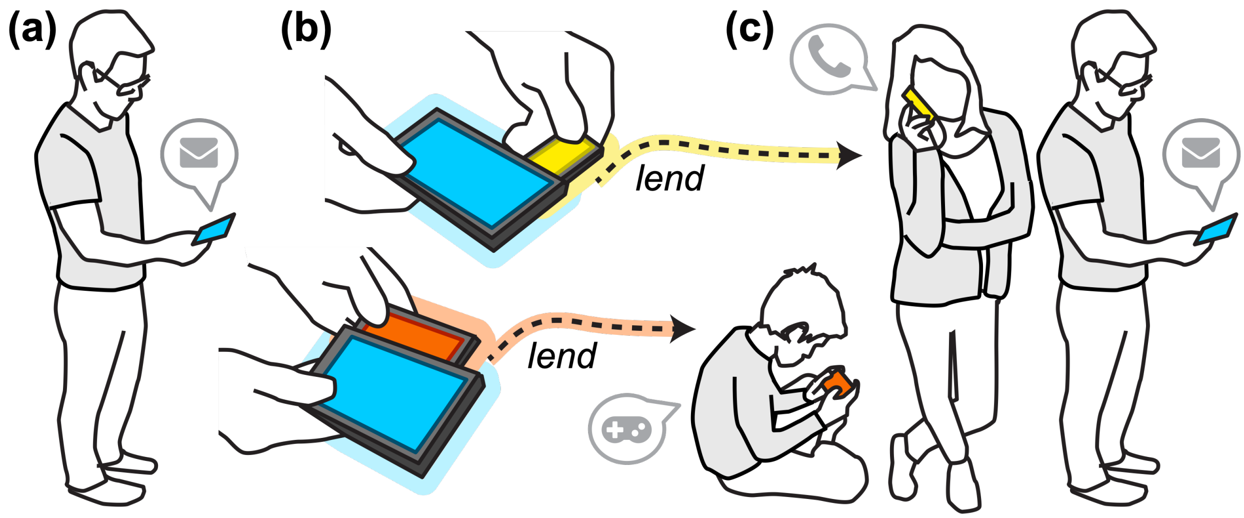


Figure 33: Modular smartphone for lending: (a) owner wishes to lend phone; (b) they slide out a module configured for limited access; (b) they hand it to their child to play a game, or to a friend to make a call, while they keep their phone.

7.2 Related Work

In HCI research, the verbs *share* and *lend* are sometimes used interchangeably, but the definition reveal a distinction: *share* means "to partake of, use, experience, occupy, or enjoy with others" and *lend* means "to give for temporary use on condition that the same or its equivalent be returned". Our focus is on lending (and the complementary verb, to *borrow*). After discussing current device lending practices, we examine trust and convenience issues, and argue why a modular device form factor addresses these issues.

7.2.1 *Current Device Lending Patterns*

Despite smartphones and other devices containing personal information, lending is commonplace (Brush and Inkpen (2007); Liu et al. (2009); Matthews et al. (2016)). Karlson et al.

(2009) found people shared their mobile phones more frequently than they initially believed, and this raised privacy concerns. Müller et al. (2012) found multiple family members shared a single tablet for tasks like playing games and information seeking. Bødker and Christiansen (2012) report that close friends and family members shared smartphones. Liu et al. (2009) found common apps used by lendees included maps, web browser, music player, and games. Device lending is especially common in developing countries due to socio-economic conditions (Murphy and Priebe (2011)). Steenson and Donner (2009) found device lending is essential in these countries, and often organized around familial relationships.

Trust is a primary factor that influences lending (Karlson et al. (2009); Matthews et al. (2016); Steenson and Donner (2009)). Trust in a lendee translates into a willingness to share, from open sharing (full access) with close family members and friends, to more limited sharing with strangers (phone calls only) (Karlson et al. (2009)). Since trust is implicitly communicated by the lender, they sometimes share more information than necessary to avoid harming established relationships (Matthews et al. (2016)). Convenience is another contributing factor for lending a device (Matthews et al. (2016)), handing over your phone to a friend so they can check a map or play a game requires very little effort. Matthews et al. (2016), identified six types of account sharing between family members: *borrowing*, *mutual use*, *setup*, *helping*, *broadcasting and accidental*. They found that convenience and trust influenced the characteristics of all six scenarios. Our lendable device focuses on supporting trust and convenience for scenarios like those described by Matthews et al. (2016).

7.2.2 *Trust and Convenience Issues*

Although trust is a factor when lending, the lender is at risk for security issues like loss of privacy (e.g. revealing sensitive information), malicious behavior (e.g., using device access to post on social networks), and accidental damage (e.g., deletion of data, changes in app settings) (Bødker and Christiansen (2012); Hang et al. (2012)). These issues can also occur for a lendee, they may forget to remove their information before returning the device, leaving their private information exposed to the lender (Hang et al. (2012)). Additionally, a lendee may find themselves in uncomfortable situations where they unintentionally view information (e.g. incoming messages from a spouse, personal photos) but still desire to respect the lender's privacy (Hang et al. (2012); Hayashi et al. (2012); Liu et al. (2009)). Liu et al. (2009) found lenders were reluctant to lend apps with personal data, like photos, videos, or messaging. This is a primary reason why lenders keep in close proximity to their device and the lendee (Bødker and Christiansen (2012); Hang et al. (2012)).

Most current mobile operating systems support multiple account profiles, or a guest mode. Using these to lend may decrease risk (Hang et al. (2012); Matthews et al. (2016)), but these mechanisms can be brittle and are underused in practice due to effort to configure them and switch between profiles (Brush and Inkpen (2007)). For example, many families share a single profile on a desktop computer or tablet, despite the availability of multiple user accounts (Egelman et al. (2008)).

Alterative solutions have been proposed to manage security and privacy more easily for sharing and lending. *xShare* provides a lender with custom access controls and a way to switch to guest mode in a user interface integrated into a lending workflow (Liu et al. (2009)). *Treasurephone* can automatically determine access control for a lendee based on application context (Seifert et al. (2009)). Our lendable smartphone also enables custom access control, a simple way to enter guest mode, and methods to leverage application context, but we accomplish this using physical manipulation of the lendable modules.

7.2.3 *Modularity*

Regardless of software improvements for lending, a lender will still be inconvenienced because they cannot access their information (e.g. time-sensitive notification) or their smartphone's functionality (e.g. make a phone call) while lending. Our approach is to lend only a piece of a phone using a modular design, so the lender is not inconvenienced.

Commercial modular smartphones like *Project Ara*, *Phoneblocks*, *Motorola Z*, and *LG G5* use modularity as a way to customize or enhance device capabilities. For example, replacing a camera module to upgrade the lens and sensor, or adding a battery module when going on a long trip. Modular approaches have even been applied to fitness wristbands and smartwatches. We use a different approach with modularity, by creating modules that are self contained and fully interactive so they can operate like simple smartphones, but under the control of a master phone.

Sharing content with interactive displays that can be attached and detached has been demonstrated on a larger tablet scale by *Codex* (Hinckley et al. (2009)), a smart watch scale by *Doppio* (Seyed et al. (2016)), and an even smaller "block" scale by *Siftables* (Merrill et al. (2007)). These systems demonstrate the feasibility of multiple interactive displays working independently or jointly, depending on physical connection. Doppio demonstrated the idea of separating one watch face to share photos with a friend. This was a primary inspiration for the concept of a lendable smartphone.

7.2.4 *Interacting with Multi-Display Devices*

Using multiple connected displays has shown a wide variety of interaction techniques. An early example is "pick and drop" which allowed a user to transfer digital items between wall-displays and a PDA (Rekimoto (1997)). Similarly, Dachselt and Buchholz (2009) used a "throwing" technique to place content onto a large display at different distances. Paay et al. (2017) describe general approaches used in cross-device interaction techniques for large displays (*pinching*, *swiping*, *swinging* and *flicking*). Bragdon et al. (2011) highlighted the impact of multi-device interactions on co-located collaboration. We build upon existing research work in multi-device interactions and apply their techniques in the context of a modular smartphone.

7.3 STUDY 1: LENDING ATTITUDES AND PERCEPTIONS

To motivate and ground our work, we conducted a survey about lending trust and convenience with current smartphones and to probe the idea of a future modular phone. Our goal was to discover current perceptions and attitudes about how devices, scenarios, and social relationships affect lending and possible directions for modular lendable devices.

7.3.1 *Study Method*

We conducted a 15-minute online survey with United States residents using SurveyMonkey. Respondents were presented with three *lending scenarios*: lending to a close friend or family member; lending to a colleague like a co-worker or classmate; and lending to a stranger. For each scenario, they commented on trustworthiness, convenience, and lending frequency with two current lending methods in mind: handing over an unlocked smartphone; and logging out to activate a guest profile. At the end of survey, we introduced the idea of sharing a piece of a modular phone and asked for comments.

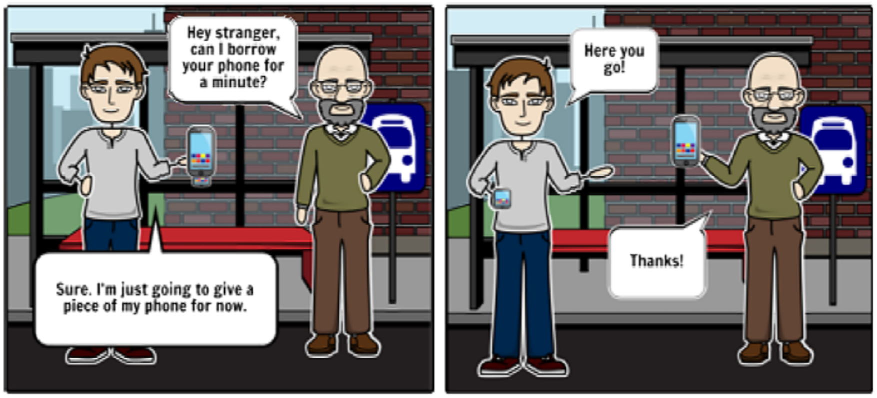


Figure 34: Example of comic strip used to explain lending scenarios (lending a piece to a stranger shown).

We used comics to demonstrate the techniques and scenarios, including the lendable modular smartphone concept (see Figure 34). The study explained it as containing a small piece equipped with a touchscreen that can be detached from the body of the smartphone for lending. Participants imagined keeping and using the small piece to control and monitor what the lendee does on the lent smartphone. They were informed that the lendable device does not reflect the final design and they should only focus on the high-level concept.

7.3.2 *Results*

We collected responses from 54 people, 29 male and 28 female, ranging from 25 to 75 years of age. Overall, qualitative findings confirm smartphone lending occurs in all three scenarios, but there are issues in current lending practices.

Confirming prior work, we found that device lending practices and willingness to lend either their smartphone or piece of a smartphone was linked to the relationship between the lender and lendee. In the scenarios where the lendee was either a colleague or stranger, using a modular design was more preferred than using the default unlocked state of a smartphone

(Figure 2). One participant noted "*…I would probably "loan" it more frequently to friends and relatives*". We also observed some unwillingness to share with strangers at all, one participant commented "*…I wouldn't lend my phone to a stranger in any circumstance*".

We also found that in scenarios where the lendee was a close friend or family member, participants preferred the unlocked (or all or nothing) approach compared to guest accounts and different profiles. This confirms previous work by Brush and Inkpen (2007); Matthews et al. (2016), and others (Liu et al. (2009); Müller et al. (2012)). Another participant based their opinion on personal experience "I would never have occasion to lend my phone to anyone other than a close friend or family and then not for their use out of my presence." This type of lending relies solely on the owners' trust in the lendee. This can be inconvenient for the lendee if they are outside of the trusted group. On the other hand, giving up the smartphone can be inconvenient for the owner as it may lead to missing phone calls or text messages. Participants saw the potential of a lendable smartphone addressing these issues, with one participant stating "*…it's a great idea to allow the use of the phone [during sharing].*"

7.3.3 *Discussion*

Our results confirm that users are forced to limit smartphone lending within a small group of highly trusted people, validating previous work and extending those results to current smartphone lending practices. Lending a device is inconvenient for both the lender and lendee. A modular smartphone has great potential to address these issues, with some participants noting "*…I like the idea better than handing over my phone*". An important takeaway for modular device design is to ensure lenders have control over who they are sharing with and always maintain a level of control of their lent piece and their information. This was echoed by participants who stated "*…as long as my personal stuff was kept safe, I could see* *myself lending...” and “...for others, I would still want to be within eyesight [and control] my phone*”. An easy-to use detachment technique is also important to ensure a positive overall user experience. Some participants noted concerns around a modular piece (or pieces) “*that could be easy to lose*” based on how they are attached and detached. We incorporate these principles into our design space and prototype designs described in later sections.

7.4 STUDY 2: LENDABLE SMARTPHONE DESIGN

The goal of this lab study is to evaluate modular designs and develop design considerations for lending interactions. Twelve people participated (ages 20 to 35) in this exploratory study using physical mockups of different form factors.

7.4.1 *Form Factor Mock-ups*

We created six physical mockups to explore module *size*, module *amount*, and module *detachment techniques*. The wood mockups are similar in size to an iPhone 5S (123.8 x 58.6 x 8.6 mm). The mockups were used to physically demonstrate the concept of lending through modularity and to guide discussion and trigger ideas during the interviews. Three classes of form factors were explored (see Figure 35):

Back-to-Back — Two identical modules are attached backto-back (each 124 × 58 × 5 mm). Lego bricks were embedded into the pieces to enable attaching and detaching. Each module resembles a very thin full-size phone.

Multi Piece (*2, 4, and 6 piece variations*) —Smaller identical pieces, each with the same thickness, attach to form a fullsize phone. The *Multi-2* mockup splits in half lengthwise, each

Figure 35: Six mockups for a lendable smartphone: each is painted wood with Lego embedded bricks for attaching.

62 × 58 × 10 mm. The *Multi-4* mockup splits into four pieces, each 62 × 30 × 10 mm. The *Multi-6* mockup splits into six pieces, each 41 × 29 × 10 mm. Embedded LEGO bricks enable attaching and detaching. The idea for 4 and 6 pieces was to enable sharing with multiple people or used together to create different module sizes.

Tray (Small and Large) — Inspired by SIM-card trays, an internal piece is slid out of the side cavity of a full-size phone. The internal piece in the *Tray-Large* mockup was 110 x 53 x 3 mm, but only 80 x 40 x 3 mm in the *Tray-Small* mockup. With the tray form factor, the main phone retains its full size, and the two modules are different sizes.

7.4.2 *Protocol*

We began by asking the participant about current device lending habits considering the scenarios from Study 1: lending to family members and close friends; lending to co-workers and classmates; and lending to strangers. Next, we explained they would manipulate different prototype form factor of modular smart phones built for lending. They were told to imagine each piece was capable of common tasks like viewing a map, making a phone call, playing games, etc.

Then, prototypes were shown one at time in counterbalanced order. For each, the participant was asked for overall impression, and to describe how they might use the prototype based on their own device lending experience. After all prototypes were examined, we led a semi-structured interview to compare all prototypes regarding how the different form factors could be used for lending, how each prototype impacted trust and convenience, and for ideas about alternative or refined designs. Finally, the participant ranked the prototypes considering trust and convenience and provided explanations. Each one-hour session was video and audio recorded with comments transcribed and placed into thematic groups.

7.4.3 *Results*

The value of the overall concept was demonstrated by comments like "*…I would definitely be more encouraged to share my phone with this*" [P2], and "*…I like the control that I have…*" [P4]. For the combined trust and convenience ranking, 41% of the participants selected TrayLarge for their top choice, followed by Back-to-Back (24%), TraySmall (19%), Multi-2 (10%), Multi-6 (4%), and Multi-4 (2%). Mockups with large pieces were also perceived to be more useful for lending since they were comparable to a full phone. We identified the following themes from observations and comments:

(T1) Retaining a Piece Increases Trust — Regardless of prototype, participants said lending felt more trustworthy because they kept a piece of their phone, particularly when lending to strangers. For example, "*I wouldn't mind as much to give a piece to a stranger, now that I can control and not worry as much*" [P12]. Several participants described previous experiences lending a phone to strangers. They found it untrustworthy and inconvenient, but noted circumstances can make lending to a stranger necessary. For example, one participant described being a stranger needing to lend a phone, "*my phone died and I needed to make a*

call… [I] was lucky to find someone to let me use their phone" [P3]. All participants said a modular design made lending to strangers more likely, for example, "*I would be more inclined to share with a stranger now that I can control everything*" [P6].

(T2) Controlling Trust by Piece Size — In all interviews, comments suggest a connection between level of trust and the size of the lent piece. The more trust there was in the lendee, the larger the piece. A typical comment: "*…I may just give my entire phone to a family member because it's easier, but for a stranger or someone I don't trust, they're getting a small piece…*" [P11].

(T3) Lender's Trust Is Visible — Several participants noted they would feel awkward lending a smaller piece since this implied "*less access to their device*" [P5]. When lending to family members or close friends, there was a potential to harm a relationship (Karlson et al. (2009); Matthews et al. (2016)). For example, "*a smaller piece, or interacting in a manner that restricts them access, feels like I am hiding something … and I don't want them to know I'm hiding anything…*" [P9]. When lending to strangers, obvious lack of trust was no issue: "*I don't really care what they think, they're not really gonna see me again*" [P12].

(T4) Monitoring Usage — Participants felt it was useful to have a sense of "*seeing what the other person was seeing*" [P9]. For example, one participant said when a lendee was viewing photos, they wanted "*to be able to see the exact same photo as them, almost like a screen share*" [P2].

(T5) Sharing Other Content — Modularity also made participants willing to lend for accessing content previously found to be hard to share conveniently (Müller et al. (2012)). One participant noted "*It's difficult to share the Gmail app or Facebook app because I have to do all this work to log out*" and later followed with "*I could easily just hand them this and they'd login to their account and I don't have to log out of mine*" [P7].

(T6) Keeping a Complete Phone — Many participants noted the Tray and Back-to-Back designs were more convenient because the lender retains a complete device *"…this sim design is cool because I still feel like I actually have a phone in my hands while someone is using part of it…"* [P11]. With other mockups, participants typically felt there was an "*incomplete device for myself and the other person*" [P12].

(T7) Master and Slave — The relationship between the pieces was also highlighted as important. Participants suggested a "*master and slave*" [P2] analogy. For the Tray designs, the main and secondary pieces had clear physical differences, "*… this bigger one is definitely the one I feel like I keep, and the thinner one inside I give…*" [P1]. With other designs participants noted there was no distinction between a master piece and the remaining piece(s).

(T8) Form Factor — All participants suggested designs merging different aspects of the mockups. For example, several suggested merging the Tray designs with the Multi-Piece concept. One participant said this flexibility would enable them to "*choose to give them a full piece or small pieces and still keep my phone … it would be less awkward sometimes*" [P8]. Several participants preferring sliding techniques of the Tray designs because "*…it feels less flimsy than breaking apart a device and is way easier to do*" [P4]. Combining aspects of sliding and different sized pieces ultimately proved to be the most common suggestion amongst participants. For nearly all participants, aspect ratio of the content on a piece played a factor in their mockup preference. Non-standard aspect ratios like Multi-2 and Multi-6 were disliked, for example: "*…I don't like the odd size of this [Multi-2], as I feel it's odd for me and the person I'm giving it too…*" [P10].

7.5 DESIGN CONSIDERATIONS

Primarily based on our study results, we generated design considerations for a lendable modular smartphone. Related survey results and study themes are noted when applicable.

Shape and Size of Modules — The shape of modules should at minimum, maintain a similar aspect ratio of the main piece (T8). Larger pieces are preferred for more trustworthy lendees, and smaller pieces for those less trustworthy or who need only limited functionality like making a calls(T2). Multiple modules of different sizes make an explicit choice of trust level possible (T1, T8).

Modularity Mechanism — The techniques to attach and detach modules affects perceived convenience (Study 1, T8). A sliding mechanism is preferred over a "breaking apart" mechanism since the lender retains fully functional (T6) pieces that could be lost even when not lending or the mental effort required to reassemble similar sized pieces (T8).

7.5.1 *Software Interface and Interaction*

A lending interaction begins by physically detaching a module, but a user interface for both the lender and lendee is required to monitor and fine tune the lending session.

Tight Integration — The concept of lending can be integrated into the operating system. The interaction design should support lending without any modifications to current apps, but app developers should also be provided with enhanced APIs to allow more nuanced control of lending content in their specific app (T5).

Simple and Configurable — Users should be able to lend with minimal or no explicit interaction beyond removing the modular piece (T7,T8) (in contrast to existing methods (Hang et al. (2012); Hayashi et al. (2012))). If desired, there should be fine-grained access control when beginning to lend, or during a lending session. This control should consider the type of content (T5) (after Liu et al. (2009)).

Lending Modes — There are different ways to lend a piece to support different lending scenarios (Merrill et al. (2007); Paay et al. (2017)) and different levels of trust in the lendee (T1,T2). For example the lending mode could restrict the lendee to a single app, grant the lendee access to multiple apps (e.g. phone, browser Liu et al. (2009)), enable the lendee to have full access to the piece, or even enable the lendee to view what the lender is doing (e.g. to give directions or share photos).

Lender and Lendee Control — The lender should feel in control of the lending session and by default, they should retain full phone functionality (T6). The interaction should enable real-time monitoring at different levels (T4), for example, lendee screen mirroring or a summary status of lendee activity. The lendee should have a way to request more access and feel confident any personal data they may view or generate will not be seen by the lender (T5).

Swapping Roles — The lender should be able to choose any piece to act as the master piece (T7). For example, the largest piece could be lent, and the smallest piece retained by the lender for simple monitoring and control. This is especially useful in *broadcasting lending scenarios* (Matthews et al. (2016)), where the primary piece is required to view content like video or play complex games. The ability to swap roles helps overcome social awkwardness in situations where the lendee feels they should be trusted, but the lender has some reservations (T3).

7.6 LENDABLE SMARTPHONE PROTOTYPE

Based on our design considerations, we built a hardware and software prototype to demonstrate and evaluate a lendable modular smartphone. This helped us more effectively explore the design space, and through several iterations of case designs, devices, software, and sensors, we learned valuable design and technical lessons for future hardware designs.

The phone has three modules to enable lending (Figure 37): a large phone, a medium phone, and a small phone. This design is a hybrid of the two Study 2 mockups: the medium and small pieces slide into the large piece like the *Tray* mockup and the large and medium pieces are complete phones like *Back-to-Back* mockup.

The large module is the *primary* piece. It is the only visible piece when not lending, and is the default piece retained by the lender. The medium module is a *secondary* for lending. It is a fully functional, self-contained smartphone with restricted access during lending. It is designed primarily to be lent to a trusted lendee (e.g. child, family member, close friend), but this role can be swapped with the large piece. The small piece is designed for lendees with less trust (e.g. a stranger) who typically require less functionality (e.g. strictly making a phone call). After technical details, we describe how it is used to support lending interactions.

7.6.1 *Hardware*

The prototype (Figure 37) uses a Google Nexus 6P (5.7”) as the large piece, a Nexus 4 (4.7”) as the *medium* piece, and a custom Android phone (2.4”) for the *small* piece.

Each component is placed into a custom-designed 3D printed resin housing. The housing for the medium piece with the Nexus 4 (13.9 × 7.3 × 1.4 cm) contains a hidden NFC tag placed on its back. Similarly, the housing for the small piece (9.5 × 5.1 × 1.7 cm) uses an

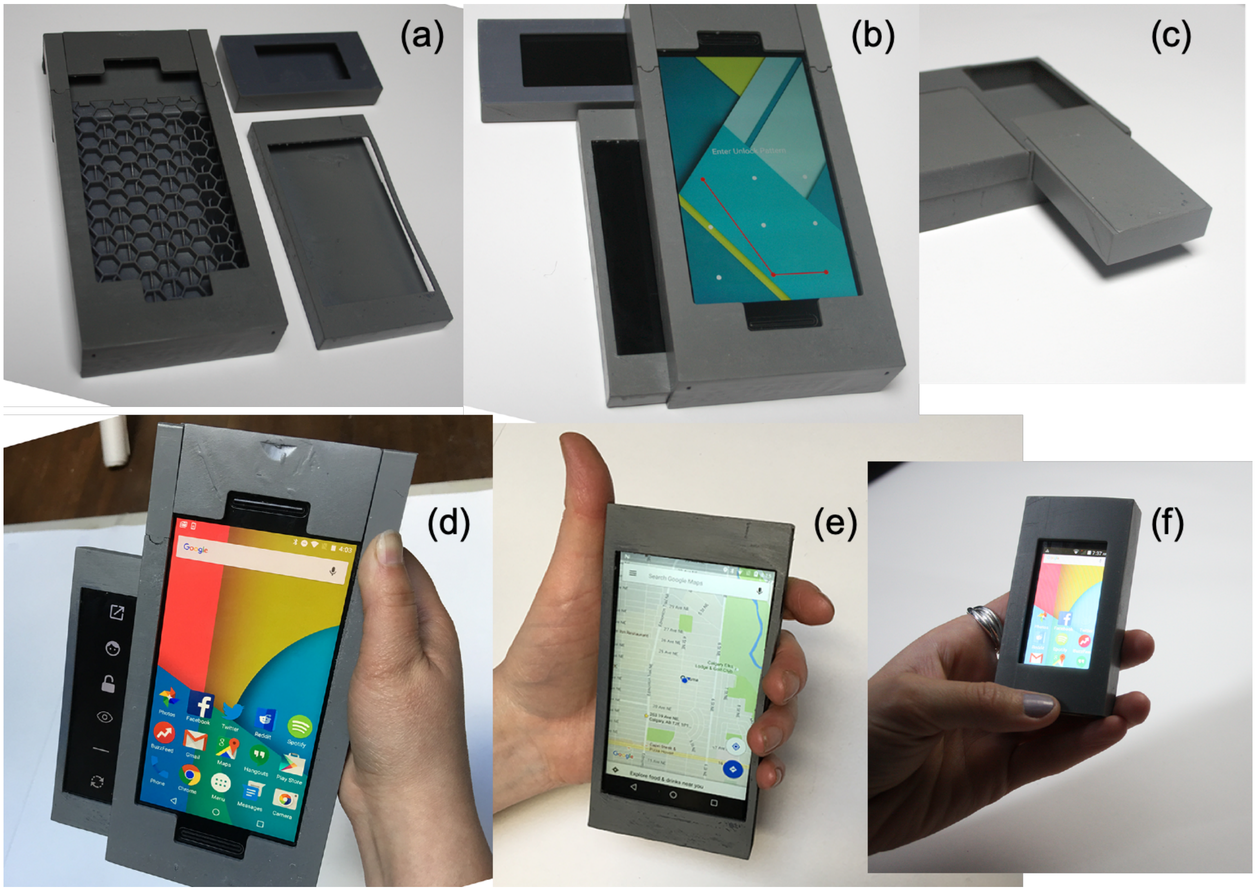


Figure 36: Hardware prototype: (a) 3D printed housings; (b) form factor showing medium and small pieces half slid out; (c) sliding mechanism; (d) large piece with medium piece half slid; (e) detached medium piece; (f) detached small piece.

NFC placed on its back. Lastly, the housing for the large piece, with the Nexus 6P ($19.4 \times 9.6 \times 3$ cm) uses a guided railing designed into the case, that allows for the Tray pieces to easily slide or be detached. The complete prototype is thicker and larger than we envision for a real device, however, it is effective and necessary for validating and demonstrating the lendable smart phone concept.

The detection of attachment of both the small and large pieces to the main piece is accomplished with the built-in NFC capabilities of the Nexus 6P. This method was accurate for detecting different configurations, as the placement of the NFC tags allowed us to detect distinct changes of state.

7.6.2 *Software*

The Nexus 6P, Nexus 4, and smaller phone all run custom Android software that passes lending state to a server via WiFi. Combined with the NFC state detection of the Nexus 6, the server determines the current lending configuration and updates the larger, medium and smaller piece. Although our prototype uses an external server to accelerate development and software design iteration, a commercial version would use the large piece as the server.

7.7 INTERACTION DESIGN

There are four types of lending interactions: starting a lending session: monitoring activity, requesting and responding, and ending the session.

7.7.1 *Starting a Lending Session*

Each lending session begins with the physical action of sliding out the medium or small piece. When partially slid out, a default lending mode is selected and a lending menu is displayed on the visible part of the piece. Using the menu, the lender can select a different lending mode, configure the currently selected lending mode, or swap roles so the large piece will be lent instead (the four lending modes explained below). Using the menu is optional — if the piece is slid out without using the menu, the default lending mode is activated and the lending session begins immediately.

This partial sliding state is designed to increase convenience and trust. A full slide is like the current practice of handing over an unlocked phone, but in our system, the piece will automatically be in a restricted lending mode. A half-slide integrates customized access control into the physical landing action. This is more convenient than manually entering a guest mode on current devices, and is more trustworthy because the lender is more likely to fine-tune access.

7.7.2 *Monitoring Activity*

The lender monitors lendee activity using a persistent lending system notification. Activity includes current app lendee is using and events like when the lendee opens a new app. The lender can expand the notification icon to view a realtime view of exactly what is on the lendee's screen. From the expanded notification, the lender may access a more detailed summary and log of all apps the lendee has used and customize lending settings. These settings are also available through a standard system menu, so lending parameters may be configured anytime.

Monitoring activity is designed to increase trust: the lender can be confident the lendee is not doing anything malicious. Although the realtime view is reasonable for many lending scenarios (e.g. a parent lending to their child to play an online game), we acknowledge it may be invasive when lending to a colleague (e.g. so they can send a private text). We imagine lendee's could be notified when the lender wishes to monitor their screen, and have the option to deny or postpone access.

Figure 37: Interaction design flows showing lender interactions on blue line and lendee interactions on orange line. Refer to text for detailed explanation and see the accompanying video for demonstrations of these interactions.

7.7.3 *Requesting and Responding*

Depending on the lending mode, the lendee may request additional permissions. For example, requesting access to another app by tapping on the app icon. This request triggers a notification on the lender's piece with option to grant or deny the request. This request and respond system is designed to make lending sessions flexible. Consider a scenario where a parent lends a piece to their child with access restricted to an educational game. After playing the game for a while, the child can request access to another game or a video app.

7.7.4 *Ending the Lending Session*

The lendee ends their session by tapping a persistent notification. This warns them if any personal data will be lost (e.g. photos taken that are still uploading to the cloud) and verifies all personal information will be cleared. The full lending session ends when the lent piece is slid back into the housing.

7.7.5 *Lending Modes*

These lending interactions are used to select and configure different lending modes: app lending; guest lending; full-access lending; and screen sharing. These modes are designed to support lending scenarios described in our previous two studies, applications typically shared during lending (Liu et al. (2009)), and the taxonomy described by Matthews et al. (2016).

App Lending — This mode supports lendees who only need access to a single app like a map or instant messaging (*borrowing* scenarios (Matthews et al. (2016))). Selecting the app for the app lending mode may be accomplished in two ways. If the lender slides out a piece while an app is open on the large piece, that app is selected for app lending mode. Alternatively, the lender can half-slide the piece, select app lending and pick the app to share. Regardless of method, completing the slide will begin a lending session with a generic version of the selected app (without any lender settings or personal information). With apps like a map, the lendee is ready to navigate. For apps like instant messaging, the lendee must sign-in to the messaging service before reading or sending texts. During the lending session, the lendee may request access to another app. The request is sent for the lender to approve or deny.

Guest Lending — This mode supports lendees who need access to a set of standard apps. By default, we guest mode provides access to common apps used during lending (Liu et al. (2009)): browser, phone, camera, and maps. Guest mode may be selected in three ways. If the lender slides out a piece with no apps open, guest lending mode is selected by default. Or, the lender can use the half slide menu to select guest mode. Finally, if a piece is slid out when the large piece is locked, guest mode is selected and the half-slide menu only provides the option to swap pieces (other lending modes may not be selected). All apps in guest mode are generic versions, no lender settings or personal information. This lending mode is useful for scenarios involving ad hoc temporary lending (*borrowing* (Matthews et al. (2016))), such as a user needing to borrow a phone in an emergency, and also prevents *accidental* (Matthews et al. (2016)) situations where a lendee accesses a lenders private information. This is also ideal for lending scenarios where a parent wants to lend their phone to their child to monitor and keep them occupied, but also prevent unnecessary applications from being purchased or installed. Several participants in Study 2 stated this to be the “killer app” for guest mode lending.

Screen Sharing — The lender may want *help* (Matthews et al. (2016)) from a lendee, such as configuring an email account or sorting locally stored photos. Or, the lender may want to share their view with others, such as showing photos from a vacation, sharing a humorous video, or going over a presentation before a meeting. To support these lending situations, a one way screen sharing mode streams a view of the lender's piece to the lendee's piece. For help scenarios, the lender and lendee can physically exchange pieces, so the lendee has full access to the phone while the lender monitors their actions. For example, a lender who is less technically adept could be helped by a family member or friend to configure settings, install applications, or demonstrate how to accomplish tasks.

Full Access Lending — A lender may choose to allow the lent piece full application access with all available functionality This is useful for scenarios involving *mutual-use* (Matthews et al. (2016)), such as when a device is equally shared by a married couple. Full access is selected explicitly using the lending menu.

7.7.6 *Small Piece Lending*

The lending menu and role swapping behaviours are different for the medium and small piece. The small piece lending menu presents fewer lending modes, chosen as suitable for the smaller piece, and the app sharing mode defaults to certain apps like phone dialing, as suggested in our studies. Given the relationship between trust and size, this functionality is thus better suited for the smaller piece, while the larger piece can be used for more trusted lendees. Therefore, we treat the secondary small piece as a touchscreen headset.

When swapping the master role to the medium piece, access to all apps, data, and functionality are swapped. When swapping the master role to the small piece, master piece functionality is focused on what is possible with the small screen: monitoring lendee activity, handling lendee access requests, and providing the lender with basic application functionality like notifications, texting, and calling.

While several of the lending modes and their associated scenarios are inspired by the broad taxonomy of (Matthews et al. (2016)) and our prior studies, focusing on a specific lending scenario, like lending to occupy a child, could have design implications. For hardware, this could mean a simpler design for the small piece, or only a small piece. For software, this could mean a simplified user interface for young children.

7.8 STUDY 3: USABILITY AND USER FEEDBACK

To gain some preliminary feedback on the prototype and elicit discussion on the topic of modular, lendable devices, we conducted a small usability study with five participants (27 to 45 years of age, 2 female, all recruited by email). The 1-hour session began with the experimenter demonstrating the prototype, then the participant used it to perform lending tasks. The 8 tasks were all based on the scenarios described in our exploratory study: lending to a close friend or family member; lending to a colleague like a co-worker or classmate; and lending to a stranger. Each task was evaluated with the participant as both lender and lendee, with the experimenter playing the alternate role.

7.8.1 *Result*

The concept was generally well received with positive feedback on the lending interactions in different social scenarios. As a lender, all participants felt that they must give a larger piece to lendees they trust, or who perceive they should be trusted confirming our earlier results.

However, trustworthiness with respect to lender information was mentioned frequently, as the distinct modes, application requests, and continuous monitoring resulted in participants feeling more secure with their data, and thus more trustworthy of others when lending the medium piece and even the primary large piece (when swapped with the small piece).

We observed that participants familiar with the Android operating system (4 of 5), appreciated the integration of lending notifications and monitoring. The remaining participant struggled initially with the software techniques due to their unfamiliarity with the android operating system, however, the interaction techniques given the scenarios was well understood. All participants commented on the large size of the prototype, and felt a smaller size would be more convenient. Some participants expressed interest in a device with only the medium piece, but also noted lending it to strangers would be less comfortable.

Comment from several participants relate to the cost benefit of carrying a modular phone. One interesting idea to address this is to make modularity optional. For example, designing the main piece to be a fully functioning lightweight and thin phone without any modules, but when necessary, enabling modular pieces to be attached when lending scenarios are expected (such as travelling with a child).

7.9 LIMITATIONS AND FUTURE WORK

We discuss limitations with future research directions.

Technical Feasibility — With further engineering efforts, the size and thickness can be reduced. Entirely custom enclosures could be used to replace the combination of stock smartphones enclosures and 3D printed cases. The smartphones would also be replaced with secondary thin-displays, acting only as clients, while a "core" module could be thicker and more powerful to drive the clients. We believe that using technology readily available today, it is possible to create a lendable, modular smartphone with a similar thickness as Project Ara (9.7 mm).

Battery Life — Other engineering improvements like a better battery size strategy should be explored. Since the medium and small pieces are not intended for all-day use, they could have small batteries that are charged when attached to the large piece. This would reduce the overall weight and thickness and reduce cost. Strategies such as using thin-secondary client displays would also reduce power battery demand.

Alternative Designs — Increased modularity on the large piece, like the *Multi-Piece* designs from Study 2, would be beneficial. Although we focused on a single prototype design, participants mentioned multi-user device lending scenarios that could benefit from alternate designs, particularly with multi-player mobile gaming. Future research should focus on exploring new device lending interactions that could be enabled by multi-user scenarios.

Other Sharing Influencers — Our concept primarily focused on *trustworthiness* and *convenience*, we did not fully explore other aspects such as cultural context, or older populations who may view device lending in a different manner.

Lending Content — We primarily focused on lending applications, with less focus on the techniques for lending content within applications. For example, a lender may also wish to share an album of photos in the photo application on the lent piece instead of a blank photos app. We also did not fully implement and explore how lendee information would be erased (or saved) when the lending session ends.

Evaluation — Understanding usability in real-world usage scenarios are necessary to identify issues not revealed in our usability study. We also did not fully explore the usability of the small piece which has more restricted capabilities.

7.10 CONCLUSION

Our work introduces the concept of a modular smartphone, designed to address issues with trust and convenience by lending only modular pieces of the device. The results of our formative studies provided additional motivation for the idea, and justifications for design choices. A proof-of-concept hardware prototype demonstrated how lending interactions can be used for common lending scenarios. Our concept may appear far-fetched at first, but we hope to have convinced the reader that there is some need and our modular strategy could be made practical. Perhaps more important, we hope our work will inspire other new ideas for future devices tailored to different needs of users, and their family, friends, and fellow citizens.

8

PIVOTING A RESEARCH CONCEPT: FROM DOPPIO TO EMOTICA

"*Once we accept our limits, we go beyond them.*"

Albert Einstein

Immediately following SITS, my entrepreneurial journey continued when I began to explore founding and running my own startups. After all, with the lessons from SITS, I have been more than amply prepared to succeed in startup life, right? Back to back "failures" can't happen!

The first venture I attempted was Slate Scale, a portable smart scale that connects to a smartphone using bluetooth. The app kept track of calories and nutritional information of the food that was being weighed on the smart scale. My co-founder and I had dubbed it the "fitbit of nutrition." Slate Scale was my first introduction into building consumer hardware, and the world of hardware manufacturing in China — albeit remotely. Despite the awards Slate Scale had won, and reasonable press coverage it received, we ultimately failed because we were unable to crowdfund the necessary amount on Indiegogo – we raised $35,000 of $50,000. If you are wondering why Slate Scale didn't continue with commercialization grants, it was clear our target market had spoken, and grants would have only delayed the inevitable. Again, having commercialization grants alongside a limited amount of customers (or none) is futile for a startup — see *Lesson 2: The Challenge with Commercialization (5.3.2)*.

Other than the value of time, the key lesson from Slate Scale was that building hardware seemed like a viable option (more on this later). Pragmatically, this was also around the same timeframe where my PhD research began incorporating aspects of hardware that I had learned when building Slate Scale. Doppio (Chapter 6) was one of my first hardware-based research projects that felt the entrepreneurial influence from Slate Scale. Being partially motivated by news coverage from places such as Gizmodo and Engadget, and continually being asked if I could make the smartwatch real and into a product, another entrepreneurial venture was born.

8.1 STARTING WITH DOPPIO

One the research side, Doppio explored how to overcome the challenges with limited input space on smartwatches, hence its dual-screened approach and tangible interactions. As a hardware product however, I was initially unclear if this was going to be good enough to compete in the smartwatch space with larger companies like Google, Apple, Huawei and others. The press Doppio had received was a tremendous help in getting some early validation, but as I learned in Slate, getting press and awards (1) doesn't necessarily translate to sales, (2) it really isn't validation for a product, and (3) it doesn't mean you have a product either.

As a result, I spent over a year trying to find different avenues finding a pitch and business case for Doppio. I even travelled to China several times, traversing the markets in Shenzhen looking at different smartwatch designs, and trying to find manufacturers in China and other Asian countries. Ultimately, I found a way to license Android Wear for Doppio with a manufacturer in Taiwan and everything was in place, which is no small feat for a student. In every practical sense, Doppio was ready to be manufactured and sold, it just needed some funding, specifically $250,000 to start. Here we go again with commercialization, right?

Instead of immediately jumping to grants to build Doppio, it was fortunate enough to be accepted into the Creative Destruction Lab (CDL)[1], in Calgary, Alberta, Canada. CDL is an accelerator for startups, designed to assist you in areas of your business to help you scale quickly, such as where does your product fit in the market, or more importantly, raising funds from venture capitalists (VCs). CDL operates in sessions, where sessions occur every 2 months, with measurable objectives being assigned to you by VCs who are your mentors. As a student, this was certainly the best part of CDL, as working alongside people who had been successful already (multiple times in many cases) was eye-opening and transformative.

For the first session in CDL, pitching Doppio was beyond humbling. As mentioned earlier, at this point I had been working on Doppio for quite some time and it was ready, or so I thought. Nearly every piece of feedback I received from a room full of VCs about Doppio was critical. Some of the feedback I received was the following:

- Is a dual-screen smartwatch even necessary, when smartwatches are a shrinking market and already highly competitive to begin with?
- Is a student even capable of creating commercial hardware? Do I have real-world experience doing-so, and make it at scale?
- Have I manufactured anything in China, ever?
- If I wasn't going after the consumer market, what other markets would Doppio fit in?

In short, Doppio as a product received feedback that I needed to hear, and not necessarily what I wanted to hear (see *Lesson 5: Product Management, Do You Have It? (5.3.5)*). As a

1 Creative Destruction Labs - http://www.creativedestructionlab.com

result of this, I was challenged to find a better market for something Doppio-like, also known as product-market fit (discussed later). This was where a pivot began, and the concept of Doppio was laid to rest. What was a research project in Doppio, was ultimately not a product (see *Lesson 1: Evolving Research to Product (5.3.1)*).

8.2 PIVOTING TO EMOTICA

After understanding the need to pivot away from a pure smartwatch consumer device, my first order of business was building a team (see *Lesson 3: Building an Organization Quickly vs. Effectively (5.3.3)*), and so a close friend of mine who was experienced and effective in many technical and business areas joined. Our cumulative skill sets were critical in moving forward to explore where and how to pivot, both on the technical side and on the business side.

We worked alongside our VC mentors from CDL, to better understand where a Doppio-like device could fit. The pivots we explored and a brief summary of the feedback for each were the following:

Medical

Medical was a clear use case for a wearable like device, but from the feedback we received, and our own research, the barrier of entry into medical devices is high. Firstly, the timelines are much longer than other markets (e.g. consumer) due to regulatory approvals which are required for medical devices. For example, getting FDA approval in the United States can take from months to years, depending on the type of device. Secondly, getting this approval can be very costly. Thirdly, and perhaps most importantly, approval processes are not the same across states, provinces in Canada, and even other countries. Thus, the cost and timelines for a medical device can be further magnified, and begs the question for an entrepreneur, is it worth exploring?

Personal Analytics

With a large number of wearables focused on monitoring for personal health and sports, we also examined how a dual-screen like device could fit into these markets. One of the unique benefits of a dual-screen device, was that it allowed us to have specialized sensors integrated into each watch face. For example, one watch face could have a heart sensor and the second could have a specialized galvanic skin response (GSR) sensor. The ability to mix and match the sensors could then be targeted towards high performance sports, gait monitoring, physiotherapy, and a number of other areas. While this was an area that was interesting, the key question for us was — despite the large number of use cases, was the target market(s) large enough to justify a large investment in new hardware and software?

Hardware Integration

The last area we explored was using a custom wearable device as an input mechanism to other platforms. For example, several virtual reality (VR) headsets rely on using mobile devices with a custom enclosure, such as the Daydream View from Google or the Gear VR from Samsung. Many of the input mechanisms for these mobile-based headsets use a one-handed, singular controller, which is not an ideal way to interact with objects VR. With a custom wearable device, we could then overcome this issue by allowing one-handed or two-handed interaction, with the second screen having the ability to be on the second wrist. Additionally, using haptics, we could then provide output capabilities for VR based content. The challenge with this hardware integration approach was that did the market need this solution, and if it did, was it big enough for a venture?

While much of Figure 38 describes hardware, we eventually moved away entirely from hardware (described in detail later) and focused specifically on using sensor data from existing wearables and providing a platform. Thus, Emotica was born.

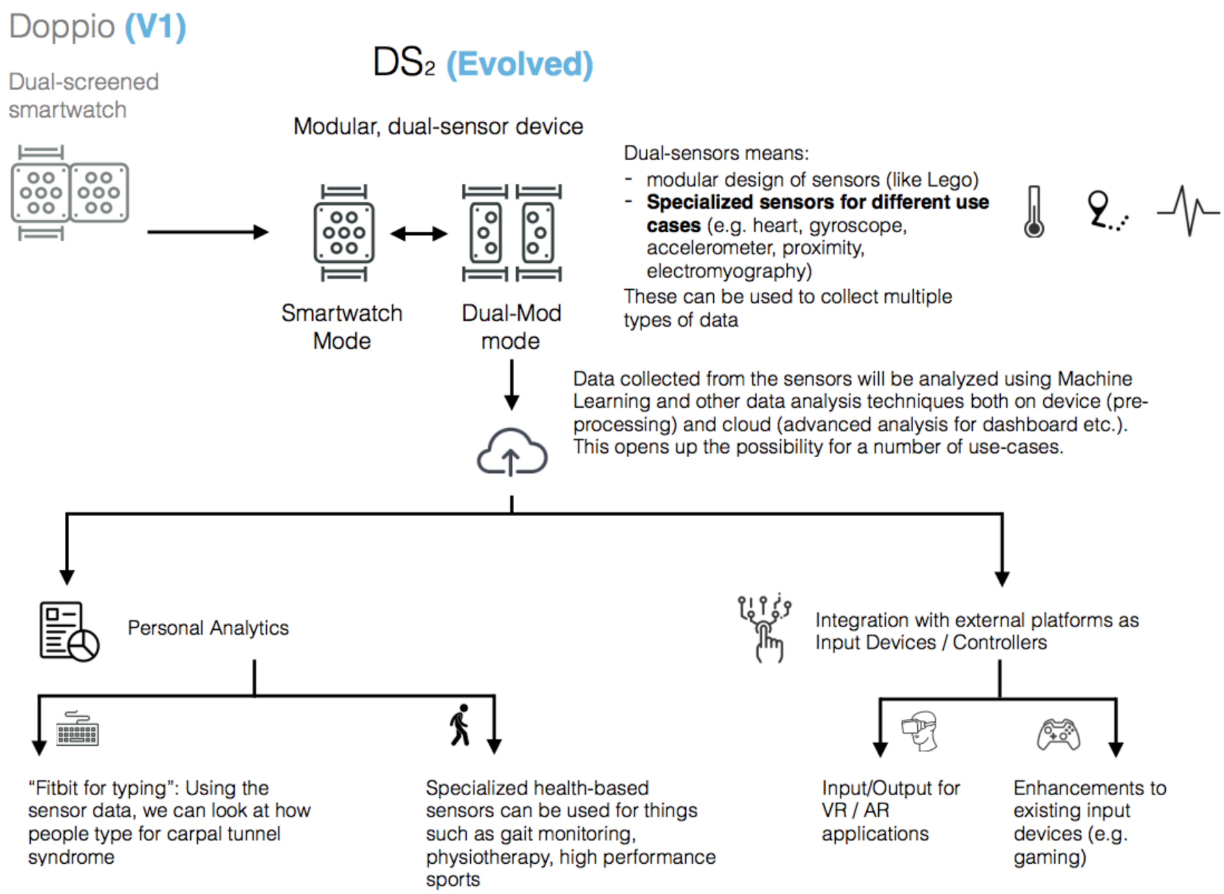


Figure 38: The process of transitioning from the ideas in Doppio to (eventually) Emotica.

Emotica arose from an extremely promising area of research we uncovered in our DS^2 explorations, AR/VR, where it is difficult for content creators (e.g. game studios) to measure engagement of their immersive content. Capturing and measuring this type of analytics is commonly known as engagement analytics. With Emotica, we aimed to better understand user engagement by supplementing traditional analytics techniques (e.g. clicks throughs, gaze tracking, surveys, etc.) with biometric data using off-the-shelf hardware (e.g. fitbits, smartwatches, etc). This ultimately meant we could quantify emotional response for immersive content, using a combination of different sensors (e.g. heartbeat, gaze tracking and/or skin response), thus providing emotional analytics for immersive content.

As you can see, starting from a consumer smartwatch to arriving at an emotion analytics platform for immersive content is quite the pivot, and journey. As a student, this illustrated several key lessons, described later in this section. Next, I describe some of my roles and responsibilities I undertook as a co-founder in this startup, named Marathoner Labs.

8.3 ROLES AND RESPONSIBILITIES

Unlike in Chapter 5 where I was working for a startup, being a co-founder was substantially more involved and required me to be more "boots on the ground" so to speak. This meant putting in substantially more hours, most of which was unpaid (at least initially) and non-glamorous (e.g. lots of emails and meetings). Below are a few of the key responsibilities my co-founder and I took on over several months. This list is not meant to be exhaustive, but rather provide a brief glimpse into some of the things my co-founder and I did when we pivoted from Doppio to Emotica.

Technical feasibility

One of our first tasks was to understand if it was *technically possible* to measure emotions in VR and how it could work in a platform. This meant first understanding the landscape of current and upcoming technologies with regards to sensors, VR headsets and machine learning (ML) techniques. For ML, which was needed to fuse different sensor information (e.g. heart beat and skin response), we examined what types of learning models and amount of data would be necessary to provide analytics. We then architected a system that could scale reasonably as users and different types of biometric sensor data was added.

As part of the technical feasibility, we built a brief demo for existing VR content and did ultimately determine this platform was possible. Part of why we did this exercise was that as strong technical co-founders, we wanted to validate the idea and be able to speak to it properly, which in plainspeak means, we didn't want to be misleading or deceptive to potential investors. From a rational standpoint, we had often seen many ideas in local startups that were vapour-ware or didn't make much (or any) technical sense. We wanted to know for ourselves that we could do what we said we could do, and so we did.

Technical feasibility

Product-Market Fit is a term coined by Marc Andreesen, and he describes it as:

"...means being in a good market with a product that can satisfy that market."

Plainly, it means if the market isn't buying a product, it isn't taking off quickly or growing, or if sales cycles are long (among many other factors), it means there is no product-market fit. Alternatively, if the product is taking off, sales are growing and so is the team behind the product, then there is likely a product-market fit.

Our approach in determining product-market fit for Emotica mirrored many of the qualitative research approaches of HCI: surveys, interviews, and questionnaires. We canvassed the top influencers of the field in VR and immersive content creation using email, LinkedIn, and Twitter, with the aim of determining if there was a market for

Emotica from experts themselves. In addition to this, we examined many of the current and upcoming market trends for the VR and immersive content and used this as a lens to determine where exactly the product could fit in the market. Combining the market research with the validation from experts enabled us to continue forward with pursuing Emotica.

Growth and Investment Strategies

Working alongside our VC mentors, we laid out growth and investment strategies. This meant planning what our team would look like in a few months, a year and several years beyond just 2 co-founders. As technical co-founders, this was interesting because at the beginning, much of the technical work was going to be done by us, as we had the background and expertise to build the product. But as we planned forward, there was going to be a need to transition out of the technical side of the startup and into the business side, thus building a strong technically capable team was important. On the investment side, we examined how and where we would spend capital, such as where and when to invest in additional product features, hiring for marketing and sales, salaries of employees and also salaries of ourselves. One interesting note here was, as a student, I was used to the mentality of not paying for myself (or my time) when working on entrepreneurial ventures. However, the truth of the matter is, you need to live, and in order to live while working on a startup, you need to pay yourself — if you can of course.

One of the first things you'll notice with this brief list of responsibilities is that much of them boiled down to *planning ahead*. From the perspective of a student, this was a fascinating approach, as these tasks helped in preparation for many of the questions (some quite difficult) that VCs ask, such as where do you see the product in 3 years, or do you plan on exiting your company in the future. These really can't be answered unless you consider them and plan accordingly.

8.4 Entrepreneurial Lessons Learned (Continued)

In the next section, I build upon the entrepreneurial lessons discussed in Chapter 5. However, these lessons are from the perspective of running a startup rather than working for a startup. Lastly, before I get into these lessons, I would like to again state this is even more of a direct reflection than those described earlier. Not all of what I experienced as a student was enjoyable, some of it was quite difficult, and thus I am conveying it as such.

8.4.1 *Lesson 7: Anyone Can Be An Advisor, But Not Everyone Can Give You The Advice You Need*

Prior to joining CDL, I heavily relied on the local Calgary ecosystem to get advice on matters of business, as well as my own intuition. After all, getting more opinions and perspectives had always worked really well for me on the research side, so it should/could work on the business side right? Wrong. If I could goldstar any of the lessons I have already discussed or will discuss later in this dissertation, it would be this one. The reason is quite simply the following:

There is a significant difference between advisors (or mentors) who have gone through the rigors of doing a startup (failed or otherwise) and those who haven't. Not being able to differentiate between the two will damage your startup and waste your time.

At first, I listened to anyone and everyone possible — as I should since I was a student and needed to learn after all. But over time, this became a significant detriment. One challenge with being a student entrepreneur is the perception your inexperience means you need constant "free" help or mentorship. What you don't realize is that this can build an unhealthy reliance on others for your entrepreneurial ideas, their validation, and over time will lead to difficult issues such as equity and ownership for your own ideas. To illustrate this point further, I will introduce and utilize a second analogy, "Alex's Apple Shop":

You are an entrepreneur named Alex, and you're just about to open up your shop that sells Apples, called Alex's Apple Shop. You have limited experience selling your apples, but you know how to grow them quite well, so you ask around your local town for additional advice to get clarity on your strategy of selling apples. You have dozens of meetings with people around town to get advice, and eventually you meet an advisor named Christopher. Christopher has experience with providing novice entrepreneurs advice and strategy, and is willing to assist you free of charge. Naively, you agree to this assistance and for a period of time, Christopher provides you advice on business strategy. At the same time, your apples are getting noticed by your town, as you have been putting in extra hours around town advertising them. Curiously however, Christopher begins taking credit for your apples which are beginning to gain traction. For a time, you ignore this thinking it is a minor issue and can be addressed later. Months later, you are ready to open your shop up, and you have the discussion with Christopher about next steps. During this discussion, Christopher asks for half of the equity of your Apple Shop. At first you are shocked, thinking how can strategy and advice equate to half of your business, when Christopher has no experience with apples to begin with. In the end, the relationship between you and Christopher breaks down, and it forces you to ask some difficult questions: what kind of help did you really need to begin with, how much was the help worth to you , and what are the motives of those who tried to you?

There are a few key takeaways from the analogy above. Firstly, the boundaries need to be set between a mentor (or advisor) and the startup. In the case of Alex, this boundary was not set nor addressed early in the venture, leading to some of his difficult realizations later. Building on this point, Alex was also naively confused into what a mentor is and what a co-founder is. A mentor should not be a co-founder at the beginning of a venture, as their role is to advise and help you grow. In Alex's case, as this boundary was not established, he was nearly taken advantage of.

A second take away from the analogy is to understand that the motives of others — such as those perceived to be mentors — is not always going to be honest nor will they always have altruistic intentions. Entrepreneurship can be messy, and being mindful is critical. In the case of Alex, he naively trusted Christopher which led to his difficult situation later. As a student entrepreneur, it is critical to not blindly trust those with more experience than you, but instead to understand the motives of those who help you, and adjust your strategy accordingly. For example, some may be purely motivated by finances (e.g. they believe you will be successful and thus help you), while others want to give back to the community by helping entrepreneurs (e.g. someone who has succeeded already). As a student, it is the latter that you should seek out, not the former.

The third takeaway concerns equity with regards to mentors or advisors. As a founder of a startup, you (and your team if you have one) control the equity in the company. While it may not seem like a big deal to start, over time if your business grows, equity becomes critical in attracting and receiving investment. In the context of Alex, if he had given equity to his advisor Christopher, he would have been significantly hampered later in attracting investment. Equity also determines the ability to make decisions in a company, and had Alex done this with Christopher, a future situation could have arisen where he could lose control of his own company. As a student, I relied heavily on the Founder / Advisor Standard Template (FAST)[2] to help novice entrepreneurs deal with mentors and advisors around the topic of equity. Below is a relevant portion of the template:

Advisor Performance Level	Stage of Startup		
	Idea Stage	Startup Stage	Growth Stage
Standard	____ (0.25%)	____ (0.20%)	____ (0.15%)
Strategic	____ (0.50%)	____ (0.40%)	____ (0.30%)
Expert	____ (1.00%)	____ (0.80%)	____ (0.60%)

Figure 39: The Founder / Advisor Standard Template (FAST) for determining equity of a mentor or advisor.

With the template, there is a clear level of equity based on the type of advisor and equity according to the stage of the startup. In the case of Alex, had he used this FAST template, it is likely he would have never proceeded with Christopher given his ultimate aim was half the equity of his apple shop. As a student, I strongly suggest having this template in hand before engaging any sort of mentor or advisor for your startup. Further, if you're going to give up more equity for strategy, you need to consider if it is really worth it for your startup, and can you get strategic advice elsewhere entirely.

For the last and (likely) most important point of this lesson, I introduce the term wantrepreneur. While there are several definitions that exist, from the perspective of a student, it was the

2 Founder / Advisor Standard Template (FAST) from the Founder Institute - http://fi.co/fast

following:

A wantrepreneur is someone who doesn't necessarily come up with (logical) business ideas, but instead follows or builds upon successful ones (or people). Common traits of wantrepreneurs including avoiding risk and putting it on others instead (i.e. you), significant focus on capital instead of the product (i.e. they will "raise" the capital, while you "build" the product), the dislike of failure (i.e. very little decision making and instead constant analysis, also known as analysis paralysis), and the constant need to be portrayed as an "expert" in an area despite not having the proper background in said area, technical or otherwise (e.g. claiming to be a machine learning expert or block-chain expert without having any experience in either of the fields whatsoever).

I came across several wantrepreneurs in my time as a student entrepreneur. I learned to recognize the aforementioned traits and avoid them as I became more experienced, but it was not easy and some very tough lessons were learned. From my experience, if you recognize a wantrepreneur, avoid them at all costs. You will be better off for it.

8.4.2 *Lesson 8: Push Instead of Pull (Your VCs)*

As a student, graduate or not, it is generally fairly easy to meet your professors, supervisors, and teaching assistants (TAs) when required. But as an entrepreneur, the same cannot be said for your VCs. The truth is, VCs are *extremely* busy, and your startup may be one of several they are investing in or spending time on. This means that you need to be aware that their time is important and to be effective with the time they are giving you.

One effective strategy that my co-founder and I used for Emotica, was that when we worked alongside our VCs, we ***pushed*** them instead of ***pulled*** them. What this means is the following: As students, we are typically trained to ask (or ***pull***) for feedback and time from our supervisors, teachers and others. But our approach with our VCs was to instead ***push*** them, by frequently providing succinct summaries of our tasks, and our areas of difficulty. The purpose of this was that when we were allotted time with our VCs, they would be ***pushed*** to be up to date to make better use of their time with us.

This is just one of many strategies one can use when working with VCs, but it ultimately boils down to this: their time is literally money, and it's worth more than yours. Use it effectively.

8.4.3 *Lesson 9: Hardware is Hard*

As mentioned earlier, I was exposed to the world of consumer hardware early in my entrepreneurial journey with Slate Scale. It was my moderately positive experience that (initially) influenced my somewhat naive lack of fear in pursuing hardware as a viable option for a startup. After all, in Slate Scale it was possible to do things remotely with a manufacturer, it can't be that much harder with a new device that uses multiple sensors, right?

The first thing we were told about Doppio in CDL by a VC, was that "hardware is hard". Then a second VC said the same thing, and a third, and then every VC began to echo this. At the time, I thought this was incorrect, and that it couldn't be that hard. But working closely with the VCs for Emotica, I learned that I had no idea about hardware at all in fact. Doing research projects with hardware is entirely different than building a consumer device.

For example, the supply chain required to make a new device, the testing rigours that it requires (e.g. FCC emission testing if you are using Bluetooth in a device, which I was planning on doing), the occurrences of defective devices and how those are supported and dealt with, along with providing tech support for a device. Overcoming these challenges also requires a significant capital investment (as mentioned earlier), making the barrier of entry already high without even determining the product-market fit.

Another substantial challenge with hardware is that iterations and pivoting are fundamentally harder to do. In software, it is generally simpler to pivot or iterate, as servers can be shut down and restarted, software redeployed, and interfaces changed. Of course, this depends on the scale of the software, but in the case of hardware, iterations are costly, and pivoting a physical design is also costly. This is why significant capital is spent in the research and development phases. Following those phases, the next challenge is to design your hardware in a manner that it can be scaled for manufacturing. For example, when the iPhone X from Apple was first released, there were supply chain issues regarding the screen that was being used, preventing their ability (at least initially) to scale manufacturing. Apple is experienced with these matters, while a novice entrepreneur, let alone a student may not be.

Hardware is also easy for potential competitors to copy. With most hardware being manufactured overseas due to cost, it means there will always be a chance your proprietary hardware design being copied and sold before you've even finished your development. While there are several tales of this happening on Kickstarter with hardware or physical products, it also practically depends on what you are doing. If it is a simple toy (e.g. a Fidget Cube [3].), then there is a high likelihood, especially if it has received some measure of success or press already.

While the aim of this lesson is to get the point across that hardware is hard, I (still) do not intend nor want to discourage its pursuit either. Hardware is hard, but not impossible. One of the strategies for Slate Scale to mitigate hardware risk — which I realized much later — was to use what already existed and modify it. This effectively meant finding off-the-shelf hardware that worked for our purposes and then asking for customizations. For Emotica,

3 Fidget Cube - www.antsylabs.com/products/fidget-cube

we also focused on building the backend, libraries and APIs for existing hardware and a platform first before even considering the hardware itself. This was also echoed by our VCs, as it mitigated the risk.

Hardware is hard, but if you're intent on doing it, make the risk minimal for yourself and your VCs. Build the data pipes for it (if it is required) such as APIs, then build the platform, and finally focus on the hardware.

8.4.4 *Lesson 10: Be Shameless, But Be Realistic*

One of the assumptions you make as an entrepreneur is that you're going to change the world, you're super motivated and you're going to make a difference, and make a profit doing so. However, the truth is most startups fail, and by most, I mean 99% of them. Very few of these become unicorns, a term often used for the billion dollar companies (e.g. Facebook, Twitter, etc.). As a student, I frequently disliked the notion that the entrepreneurial life was glamorous. It is not.

This brings me to the point of this lesson. Currently, there is a big push to make anyone and everyone an entrepreneur, but this needs to be grounded in reality. And this reality is that you need to balance being realistic with being shameless.

What I mean by mean by being shameless is that you need to fundamentally acknowledge that you don't really know much, especially as a student. Whether it is business plans, finances, marketing, hardware, or any of the other issues I've discussed in the lessons already, being shameless in admitting this is important. As a student, coming to this constant realization was the key to being able to better navigate entrepreneurship. Finding the balance between being realistic and being shameless was the key success factor for my co-founder and I in Emotica.

For Emotica, being realistic meant several things — some of which have been touched upon already. Working with our VCs, we addressed many of the areas we didn't understand. Unless you are willing to admit you don't know something, you will never have the opportunity to learn it. Our VCs also made us realize that some of our assumptions for our business were incorrect, and that it was OK to be wrong. For example, we assumed initially that the immersive content market could support a new device to measure emotions, but after several weeks of research, we learned this was not the case. Our VCs told us this initially, and had we listened (and realistic), we would have saved a significant amount of time.

We also were very pragmatic about how to receive investment for Emotica and how we dealt with VCs beyond those we were already working with. Much like *Lesson 7: Anyone Can Be An Advisor, But Not Everyone Can Give You The Advice You Need (8.4.1)*, the same notion can be applied when it comes to investors. There are those that talk about investment, and those that actually invest, and differentiating between the two is important. But before getting to that point, it is critical to be realistic in understanding if you need investment to begin with.

On the other hand, being shameless in Emotica, meant we always (continually) asked the difficult questions to our VCs such as, what would make them invest, what their concerns were in investing, and how do we alleviate those concerns. While being realistic means understanding our entrepreneurial limitations (as discussed earlier), being shameless means asking as many questions as possible to those who would listen.

One specific example of how we did this for Emotica was how we validated our concept. Over a period of 4 days, we compiled a list of nearly 500 people who were considered to be major influencers in AR/VR and immersive content. We then proceeded to email and tweet at 500 of these people with custom messages about the Emotica concept, and our backgrounds. Of the 500 emails that we sent out, we only received 4 responses back, or a

0.008% response rate. That being said, the 4 responses we received back were extremely valuable, such as one from NASA, all of which validated our idea and were ready to be signed up as alpha customers for our platform — with a bit of additional work of course. The ultimate lesson in this example for us, was to be shameless in how you reach to others for things like validation, because you never know who will reply and who will be interested. Sometimes getting no reply back is good because you've made an attempt in reaching out.

Last but not least, as a student, being realistic also means being aware of the entrepreneurial environment. It took me several years to realize this, but sometimes, an idea or venture is not in the right environment to succeed, not that the idea is wrong itself.

8.5 FINAL REFLECTIONS

If the impression this section has given you is that entrepreneurship as a student is difficult, then it has successfully done its job. Several very difficult lessons were learned over a period of few years, as I have hinted at continually throughout this section. To date, Doppio has been my favorite project on the research side of my dissertation, but seeing its death as a product concept was difficult to reconcile at first. Over time, as I had transitioned into Emotica and now a few years later — as I write this — I realize how naive I was in considering its commercialization. However, the journey to this realization was another fundamental step in my growth as an entrepreneur.

Part III

PROTOTYPING TOOLS FOR NOVICES

In the third part of this dissertation, I explore creating tools to enable non-technical users to learn and prototype interactions with technology. In the research sections, in both Chapter 9 and Chapter 10, I explore different domains in this context. In the entrepreneurial sections, I build upon the concepts and lessons from this research work, by creating a commercially available product which I reflect upon in Chapter 11.

9

MANNEQUETTE: UNDERSTANDING AND ENABLING COLLABORATION AND CREATIVITY ON AVANT-GARDE FASHION-TECH RUNWAYS

Drawing upon multiple disciplines, avant-garde fashion-tech teams push the boundaries between fashion and technology. Many are well trained in envisioning aesthetic qualities of garments, but few have formal training on designing and fabricating technologies themselves. We introduce *Mannequette*, a prototyping tool for fashion-tech garments that enables teams to experiment with interactive technologies at early stages of their design processes. *Mannequette* provides an abstraction of light-based outputs and sensor-based inputs for garments through a DJ mixer like interface that allows for dynamic changes and recording/playback of visual effects. The base of *Mannequette* can also be incorporated into the final garment, where it is then connected to the final components. We conducted an 8-week deployment study with eight design teams who created new garments for a runway show. Our results revealed *Mannequette* allowed teams to repeatedly consider new design and technical options early in their creative processes, and to communicate more effectively across disciplinary backgrounds.

9.1 INTRODUCTION

Avant-garde describes artistic, intellectual and cultural movements that are characterized by experimental, radical and unorthodox approaches (Gongini (2019)). The capacity in which something goes beyond the boundaries of convention through experimentation and proposes a new way in the creative arts – e.g. art, dance, fashion – is what makes it avant-garde (Gongini (2019)). In the fashion industry, avant-garde has impacted the way designers think and create work, especially around the form, shape and volume of outfits worn today. The fashion industry, along with avant-garde, influences culture and global trends (Torán et al. (2018)), drives social and economic change, and is a fundamental component of everyday life. Today, both are rapidly being transformed with modern technological innovations – such as factory robots which handle garment construction and manufacturing processes, algorithms which predict trends in style and even VR mirrors and installations (Liu et al. (2016);Vaccaro et al. (2018)). The expression of fashion (i.e. its *communication mediums*) can be divided into two categories: the *street* (or consumer market) and the *runway* (or catwalk) (Torán et al. (2018)). Avant-garde runways have incorporated numerous technologies (e.g. robotics, sensors, fiber optic fabrics) for several years, with designers frequently collaborating with technologists, and other disciplines, forming teams that create garments that are infused with technology. Throughout the collaboration and creation processes for a team creating avant-garde fashion-tech garment – *from sketch to runway* – designers, technologists and others, often draw upon a number of different disciplines, such as fabrication, electronics, and

***Note: The text in this chapter appears in the following publication:**
Teddy Seyed, Anthony Tang (2019). Mannequette: Understanding and Enabling Collaboration and Creativity on Avant-garde Fashion-Tech Runways. In *Proceedings of the 2019 Designing Interactive Systems Conference (DIS '19),*, ACM, New York, NY, USA.
🏆 *Best Paper Honorable Mention*

programming (among many others). Several tools can also be used to help novices in learning these disciplines that are essential for prototyping and creating avant-garde fashion-tech garments (adafruit (2012);Wang et al. (2017)). However, these tools also present numerous challenges within creative and collaborative processes of multi-disciplinary teams that create avant-garde fashion-tech garments which appear on runways. Within these teams, designers, technologists (and others) often operate with unique skill sets, knowledge, and vocabularies making collaboration sometimes difficult. Thus, the problem is that these collaborators do not have a common medium to express creative ideas, experiment with sometimes complex technology (e.g. sensors, lights, augmented reality) and communicate with one another about challenges and limitations, particularly in the early stages of constructing an avant-garde fashion-tech garment.

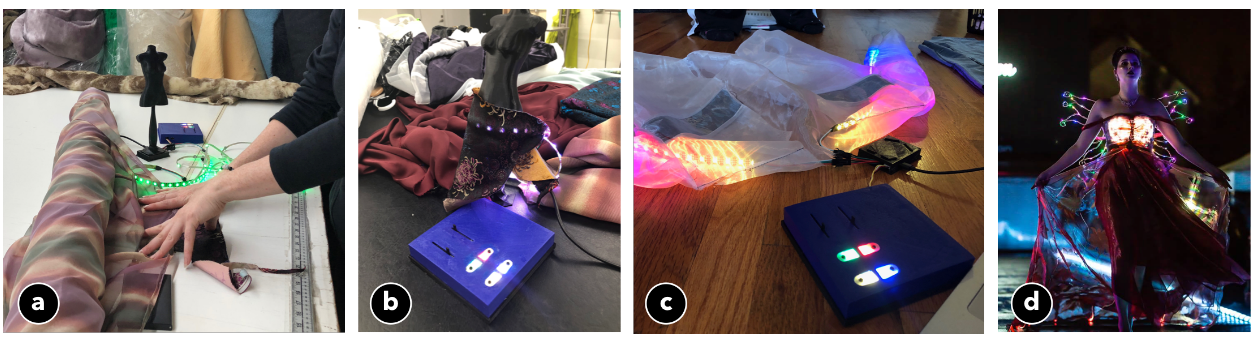


Figure 40: The different usage scenarios of Mannequette. **(a)** In-situ, iterating and deciding upon fabrics at a market; **(b)** Prototyping light (LED) patterns and sensors on a miniature dress form with fabrics by using the mixer; **(c)** Integrating a previously prototyped and now completed pattern and sensor interactions into an assembled garment; **(d)** A completed garment created using Mannequette on a runway.

Our approach is to support facilitated communication and prototyping between team members (e.g. fashion designers, technologists) by the introduction of the *Mannequette* (Figure 40). Mannequette is a modular, miniaturized mannequin system, designed to help cross-disciplinary teams in prototyping and experimenting with technology – specifically lights and sensors – from the onset of the garment construction process. With Mannequette, a team can engage in low-fidelity prototyping of sensors and lighting patterns using a simple DJ-mixer like control panel, without the immediate reliance of a technologist or technical skills. A technologist can also adjust the sensors and light patterns created by a fashion designer (or other team members) with the control panel through programming, for demonstrating more complex interactions and effects. Thus, Mannequette serves as a communication medium between team members, as it can be used to create and illustrate temporal effects that may otherwise be difficult to communicate.

To evaluate how avant-garde fashion-tech teams appropriate and use Mannequette in their design process, we conducted a deployment study with Mannequette in the context of an avant-garde fashion-tech runway show. Our research questions focused on (1) *how teams would use it in their early and more tangible processes*; (2) *how teams would use it to translate effects into more concrete concepts*; (3) *how such a tool would impact the communication between cross disciplinary team members*; and (4) *what benefits, and challenges cross-disciplinary teams would find in such a system for their processes*. Our deployment was conducted with eight teams for 8 weeks. Our findings revealed that teams found Mannequette valuable and used it as a means of facilitating and supporting communication very early in the process of constructing avant-garde fashion-tech garments. They also used it to establish new prototyping and communication routines within their team and even with those in other environments who were not formally part of building the garment. For example, we observed fashion designers using the portable version in fabric stores where with the help of store employees, they quickly iterated and decided upon fabrics for their garment based on different temporal effects and lights (in-situ) without the immediate reliance of a technologist.

In summary, our contributions are: (1) the Mannequette tool that facilitates the prototyping and collaborative processes of the design and creation of Avant-garde fashion-tech garments; (2) findings demonstrating the effectiveness of Mannequette through an 8 week long deployment study that culminated in an avant-garde fashion-tech runway show featuring garments which used Mannequette; and (3) a discussion of how avant-garde fashion-tech can inform the design of wearable (and other) toolkits.

9.2 RELATED WORK

In the context of designing and creating tools to facilitate and support the process of constructing of avant-garde (and fashion-tech) garments, we drew upon two primary research areas to inform our design: (i) fashion technology industry and research, (ii) programming tools and construction kits for wearables. We also position our contributions within the space of wearables, fashion tech and existing tools.

9.2.1 *Fashion and Technology*

The fusion of fashion and technology has transformed garment creation and fabrication processes (Dabby et al. (2017);Posch and Fitzpatrick (2018)), now making it a more critical component and influencer of the consumer wearables industry, and vice versa. One major area in fashion seeing this influence is e-textiles, which incorporate electronics that are woven into the fabric and has a number of applications (Holleis et al. (2008);Karrer et al. (2011);Kim et al. (2009);Lepinski and Vertegaal (2011);Posch and Fitzpatrick (2018);Poupyrev et al. (2016)). Technologies such as 3D printing sensors, actuators and other components have also begun being incorporated into clothing (Pece et al. (2017);Poupyrev et al. (2016);Rivera et al. (2017);Roinesalo et al. (2017)) and even on the skin (Dierk et al. (2017);Lo et al.

(2016);Nittala et al. (2018);Wang et al. (2017);Weigel et al. (2017)). In this context however, several fundamental challenges still exist, such as the need for technology to be lightweight or finding sufficient and long-lasting sources of power. These challenges and the ability to solve them creatively, rely on a cross pollination of skills between design, engineering and other disciplines. In the world of avant-garde and fashion tech, the experiences and lessons learned in creating garments can lead to or inspire more innovative wearable solutions to these challenges.

In the broader context of HCI research, prior work has examined several application areas in fashion using technology, such as in-store environments (Cremonesi et al. (2016)), while other researchers have focused on the relationships between fashion and clothing (Juhlin et al. (2013)), understanding the design of electronics with fashion lenses (Juhlin and Zhang (2011)) and design principles from fashion itself (Pan and Blevis (2014)). (Pan and Stolterman (2015)) described scenarios for HCI driven by fashion from different perspectives, such external fashion concepts influencing the field, as well as fashion concepts that go in and out of fashion within the field itself. Okerlund et al. (2018) ran an interdisciplinary maker fashion show at a university campus as a lens to study empowering a campus Maker community. We build upon the approach of using a fashion show as the context for conducting research, as well as the work of Pan and Stolterman, but focus on the underexplored avant-garde fashion-tech area and its cross disciplinary collaborations.

9.2.2 *Toolkits for Designing Wearables*

A large number of electronics and wearable construction kits, such as the Adafruit Flora (adafruit (2012);noa (2019b)), the LilyPad (Buechley et al. (2008);Buechley and Hill (2010)), the BBC Micro:bit (Ball et al. (2016)) and other Arduino-based electronics kits (noa (2019j)) have opened up a new space for children and adult hobbyists to begin exploring and building new kinds of wearables and integrate electronics into clothing. However, many of these toolkits and electronics still have high barriers of entry, requiring knowledge in areas such as electronic circuits, an understanding of lower-level digital and analog I/O (especially if sensors for interactivity are used) as well as programming. Several toolkits have been designed to overcome these issues especially with regards to programming and electronics (noa (2019j)). However, only a limited subset is aimed towards avant-garde fashion-tech, and even less are appropriate for the constraints that come with designing for and building garments that are worn a runway. For example, using conductive threads to link components, a common approach for several toolkits (e.g. Buechley et al. (2008)) is not ideal due to: (i) movement of garments on a runway; (ii) the wear and tear of a garment over its lifetime as it travels to different runway shows, while also being worn by different models.

One approach to address the difficulties in programming, electronics and runway constraints is to consider tangible and modular approaches. MakerWear is an example of using a tangible and modular approach for wearables (Kazemitabaar et al. (2017)), similar to littleBits and others (Ball et al. (2016);Bdeir and Ullrich (2011);noa (2019f);noa (2019a)). While MakerWear is targeted towards building wearables for children and demonstrated value in enabling children to begin working with basic programming and sensors for wearables, its focus and design is not targeted towards fashion designers, technologists (i.e. avant-garde fashion-tech teams) or the runway environment. We build upon this and prior work in tangible construction kits (adafruit (2012);Buechley and Hill (2010);Kazemitabaar et al. (2017);noa (2019a)), where the construction experience (both user input and output) for Mannequette is tangible, serving as a means of communication, creative expression and experimentation without many barriers at the onset of constructing an avant-garde fashion-tech garment.

9.3 DESIGN CONTEXT

To ground our research within the context of real-life practices of avant-garde fashion-tech runways, we collaborated with a fashion-tech collective (MakeFashion (noa (2019i))) that produces multiple international avant-garde runway shows per year. We built our relationship with MakeFashion over a 2.5-year period through 7 separate runway shows in Canada, USA, Germany, Ireland and China. Each show contained 8-15 design teams, where each team created up to three avant-garde garments per runway show. Design teams were typically comprised of at least one self-identified fashion designer, and one self- titled technologist. The construction of teams (and their selfidentification with titles) are specific to the context of MakeFashion; thus, the bifurcation of technologists and fashion designers does not necessarily reflect the broader community of avant-garde fashion-tech, many of whom have cross-disciplinary skills. For the purpose of this paper, we use these labels to describe and reflect the people we studied. We return to the issue of generalizing beyond this community later in our discussion.

To understand the challenges and process of avant-garde fashion-tech and runways, we embedded ourselves in three key roles for runway shows: first as a *technologist*, then as a *fashion designer*, and finally as a *producer* and *director*. Thus, the lead author used a participant observer process in different roles within this organization to understand the practical problems faced by members of each group. As a trained computer scientist, the lead author began involvement in the organization as a technologist within a design team. After three runway shows in this role, the lead author then organized a new design team, and played the role of a fashion designer on the team for another three runway shows. In total, the involvement as a technologist and designer resulted in 14 garments, and interaction with 30 other designers and technologists. Since then, the lead author has been a producer and technical director for MakeFashion, managing and meeting with teams over the course of 4 months prior to the runway show. As a consequence, the lead author has worked with over 20 teams, producing over 40 avant-garde fashion-tech garments.

Next, within the context of MakeFashion, we describe in detail, our observations of common practices in avant-garde fashion-tech garment creation.

9.3.1 *Observed Practices of Avant-garde Fashion-Tech*

Based on our observations and experiences over the 2.5-year period, we saw three distinct phases based on increasing technical complexity and involvement of a technologist. This process is briefly summarized in Figure 41, where each of the lettered phases circumscribes different processes that are commonly happening within a cross-disciplinary team that typically consists of at least one self-identified fashion designer and one self-identified technologist.

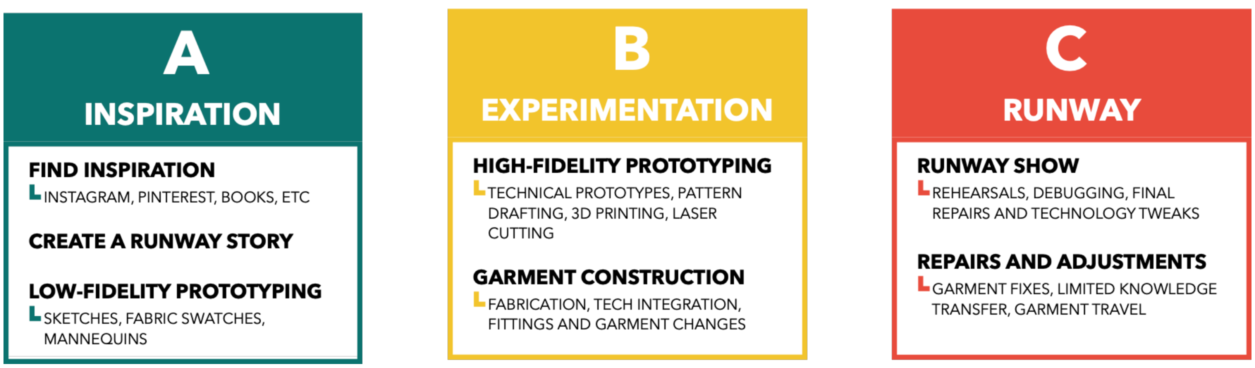


Figure 41: Our observed practices of avant-garde fashion-tech.

Phase A: Inspiration: Before creating a garment, a team (typically the fashion designer) explores different sources of *inspiration* — Instagram, Pinterest, magazines, pattern drafting books, music—with the goal of creating a story that describes the garment on a runway. Story plays a critical role for avantgarde garments on a runway, as it is ultimately translated into the design of the garment, the technology used to support it, the music chosen for the runway, as well as how the model walks and poses on the runway. Furthermore, while fashion has a long history and set of materials to draw upon for inspiration, avant-garde fashion-tech is a relatively newer space, thus fashion designers and teams are more limited in examples and materials they can draw upon for inspiration.

Once a concept has been settled upon, *low-fidelity experimentation* is used to explore possibilities for implementing the idea. Tangible and physical techniques are also used to prototype different elements of a garment, such as testing different fabrics and materials to understand how well lights diffuse them, color combination of fabrics and the draping of fabrics on a mannequin. Fashion designers typically create sketches for the concept of the garment, which is occasionally combined with a limited description of what is expected from technology.

This phase is typically characterized by limited involvement from a technologist, and any progress is revisited once the technologist becomes more deeply involved. In this phase, limited technical engagement creates two challenges: (i), since sketches and creative ideas are initially developed without a clear understanding of technical limitations, ideas often need to be scaled back drastically later in the design or construction process; (ii), since designers are rarely well versed in how or what technologies could be deployed, their ability to communicate is limited, meaning communication opportunities are lost.

Phase B: Experimentation: Once a garment (story and concept) has been defined, a fashion designer and a technologist begin using high-fidelity techniques and technical "prototypes" are also built — often independently. Teams typically face challenges when trying to integrate these prototypes, owing to situations where intentions of the design were unclear, or limitations of the technology were not clearly communicated earlier.

Here, fashion designers develop multiple iterations and prototypes until they are satisfied with the outcome of their pattern and chosen fabrics. They begin with sketches, and draft with patterns appropriate to the design. Those with access to fabrication skills (i.e. laser cutting and 3D printing) use them to speed up or augment different parts of their pattern drafting process (e.g. using laser cutting to quickly prototype modifications to a pattern).

In parallel, technologists' experiment with different electronics and components (e.g. motors, haptics, lights, sensors), create programs, and occasionally design new electronics entirely for a garment. For example, with a light strip, they may explore different light patterns, their timing or colors. Typically, a technologist's role does not have a large contribution to the design itself, and thus it is extremely common for the technologist to misunderstand an idea articulated by the fashion designer. Thus, the technologist might develop the technical prototype in ways that are unexpected or unwanted by fashion designers. Once the designer and technologist are respectively satisfied with their choices, they begin *garment construction* and integration. Teams run into significant problems in this step, many of which are due to poor communication practices, and some due to practical problems of deployment. Typically, the technology integration (i.e. microcontrollers, wiring, sensors) occurs after the garment has been constructed, which has negative consequences. For example, many teams fail to consider where sensors should be placed (or sewn into the fabric), or where (or how many) batteries must be placed on the body. This means that garments require additional work, such as constructing pockets for batteries or sensors, potentially designing new 3D printed and sewable casings for microcontrollers or building housing for extremely messy wiring. This problem is even more difficult for novice teams of fashion designers and technologists, who have limited experience designing for wearability on the runway— sometimes, their early fixes to the garment are inadequate (e.g. gluing LED pixels vs. building an LED layer), meaning they make harmful compromises to the garment itself.

Phase C: Runway: An entirely new set of challenges present themselves when the garment is worn by a model in preparation for a *runway show*. While the garment may be functional, teams can encounter problems during rehearsals. Here, practical challenges of wearability present themselves again—for instance, was the fabric chosen for the design flexible enough to accommodate the model's walking movement or poses, is there enough slack in the wiring for electrical signals to pass through during choreographed movement (as may be required by the story), and does sweat from the model damage or cause problems for the technology used?

We frequently observed that when a garment was involved in dance or other choreography, electronics were at risk. These issues were typically uncovered during rehearsals. For example, wires may have broken or stretched underneath a casing, or the movement programmed into accelerometer to cause an effect, is too sensitive for the walk of a model. This meant that designs adjustments and technical adjustments needed to be debugged and made on the fly. This was especially challenging if the design of the garment did not account for the ability to make necessary changes when something breaks down, such as a wire or sensor. For novice teams of fashion designers and technologists, these issues were magnified because of the late integration of technology, with consequences now including the removability and reparability of the garment. Overall, many teams also do not have prior experience in understanding potential runway show issues for avant-garde fashion-tech and their solutions.

Following a runway show, a garment typically requires *repairs and adjustments* which might again require taking apart the piece. Furthermore, garments commonly travel between shows and are worn by different models, presenting additional challenges. For instance, garments often need to be put on a model in a certain way or in a certain order (e.g. some pieces have multiple layers) and transferring this knowledge to others who do have not the necessary background or experience with garments incorporating technology is difficult. Additionally, technical knowledge of avant-garde fashion-tech garments is not well maintained or documented. This becomes a problem when a garment requires repairs, and its creators have not travelled with it. This also creates several issues around communication between different groups of technical and fashion expertise and can substantially hinder the success of the garment in the highly-competitive avant-garde fashion world.

9.3.2 *The Challenges of Avant-Garde Fashion-Tech*

Traditional fashion runway shows are messy and hectic by their very nature and by combining avant-garde culture with technology to the mix, a variety of challenges have been created. We distill many of these challenges we observed previously into three themes.

Limited Technology Literacy. A wide range of technology – from sensors, to augmented reality, to artificial intelligence – is available for teams to create garments. However, because some members of the team may have limited technology awareness or skills, it is difficult for them to come up with concepts that use technology. Also, the gap in technology literacy between a designer and technologist makes it difficult for them to communicate concepts in an effective manner. For example, communicating different technical choices to a fashion designer can be difficult, particularly when they may not understand the alternatives and their implications on the design.

Inadequate Vocabulary for Creative Expression. Avant-garde fashion is an extremely expressive form, with fabrics, colors and other aspects serving as the fundamental language to tell the story of a garment. For avant-garde fashion-tech, creative expression and language comes in many forms due to technology (e.g. robotics, VR, lights, sensors). However, for fashion designers (especially those that are novice), the ability to experiment with different languages and expressions through technology is difficult, particularly when there is limited technical literacy. Designers often rely heavily on technologists to assist in creative expression (e.g. creating custom electronics or light), emphasizing the importance of facilitated communication tools. For technologists, this also means there is a lack of contribution to design choices for the garment itself, as their involvement in the construction process typically occurs much later.

Balancing Wearability and Troubleshooting. Building garments for runway conditions is difficult, particularly when garments are also worn by different models in its lifetime. This makes designing for wearability extremely important when considering the types of sensors, fabrics, lights and batteries that can be used in constructing a garment. However, a balance needs to be struck between the wearability and the ability to fix the garment, as they break down and need repair overtime. Because technology is incorporated later in the design process, these considerations are not thoroughly accounted for by the designer or technologist in the early stages, resulting in garments that are wearable but not repairable (or vice versa).

9.4 MANNEQUETTE

Currently, wearable kits support a wide range of sensors and outputs, but they do not necessarily support or properly consider the range of described activities in avant-garde fashion-tech, nor the audience: multidisciplinary teams consisting of fashion designers, technologists and others. We envisioned designing a tool built around the processes we observed while also building upon prior research (Buechley and Hill (2010);Devendorf et al. (2016);Dierk et al. (2018);Kazemitabaar et al. (2017)). Thus, we designed a miniature mannequin-based tool called *Mannequette*, with the name referring to mannequins (of all sizes) that fashion designers build around, combined with a marionette which is controllable by a puppeteer. Thus, Mannequette is a mannequin with a controllable piece.

Mannequette allows teams (especially fashion designers) to test, experiment and quickly iterate upon light patterns, interactions and fabric techniques (e.g. diffusion testing with light or dark fabrics) in a rapid and ad-hoc manner without technical knowledge, at the onset of the construction process. Certainly, the space of availability of technologies for creative expression in avant-garde fashion-tech is large and while examples of avant-garde fashion-tech garments do exist using different methods of expression (such as Berglund et al. (2018)), we specifically chose to focus our tool on *sensors* as the means of input, and *lights* as the primary means of output for garments. This is because in the 2.5 years of our work with MakeFashion, lights and sensors were the most common form of input and output chosen by design teams for their garments shown on their runways, due to their visual nature. Additionally, teams working with illuminated materials (and their interactions) contributed to some of the challenges we observed. Mannequette also facilitates communication and creativity between the team members through a shared and tangible medium of communication. Next, we describe our design principles for tools within the avant-garde fashion-tech space and our Mannequette system, and how it works.

9.4.1 *Design Principles and Goals*

Informed by prior informal interviews through our relationship with MakeFashion, as well as our experiences with avant-garde fashion-tech runways, and their associated technologies (including (adafruit (2012);Buechley et al. (2008);Buechley and Hill (2010);noa (2019b))), as well as relevant prior work (e.g.,(Kazemitabaar et al. (2017);Resnick and Silverman (2005);Vossoughi and Bevan (2014);Wyld and Dierking (2015))), we synthesized the following key goals for a tool designed to facilitate communication and creativity between multi-disciplinary teams:

Leverage existing models and processes. While several toolkits for wearables exist and provide a rich number of components (Ball et al. (2016);noa (2019b);noa (2019a)), many don't consider the creative and construction processes within avant-garde (for both designers and technologists) or runway environments. In contrast, we aim to focus our tool on leveraging specific processes that exist within avant-garde and fashion – for example, a tool could be designed around a mannequin (and the practice of maquette, which plays a role in the areas of ideation, prototyping and discussion in art and fashion.

Tinkerability. Avant-garde fashion-tech teams rely on their combined team skills, supplemented with additional skills from others (e.g. craft, fabrication, design, programming) to create a garment. Similar to wearable toolkits for children, emphasis should be placed on allowing quick tinkering and prototyping regardless of available skills on a team (Vossoughi and Bevan (2014);Wyld and Dierking (2015)), especially in the areas of lighting (LED) patterns and sensors. Tinkering occurs heavily in the early phases of the garment construction process we observed (e.g. when choosing fabrics), so tools must be able to accommodate for a wide range of tinkering tasks within the construction process.

Low floors, high ceilings, wide walls. Building upon design cues from (Kazemitabaar et al. (2017);Resnick and Silverman (2005)), tools must support teams in the creation of increasingly complex designs as they gain experience. Not all avant-garde fashion-tech teams have technical experience in relation to garment construction, so accommodating differing levels of skills with multi-disciplinary teams is important in designing a tool.

Augmenting avant-garde. Avant-garde fashion-tech is heavily focused on aesthetics, story and culture, much more than traditional wearables which focus on practicality. Given the highly experimental nature of Avant-garde, we aim to support and augment as wide a variety of designs as possible that emphasize look (and occasionally simplicity) while also allowing for creative exploration and incorporation of sensors and lighting in a meaningful way.

Modularity. Modular wearable kits and systems have proven to be successful in the past (Kazemitabaar et al. (2017)). For the processes we observed, where dress forms, fabrics, and tools are continually swapped, building upon these principles can provide value. Fashion designers (and technologists) should be able to quickly swap between components (i.e. sensors and lights) and interactions. Additionally, using a modular approach from a fashion and technical perspective, allows a team to collaboratively create garments in a more systematic manner (e.g. a dedicated and removeable technology layer, removeable casing for wiring). This also helps with adjustment or repairs for a garment if and when technology breaks down in the garment (e.g. loose wiring, or an electrical short), compared to unsewing or entirely redoing parts of a garment. Furthermore, for runway conditions when the environment is stressful for a design team, a modular system can enable quicker and easier debugging.

9.4.2 *Mannequette System*

Based upon our design goals, we created Mannequette (Figure 42), which is comprised of three parts: (iii) a custom I^2C-based DJ mixer-like interface that allows designers and technologists to dynamically prototype behaviors with (virtual) sensors and light patterns, which can then be integrated and used by a proper physical sensor in the final garment; (ii) a tech-infused base that supports a plug and play system of grove-based I^2C sensors which are used to create interactive behaviors; (iii) a swappable miniaturized dress forms. The dress forms are 3D printed and scaled from open source models, and minimal modification is required for additional 3D printed dress form models.

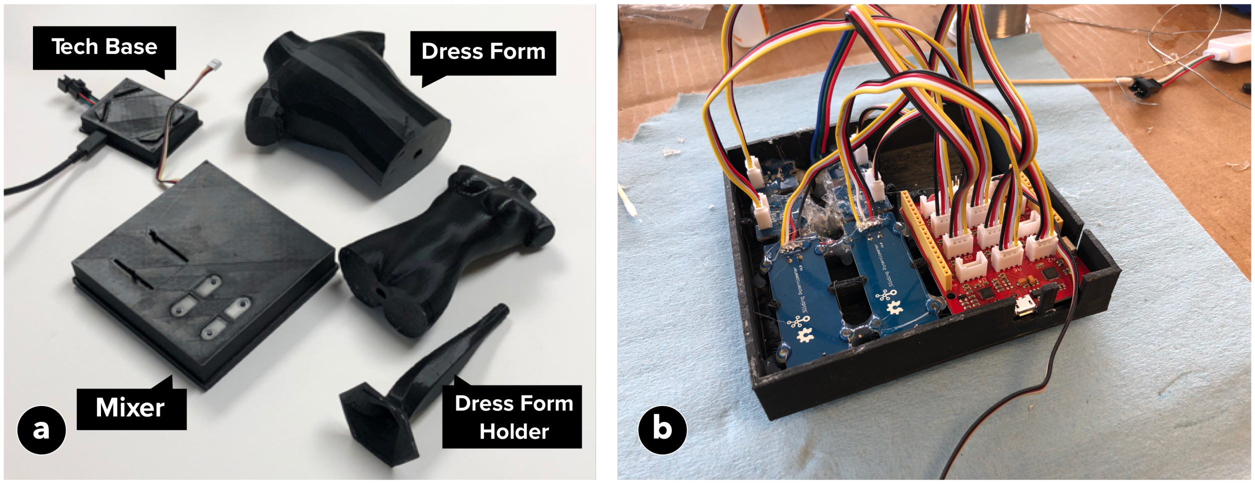


Figure 42: (a) The components of Mannequette; (b) Seeeduino based internals of the mixer control panel.

Mixer. The *mixer control panel* for Mannequette serves as the hub for teams to communicate and quickly experiment with a number of light patterns and interactions. Inside is a Seeeduino Lotus, 4 Seeed Grove I^2C-based buttons and 2 Seeed Grove I^2C-based potentiometers, enclosed in a custom-designed 3D-printed case (Figure 3b). Each button and slider is mapped to specific functionality regarding a pattern and interaction: saving, recording, play back, and loads/saves/clear, as well as adjusting brightness and sensor values (Figure 43). The control panel connects to the base using an I^2C cable, and a custom I^2C communication protocol that was written to enable customization of patterns and interactions that are saved/loaded/cleared onto the tech base. We initially explored using a mobile application, as well as a WIFI or Bluetooth-based approaches (e.g. a mobile phone app) but we aimed to support the full set of activities described earlier, particularly runway environments where debugging is critical and Bluetooth and WIFI does not work sufficiently – due to changing or noisy environments– based on our experiences.

Tech Base and Dress Forms. The *tech base* consists of a customized ATMega32U4-based Arduino board inside a custom 3D printed casing, with a slot on its top where the *dress form*

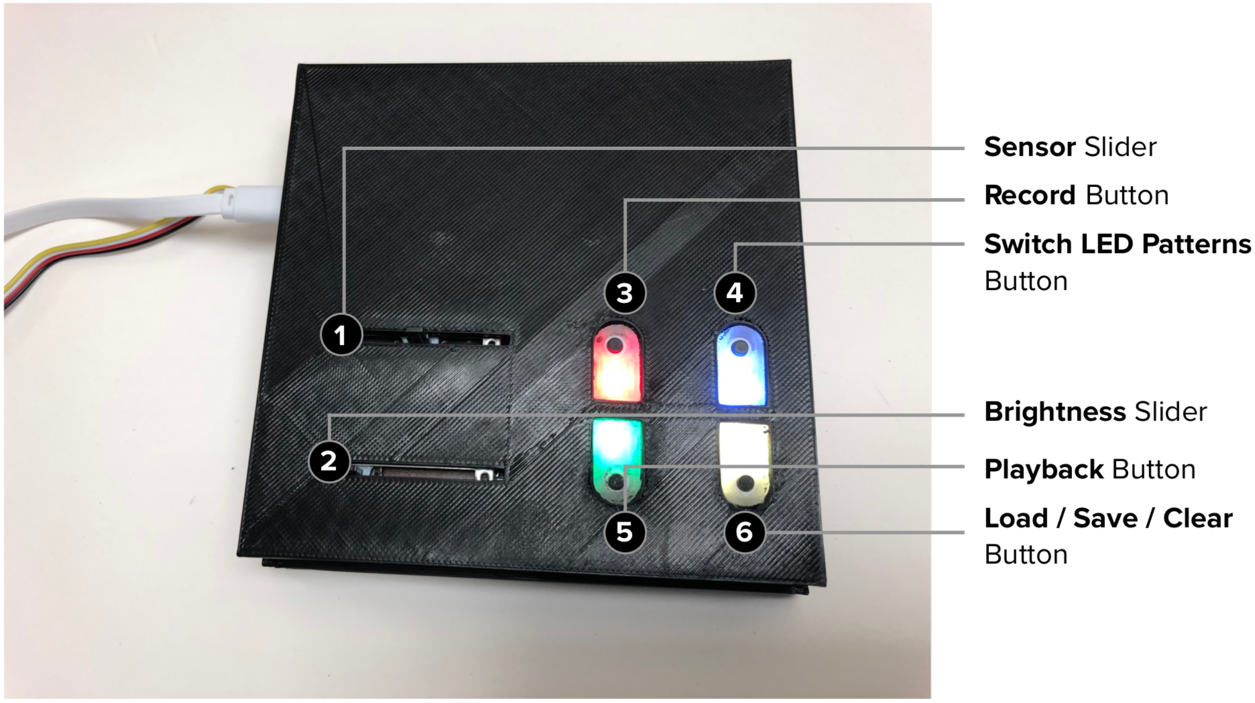


Figure 43: The mixer control panel's button configuration.

holder snaps into place, with *swappable dress forms*. The tech base supports 2 outputs for WS2812b (or similar) LED strips and 1 I^2C port, which the mixer uses to program the tech base and is also used for the sensors. The information coded by the mixer (e.g. light patterns and sensors) is stored on the tech base and saved in memory, allowing for simple modifications to the light patterns or sensor values already stored. The base can also be integrated into the final garment when prototyping is completed.

Supported Sensors. Selecting and abstracting a set of sensors is crucial for enabling teams to create garments. In addition to supporting standard modules used by makers who create wearables (e.g., WS2812b LED strips for lighting), we also focused on supporting sensors that could quickly assist in interactivity for the runway environment (noa (2019c)). A runway environment constrains the effectiveness of certain sensors (e.g. Light or Sound) and communication technologies (e.g. Bluetooth or WIFI) for garments because as they travel, these conditions cannot be guaranteed, nor is it guaranteed that the team handling a garment at a show will be the same as the original design team. Thus, we emphasized sensors and interactions we knew have worked previously in runway conditions, by examining 5 years' worth of garments from MakeFashion teams, as well as informal interviews. We supported: *body movement* (proximity, motion, accelerometer), *environment* (light, sound, temperature, heart) and *input* (touch, button, potentiometer and rotation) to start. Our intention was not to immediately explore all sensor possibilities, but to instead begin with those frequently used or requested by teams in MakeFashion.

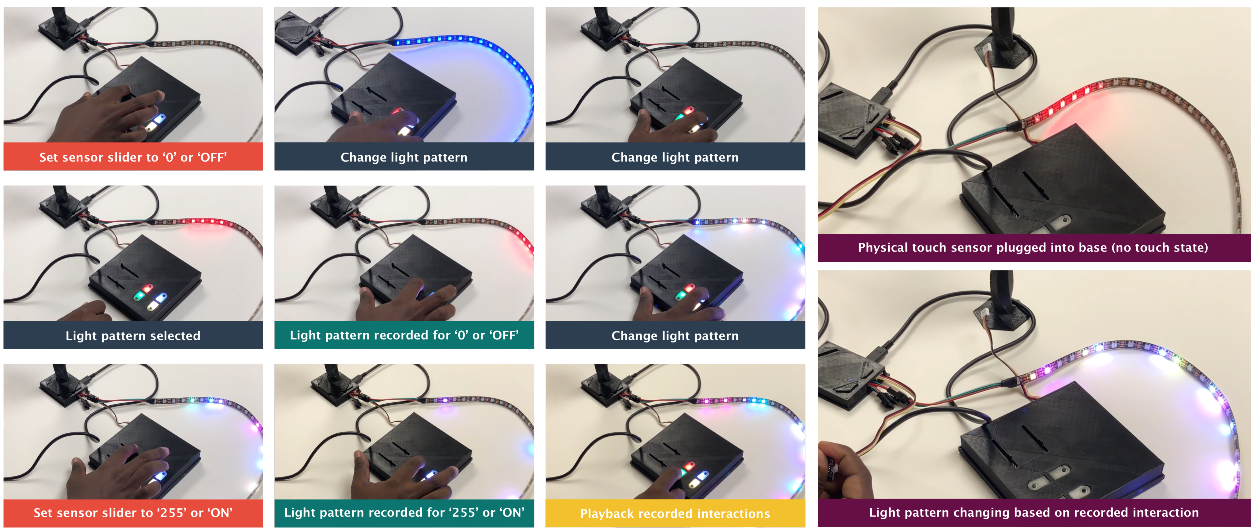


Figure 44: An example of how to prototype and finalize a touch sensor interaction with lights using Mannequette.

Prototyping and Finalizing Interactions. Creating different sensor and light interactions generally occurs by using the mixer (see Figure 44). First, a team chooses a predefined light pattern from the mixer. We provided 23 predefined (but customizable) light patterns chosen based on our prior examination of garments created in MakeFashion. The brightness of light patterns can be adjusted by using the brightness slider on the mixer. After the team (or specifically the fashion designer) has completed experimenting with different light patterns, they configure different sensor interactions. Sensor interactions are mapped to the sensor slider where the values of sensors are mapped to a value in a range. For example, a button (or touch) interaction is mapped to *'off' (or 0)* on the slider, while *'on' (or 100)* is mapped to maximum on the slider. Similarly, for sensors such as light, sound or proximity, ranges can be created for *low, medium, and high.* For each value (or range) on the slider, an associated light pattern can be recorded and played back. If the team does not like what has been recorded, it can be cleared, or they can save their creation for further (or final) modifications on a garment. The mixer supports 5 ranges.

After a design team is satisfied with their prototyped interaction and it has been saved, the light patterns can be further customized, and a supported sensor can be used, as well as tweaked. To modify a light pattern, the tech base is plugged into a computer running Arduino, and a pre-defined region of Arduino code for its color and other properties (e.g. speed, delay, number of pixels) is modified. Each of the 23 light patterns we provide are modifiable with their own properties. Additionally, the light patterns are templated in a manner such that additional light patterns can be added to the Mannequette system by a technologist (or designer) if they chose. To use a sensor with the interaction that was recorded and saved, the associated sensor is plugged into the I^2C port of the tech base. As the sensor values were mapped to the interaction slider, if the sensor values need to be tweaked (e.g. volume level or distance), a predefined section of provided Arduino code can be adjusted by a technologist (or fashion designer) to tweak these recorded values.

9.5 DEPLOYMENT STUDY

To gain insight into *how* multi-disciplinary teams consisting of (self-identified) fashion designers and technologists can use, understand and communicate with Mannequette throughout the process of creating an avant-garde fashion-tech garment, we conducted an 8-week deployment study.

9.5.1 *Method*

We recruited eight teams through our partnership with MakeFashion, which hosted an avant-garde fashion-tech runaway show in Fall 2018. Each team was self-selected and varied in experience with avant-garde fashion-tech (both garment and technical), as well as the number of (self-identified) technologists/designers for the team. We also asked each team their perceived level of expertise with technology in fashion, with respect to areas such as 3D printing, laser cutting, electronics and programming. They were asked to rank this in terms of *limited, medium* and *strong experience*. Figure 45 describes these teams in detail.

Team	Composition	Experience	Prior
1	2 designers (both female), both technologist	Limited	10
2	3 designers (all female), 0 technologists	None	4
3	2 designers (all female), 0 technologists	None	4
4	2 designers (both female), both technologists	Medium	30
5	1 designer (female), 1 technologist (male)	Limited	10
6	1 designer (male, also a technologist)	Medium	10
7	1 designer (female), 0 technologists	Limited	2
8	1 designer (female), 1 technologist (male)	Strong	10

Figure 45: A breakdown of each team and their experience.
I

9.5.2 *Procedure*

Over the course of the 8-week period, we engaged our design teams in bi-weekly interviews and discussions, tracking their progress and processes of constructing their garments. We also asked teams to document their creation process. When teams completed their garments and felt they were runway ready (i.e. ready to be worn, technology had been integrated/tweaked and ready for dress rehearsal), we ran a final interview. We did not track team progress beyond the first appearance of garments on the runway.

We deployed a Mannequette to each of these teams following their selection to feature in their annual runway show. We provided each team with the Mannequette system, a 5cm strip of WS2812b LEDs and one USB-C cable for the tech-base. Teams were given 10 tutorial videos for Mannequette that covered topics including assembly, how to use the mixer and how to modify pre-defined code for sensors and lights. Finally, teams were provided sensors and additional LED lighting strips by MakeFashion when needed for experimentation, as well as for their final garments.

9.5.3 *Data and Analysis*

We employed a mixed-methods approach to assess the effectiveness of Mannequette, as well as the challenges teams faced when constructing their garments using Mannequette. The first interview with design teams collected information about prior experience in avant-garde fashion-tech, and the last collected their overall experience, and how well they felt they accomplished their design (and technical) goals with Mannequette. All interviews were captured and transcribed, and we used a thematic analysis approach to analyze our interview data.

9.5.4 *Findings*

In total, the eight design teams created a total of 15 garments using Mannequette. Figure 46 also provides a brief breakdown of the garments each team constructed and the types of interactions (i.e. sensors) used in the garments.

Team	Made	Garment Description
1	2	1 used a **temperature sensor**, 1 used a **heartbeat sensor**
2	2	1 used a **motion sensor,** 1 used a **touch sensor**
3	1	1 used a **touch sensor**
4	2	2 used a **touch sensor**
5	5	3 used a **motion sensor,** 2 used **no sensor**
6	1	1 used a **proximity sensor**
7	1	1 used a **touch sensor**
8	1	1 used a **magnetic sensor**

Figure 46: A breakdown of the garments made for each team.

We first frame our findings in the context of the process described earlier (and MakeFashion) and discuss themes and common patterns we observed with teams' use of the Mannequette. To illustrate these themes, we highlight individual examples, though stress that these examples are not outliers; rather, they represent common patterns of Mannequette uses including: (i) how and what types of garments were constructed; (ii) the changes in design processes for teams; and (iii) how different teams progressed in their understanding, use and modification of Mannequette.

Inspiration with Mannequette. All teams immediately began incorporating Mannequette in their design processes in a similar manner to traditional mannequins. Several teams also introduced the Mannequette into unexpected environments, such as the fabric market, which allowed them to begin thinking about the impact of the electronics on their final garment designs earlier. As articulated by a member of Team 2: "*...we took the control panel and base with us to the fabric market when we went shopping for fabrics. Since it's small, it fit in my purse, and it was useful for us to try out so many different fabrics suggested by employees in the market with different light patterns immediately. We changed our design because of this, especially the diffusing fabric!*"

This early engagement with the light patterns allowed teams to be even more active in generating and considering new ideas. For instance, we frequently observed teams (especially fashion designers) using the mixer and base to test different fabrics and materials with the provided light patterns. Similarly, because fashion designers were able to quickly iterate through fabrics and wanted to experiment with more complex light patterns at an earlier stage, technologists became involved very early in the design phase, modifying light patterns immediately. A technologist from Team 5 described: "*...in a sense, I am also a designer now too since I can see how things work with the fabrics and make suggestions and modifications right away.*"

A new communication practice enabled by Mannequette was the ad hoc creation of videos to illustrate temporal effects. Fashion designers would create video recordings of patterns to communicate ideas about more complex temporal patterns to technologists. These could be visualized with the Mannequette and articulated in-place with the tool, rather than being described verbally or through paper sketches. Some teams even shared their patterns with other teams who were not using Mannequette but were also creating garments for the MakeFashion show. Thus, these videos of the Mannequette acted as inspiration for other teams. A Team 3 member: "*...I thought the pattern I made was very pretty and something* *that [another] team could modify, since they make interesting things and it would make theirs so much cooler*."

We also observed teams draping fabrics and components of using the dress form of the Mannequette. This allowed them to see how the fabrics would interact with sensors and LEDs early in the design process, as well as plan for the placement of components and batteries. Unlike a traditional mannequin, however, the Mannequette did not allow teams to pin fabrics. A designer from Team 3 stated: "*...the miniature size of the dress forms is great, but we can't pin to it directly which makes it a bit challenging sometimes*."

Experimenting with Mannequette. Prior to the introduction of Mannequette, MakeFashion teams only began experimenting with technology late in the process (i.e. distinctively after working through the inspiration phase). With Mannequette, this process changed dramatically: teams (especially fashion designers) experimented with different interaction concepts very early on in their design process, where the delineation between the inspiration and experimentation phases was considerably less distinct. This seems to have been made possible by how Mannequette makes interacting and testing ideas easier.

For example, we observed teams working collaboratively to rapidly iterate through sensor interactions on the miniature dress form (Figure 47). A fashion designer from Team 8 states: "*[the Mannequette] was great because we discussed and prototyped three different sensor ideas before settling on something much simpler. It really made our lives easier before we started doing any of the more difficult work*."

Because teams had Mannequette, they incorporated more conceptual planning around placement of batteries, sensors and general wearability using the dress forms. Previously, teams would generally not consider these issues until quite late in the design process. Both designers from Team 3 stated: "*...we were really new to the tech side of all of this but being able to actually see and plan our piece with tech was helpful for us in designing the casing*

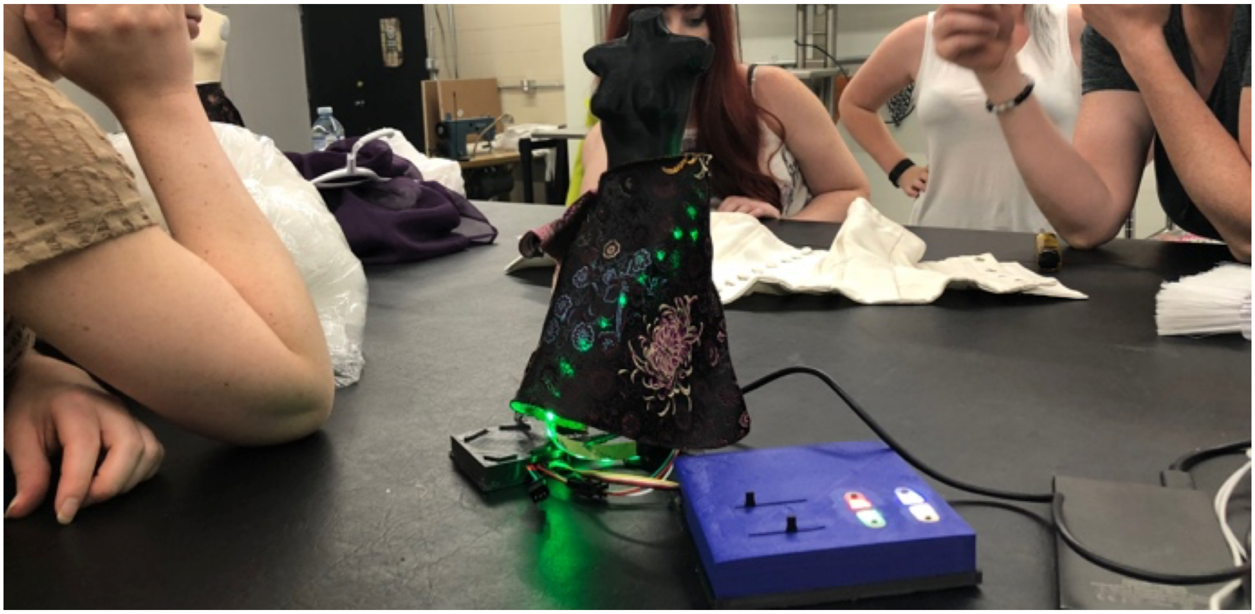


Figure 47: A design team collaborating using Mannequette.

and pouches that we placed in the garment. It actually makes it so much easier for it to travel as well."

One of the biggest benefits of Mannequette was the impact on how and when teams integrated technology into their garments. Because several teams became increasingly technology literate for their garment earlier in the process, and collaborated with their technologist, less time was spent on integration and requirements than garments they previously made for other runway shows. As a result, teams were more satisfied overall and spent more time on the creative aspects of the garment (e.g. nicer light patterns, adding accessories like wings) (Figure 48). A more experienced technologist from Team 8 stated: "*...being able to use the base as a prototyping tool and then just putting it into the final garment was brilliant. We spent less time trying to get things to work vs. knowing it already worked, which meant we spent more time making things look even nicer on the design side... which was a refreshing experience.*"

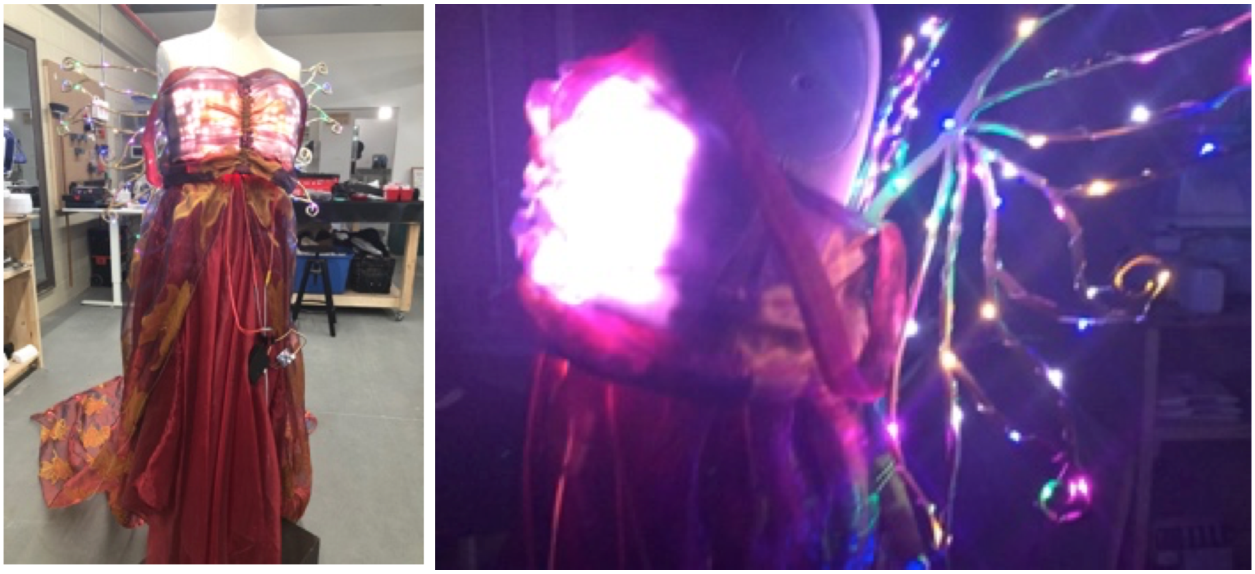


Figure 48: A fully completed avant-garde fashion-tech garment.

Because Mannequette gave fashion designers earlier access to technology, it impacted the traditional fashion designer vs. technologist role division. Specifically, since it simplified technical work with the sensors, some technologists with prior experience in design teams felt increasingly unclear with their role. A MakeFashion technologist on Team 7 stated: "*...sure, it's great that the designer can do way more now and it's simpler code modifications for me in the end, but then I don't really feel like I contribute as much as I am used too. If anything, my contributions are more design focused than tech focused now, which is quite interesting!*"

The Runway with Mannequette. Teams used the Mannequette (specifically the mixer) to fine-tune their garments for runway conditions, and those with more complex interactions and sensors benefited the most from having the Mannequette. For example, Team 6 (consisting of a single person who was both a designer and technologist) used a proximity sensor, where the concept was that anytime someone approached the model, their garment would react. The proxemic interactions were prototyped and constructed without a clear sense of the runway size, timing for music, and distance that the garment would need to react with another model on the runway. During the dress rehearsal, where garments were tested, this team adjusted their proxemic values ad-hoc using the Mannequette while on the runway itself. They stated: "*…tweaking the ranges to the runway [live] took a matter of minutes versus how I've done things before, where I've had to carry around a laptop, plug it into a garment and continually upload to an Arduino board and tweak. It saved me so much time.*"

We also observed teams begin to introduce modular concepts into their design practices and how they fixed garments. For example, Team 3, who specifically designed a removeable technology layer, found a small short in the wiring for an LED strip. This was discovered after a model wore the garment. As they describe: "*…we're really glad we caught this at the rehearsal and that we could diagnose it so quickly and fix it the way we did. Having the mixer allowed us to isolate the problem with the help of a technologist.*"

9.6 DISCUSSION

Mannequette provides one solution for some of the issues we described in the avant-garde fashion-tech process, but it is not a silver bullet. Our aim was to introduce a complementary tool within this process to aid in overcoming some key challenges in designing the garments — communication and creativity. We discuss the key takeaways when designing tools like Mannequette within the broader scope of wearables, as well as the impact of facilitated communication within avant-garde fashion-tech.

Redesigning Tools for Wearables. Future tools for designing wearables must be inclusive to other design processes and communities to enable richer explorations (noa (2019b)). While several toolkits for wearable fashion exist, they seem overly focused on maker communities or education. And, although these toolkits have been wildly popular within those communities, they have had comparatively limited uptake in the avant-garde fashion-tech community, particularly because they do not consider pragmatic concerns such as weight, robustness, and power. Tools in the fashion-tech space can incorporate lessons learned from prior work in construction kits (Berglund et al. (2018);Buechley and Hill (2010);Kim et al. (2009);Pan and Stolterman (2015)) to lower the barriers of entry into technical areas. With Mannequette, we still observed some discomfort in programming by designers, despite some becoming interested or even empowered in those activities.

Within and Beyond the Design Team. Maker communities often openly share designs and collaborate, with novices able to build from designs depending on what tools are available to them (Gongini (2019)). With Mannequette, we observed a small step towards this within avant-garde culture, as many designers became increasingly confident and empowered in communicating their ideas with technologists. Designers began to use their own documentation as a means of explaining and comparing temporal, and also demonstrating their work to others (through video and pictures). Several even began to share their creations and construction processes to other designs teams in MakeFashion, as well as the broader avant-garde fashion-tech community through social media (e.g. Twitter, Instagram). This is important because one of the main challenges in the observed process discussed earlier was the lack of examples for novice teams to draw upon. Furthermore, the video and photo documentation created and shared can be used as a means of documentation for the garment itself, especially useful when the garment is handled by other designers and technologists at different runway shows. Ultimately, we envision this approach of using a tool to create expressive visual artifacts, coupled with documentation processes, a valuable way to grow the avant-garde fashion-tech community, similar to (Buechley et al. (2008)).

9.7 LIMITATIONS AND FUTURE WORK

We discuss the limitations of our work and suggest future research directions for the avant-garde fashion-tech space.

Design Modularity — Our concept primarily focused on creativity and enabling interactions, but we did not fully explore the modular aspects of our design and its benefits in extensive detail. Furthermore, we used existing off the shelf components (noa (2019c)) which were already modular but not always ideal for garments (e.g. bulky connectors).

Team Dynamics — The lead author was deeply embedded in the MakeFashion organization, which has unique team dynamics between (self-identified) fashion designers and technologists. MakeFashion specifically creates or finds teams that contain fashion designers and technologists who create garments. Much of our observed and described processes is built upon this, and we recognize this may not be reflected in other organizations and runways. In our future work, we will explore working with other organizations that may have different team dynamics and processes.

Evaluation — While we deployed Mannequette in a real-world setting with design teams, our period of evaluation was limited. We followed the progress of teams from concept to execution, but we were not able to explore the later phases when the garment travels along with its Mannequette.

Alternative Forms— While Mannequette focused on using a whole-body mannequin for prototyping, other singular forms do exist, such as arms, or heads. Exploring these forms individual may enable designers to have even more interesting experimentations and explorations into wearable technologies. We also focused on normative binary body shapes for this initial work, but our future work will explore using different shapes of different bodies for designers.

9.8 CONCLUSION

We introduce a prototyping tool for avant-garde fashion-tech garments, designed to address the challenges that arise from designing and experimenting with interactions, sensors and outputs at the earliest stages of the design process. It does this by providing different detachable dress forms, support for different sensors, and a custom baseboard paired with a simple DJ mixer. We first embedded ourselves within a fashion-tech organization, before designing and building our tool, Mannequette. We conducted an 8-week deployment study with eight teams who designed garments, from concept to runway. Our results provide insight into how to design tools that incorporate the avant-garde process and facilitate interdisciplinary communication and creativity

10

MAKERARCADE: USING GAMING AND PHYSICAL COMPUTING FOR PLAYFUL MAKING, LEARNING, AND CREATIVITY

The growing maker movement has created a number of hardware and construction toolkits that lower the barriers of entry into programming for youth and others, using a variety of approaches, such as gaming or robotics. For constructionist-like kits that use gaming, many are focused on designing and programming games that are single player, and few explore using physical and craft-like approaches that move beyond the screen and single player experiences. Moving beyond the screen to incorporate physical sensors into the creation of gaming experiences provides new opportunities for learning about concepts in a variety of areas in computer science and making. In this early work, we elucidate our design goals and prototype for a mini-arcade system that builds upon principles in *constructionist gaming* – making games to learn programming – as well as physical computing.

***Note: The text in this chapter appears in the following publication:**
Teddy Seyed, Peli de Halleux, Michal Moskal, James Devine, Joe Finney, Steve Hodges, and Thomas Ball. 2019. MakerArcade: Using Gaming and Physical Computing for Playful Making, Learning, and Creativity. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems (CHI EA '19).* ACM, New York, NY, USA, Paper LBW0174

Over the past several years, efforts to promote 'making' – cross disciplinary activities that involve DIY culture, electronics, science, engineering and craft – have expanded greatly (Brahms (2014)). Building upon these efforts, a number of construction toolkits have been created (e.g. LilyPad (Buechley et al. (2008)), Lego Mindstorms noa (2019d)) to lower barriers of entry into programming and introduce computing concepts across using a variety of contexts (and domains), such as robotics or e-textiles. Many of these toolkits and subsequent efforts have shown significant success in attracting, engaging and teaching youth in STEAM (Science, Technology, Arts, Engineering and Mathematics) activities (Buechley and Hill (2010))).

One rapidly growing area for these efforts utilizes gaming. Due to the ever-growing popularity of gaming among children today (especially for entertainment), researchers and educators have begun using gaming to facilitate positive learning experiences (Gee (2007)), building upon core game design principles (scaffolding, interactivity and productive failure) (Kapur (2008)). A more recent approach within this context that has already shown promise is constructionist gaming, which involves students (and others) making or programming their own game for learning (Kafai (2006)). Some tools that embody this approach (e.g. Scratch, Kodu Game Lab (noa (2019e))) enable those with limited game development or programming skills to create games. By enabling children and others to learn by both playing and making games, computational thinking and problem solving can be improved (Kafai and Ching (2001)).

Although constructionist gaming approaches, tools and associated research is growing rapidly, there is still a large focus on designing around a singular screen (or gaming) experience (Kafai and Vasudevan (2015)). Alternatively, commercial gaming has already begun moving beyond the screen into the physical world (Eisenberg et al. (2009)). For example, the Nintendo Labo platform uses different approaches to combine elements of making and construction into gaming experiences. More recently, construction kits have been used to link the design of games with the design of gaming interfaces, and this combination approach has shown promise in fostering collaborative learning, creative expression, and learning computational concepts (Kafai and Vasudevan (2015)). This combination approach has revealed challenges with designing around constructionist gaming, especially with regards to construction kits. Specifically, there is not an equal emphasis on both computing and crafting, meaning designing on and off the screen experiences haven't been made equally important for constructions kits for gaming, which differs from other approaches such as those for e-textiles, where screen-based activities are (typically) limited to programming and downloading code onto a wearable (Kafai and Vasudevan (2015)).

To begin investigating and addressing some of the challenges in designing and building a constructionist kit for gaming, we created the MakerArcade system. MakerArcade draws upon concepts of physical computing using manipulatives, similar to (Girouard et al. (2007)), as it provides several benefits for novices (e.g. collaboration], visibility of work) ((Kafai and Vasudevan (2015))). It also draws upon programming concepts taught using simplified graphical, block-based user interfaces like MakeCode, as they simplify programming concepts, and block-shaped constraints help teach and enforce syntactically correct programming statements (Touretzky (2014)). In this paper, we describe our preliminary system (Figure 49) that builds upon the Microsoft MakeCode Arcade (beta) software platform (noa (2019g)). We believe this is an interesting step towards designing and building constructionist gaming experiences that incorporate both hardware and software aspects from the ground up.

Figure 49: An example of a completed MakerArcade system in the form of a miniature Arcade Cabinet running a user-created Flappy-bird like game with a default set of buttons mapped. This game (and others) can be mapped to different sensors (like an accelerometer) that a user can program and map as an alternative controller that is plugged into the cabinet which contains a headphone jack and uses our custom headphone jack-based hardware protocol.

10.2 DESIGN GOALS

Informed by our own experiences in designing, building and gathering feedback for MakeCode Arcade and activities of our users from MakeCode Maker noa (2019h), as well as prior work in relevant areas of constructionist toolkits, tangibles and physical interfaces (Kazemitabaar et al. (2017)), we describe our design principles for a constructionist gaming system designed to facilitate play, learning and creativity.

10.2.1 *Leverage the Gaming Domain*

Several construction kits have used their respective domains for design and inspiration, such as MakerWear (Kazemitabaar et al. (2017)) which utilized wearability and mobility for the design and experience of the platform. Similarly, we aim to draw upon and provide components that incorporate and are inspired by gaming, specifically retro gaming and the arcade culture. Furthermore, retro gaming has begun returning in popularity recently, with re-releases of consoles such as the Nintendo Classic Mini, providing a unique opportunity to blend playful making and learning.

10.2.2 *Modularity*

Prior work in modular kits for wearables has shown benefit (Kazemitabaar et al. (2017)). Similarly, providing a system that supports plug and play with sensors and electronics, enables a broad range of activities for both on and off experiences for constructionist gaming. For example, a user can design, program and physically construct a custom controller for a game (which they've also designed and programmed) using different sensors, like an accelerometer and proximity sensor.

10.2.3 *Equality in Physical and Digital Making*

A key lesson from (Kafai and Vasudevan (2015)) with regards to constructionist gaming and the evaluation of existing tools was that there needs to be an emphasize on equality between crafting and computing for both on the screen and off the screen activities in constructionist gaming. This notion should be incorporated both on the physical (activities with electronics) and digital side (activities with the software) when considering the design of a constructionist gaming kit.

10.2.4 *Low-Ceilings, High Walls*

Building upon Resnick and Silverman's design cues (Kafai and Vasudevan (2015)), a constructionist gaming system should support users in the creation of increasingly complex games that combine both the physical and digital aspects, as further experience is gained.

10.2.5 *Low-Ceilings, High Walls*

In wearable toolkits, an emphasis was placed on rapid tinkering and prototyping (Buechley et al. (2008); Kazemitabaar et al. (2017)). Similarly, our system should enable rapid tinkering and experimentation, as we aim to enable users (especially children) to easily experiment and play with different components. Given that expression is an important aspect of the constructionist approach (Kafai (2006); Kafai and Vasudevan (2015)), focusing on a wide range of tinkering activities (e.g. adjusting effects) in the creation and play of a game is important.

10.2.6 *Fostering Collaboration*

Another benefit seen from constructionist gaming activities, is the occurrences of collaboration and their associated artifacts (e.g. components customized with crafts), particularly in classroom settings (Fowler (2017)). This means that a system should both enable and capture (digitally and physically) these artifacts to foster collaboration.

10.3 MAKERARCADE

Our current prototype of the MakerArcade system is comprised of: (i) custom software experiences built upon the Microsoft MakeCode framework (?), (ii) a mini arcade cabinet that houses electronics and (iii) custom modular blocks that can be programmed. MakerArcade is heavily inspired by the retro gaming and DIY culture (through the cabinet), as well as prior work in modular blocks and devices, such as (Merrill et al. (2007)). Next, we briefly describe our current implementation.

10.3.1 *Software*

The software components of MakerArcade use customized (beta) versions of Microsoft MakeCode Arcade (Figure 50) and Microsoft MakeCode Maker (Figure 51). Both are open-source and are built upon the Microsoft MakeCode web framework, which enables a web-based programming experience for novices. It's editor uses block-based programming and code can be executed within the browser itself (Devine et al. (2018)). The MakeCode Arcade editor enables novices to learning programming by creating games in the browser using block-based programming (similar to (noa (2019e))). Games are stylized in a retro 8-bit manner, with a small view dynamically rendering the game as it is being programmed. The MakeCode Maker editor allows novices to learn how to both program and prototype (via breadboard) with a variety of different sensors and electronics components.

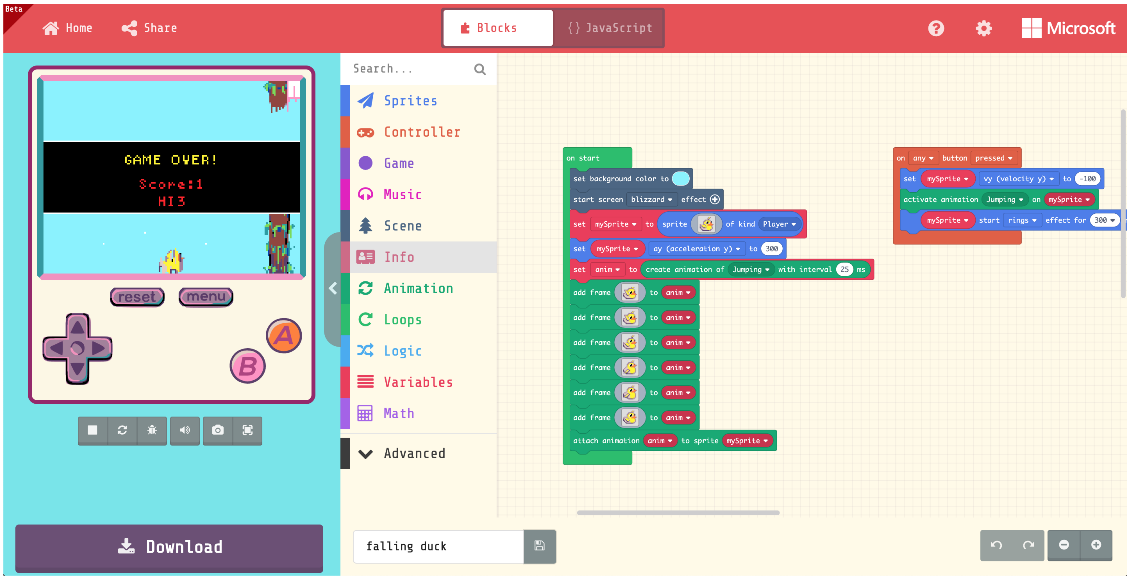


Figure 50: Microsoft MakeCode Arcade (beta) that allows a user to create a game using block-based programming concepts. We provide packages that allow a user to program custom physical controllers that use buttons or sensors (e.g. sound) and map input to the game being programmed. For example, a user can map shaking input from an accelerometer (using a block) to the flapping of the bird. When the game is run on MakerArcade, the sensor is plugged in to play.

For our MakerArcade prototype, a user creates their game using MakeCode Arcade, which can then be loaded onto it (using USB) and played immediately. MakerArcade by default comes with 3 buttons (mapped to Start / Reset / B key), however, several games require more than 1 button for interaction (e.g. a Pacman-like game requires multiple keys for directions). This is where our software customizations arise, combining digital and physical construction through the use of Arcade and Maker. Using customized packages in Arcade, and Maker, users are able to program custom plug and play input using sensors (or buttons) for their game. The mapping of custom input created by a user in Arcade is done automatically using our packages. This means that a user can not only create their game, but they can also create how they want their game to react with a chosen form of input. Given that sensors (and buttons) can be used dynamically, it allows a user to build their own controller

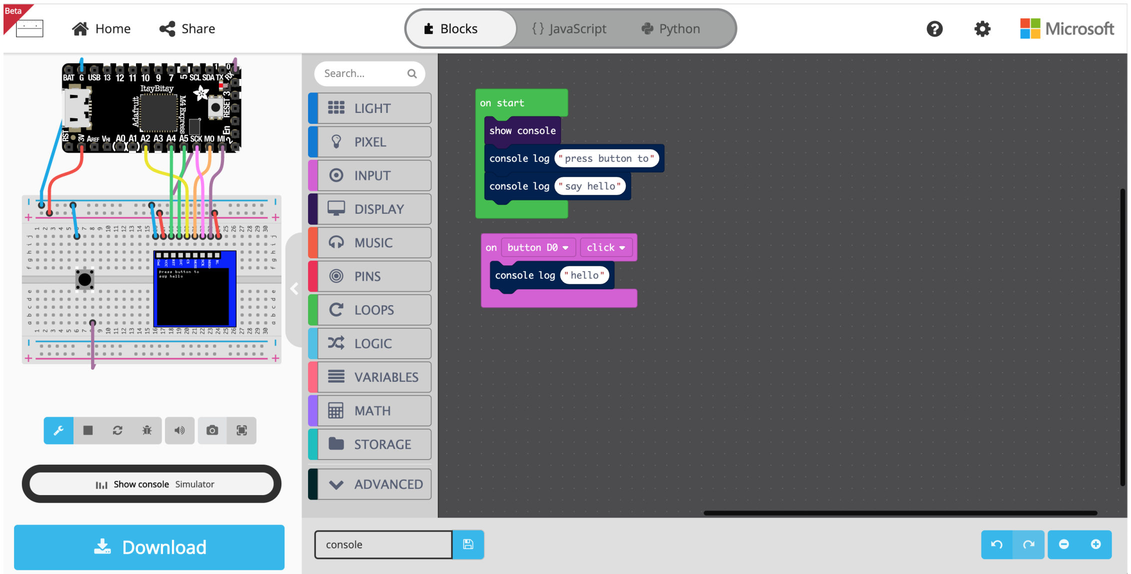


Figure 51: Microsoft MakeCode Maker (beta) that allows a user to program and prototype electronics. We provide packages which show a user how to physically create an arcade (with a screen) and buttons mapped to a microcontroller. We also provide mappings on how to create additional plug and play components (for custom controllers, sensors or modules), that use a custom headphone jack-based hardware bus protocol. These mappings when programmed by a user, are auto-detected when plugged into our MakerArcade system.

experience that plugs into the MakerArcade cabinet, or even further customize their own MakerArcade, as instructions are provided on Maker that describe how to build an arcade using a screen, microcontroller and buttons on Maker (full list of supported hardware available on our website). In summary, we allow users to fully customize a gaming experience, not just with their programmed game, but also with their own plug and play controllers (built using their own choice of input) for our provided Mini-arcade cabinet, which can also be further customized.

10.3.2 *Hardware*

Our provided MakerArcade physical prototype consists of (1) a SAMD51-based microcontroller (Adafruit Trinket M4 Express), (2) a ST7735-based 1.8" TFT LCD, (3) 3 analog buttons and (4) an audio-jack port mapped to a pin on the microcontroller, all of which is housed in a laser-cut acrylic Arcade cabinet. The SAMD51-based microcontroller (and other supported microcontrollers) run custom bootloaders that enable it to both run games compiled on MakeCode Arcade, as well as the screen itself. The audio-jack port plays a critical role in the plug-and-play architecture for the constructionist components of gaming. We built upon JACDAC (noa (2019k)), a bus-based hardware protocol that utilizes audio-jack cables.

Generally, the supported parts serve as the base components for our designed MakerArcade system (and cabinet), but users are also able to design and build their own housing for these components, emphasizing more of the playful making and physical computing aspects. For example, a user could create a console-like design (e.g. Nintendo Classic) to house the base components.

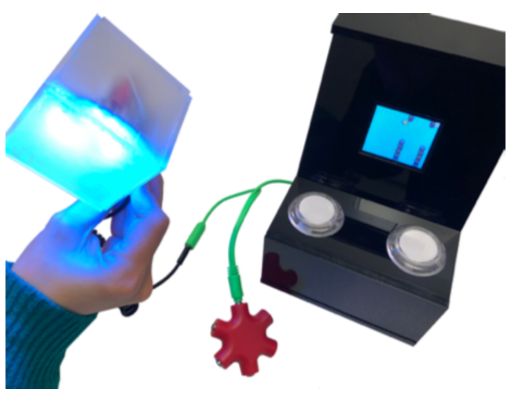


Figure 52: This custom designed accelerometer controller plugged into additional inputs using a headphone jack hub. This lets a user build their own controller and interactions for their game using MakerArcade.

Figure 53: Buttons from an Adafruit Circuit Playground Express (CPX) used as a controller for the Flappy-bird game. The CPX is coded using blocks (and a custom plug-and-play headphonejack bus protocol) within MakeCode Maker. On MakeCode Arcade, a user simply programs additional buttons using our packages, which are auto-detected when plugged into the headphone jack hub.

With MakerArcade, a user can plug and play their own custom designed inputs (with mappings provided and created in Maker) into our supplied cabinet. These inputs use the JACDAC protocol which requires headphone jack cables, and with a Headphone jack hub, multiple custom inputs become plug and play (Figure 52). Furthermore, microcontrollers that are supported on MakeCode Maker (Figure 53) can serve as input (and eventually output) for games created on MakeCode Arcade. This means that users can fully prototype and build their own MakerArcade controller experiences with a variety of popular microcontrollers.

10.4 Conclusion and future work

We introduced our early work towards building a constructionist gaming system that emphasizes designing, remixing, and customizing physical and digital components in building games. As it is early work, there are numerous directions and opportunities we intend to explore, including: (1) providing alternative and more customizable designs (e.g. console-like, hand-held or a wearable form factor); (2) explore providing modules with outputs that can be used in the creation and experience with a user programmed game (e.g. sound, movement); (3) participatory design sessions with children of different age groups and makers and (4) running user studies with a similar group to see how they use MakerArcade and what games, controllers and experiences they design, ultimately creating guidelines for creating constructionist gaming systems and kits.

11

STITCHKIT AND THE PERILS OF CROWDFUNDING

"*Success is walking from failure to failure with no loss of enthusiasm.*"

Sir Winston Churchill

One of the biggest challenges I faced as a student entrepreneur, was being able to blend my research with the entrepreneurial side. For the most part, they had been in parallel and overlapped somewhat, but not fully. A key reason for this was timing, as it was sometimes not feasible to synchronize my academic deadlines (e.g. a paper deadline), with entrepreneurial ones that can occur at any given time of day (e.g. meetings, pitches, proposal writing, etc).

Following Emotica, as a student I felt my entrepreneurial experience was incomplete because my co-founder and I weren't able to raise the necessary capital from VCs. One of the hallmarks of being a good entrepreneur is to accept failure and continue, as the hope is, eventually you will succeed.

In my case, before continuing with another startup, I took a very self-critical look at how I had participated in startups as an employee, how I operated them as a co-founder, and how I was blending research and entrepreneurship. From building a team, to de-risking hardware, to finding different ways to fund a startup, I felt ready for the next challenge, and thus I co-founded another startup, but this time significantly more aligned with the research in my dissertation.

11.1 THE WORLDS COLLIDE

In Chapter 6 and Chapter 8, I described the Doppio and the work behind its commercialization. One detail I did not fully elaborate on was how I attempted to address my uncertainty if a watch I designed was going to be good enough to compete with those from Google, Apple, Huawei and others. Pragmatically, a lone student cannot compete with multi-billion dollar companies, but I made an attempt regardless. I began to think about what I could do differently than what was done before, other than the unique features of Doppio.

Thinking about how to design a better smartwatch, and by extension Doppio, led me into the world of Fashion and the reason was simple: smart watches, and other types of wearables are worn on the body, and if they're doing to be designed well and look beautiful, I should then get some input from fashion designers. As fashion already makes things worn on the body, this seemed like a logical step, and thus I initially gathered informal feedback about wearable tech, smart watches and other devices. What I discovered was an interesting area, both entrepreneurially and research wise.

As demonstrated in Chapter 9, there are several issues with fashion designers who want to create wearable tech garments. On the research side, Mannequette addressed issues of facilitated communication between designers and technologists. But, as I worked in this space for nearly 3 years, I observed things like the following:

- Combining fashion and technology is very impactful from a Science, Technology, Engineering, Arts and Mathematics (STEAM) learning perspective. Many of the children and fashion designers I worked alongside were learning technical concepts in an easy to digest manner, and having fun doing so.

- From a technical standpoint, there is a lot of work to be done in making the electronics side of fashion technology simpler, such as minimizing the amount of soldering, or making electronics components (e.g. sensors) follow a more plug and play approach.
- Programming a technology infused garment to do something simple, like turning light patterns on and off with a sensor can be a big challenge for novices.
- One of the most common tasks, turning on hundreds of LED lights, is difficult for novices, because there are technical limitations (i.e. LED lights are power hungry, and the more they are used, the more power is required).

As an entrepreneur, the solution was quite simple. Create a product that addressed some of these challenges. This product and some of the research behind it became StitchKit, a fashion technology kit designed to address some of the aforementioned challenges. Additionally, on the research side of my dissertation, I used StitchKit as the electronics component that enabled Mannequette to have its plug and play ability with components, it's ability to interface with the DJ-like mixer. In essence, StitchKit was where both research and entrepreneurship finally met, as it influenced the design of Manneuqette and vice versa.

11.2 WHAT IS STITCHKIT?

StitchKit was a fashion technology kit designed for a broad audience — from young students, to students, cosplayers, fashion designers, and generally anyone interested in incorporating technology into something wearable. It consisted of a custom programmable Arduino board and a few electronics components (e.g. accelerometers, sound, or proximity sensors), which varied depending on the type of kit. See Figures 55 - 55

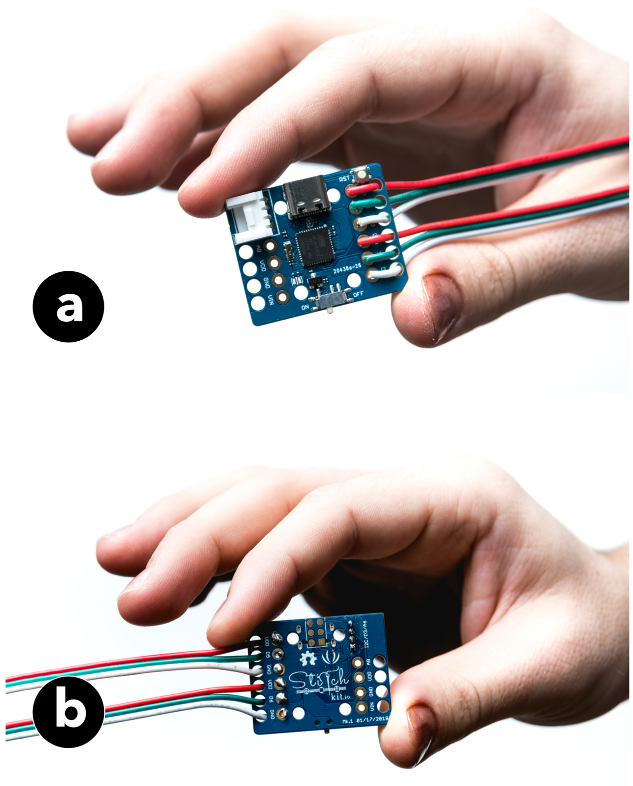


Figure 54: The StitchKit Arduino board. (a) Front of the StitchKit board. (b) Back of the StitchKit board.

One of the differentiating features of the custom StitchKit board compared to other competitors (such as the Lilypad Buechley et al. (2008)) was the built in stress-relief holes. These were important because components such as wiring, LED lights or accelerometers would break frequently when worn on the body, which we had observed in many fashion-tech garments. These components were often soldered, which also presented challenges for those who were novices, and making this easier meant lowering one specific barrier of entry into fashion-tech and wearables. Next, a Grove sensor port was added, to support a wide

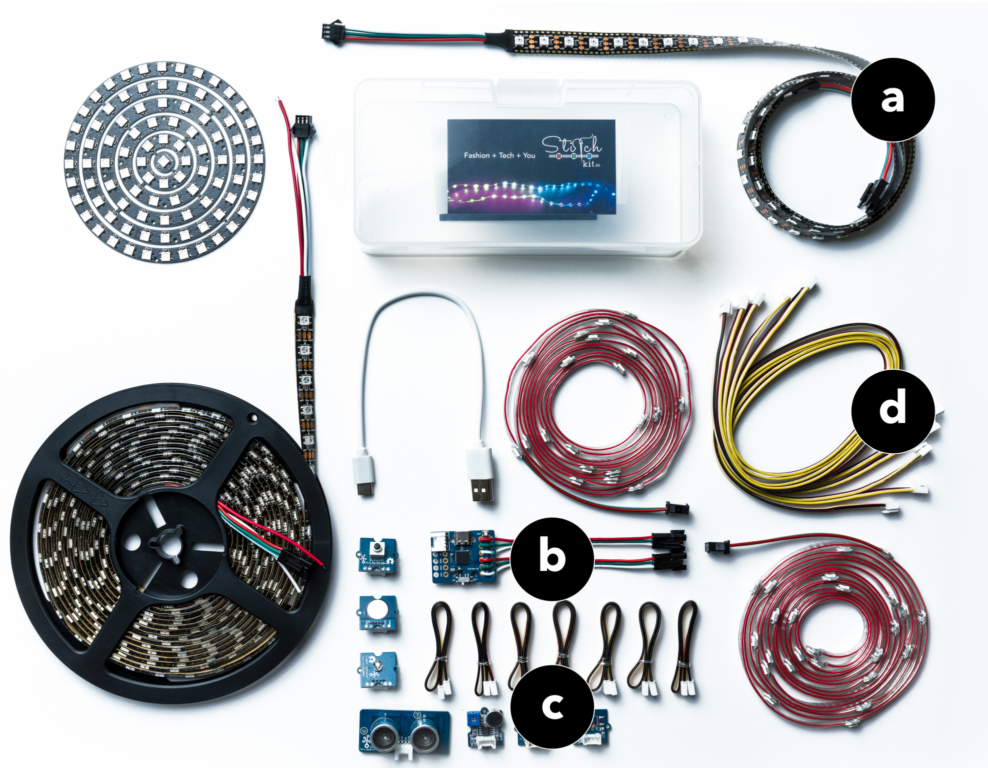


Figure 55: An example of one of the contents of a StitchKit package. (a) Multiple types of LED strips. (b) The StitchKit Arduino board. (c) Different types of Grove sensors from Seeed Studio, including accelerometers, proximity, touch and sound.

range of grove sensors from Seeed Studio[1]. The Grove Sensor port enabled sensors to become plug-and-play which could be programmed using the Arduino IDE.

11.3 STITCHKIT ON KICKSTARTER

Unlike Emotica, where we focused on attempting to raise capital from VCs, or SITS which was primarily funded with numerous government grants, StitchKit instead focused on crowdfunding. Crowdfunding is an alternate method of funding a venture (or product) by raising money on the Internet from a number of different people around the world.

1 Seeed Studios Grove Sensor System - http://wiki.seeedstudio.com/Grove_System

While my previous experience with crowdfunding Slate Scale was unsuccessful, I still preferred this method for StitchKit, as it was less complicated from a technical standpoint to manufacturer at scale (a customized Arduino microcontroller vs. a consumer electronics product). Additionally, the StitchKit team, which consisted of myself and 2 other co-founders, also intended on using a successful Kickstarter campaign (and its associated funds) as a means of validating our target market, provide finances for an initial manufacturing run of microcontrollers for the kit (i.e. the StitchKit board), and lastly, to gain some product traction (e.g. it gets adopted in schools locally in Calgary) and press.

As mentioned previously, the team for StitchKit was comprised of myself and 2 other co-founders. Our combined experienced covered important areas such as fashion, technology, manufacturing and engineering. We also had a strong team of volunteers to assist us as needed. While I have described a mostly abbreviated timeline of StitchKit, the team took a bit longer to form, as we all began on this track at different points career-wise. For myself, I became involved when I was investing fashion designers to assist with creating Doppio. Overall, we were all realistic about what was necessary from ourselves and our volunteers to succeed on a Kickstarter campaign (see *Lesson 3: Building an Organization Quickly vs. Effectively (5.3.3)*). The full timeline is provided in Figure 56.

StitchKit launched on Kickstarter on December 13, 2017, and achieved a number of milestones in a relatively short time frame:

- Within 1 day, it 50% of its $10,000 CAD goal.
- By December 19, 2017, it hit 100% of the funding goal, and on the final day of the campaign, January 27, 2018, the funds raised totaled $30,011.
- StitchKit was successful in attracting press, as it appeared in places such as Daily Planet on Discovery Channel, and magazines such as Make:.

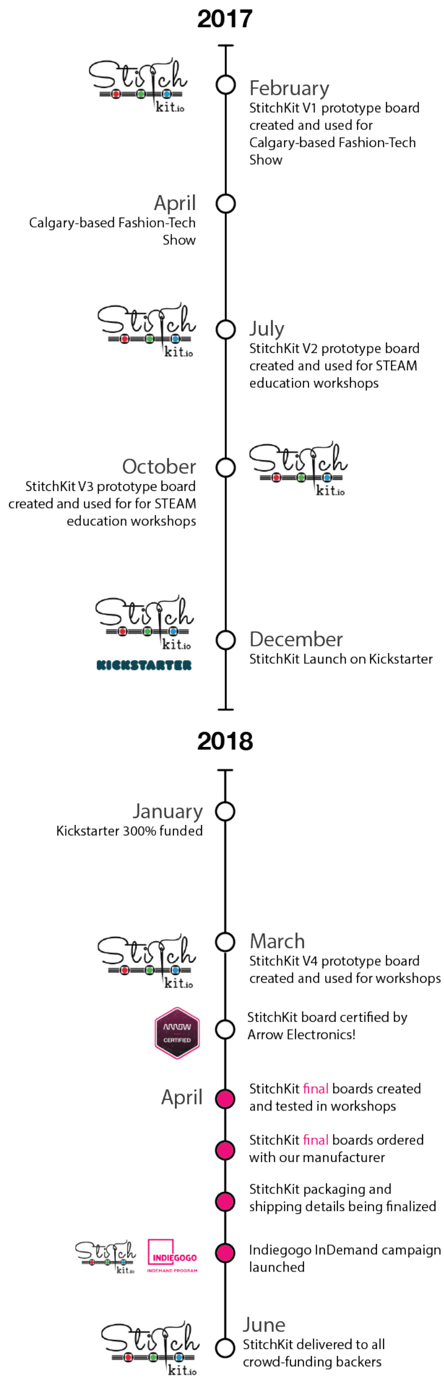


Figure 56: The product timeline for StitchKit.

- StitchKit also became a 'Project We Love' Kickstarter, a designation given to projects that Kickstarter staff believes are creative and standout from others on their platform.

For all intents and purposes, the Kickstarter for StitchKit was successful. But as I describe later, there were several challenges that occurred after the campaign.

11.4 ROLES AND RESPONSIBILITIES

Planning and launching a Kickstarter project was not a simple task. Operating a successful one is even harder, especially with the work that follows (which I discuss later). Ultimately, there is no silver bullet for having a successful campaign, but below is an abbreviated list of some of the responsibilities I undertook as one of the co-founders of StitchKit. I have broken them into different sections according to the point they occurred in the campaign.

Pre-Launch: These are some of the activities that were necessary before launching the campaign on Kickstarter. Without accomplishing these, launching the Kickstarter campaign would not have been possible:

- *Building A Backer List*

  One of my experiences in Slate Scale which I brought to StitchKit was to build a list of emails for people who would be interested in purchasing the product (also known as backers). While Kickstarter is a crowdfunding campaign, you still need to attract a crowd, and thus building a backer list meant when the campaign launched, there was a group of people ready to purchase StitchKit.

- *Designing Campaign Materials*

  To build the backer list, a website and other materials needed to be designed that was attractive, simple, and encouraged people to sign up to purchase a kit. This was not a simple task as we had plenty of video and image material that needed to be modified and distilled to capture the The Fashion Technology Kit for Everyone tagline.

- *Supplies and Manufacturing in China*

  Before launching our campaign, we travelled to Shenzhen, China to look at how to manufacturer our StitchKit board. Shenzhen is one of the top locations in the world for manufacturing electronics in a quick manner. Over multiple trips, we found a manufacturer, produced a few samples, and travelled through several electronics districts to find the components that would comprise the kit. The ultimate aim was to have much of the manufacturing and components already in place before the campaign began, as a way of minimizing our risk and the risk of our potential customers. As a side note, this was something my co-founder and I did not do for Slate Scale, and in retrospect, it should have been done. Comparatively, the amount of time and money saved by going to China is worth it if a hardware project is being attempted.

- *Campaign Finances*

  One of the most difficult aspects of any crowdfunding campaign is determining what the funding goal is going to be. For StitchKit, there were many things we needed to consider, such as the per unit cost of the kit, the per unit price of the StitchKit arduino board (also known as bill of materials or BOM), the average price to ship one of the kits to different parts of the world, how much we were going to be paying ourselves for our time, social media costs (e.g. running ads on Facebook and Instagram), cost of warehousing, just to name a few.

  With many of these costs in mind, we still aimed for a lower funding goal on our Kickstarter because we wanted to be able to reach our goal successfully. This meant not paying ourselves for our time and effort in making the kits, or the trips to Asia for manufacturing. We went for the absolute minimum cost for the Kickstarter campaign, and thus it ended up being $10,000 CAD.

- *Social Media*

  One of my other pragmatic lessons learned in Slate Scale, and even SITS, is that Social Media is a critical aspect of making a campaign successful. After all, if nobody has seen or heard about your product, nobody can buy it. For StitchKit, Instagram was the most popular form of attracting our customers, however, we also engaged people by creating a Facebook group and facilitating conversations around fashion-tech, wearables and other types of electronics. Through all of these campaigns and startups, I also learned and had to admit to myself that I am not proficient at Social Media (see Lesson 10: Be Shameless, But Be Realistic (8.4.4)), but having a strong team helped in this regard (see Lesson 3: Building an Organization Quickly vs. Effectively (5.3.3)).

Post-Launch: These are some of the activities that were necessary after launching the campaign on Kickstarter. It is also important to note, while these are post launch activities, they are in fact continuous. This is because, once a campaign is successful, you effectively have a startup running with real customers, some amount of finances, neither of which are not "one and done" so to speak.

- *Managing Customers*

  Following the completion of our Kickstarter campaign, we had over 300 customers to manage and communicate frequently with. This was challenging because not all 300 customers were in the same time zone, some did not speak English, and not all customers ordered the same type of kits. We frequently posted updates on our Kickstarter page to give customers updates on things like when their product would be shipping, manufacturing status, additional kit details and other types of information.

  We also received a lot of messages from customers regarding support, and word of mouth from our existing customers also attracted new customers. This led to us continuing the StitchKit campaign on another crowdfunding platform, Indiegogo[2]. In the end, customer communication was a 24/7 job, and needs to be properly supported. After all, the backers on crowdfunding platforms for a product are the early adopters, and their influence plays a critical role in the future success of a product.

- *Packaging*

  Perhaps one of the most surprising aspects of StitchKit was the challenges we had in creating packaging. One of our biggest challenges was that we had a number of different kits that we had sold in the campaign. This meant creating different types of packaging for each, and as a result, this became an added cost to the campaign which we had not initially considered. Personally, I had wanted to follow in the footsteps of the nice aesthetics of products typically sold in Apple stores, but the costs were quite high. In the end, one of the key lessons we learned for packaging, was to minimize the number of packaging styles you need from the start, and to keep it as simple as possible. We ultimately ended up using simplistic cardboard boxes, with paper sleeves for packaging, as it was significantly more cost-effective.

- *Managing Shipping Expectations*

  While the campaign for StitchKit concluded on January 28, 2018, it was not until early July 2018 that the kits began to ship. Our estimated shipping time when we launched the campaign was March 2018, and thus we were significantly off in our estimates. One of the reasons we had done prior work with manufacturing in China was to mitigate the risk of shipping our product and to have reasonable delivery dates for our backers. The problem we faced, however, was that a microcontroller that was necessary for producing the StitchKit boards suddenly became out of stock worldwide. This caused

2 StitchKit Indiegogo - http://www.indiegogo.com/projects/stitchkit-a-fashion-technology-kit-for-everyone

a significant delay in our shipping and required a substantial amount of expectation management for our backers.

One downside to crowdfunding is that there are several misconceptions that products from crowdfunding are never delivered, or are delivered several months and sometimes even years later. Both Kickstarter and Indiegogo have reduced this risk for those who back projects by introducing projects requirements, such as demonstrating prototypes, and providing proof that a product exists (which we also went through). However, this does not fully guarantee delivery, and thus it is important to constantly communicate and manage shipping expectations for backers.

- *Sample Code*

  As with any Arduino based (or even technical) product, sample code, demo apps and videos are critical in helping people get started. For StitchKit, we wrote several demo applications that were posted on GitHub. One of the challenges with this however, was that many of our customers had never ever used GitHub before.

For the most part, I have provided a very high level overview of some of the tasks that were involved with running the StitchKit campaign. A more detailed examination at different tasks and strategies for running a successful crowdfunding campaign, I highly recommend taking a look at Kickstarter Launch Formula: The Crowdfunding Handbook for Startups, Filmmakers, and Independent Creators by Salvador Briggman.

11.5 ENTREPRENEURIAL LESSONS (CONTINUED)

In Chapter 5, I described working in a startup and provided reflections, while in Chapter 8 I provided reflections of doing a startup and working with VCs from a local accelerator

(CD). In this last section, I instead provide reflections of building a startup that involves hardware, by using crowd-funding (e.g. IndieGogo or Kickstarter). While my prior experience in crowdfunding with Slate Scale was extremely informative, StitchKit was a much more in-depth hardware startup due its success in achieving its funding goal on Kickstarter[3].

While the Kickstarter campaign was successful, there were a number of challenges the team faced after. These challenges resulted in several difficult entrepreneurial lessons being learned, which I will describe next.

11.5.1 *Making Hardware Sustainable is Even Harder*

As I reflected in *Lesson 9: Hardware is Hard*, creating hardware is not a trivial task. Despite this, I still continued on the hardware path with StitchKit, with the assumption that if I could overcome the barriers I uncovered in Slate Scale, the next attempt would be successful. Instead of doing remote discussions with manufacturers, my co-founders and I went to China and dealt with them directly. In this aspect, Stitch Kit was successful, however, as we later learned, there are several levels to making hardware as a business sustainable.

As described earlier, the StitchKit board is a custom Arduino board. At the time of its design, my co-founders and I had limited experience with designing PCBs, let alone designing them for manufacture. We relied heavily on a 3rd party vendor (i.e. SeeedStudios) to assist in our design, and help us in other areas such as packaging and shipping from Asia. Prior to our launch, we worked alongside them to create sample kits, but the challenge we later faced after the success of the campaign, was making these kits sustainable to produce. For the StitchKit board, we unfortunately did not have a low bill of materials (or general cost),

3 StitchKit Kickstarter Campaign - http://www.kickstarter.com/projects/stitchkit-io/stitchkit-the-fashion-technology-kit-for-everyone

making them priced fairly high, and thus resulting in a lower margin per board. As a side note, it was this experience that led to me becoming self taught in PCB design and other necessary electronics engineering concepts to be able to it myself the next time.

We also faced the same challenge for the other components in the kit, such as LED lights, and various sensors. Overall, the margins for the hardware in StitchKit were fairly low, meaning it was not possible to make a significant amount of money from selling kits alone, thus making it unsustainable.

With Emotica, one of the key lessons I learned was to not rely solely on hardware, but also have it supplemented with a software component. As StitchKit was an educational product, this meant not only building a platform that could support our hardware and it's fashion-tech type activities, but one that we could make ongoing sales from. We initially focused our pitch on Science, Technology, Engineering, the Arts and Mathematics (STEAM) K-12 education, and began working closely with teachers, gathering feedback, and understanding how we could create curriculum around StitchKit. We then provided several STEAM education companies our kits both to sell, and to create curriculum for, but this also became a difficult and self created challenge. Do we just become a provider of kits (which we already knew to be an unsustainable business model) and allow others to create curriculum and STEAM education camps around our kits, or do we ourselves create STEAM education camps using our kits and scale our content and kits from there? We were never able to answer this question effectively, and it eventually led to the demise of StitchKit.

By no means is producing hardware easy, but the follow up ancillary tasks and questions which need to be answered, such as those I have described in StitchKit are just as important, if not more important.

11.5.2 *Timing is Everything*

Would you launch a new fitness product in the winter, or just before the summer? Would you launch an educational product during the winter, or in the Fall, just before most schools open? These are some of the critical questions that were asked when Slate Scale and StitchKit were going to be launched in their respective crowdfunding campaigns. Making the correct choice for when a product launches is important, and can play a critical role in its success or demise.

For Slate Scale, we launched in the winter. This was simply due to the reason that my co-founder and I were working part-time on Slate Scale, and as a result it took us longer to complete what was necessary to launch our campaign. This timing coincided with closer to the holidays, so we took advantage of that and marketed Slate Scale as a health and fitness Christmas gift. As I reflect back, this was a critical mistake because it was appearing at an off time for the market and its audience. For a fitness product, it would have made substantially more sense during the Spring or Early Summer, when “people are preparing to go to the beach or vacations and want to get into shape” as one of our potential users had told us.

One of the other challenges with launching during the holidays was capturing press was difficult. From my experience, during the holidays, news outlets preferred to focus on stories that were about the holidays rather than another gadget or new technology. This was further made difficult by the Consumer Electronics Show (CES[4]), which often occurs in January. At CES, there is a large number of products which are unveiled, from both startups and corporations alike. As a result, this meant that it was very difficult to get Slate Scale noticed in all the other announcements, and it was thus lost in the noise of other new products.

4 Consumer Electronics Show - http://www.ces.tech/

Unfortunately, StitchKit repeated the same process for its launch, which also occurred in December, just before the Christmas holidays. The same challenges occurred in capturing the attention of major press, although this time it was easier to create a holiday story around the product, by simply making Christmas sweaters embedded with technology.

Although StitchKit hit its completion goal in 6 days, its success to me was always slightly deceiving. Despite achieving its goal, and surpassing 300% of our funding, it is unknown if and how the campaign would have gone if it had launched in Late July or August, also known as back to school season, or if it had launched just before summer education camps began. We may have ended up with substantially more traction, a viral campaign and potentially more significant press than we had received.

In both the cases of StitchKit and Slate Scale, we launched during the holidays. If you have a product that relates to the holidays, then launching it then should be fine, otherwise, understand your market and choose the appropriate time to launch

11.5.3 *Define Success Within Your Team*

Earlier in Chapter 5, I talked about building an organization and its importance (see *Lesson 3: Building an Organization Quickly vs. Effectively (5.3.3)*). With both Emotica and StitchKit, it was easier to build a small but effective team because of the network I had around me. However, as I learned over time, just building an effective team was not enough, but also having a team that had a similar definition of what success is and looks like was just as important.

As I mentioned earlier, there will always be failures in startups, but there will also be successes. It is a continual learning process. But for StichKit, the challenge became that different parts of the organization had different definitions of success. To frame this point further, I will introduce the fabric shop analogy.

Ben, Klaus, Miele, Reggie and Marion are all good friends who have been working together for a few years now. All have similar interests and worked in clothing and creating fabrics before, and so they decide to open a fabric store. The team spends a year preparing for the launch of the store, shopping and purchasing fabrics overseas, creating a social media following and finding the right store venue. Fast forward to the launch of the store, and the team is excited. A few days after the launch of the store, they completely sell out of fabrics, which concerns Ben and Klaus. They are concerned about their revenue stream, as despite the success of selling out their fabric, they still need more revenue to purchase fabric to sustain the store and its customers. At the same time, Miele, Reggie and Marion are excited with the success of the local store selling out of fabrics, and partner with another organization in Asia to open another fabric store. Over time, Ben and Klaus become increasingly concerned as to whom the fabrics the business they started with Leonard, Reggie and Marion, is for. One day, the team has a meeting to discuss what would be considered successful in the business. For Ben and Klaus, success to them meant having one sustainable store that was making profit first, followed by expansion. For Miele, Reggie and Marion, it meant immediately partnering with other organizations that sometimes competed with their main store to sell their fabric. Their view was that they could get the business to profit quicker, but it also required everyone in the organization put it their own personal funds to do this. Due to the views on success and the path forward not being aligned, Ben, Klaus, Miele, Reggie and Marion, decided to move their separate ways.

Much like the analogy above, StitchKit faced similar issues as to what was considered successful within our team. One of the domino effects of Lesson 11: Making Hardware Sustainable as A Business is Even Harder, was that the path forward post-Kickstarter campaign was difficult. A number of tough decisions needed to be made, such as were we going to become a vendor of kits or partner with organizations, or did we want StitchKit to become a software platform and entirely disregard hardware, which meant pivoting away from a successful Kickstarter campaign. Broadly, the team had different measures of success for what StitchKit should be both in the short term, and in the long term, which meant that the vision for the product never truly aligned to begin with.

On the research side, I always felt it was easy to determine success, whether it was a top quality publication or presentation, but on the entrepreneurial side, success seemed to mean different things, as it depended on the type of business, personalities involved and other external factors. At the end of the day, one of the biggest lessons for me in StitchKit, was that short term and long term success needs to be clearly defined as a team before ever properly beginning a startup.

11.5.4 *Responsibility*

Throughout my reflections, I have referenced *Lesson 3: Building an Organization Quickly vs. Effectively continuously*, emphasizing its importance in different areas in a startup. One of the last areas I'd like to touch upon in its importance relates to responsibility.

Regardless of any metric of success for a startup, in order to have success there are a number of tasks that the startup is responsible for in order to succeed. For SITS in Chapter 5, my role and its responsibilities were quite clear, especially as I had several titles (e.g. product manager). But for startups like Slate Scale and Emotica, the roles and responsibilities between my co-founder and I were clear, we defined the division of tasks based on our preferences and skillset. Additionally, in both startups, accountability for responsibilities was maintained with a paper trail of our progress in tasks, decision making processes, our meetings with each other as well as VCs, mentors and others. Keeping a paper trail made a significant difference in ensuring our startup was responsible, regardless of the difficult decisions that needed to be made.

On the flip side, one of the challenges for StitchKit was the difficulty in ensuring this type of responsibility for some of the tasks and areas described earlier in this section. Often when working on a startup part-time, which is often the case for a new co-founding team, having synchronized schedules, consistent meeting times (both physical or digital), and keeping track of tasks can be difficult. To frame this point further, I will re-use the fabric shop analogy.

Ben, Klaus, Miele, Reggie and Marion are a few months into the opening of their fabric shop. The entire team has been working hard for the shop, but all also have other part-time jobs and often split their tasks between each other. At first, they are able to manage their tasks and responsibilities. However, as Miele, Reggie and Marion began to have an interest in expanding the fabrics and shop to overseas markets, they then spend an increasing amount of time away from the shop, leaving Ben and Klaus to manage more responsibilities. Ben and Klaus at first try to offset the challenges that come with more responsibilities by trying to work with Miele, Reggie and Marion, remotely. The time differences between the team becomes a major challenge in managing their responsibilities, as it becomes difficult to track both locally and remotely.

Similar to the analogy above, the responsibilities for StitchKit were distributed amongst a team that was both local and remote at times. This created a challenge for us because we were not able to manage our responsibilities from the beginning, which meant as more time passed and more tasks and responsibilities arose, there was a time cost. Often this increasing time cost meant that a co-founder would have to make a difficult choice between StitchKit and something else (eg. family, academia, or even other businesses).

Without building a clear accountability and responsibility structure for a startup in the beginning, there will be a number of challenges that may ultimately lead it to failure. Thus, always document everything from the start, and keep track of progress, as it will be one less thing to worry about when doing a startup.

11.5.5 *Always Protect Yourself*

The last lesson I describe in my entrepreneurial journey is not a simple one. On the academic side, as a student, I was fortunate enough to work with amazing people and could grow and learn, and for the most part, it was similar on the entrepreneurial side. But in StitchKit, towards the latter part of its eventual end, a number of difficult conversations arose towards its end, part of which I hinted at in the previous lesson.

The most difficult of these conversations was how to split the assets of StitchKit in a fair and equitable manner between the team in the startup. This type of conversation is one that can quickly turn friendships into a nightmare. To avoid these types of situations, one of the most critical things to do when beginning a startup, regardless of whether it is with friends or colleagues, is to protect yourself (and everyone else), by investing in a proper shareholders agreement. A shareholders agreement is critical in determining how decisions are made within an organization (e.g. by majority, unanimous decision, etc), in addition to deciding how stocks are assigned to the founders, and a host of other areas.

Further, a shareholders agreement can break deadlocks and hold everyone accountable in a startup, which is particularly important when difficult situations arise in a startup, such as determining equity, when founders want to exit, or if a founder is becoming problematic. As a student, I strongly believe that it is necessary to understand and spend time on creating shareholder agreements that protect you. Over the years, I learned that not everyone has your best interest in mind, regardless of how they present themselves at the beginning. One of the most effective methods to combat this is, is to have a solid shareholders agreement in place. You should always hope for the best in a startup you co-found, but you should also always protect yourself as well.

11.6 FINAL REFLECTIONS

Of all the startups thus far in the entrepreneurial journey described in this dissertation, StitchKit was simultaneously the most successful but also the most difficult. Some of these difficulties are described in the in this section, but as I reflect back now, it took me a significant amount of time to see the warning signs for StitchKit and its eventual failure post-Kickstarter campaign. Following StitchKit, I immediately began a Not-For-Profit (NFP) to continue some of the more important work I was trying to accomplish with StitchKit, and that was, to help people learn and be inspired using technology combined with fashion as the vehicle. Without StitchKit and its difficult lessons and journey, I would not have been able to distill what was important to me in what the product was trying to do, my vision of what fashion-tech could and should be, as well as the entrepreneur I wanted to become in the future. It ultimately had a significant impact on the work that proceeded it.

Part IV

TO WRAP UP

In this last section of the dissertation, I first provide an abridged and visual summary of the entrepreneurial lessons described throughout this dissertation, in Chapter 12. I then conclude with a discussion on future research and entrepreneurial directions, as well as a final reflection on entrepreneurship and research in Chapter 13

12

MY ENTREPRENEURIAL PRIMER

"By three methods we may learn wisdom: First, by reflection, which is noblest; Second, by imitation, which is easiest; and third by experience, which is the bitterest."

Confucius

In Chapters 5, 8 and 11, I detailed a number of reflections and lessons that I encountered and built upon as a student. As is evident, my experience was varied, with both positive and negative aspects. In Figure 57 I summarize the lessons from Chapter 5, while in Figure 1 I summarize the lessons from Chapter 8. Lastly, in Figure 59, I summarize the lessons from Chapter 11.

By no means are my reflections complete, nor my lessons an exhaustive list. After all, I have only experienced the entrepreneurial journey as a student thus far. There will always be more to learn, and thus more to add to a list such as this. As a student, it is critical to use these lessons and reflections as a starting point to avoid many of the challenging pitfalls that resonated with me. As I move forward in my entrepreneurial career beyond academia, I will certainly be building upon this list.

Figure 57: Lessons from Chapter 5 - Beyond SkyHunter: Reflecting on Commercialization.

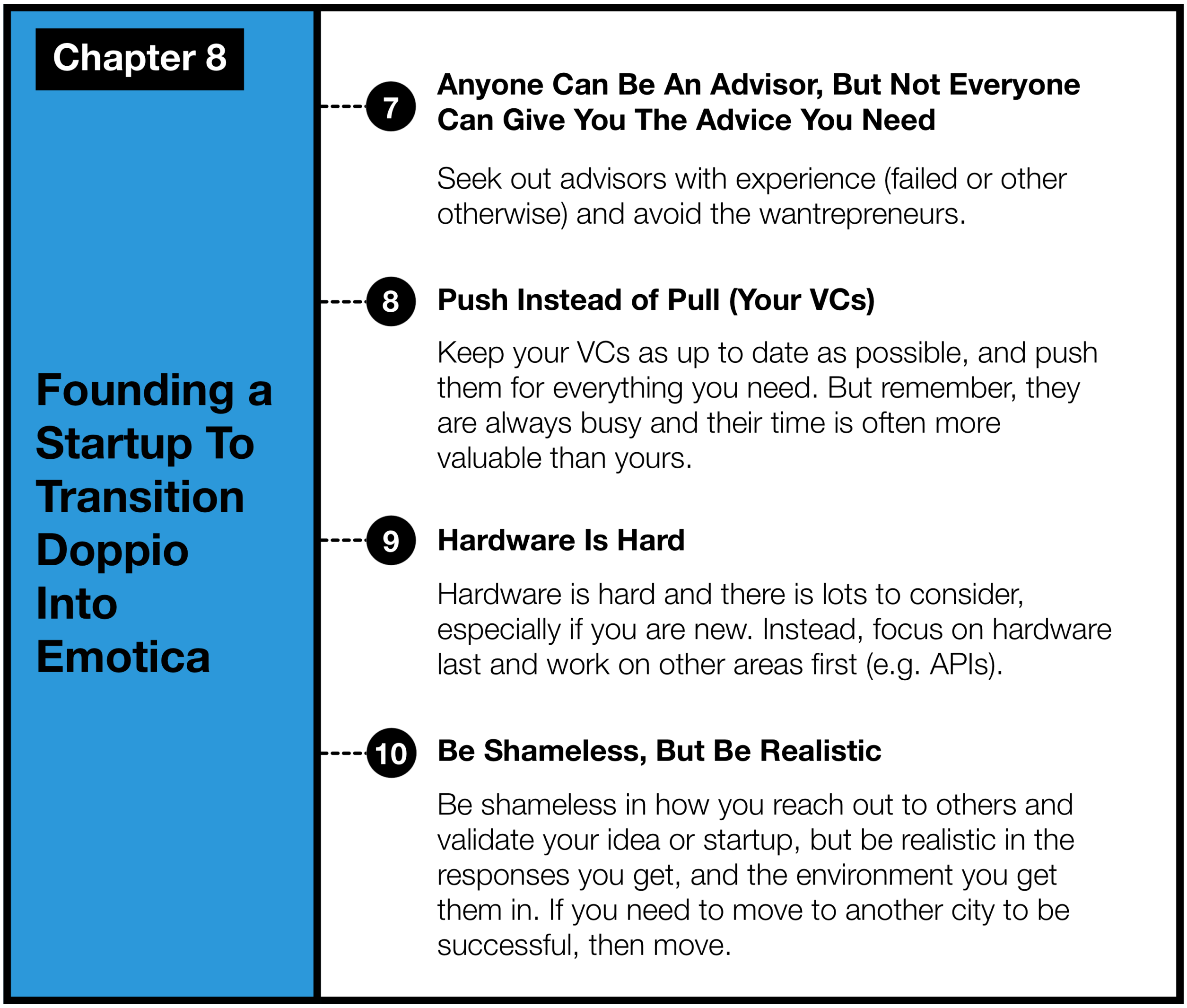


Figure 58: Lessons from Chapter 8 - Pivoting a Research Concept: From Doppio to Emotica.

Chapter 11

Founding StitchKit which ran a successful Kickstarter campaign

11 Making Hardware Sustainable is Even Harder

Assuming you've made hardware successfully, you need to consider things like overall cost, and all the tasks after it (e.g. support, testing) to make a better judgment if it is actually sustainable to continue.

12 Timing is Everything

Don't rush into your launching your product because you must. Instead, understand the market for it and then decide accordingly. If you have to delay, delay.

13 Define Success Within Your Team

Success means different things to different people. When building your team, define what it means to everyone individually and for the startup.

14 Responsibility

As a team, document and track tasks and responsibilities in the startup. When something isn't working or is wrong, hold each other accountable.

15 Always Protect Yourself

Things like shareholder agreements are important in protecting yourself and others from bad startup situations. Not everyone you meet will always have your best interest in mind afterall.

Figure 59: Lessons from Chapter 11 - StitchKit and The Perils Of Crowdfunding.

13

CONCLUSION AND FUTURE WORK

Ubiquitous Computing enables us to take advantage of the wider world around us, and use computing in a number of different ways. However, in order to enable the interactions and environments necessary to make this happen, a number of challenges need to be overcome. These include the technical challenges required for integrating different sensors and devices, to designing interactions for the bevy of new devices with different form factors, to the overall design of the environment itself. The ultimate goal in overcoming these challenges is to have our devices and technology being "*indistinguishable*" in the environment.

Throughout this dissertation, I have explored and demonstrated solutions and entrepreneurial reflections in several of these areas and more, which has resulted in a collection of work that I believe significantly advances Ubiquitous Computing as a whole, defines the state of the art for novel everyday devices, such as smartwatches and smartphones, and provides new perspectives, prototyping tools, and research methodologies for increasingly important domains in the context of Ubiquitous Computing, such as fashion-technology.

I began in Chapter 2, where I demonstrated an example of a Ubiquitous Computing toolkit (SoD Toolkit) and then systematically (and pragmatically) explored them in two real-world examples, shown in Chapter 3 and 4. In Chapter 5, I further explored a Ubiquitous Computing environment (specifically a multi-surface environment) through the lens of entrepreneurship, demonstrating and reflecting upon several valuable lessons. In Chapters 6 and 7, I explored novel types of everyday devices, demonstrating that the rethinking of devices in the context of our daily tasks, and environment, is feasible and within reach. In Chapter 8, I further explored commercializing one of these devices in a startup, and reflect upon the challenges for commercializing technology in the consumer space. In Chapters 9 and 10, I then explored how to design tools to facilitate non-technical users to participate in the design of ubiquitous systems that incorporate fashion, and gaming. Finally, in Chapter 11 I provided a final reflection on a product I co-founded, that was also designed to aid non-technical users in prototyping ubiquitous systems, specifically those that involve fashion. I conclude my entrepreneurial reflections and lessons with a visual summary of the lessons learned in my entrepreneurial journey in Chapter 12.

In this final chapter, I conclude by summarizing the major contributions of this dissertation, and highlight several promising areas for future exploration for research and entrepreneurship.

13.1 SUMMARY OF CONTRIBUTIONS

13.1.1 *Society of Devices Toolkit and a Multi-Domain Exploration*

In Chapter 2, I introduced the Society of Devices (SoD) Toolkit, which combined sensors (such as Microsoft Kinects or iBeacons) and devices (such as tablets and smartphones) to create interactive Multi-Surface Environments (MSEs), which are a type of Ubiquitous Computing environment. The SoD Toolkit provided a number of novel technical innovations — application programming interfaces (APIs) to make it easier to use different software platforms, a plug and play architecture for popular off-the-shelf hardware (sensors and devices) that is extensible, programming abstractions for complex sensor fusion and algorithms that are necessary to create MSEs, and the ability to virtually prototype an interactive environment without the need for physical hardware.

Furthermore, SoD Toolkit was designed to enable real-world deployments in different domains, which I explored in Chapter 3 with SkyHunter, and Chapter 4 with ePlan. In SkyHunter, the SoD Toolkit was used to explore the oil and gas domain, where its exploration revealed the necessity for more simplified interactions, the strengths of multi-device collaboration and the challenges with deploying Ubiquitous Computing environments. With ePlan, I explored the emergency response planning domain, and revealed the critical need for privacy and security in MSEs. The research insights learned from these domains and deployment, also influenced the entrepreneurial work described in Chapter 5.

13.1.2 *Novel Devices and Interaction Techniques*

In Chapters 2, 3, and 4, I did not explore how to better incorporate devices such as smartwatches and smartphones in MSEs, which have their own inherent challenges. Some of the interaction challenges for these devices are limited input space (for the smartwatch) or the ability to share certain types of information on a device (for all devices).

In Chapter 6 I presented Doppio, a re-configurable dual screened smartwatch that was designed to address the limited interaction space of smartwatches. Using the dual screens — one of which is detachable — I introduced a number of novel tangible interaction techniques. These techniques and their feasibility were demonstrated in a fully-functional hardware prototype and a number of demonstration applications built on top of Android Wear. The hardware design, approach to combining the two screens, and the implementation of 124 different interactions are all novel components from a technical standpoint, and will be valuable for the design of future smartwatches and devices that incorporate multiple screens.

In the Lendable phone described in Chapter 7, I explored issues with trust and convenience in our everyday devices. How we share these devices and their information — physically and otherwise — is an underexplored area, both in the context of MSEs and the wider research space. Through this lens, I examined typical lending scenarios for everyday devices, specifically the smartphone. I then demonstrated how to consider trust and convenience in everyday lending scenarios (e.g. lending your phone to a family member to look at photos) by redesigning the smartphone itself. This was accomplished through a fully-functional (modular) hardware prototype built on top of Android, where individual modules could be detached and shared with another person. The functionality of each piece is dependent upon the lending scenario initiated by the owner of the device (or lender). The exploration of designing for lending (or sharing), the software and hardware approach in combining multiple screens into a singular device, as well as the interaction techniques for detachment are all novel components from a technical and HCI perspective. This is also the first known example of redesigning an everyday device specifically for lending, and is a fundamental first step into providing insight for future devices (e.g. foldables) where lending will become more commonplace. While both Doppio and the Lendable phone are prototypes, each demonstrates *"indistinguishable"* by incorporating different interaction techniques around lending and everyday practices into the devices themselves.

13.1.3 *Facilitated Prototyping Tools for Novices*

An alternate approach to designing a wearable device like Doppio is to consider the domain of fashion. Fashion has a wealth of knowledge in designing clothing and other accessories worn on the body, and this knowledge can and should be applied to the design of wearables (and other devices), especially as the incorporation of digital technologies (e.g. sensors) can contribute to a more *"indistinguishable"* Ubiquitous Computing environment.

In Doppio, I did not explore or initially consider fashion. However, on the entrepreneurial side, I explored fashion in its redesign, as having a well designed device (wearable or not) was important for critical and commercial success. One of the challenges I uncovered when working with fashion designers to create a well-designed dual-screened smartwatch, was that they were very creative, but lacked the technical skills and verbiage to collaboratively assist and communicate in the design of my new smartwatch. In the context of Ubiquitous Computing, this led to an interesting and pragmatic research question: How do we enable creative, but non-technical users (e.g. fashion designers) who have valuable skills, to fundamentally contribute and redefine the space.

I first explored this by embedding myself in a specific domain to start, specifically a fashion-technology runway organization for 2.5 years, where I served in various roles. I observed how fashion designers with different levels of technical skill, created garments that were augmented with technology, and how they created interactions for them. As mentioned earlier, one of the key challenges these designers have is the difficulty in communicating technical concepts for their wearables (i.e. garments). To address this, I provided the first documented description of the design processes for fashion-technology garments that appear in highly expressive fashion-tech runway shows. Using this defined process, I then explored how to address several of the communication challenges, by designing a novel prototyping tool called Mannequette, presented in Chapter 9. Mannequette enables a plug and play approach to designing and experimenting with interactions, sensors and outputs at the earliest stages of the aforementioned design process, in a substantially more collaborative manner. Mannequette was used in an 8-week deployment study with several teams who conceptualized and realized multiple types of garments for a runway show. These explorations of garments that integrated technology are a step forward for the “indistinguishable” Ubiquitous Computing environment, as sensors which can contribute to the environment, are now hidden.

The description of the avant-garde fashion-tech process, the approach and design of Mannequette, its technical implementation (both hardware and software), as well as the methodology of using a fashion show as the context for its research are all novel contributions. As fashion and digital technologies begin to blend further, much of the insight Mannequette provides will be critical in how future wearables that incorporate fashion are designed in the future.

In Mannequette, one of the other lessons learned was the difficulty non-technical users faced when attempting to program the sensors and devices beyond the interactions that were provided. From a research perspective, this meant an initial exploration was necessary to understand alternate methods of introducing programming with sensors and interactions to novices, regardless of domain. My initial exploration into this research area was MakerArcade — described in Chapter 10 — which incorporated block-based programming and sensors into the design of video games, another creative domain. MakerArcade makes it easier to prototype and design video games (using block-based programming) that use existing plug and play sensors, beyond those that were used in Mannequette. It emphases designing, remixing, and customizing physical and digital components, a common task in many design processes beyond video games. Ultimately, MakerArcade provides a number of novel technical contributions, from the integration of a block-based programming platform and different sensors, to the initial exploration in video game design.

13.1.4 *A Reflection on Entrepreneurship and HCI*

In Chapters 5, 8 and 11, I provided a number of entrepreneurial reflections and lessons learned as a student. While there are a countless number of publications, books, blogs and other forms of media that provide entrepreneurial reflections, none have particularly charted the path of commercialization from a research concept in HCI, to a product and beyond, let alone from the perspective of a student. My aim was to provide a critical assessment of the pitfalls of my entrepreneurial journey to date for future students who wish to combine entrepreneurship with their HCI studies. As multidisciplinary approaches become more commonplace in educational institutions as a whole, the reflections and lessons I have provided are the first to have woven research and entrepreneurship in this manner for a dissertation. Lastly, my reflections also provide value for professionals who wish to combine HCI and business, as my reflections are from the perspective of an HCI practitioner.

13.2 FUTURE WORK

Each contribution area that I have completed has opened up additional research questions and possibilities to explore. In this section, I highlight and describe promising research and entrepreneurial questions which can further augment our capabilities in the rapidly growing space of Ubiquitous Computing.

13.2.1 *Combining Mixed-Reality Environments with Multi-Surface Environments*

The SoD Toolkit (Chapter 2) was completed at a time where devices and sensors such as digital tabletops and iBeacons were novel. Since that time, a number of technologies have emerged and come to the forefront of Ubiquitous Computing, specifically Augmented Reality (AR) and Mixed Reality (MR). AR blends the digital world with the physical world, while MR takes this blending even further, mixing physical and digital interactions with objects. The next research steps for the work introduced in SoD-Toolkit, is to incorporate AR and MR. From a technical standpoint, the APIs, to the extensible architecture design, to the new interactions enabled by AR and MR, provide the basis for a number of technical contributions. Additionally, how AR and MR can work from an interaction perspective, in environments with multiple displays, provides an interesting area of research, especially when considering the need for interfaces needing to be redesigned, let alone exploring them in the context of a domain, such as emergency response planning.

13.2.2 *Exploring Other Devices That Can Be Shared*

In Doppio and the Lendable phone (Chapter 6 and 7), I began the first steps in exploring how we can rethink and redesign everyday devices to be shared. For both the phone and smartwatch, a number of different form factors, modalities and even scenarios are yet to be explored. For example, if Doppio was designed to be a dual-faced gaming smartwatch, how would its interaction techniques change, or if the Lendable phone was designed to share content, and not just applications as it currently does, would its modular design change. Additionally, I did not explore how these devices would be incorporated into MSEs, and the interactions they would enable with other devices such as digital tabletops or wall displays.

Lastly, smartwatches and smartphones are just two examples of devices that I explored. There are other devices we share, such as laptops and tablets, each of which has their own design considerations and scenarios, particularly when it comes to trust and convenience. Rethinking these devices in this context can lead to highly novel devices that influence future devices beyond Doppio and the Lendable phone.

13.2.3 *Considering The Design of Hardware For Ubiquitous Computing*

One area I did not fully explore, but briefly touched upon in the work of Mannequette and MakerArcade (Chapter 9 and 10), was the design of plug and play hardware. For Fashion-tech, many of the existing hardware components are not properly designed to be worn on the body, and more importantly, do not properly consider the fashion-tech design process, as mentioned in Chapter 9. To move the space of fashion-tech forward, especially in the context of providing more appropriate prototyping tools for non-technical users, the design of novel plug and play hardware should be explored. Techniques such as flexible PCBs, and different methods for connecting hardware (beyond the audio cables used in MakerArcade), as well as software integrations for block-based programming are key research opportunities that can significantly advance the state of the art in fashion-tech and the broader space of prototyping the "*indistinguishable*" environment involving clothing in Ubiquitous Computing.

13.2.4 *To Be or Not to Be (an Entrepreneur)*

As evidenced by my entrepreneurial reflections, being an entrepreneur and engaging in its activities is not easy. There are several challenges, in addition to the typical challenges a student researcher faces (e.g. lack of resources). However, with this in mind, one interesting and meaningful avenue of work is to take the lessons I have described in this dissertation and undertake a full exploration from start to finish, and compare the results with those of this dissertation. For the most part, my entrepreneurial journey overlapped with my research, but it was also slightly disjointed as it was not my only focus. I also did not have a list of entrepreneurial reflections from the perspective of a student to guide me. I fully encourage and even challenge future students to take these reflections, and build upon them if they decide to pursue entrepreneurship. In doing so, I believe future students will be more successful than I was, and thus answer the question of to be or not to be (an entrepreneur).

13.3 CONCLUSION

Ultimately, the journey undertaken in this dissertation was to enable our everyday devices to "*weave themselves into the fabric of everyday life until they are indistinguishable from it*" (Weiser (1999)). Certainly, the landscape of computing today accomplishes many of the goals of Ubiquitous Computing that Weiser had laid out. With every new device, the raw computing power of the environment is also increased.

However, this potential is untapped if it is not truly a part of the environment in which we accomplish our daily tasks. For the most part, how we have used our devices has followed a distinct pattern: we are either using it, or we are interacting with the environment. Devices such as smartphones and smartwatches have further enhanced this pattern, as they often require the attention of a user. But in a Ubiquitous Computing environment, this is a limitation.

This dissertation offers a cross-disciplinary exploration in how to design and enable Ubiquitous Computing from the perspective of different domains, devices, and the users in them.

By exploring the space in this manner, it demonstrated a critical feasibility in how to shape Ubiquitous Computing as new technologies begin to emerge. I explored this space – both through research and entrepreneurial activities – not to replace existing devices or technologies, but instead to augment our current capabilities, and enable new forms of interactions.

As we move forward, undoubtedly the computing paradigms we are familiar with today will change. How we think about our devices, how we use them, will also change. What will remain constant is that these computing elements will be the building blocks of our environment, and eventually, we'll never even know they were there to begin with.

Part V

BIBLIOGRAPHY

BIBLIOGRAPHY

Lenovo's 'Magic View' smartwatch concept hides a private display. (Cited on page 97.)

Men's rose gold watch - Piaget Luxury Watch G0a35153. (Cited on page 97.)

Reverso watch for men and women. (Cited on page 97.)

Titanium Two Face Watch, 20mm (Juniors). (Cited on page 97.)

(2011). Porsche Design P'6520 Heritage Compass Watch Hands-On. (Cited on page 97.)

(2019a). CH MAKER Ed | tech | fun | edu. (Cited on pages 184 and 193.)

(2019b). FLORA - Wearable electronic platform: Arduino-compatible [v3] ID: 659 - $14.95 : Adafruit Industries, Unique & fun DIY electronics and kits. (Cited on pages 183, 192, 193, and 206.)

(2019c). Grove System. (Cited on pages 196 and 208.)

(2019d). Homes - Mindstorms LEGO.com. (Cited on page 211.)

(2019e). Kodu | Home. (Cited on pages 211 and 216.)

(2019f). littleBits: Award-winning electronic building blocks for creating inventions large and small. (Cited on page 184.)

(2019g). MakeCode Arcade. (Cited on page 212.)

(2019h). MakeCode Maker - Blocks / Javascript editor. (Cited on page 213.)

(2019i). MakeFashion. (Cited on page 185.)

(2019j). Stitch Kit. (Cited on pages 183 and 184.)

(2019k). What is JACDAC? (Cited on page 219.)

adafruit (2012). Announcing the FLORA, Adafruit's wearable electronics platform and accessories. (Cited on pages 180, 183, 184, and 192.)

Ashbrook, D., Lyons, K., and Starner, T. (2008). An Investigation into Round Touchscreen Wristwatch Interaction. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '08, pages 311–314, New York, NY, USA. ACM. event-place: Amsterdam, The Netherlands. (Cited on pages 96 and 97.)

Aug 19, C. N. P., August 19, . . P. M. |. L. U., and 2013 (2013). Calgary police keeping cameras on uniforms | CBC News. (Cited on pages 69 and 71.)

Azazi, A., Seyed, T., Botros, F., Sabourin, D., Chan, E., and Maurer, F. (2013). Using multiple kinects to build larger multi-surface environments. In *Workshop on collaboration meets interactive surfaces in ACM interactive tabletops and surfaces*. Citeseer. (Cited on page 63.)

Bakker, S., van den Hoven, E., and Eggen, B. (2010). Design for the Periphery. *EuroHaptics 2010*, 71. (Cited on page 57.)

Ball, T., Protzenko, J., Bishop, J., Moskal, M., Halleux, J. d., Braun, M., Hodges, S., and Riley, C. (2016). Microsoft touch develop and the BBC micro:bit. In *Proceedings of* *the 38th International Conference on Software Engineering Companion*, pages 637–640, Austin, Texas. ACM. (Cited on pages 183, 184, and 193.)

Ballendat, T., Marquardt, N., and Greenberg, S. (2010). Proxemic Interaction: Designing for a Proximity and Orientation-aware Environment. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 121–130, New York, NY, USA. ACM. event-place: Saarbrücken, Germany. (Cited on pages 8 and 11.)

Bdeir, A. and Ullrich, T. (2011). Electronics as material: littleBits. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, pages 341–344, Funchal, Portugal. ACM. (Cited on page 184.)

Berglund, M. E., Foo, E., Molla, M. T. I., Sudheendra, S. M., Compton, C., and Dunne, L. E. (2018). MAKE IT BLUE: A Controllable, Color-changing Dynamic Costume. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, ISWC '18, pages 236–241, New York, NY, USA. ACM. (Cited on pages 192 and 207.)

Blasko, G. and Feiner, S. (2004). An Interaction System for Watch Computers Using Tactile Guidance and Bidirectional Segmented Strokes. In *Proceedings of the Eighth International Symposium on Wearable Computers*, ISWC '04, pages 120–123, Washington, DC, USA. IEEE Computer Society. (Cited on page 96.)

Blasko, G., Narayanaswami, C., and Feiner, S. (2006). Prototyping Retractable String-based Interaction Techniques for Dual-display Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 369–372, New York, NY, USA. ACM. event-place: Montréal, Québec, Canada. (Cited on pages 96 and 100.)

Bortolaso, C., Oskamp, M., Graham, T. N., and Brown, D. (2013). OrMiS: A Tabletop Interface for Simulation-based Training. In *Proceedings of the 2013 ACM International*

Conference on Interactive Tabletops and Surfaces, ITS '13, pages 145–154, New York, NY, USA. ACM. event-place: St. Andrews, Scotland, United Kingdom. (Cited on pages 57 and 58.)

Bowering, D. (1999). Data Visualization Centers. *Oil & Gas Executive*, 2(2):32–35. (Cited on page 37.)

Bragdon, A., DeLine, R., Hinckley, K., and Morris, M. R. (2011). Code Space: Touch + Air Gesture Hybrid Interactions for Supporting Developer Meetings. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, pages 212–221, New York, NY, USA. ACM. event-place: Kobe, Japan. (Cited on page 136.)

Brahms, L. (2014). *Making as a learning process: Identifying and supporting family learning in informal settings*. PhD thesis, University of Pittsburgh. (Cited on page 211.)

Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. A. (2000). EasyLiving: Technologies for Intelligent Environments. In *Proceedings of the 2Nd International Symposium on Handheld and Ubiquitous Computing*, HUC '00, pages 12–29, London, UK, UK. Springer-Verlag. event-place: Bristol, UK. (Cited on page 12.)

Brush, A. J. B. and Inkpen, K. M. (2007). Yours, Mine and Ours? Sharing and Use of Technology in Domestic Environments. In Krumm, J., Abowd, G. D., Seneviratne, A., and Strang, T., editors, *UbiComp 2007: Ubiquitous Computing*, Lecture Notes in Computer Science, pages 109–126. Springer Berlin Heidelberg. (Cited on pages 130, 131, 132, 134, and 138.)

Buechley, L., Eisenberg, M., Catchen, J., and Crockett, A. (2008). 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*, CHI '08, pages 423–432, New York, NY, USA. ACM. (Cited on pages 183, 184, 192, 207, 211, 215, and 225.)

Buechley, L. and Hill, B. M. (2010). LilyPad in the wild: how hardware's long tail is supporting new engineering and design communities. pages 199–207. ACM. (Cited on pages 183, 184, 191, 192, 207, and 211.)

Burstyn, J., Strohmeier, P., and Vertegaal, R. (2015). DisplaySkin: Exploring Pose-Aware Displays on a Flexible Electrophoretic Wristband. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, pages 165–172, New York, NY, USA. ACM. event-place: Stanford, California, USA. (Cited on page 97.)

Bødker, S. and Christiansen, E. (2012). Poetry in Motion: Appropriation of the World of Apps. In *Proceedings of the 30th European Conference on Cognitive Ergonomics*, ECCE '12, pages 78–84, New York, NY, USA. ACM. event-place: Edinburgh, United Kingdom. (Cited on pages 130, 133, and 134.)

Chen, K.-Y., Lyons, K., White, S., and Patel, S. (2013). uTrack: 3d Input Using Two Magnetic Sensors. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 237–244, New York, NY, USA. ACM. event-place: St. Andrews, Scotland, United Kingdom. (Cited on pages 100 and 118.)

Chen, X. A., Grossman, T., and Fitzmaurice, G. (2014a). Swipeboard: A Text Entry Technique for Ultra-small Interfaces That Supports Novice to Expert Transitions. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 615–620, New York, NY, USA. ACM. event-place: Honolulu, Hawaii, USA. (Cited on page 95.)

Chen, X. A., Grossman, T., Wigdor, D. J., and Fitzmaurice, G. (2014b). Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 159–168, New York, NY, USA. ACM. event-place: Toronto, Ontario, Canada. (Cited on pages 13, 94, and 96.)

Chi, P.-Y. P. and Li, Y. (2015). Weave: Scripting Cross-Device Wearable Interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3923–3932, New York, NY, USA. ACM. event-place: Seoul, Republic of Korea. (Cited on pages 14 and 28.)

Chokshi, A., Seyed, T., Marinho Rodrigues, F., and Maurer, F. (2014). ePlan Multi-Surface: A Multi-Surface Environment for Emergency Response Planning Exercises. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ITS '14, pages 219–228, New York, NY, USA. ACM. event-place: Dresden, Germany. (Cited on page 25.)

Corbett, P., Williams, G., Gosselin, O., Coleau, T., and Christie, M. (2011). Interactive Exercise on a Field Data Set. (Cited on page 34.)

Cosentino, L. (2001). *Integrated Reservoir Studies*. Editions Technips, Paris. (Cited on page 37.)

Cremonesi, P., Rienzo, A. D., Garzotto, F., Oliveto, L., and Piazzolla, P. (2016). Dynamic and Interactive Lighting for Fashion Store Windows. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 2257–2263, San Jose, California, USA. ACM. (Cited on page 183.)

Dabby, N., Aleksov, A., Lewallen, E., Oster, S., Fygenson, R., Lathrop, B., Bynum, M., Samady, M., Klein, S., and Girouard, S. (2017). A scalable process for manufacturing integrated, washable smart garments applied to heart rate monitoring. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 38–41, Maui, Hawaii. ACM. (Cited on page 182.)

Dachselt, R. and Buchholz, R. (2009). Natural Throw and Tilt Interaction Between Mobile Phones and Distant Displays. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 3253–3258, New York, NY, USA. ACM. event-place: Boston, MA, USA. (Cited on pages 36, 58, and 136.)

Devendorf, L., Lo, J., Howell, N., Lee, J. L., Gong, N.-W., Karagozler, M. E., Fukuhara, S., Poupyrev, I., Paulos, E., and Ryokai, K. (2016). "I Don't Want to Wear a Screen": Probing Perceptions of and Possibilities for Dynamic Displays on Clothing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 6028–6039, New York, NY, USA. ACM. (Cited on page 191.)

Devine, J., Finney, J., de Halleux, P., Moskal, M., Ball, T., and Hodges, S. (2018). MakeCode and CODAL: Intuitive and Efficient Embedded Systems Programming for Education. In *Proceedings of the 19th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems*, LCTES 2018, pages 19–30, New York, NY, USA. ACM. (Cited on page 216.)

Dierk, C., Nicholas, M. J. P., and Paulos, E. (2018). AlterWear: Battery-Free Wearable Displays for Opportunistic Interactions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–11, Montreal QC, Canada. ACM. (Cited on page 191.)

Dierk, C., Vega Gálvez, T., and Paulos, E. (2017). AlterNail: Ambient, Batteryless, Stateful, Dynamic Displays at Your Fingertips. In *Proceedings of the 2017 CHI Conference on*

Human Factors in Computing Systems, CHI '17, pages 6754–6759, New York, NY, USA. ACM. (Cited on page 182.)

Edge, D. and Blackwell, A. F. (2009). Peripheral Tangible Interaction by Analytic Design. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI '09, pages 69–76, New York, NY, USA. ACM. event-place: Cambridge, United Kingdom. (Cited on page 57.)

Egelman, S., Brush, A. B., and Inkpen, K. M. (2008). Family Accounts: A New Paradigm for User Accounts Within the Home Environment. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, CSCW '08, pages 669–678, New York, NY, USA. ACM. event-place: San Diego, CA, USA. (Cited on page 134.)

Eisenberg, M., Elumeze, N., MacFerrin, M., and Buechley, L. (2009). Children's Programming, Reconsidered: Settings, Stuff, and Surfaces. In *Proceedings of the 8th International Conference on Interaction Design and Children*, IDC '09, pages 1–8, New York, NY, USA. ACM. (Cited on page 211.)

Fowler, A. (2017). Engaging young learners in making games: an exploratory study. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, pages 1–5, Hyannis, Massachusetts. ACM. (Cited on page 216.)

Gee, J. P. (2007). *Good video games+ good learning: Collected essays on video games, learning, and literacy*, volume 27. Peter Lang. (Cited on page 211.)

Girouard, A., Solovey, E. T., Hirshfield, L. M., Ecott, S., Shaer, O., and Jacob, R. J. K. (2007). Smart Blocks: a tangible mathematical manipulative. pages 183–186. ACM. (Cited on page 212.)

Gomes, A. and Vertegaal, R. (2015). PaperFold: Evaluating Shape Changes for Viewport Transformations in Foldable Thin-Film Display Devices. pages 153–160. ACM. (Cited on page 98.)

Gongini, B. I. (2019). Avant-Garde Fashion - A Modern Definition. (Cited on pages 179 and 207.)

Gov.UK. Emergency planning and preparedness: exercises and training. (Cited on pages 55, 56, and 57.)

Greenberg, S., Marquardt, N., Ballendat, T., Diaz-Marino, R., and Wang, M. (2011). Proxemic Interactions: The New Ubicomp? *Interactions*, 18(1):42–50. (Cited on pages 8, 11, and 15.)

Hall, E. T. (1990). *The Hidden Dimension*. Anchor, New York, reissue edition edition. (Cited on page 11.)

Hamilton, P. and Wigdor, D. J. (2014). Conductor: Enabling and Understanding Cross-device Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2773–2782, New York, NY, USA. ACM. event-place: Toronto, Ontario, Canada. (Cited on pages 14, 27, and 31.)

Hang, A., von Zezschwitz, E., De Luca, A., and Hussmann, H. (2012). Too Much Information!: User Attitudes Towards Smartphone Sharing. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, NordiCHI '12, pages 284–287, New York, NY, USA. ACM. event-place: Copenhagen, Denmark. (Cited on pages 130, 134, and 144.)

Harrison, C. and Hudson, S. E. (2009). Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. pages 121–124. ACM. (Cited on pages 94, 96, and 118.)

Hayashi, E., Riva, O., Strauss, K., Brush, A. J. B., and Schechter, S. (2012). Goldilocks and the Two Mobile Devices: Going Beyond All-or-nothing Access to a Device's Applications. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, SOUPS '12, pages 2:1–2:11, New York, NY, USA. ACM. event-place: Washington, D.C. (Cited on pages 131, 134, and 144.)

Herald, C. Calgary flood: EOC bigger and better than ever five years after 2013 | Calgary Herald. (Cited on pages 68 and 72.)

Hinckley, K. (2003). Synchronous Gestures for Multiple Persons and Computers. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 149–158, New York, NY, USA. ACM. event-place: Vancouver, Canada. (Cited on page 12.)

Hinckley, K., Dixon, M., Sarin, R., Guimbretiere, F., and Balakrishnan, R. (2009). Codex: A Dual Screen Tablet Computer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1933–1942, New York, NY, USA. ACM. event-place: Boston, MA, USA. (Cited on pages 94, 98, and 135.)

Hofstra, H., Scholten, H., Zlatanova, S., and Scotta, A. (2008). Multi-user tangible interfaces for effective decision-making in disaster management. In Nayak, S. and Zlatanova, S., editors, *Remote Sensing and GIS Technologies for Monitoring and Prediction of Disasters*, Environmental Science and Engineering, pages 243–266. Springer Berlin Heidelberg, Berlin, Heidelberg. (Cited on page 57.)

Holleis, P., Schmidt, A., Paasovaara, S., Puikkonen, A., and Häkkilä, J. (2008). Evaluating capacitive touch input on clothes. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 81–90, Amsterdam, The Netherlands. ACM. (Cited on page 182.)

Horowitz, B. (2014). *The Hard Thing About Hard Things: Building a Business When There Are No Easy Answers*. HarperBusiness. (Cited on page 82.)

Houben, S. and Marquardt, N. (2015). WatchConnect: A Toolkit for Prototyping Smartwatch-Centric Cross-Device Applications. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1247–1256, New York, NY, USA. ACM. event-place: Seoul, Republic of Korea. (Cited on pages 9, 13, 27, 29, 30, and 96.)

Howarth, B. The pioneers of the Internet of Things. (Cited on pages 70 and 71.)

Huang, D.-Y., Lin, C.-P., Hung, Y.-P., Chang, T.-W., Yu, N.-H., Tsai, M.-L., and Chen, M. Y. (2012). MagMobile: Enhancing Social Interactions with Rapid View-stitching Games of Mobile Devices. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pages 61:1–61:4, New York, NY, USA. ACM. event-place: Ulm, Germany. (Cited on page 14.)

Huang, D.-Y., Tsai, M.-C., Tung, Y.-C., Tsai, M.-L., Yeh, Y.-T., Chan, L., Hung, Y.-P., and Chen, M. Y. (2014). TouchSense: Expanding Touchscreen Input Vocabulary Using Different Areas of Users' Finger Pads. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 189–192, New York, NY, USA. ACM. event-place: Toronto, Ontario, Canada. (Cited on page 95.)

Izadi, S., Brignull, H., Rodden, T., Rogers, Y., and Underwood, M. (2003). Dynamo: A Public Interactive Surface Supporting the Cooperative Sharing and Exchange of Media. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 159–168, New York, NY, USA. ACM. event-place: Vancouver, Canada. (Cited on page 36.)

Jakobsen, M. R., Haile, Y. S., Knudsen, S., and Hornbæk, K. (2013). Information Visualization and Proxemics: Design Opportunities and Empirical Findings. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2386–2395. (Cited on page 11.)

Johanson, B., Fox, A., and Winograd, T. (2002). The Interactive Workspaces project: experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, 1(2):67–74. (Cited on page 36.)

Juhlin, O. and Zhang, Y. (2011). Unpacking social interaction that make us adore: on the aesthetics of mobile phones as fashion items. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 241–250, Stockholm, Sweden. ACM. (Cited on page 183.)

Juhlin, O., Zhang, Y., Sundbom, C., and Fernaeus, Y. (2013). Fashionable Shape Switching: Explorations in Outfit-centric Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1353–1362, New York, NY, USA. ACM. (Cited on page 183.)

Kafai, Y. B. (2006). Playing and Making Games for Learning: Instructionist and Constructionist Perspectives for Game Studies. *Games and Culture*, 1(1):36–40. (Cited on pages 211 and 215.)

Kafai, Y. B. and Ching, C. C. (2001). Affordances of Collaborative Software Design Planning for Elementary Students' Science Talk. *Journal of the Learning Sciences*, 10(3):323–363. (Cited on page 211.)

Kafai, Y. B. and Vasudevan, V. (2015). Constructionist Gaming Beyond the Screen: Middle School Students' Crafting and Computing of Touchpads, Board Games, and Controllers. In *Proceedings of the Workshop in Primary and Secondary Computing Education*, WiPSCE '15, pages 49–54, New York, NY, USA. ACM. (Cited on pages 211, 212, 214, and 215.)

Kansa, P. (2012). REVIEW: Ritmo Mundo Persepolis (Part 1). (Cited on page 97.)

Kapur, M. (2008). Productive Failure. *Cognition and Instruction*, 26(3):379–424. (Cited on page 211.)

Karlson, A. K., Brush, A. B., and Schechter, S. (2009). Can I Borrow Your Phone?: Understanding Concerns when Sharing Mobile Phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1647–1650, New York, NY, USA. ACM. event-place: Boston, MA, USA. (Cited on pages 130, 132, 133, and 142.)

Karrer, T., Wittenhagen, M., Lichtschlag, L., Heller, F., and Borchers, J. (2011). Pinstripe: eyes-free continuous input on interactive clothing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1313–1322, Vancouver, BC, Canada. ACM. (Cited on page 182.)

Kazemitabaar, M., McPeak, J., Jiao, A., He, L., Outing, T., and Froehlich, J. E. (2017). MakerWear: A Tangible Approach to Interactive Wearable Creation for Children. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 133–145, New York, NY, USA. ACM. (Cited on pages 184, 191, 192, 193, 194, 213, 214, and 215.)

Kim, H., Kim, Y., Kim, B., and Yoo, H.-J. (2009). A Wearable Fabric Computer by Planar-Fashionable Circuit Board Technique. In *Proceedings of the 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, pages 282–285. IEEE Computer Society. (Cited on pages 182 and 207.)

Kim, J., He, J., Lyons, K., and Starner, T. (2007). The Gesture Watch: A Wireless Contact-free Gesture based Wrist Interface. In *2007 11th IEEE International Symposium on Wearable Computers*, pages 15–22. (Cited on page 96.)

Kobayashi, K., Narita, A., Hirano, M., Kase, I., Tsuchida, S., Omi, T., Kakizaki, T., and Hosokawa, T. (2006). Collaborative Simulation Interface for Planning Disaster Measures. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 977–982, New York, NY, USA. ACM. event-place: Montréal, Québec, Canada. (Cited on page 58.)

Kortuem, G., Kray, C., and Gellersen, H. (2005). Sensing and Visualizing Spatial Relations of Mobile Devices. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 93–102, New York, NY, USA. ACM. event-place: Seattle, WA, USA. (Cited on page 13.)

Kray, C., Rohs, M., Hook, J., and Kratz, S. (2008). Group coordination and negotiation through spatial proximity regions around mobile devices on augmented tabletops. In *2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, pages 1–8. (Cited on page 13.)

König, W. A., Rädle, R., and Reiterer, H. (2009). Squidy: A Zoomable Design Environment for Natural User Interfaces. In *CHI '09 Extended Abstracts on Human Factors in Com-*

puting Systems, CHI EA '09, pages 4561–4566, New York, NY, USA. ACM. event-place: Boston, MA, USA. (Cited on page 14.)

Laibowitz, M. and Paradiso, J. A. (2004). The UbER-Badge, A Versatile Platform at the Juncture Between Wearable and Social Computing. In *Advances in Pervasive Computing*, pages 363–368. Oesterreichische Computer Gesellschaft. (Cited on page 12.)

Laput, G., Xiao, R., Chen, X. A., Hudson, S. E., and Harrison, C. (2014). Skin Buttons: Cheap, Small, Low-powered and Clickable Fixed-icon Laser Projectors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 389–394, New York, NY, USA. ACM. event-place: Honolulu, Hawaii, USA. (Cited on pages 94 and 96.)

Lawrence-Wilkes, L. and Ashmore, L. (2014). A Critical Framework. In Lawrence-Wilkes, L. and Ashmore, L., editors, *The Reflective Practitioner in Professional Education*, pages 56–67. Palgrave Macmillan UK, London. (Cited on pages viii and 3.)

Lepinski, J. and Vertegaal, R. (2011). Cloth displays: interacting with drapable textile screens. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, pages 285–288, Funchal, Portugal. ACM. (Cited on page 182.)

Liu, Y., Jia, J., Fu, J., Ma, Y., Huang, J., and Tong, Z. (2016). Magic Mirror: A Virtual Fashion Consultant. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, pages 680–683, New York, NY, USA. ACM. (Cited on page 179.)

Liu, Y., Rahmati, A., Huang, Y., Jang, H., Zhong, L., Zhang, Y., and Zhang, S. (2009). xShare: Supporting Impromptu Sharing of Mobile Phones. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, pages

15–28, New York, NY, USA. ACM. event-place: Kraków, Poland. (Cited on pages 130, 131, 132, 133, 134, 138, 145, 151, and 152.)

Lo, J., Lee, D. J. L., Wong, N., Bui, D., and Paulos, E. (2016). Skintillates: Designing and Creating Epidermal Interactions. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, DIS '16, pages 853–864, New York, NY, USA. ACM. (Cited on page 182.)

Lucero, A., Keränen, J., and Korhonen, H. (2010). Collaborative Use of Mobile Phones for Brainstorming. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, pages 337–340, New York, NY, USA. ACM. event-place: Lisbon, Portugal. (Cited on pages 12 and 14.)

Lyons, K., Nguyen, D., Ashbrook, D., and White, S. (2012). Facet: A Multi-segment Wrist Worn System. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 123–130, New York, NY, USA. ACM. event-place: Cambridge, Massachusetts, USA. (Cited on pages 94 and 97.)

Marquardt, N., Ballendat, T., Boring, S., Greenberg, S., and Hinckley, K. (2012a). Gradual Engagement: Facilitating Information Exchange Between Digital Devices As a Function of Proximity. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 31–40, New York, NY, USA. ACM. event-place: Cambridge, Massachusetts, USA. (Cited on pages 11 and 13.)

Marquardt, N., Diaz-Marino, R., Boring, S., and Greenberg, S. (2011). The Proximity Toolkit: Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 315–326, New York, NY, USA. ACM. event-place: Santa Barbara, California, USA. (Cited on pages 11, 14, 19, 28, and 31.)

Marquardt, N. and Greenberg, S. (2012). Informing the Design of Proxemic Interactions. *IEEE Pervasive Computing*, 11(2):14–23. (Cited on pages 11 and 28.)

Marquardt, N., Hinckley, K., and Greenberg, S. (2012b). Cross-device Interaction via Micro-mobility and F-formations. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 13–22, New York, NY, USA. ACM. event-place: Cambridge, Massachusetts, USA. (Cited on page 22.)

Matthews, T., Liao, K., Turner, A., Berkovich, M., Reeder, R., and Consolvo, S. (2016). "She'll Just Grab Any Device That's Closer": A Study of Everyday Device & Account Sharing in Households. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5921–5932, New York, NY, USA. ACM. event-place: San Jose, California, USA. (Cited on pages 130, 132, 133, 134, 138, 142, 145, 151, 152, 153, and 154.)

Mayer, S. and Sörös, G. (2014). User Interface Beaming – Seamless Interaction with Smart Things Using Personal Wearable Computers. In *2014 11th International Conference on Wearable and Implantable Body Sensor Networks Workshops*, pages 46–49. (Cited on page 13.)

Merrill, D., Kalanithi, J., and Maes, P. (2007). Siftables: Towards Sensor Network User Interfaces. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, TEI '07, pages 75–78, New York, NY, USA. ACM. (Cited on pages 94, 98, 135, 145, and 216.)

Murphy, L. L. and Priebe, A. E. (2011). "My co-wife can borrow my mobile phone!": Gendered Geographies of Cell Phone Usage and Significance for Rural Kenyans. *Gender, Technology and Development*, 15(1):1–23. (Cited on page 133.)

Müller, H., Gove, J., and Webb, J. (2012). Understanding Tablet Use: A Multi-method Exploration. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, pages 1–10, New York, NY, USA. ACM. event-place: San Francisco, California, USA. (Cited on pages 133, 138, and 142.)

Nebeling, M., Mintsi, T., Husmann, M., and Norrie, M. (2014a). Interactive Development of Cross-device User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2793–2802, New York, NY, USA. ACM. event-place: Toronto, Ontario, Canada. (Cited on pages 14, 27, 29, and 31.)

Nebeling, M., Teunissen, E., Husmann, M., and Norrie, M. C. (2014b). XDKinect: Development Framework for Cross-device Interaction Using Kinect. In *Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '14, pages 65–74, New York, NY, USA. ACM. event-place: Rome, Italy. (Cited on pages 10, 14, 28, and 31.)

Nittala, A. S., Withana, A., Pourjafarian, N., and Steimle, J. (2018). Multi-Touch Skin: A Thin and Flexible Multi-Touch Sensor for On-Skin Input. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, Montreal QC, Canada. ACM. (Cited on page 183.)

Oakley, I. and Lee, D. (2014). Interaction on the Edge: Offset Sensing for Small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI

'14, pages 169–178, New York, NY, USA. ACM. event-place: Toronto, Ontario, Canada. (Cited on page 96.)

of Edmonton, C. (2019). Office of Emergency Management. (Cited on pages 56 and 68.)

Okerlund, J., Dunaway, M., Latulipe, C., Wilson, D., and Paulos, E. (2018). Statement Making: A Maker Fashion Show Foregrounding Feminism, Gender, and Transdisciplinarity. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pages 187–199, Hong Kong, China. ACM. (Cited on page 183.)

Olberding, S., Yeo, K. P., Nanayakkara, S., and Steimle, J. (2013). AugmentedForearm: Exploring the Design Space of a Display-enhanced Forearm. In *Proceedings of the 4th Augmented Human International Conference*, AH '13, pages 9–12, New York, NY, USA. ACM. event-place: Stuttgart, Germany. (Cited on pages 94 and 97.)

Olsen, Jr., D. R. (2007). Evaluating User Interface Systems Research. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 251–258, New York, NY, USA. ACM. event-place: Newport, Rhode Island, USA. (Cited on pages 27 and 28.)

Oney, S., Harrison, C., Ogan, A., and Wiese, J. (2013). ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2799–2802, New York, NY, USA. ACM. event-place: Paris, France. (Cited on page 95.)

Paay, J., Raptis, D., Kjeldskov, J., Skov, M. B., Ruder, E. V., and Lauridsen, B. M. (2017). Investigating Cross-Device Interaction between a Handheld Device and a Large Display. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 6608–6619, Denver, Colorado, USA. ACM. (Cited on pages 136 and 145.)

Pan, Y. and Blevis, E. (2014). Fashion Thinking: Lessons from Fashion and Sustainable Interaction Design, Concepts and Issues. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS '14, pages 1005–1014, New York, NY, USA. ACM. (Cited on page 183.)

Pan, Y. and Stolterman, E. (2015). What if HCI Becomes a Fashion Driven Discipline? In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2565–2568, Seoul, Republic of Korea. ACM. (Cited on pages 183 and 207.)

Pasquero, J., Stobbe, S. J., and Stonehouse, N. (2011). A Haptic Wristwatch for Eyes-free Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 3257–3266, New York, NY, USA. ACM. (Cited on pages 94 and 96.)

Pece, F., Zarate, J. J., Vechev, V., Besse, N., Gudozhnik, O., Shea, H., and Hilliges, O. (2017). MagTics: Flexible and Thin Form Factor Magnetic Actuators for Dynamic and Wearable Haptic Feedback. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 143–154, Québec City, QC, Canada. ACM. (Cited on page 182.)

Perrault, S. T., Lecolinet, E., Eagan, J., and Guiard, Y. (2013). Watchit: Simple Gestures and Eyes-free Interaction for Wristwatches and Bracelets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1451–1460, New York, NY, USA. ACM. event-place: Paris, France. (Cited on pages 94 and 96.)

Perry, R. W. and Lindell, M. K. (2003). Preparedness for emergency response: guidelines for the emergency planning process. *Disasters*, 27(4):336–350. (Cited on page 55.)

Pohl, N., Hodges, S., Helmes, J., Villar, N., and Paek, T. (2013). An Interactive Belt-worn Badge with a Retractable String-based Input Mechanism. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1465–1468, New York, NY, USA. ACM. event-place: Paris, France. (Cited on page 100.)

Posch, I. and Fitzpatrick, G. (2018). Integrating Textile Materials with Electronic Making: Creating New Tools and Practices. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 158–165, Stockholm, Sweden. ACM. (Cited on page 182.)

Poupyrev, I., Gong, N.-W., Fukuhara, S., Karagozler, M. E., Schwesig, C., and Robinson, K. E. (2016). Project Jacquard: Interactive Digital Textiles at Scale. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4216–4227, New York, NY, USA. ACM. (Cited on page 182.)

Prante, T., Streitz, N. A., and Tandler, P. (2004). Roomware: computers disappear and interaction evolves. *Computer*, 37(12):47–54. (Cited on page 36.)

Qin, Y., Liu, J., Wu, C., and Shi, Y. (2012). uEmergency: A Collaborative System for Emergency Management on Very Large Tabletop. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 399–402, New York, NY, USA. ACM. event-place: Cambridge, Massachusetts, USA. (Cited on page 57.)

Ramakers, R., Schöning, J., and Luyten, K. (2014). Paddle: Highly Deformable Mobile Devices with Physical Controls. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 191–192, New York, NY, USA. ACM. event-place: Toronto, Ontario, Canada. (Cited on pages 94 and 98.)

Rekimoto, J. (1997). Pick-and-drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, pages 31–39, New York, NY, USA. ACM. event-place: Banff, Alberta, Canada. (Cited on pages 12, 36, 58, and 136.)

Rekimoto, J. (2001). GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. page 21. IEEE Computer Society. (Cited on pages 94 and 96.)

Rekimoto, J. and Saitoh, M. (1999). Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 378–385, New York, NY, USA. ACM. event-place: Pittsburgh, Pennsylvania, USA. (Cited on page 36.)

Resnick, M. and Silverman, B. (2005). Some reflections on designing construction kits for kids. In *Proceedings of the 2005 conference on Interaction design and children*, pages 117–122, Boulder, Colorado. ACM. (Cited on pages 192 and 193.)

Rivera, M. L., Moukperian, M., Ashbrook, D., Mankoff, J., and Hudson, S. E. (2017). Stretching the Bounds of 3d Printing with Embedded Textiles. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 497–508, Denver, Colorado, USA. ACM. (Cited on page 182.)

Roinesalo, P., Lappalainen, T., Colley, A., and Häkkilä, J. (2017). Breaking of the dawn jacket: light in the Arctic Winter. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 232–237, Maui, Hawaii. ACM. (Cited on page 182.)

Rädle, R., Jetter, H.-C., Marquardt, N., Reiterer, H., and Rogers, Y. (2014). HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ITS '14, pages 45–54, New York, NY, USA. ACM. event-place: Dresden, Germany. (Cited on pages 15 and 28.)

Schmidt, D., Seifert, J., Rukzio, E., and Gellersen, H. (2012). A Cross-device Interaction Style for Mobiles and Surfaces. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, pages 318–327, New York, NY, USA. ACM. event-place: Newcastle Upon Tyne, United Kingdom. (Cited on page 11.)

Schreiner, M., Rädle, R., Jetter, H.-C., and Reiterer, H. (2015). Connichiwa: A Framework for Cross-Device Web Applications. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pages 2163–2168, New York, NY, USA. ACM. event-place: Seoul, Republic of Korea. (Cited on page 14.)

Scott, S., Allavena, A., Cerar, K., Franck, G., Hazen, M., Shuter, T., and Colliver, C. (2010). Tabletop Interfaces for Supporting Collaborative Decision Making in Maritime Operations. In *Investigating Tabletop Interfaces to Support Collaborative Decision-Making in Maritime Operations*, Santa Monica, CA, USA. (Cited on page 36.)

Scott, S. D., Grant, K. D., and Mandryk, R. L. (2003). System Guidelines for Co-located, Collaborative Work on a Tabletop Display. In Kuutti, K., Karsten, E. H., Fitzpatrick, G., Dourish, P., and Schmidt, K., editors, *ECSCW 2003*, pages 159–178. Springer Netherlands. (Cited on page 76.)

Seifert, J., De Luca, A., and Conradi, B. (2009). A Context-sensitive Security Model for Privacy Protection on Mobile Phones. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages

68:1–68:2, New York, NY, USA. ACM. event-place: Bonn, Germany. (Cited on pages 131 and 134.)

Seyed, T., Azazi, A., Chan, E., Wang, Y., and Maurer, F. (2015). SoD-Toolkit: A Toolkit for Interactively Prototyping and Developing Multi-Sensor, Multi-Device Environments. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*, ITS '15, pages 171–180, New York, NY, USA. ACM. event-place: Madeira, Portugal. (Cited on page 68.)

Seyed, T., Burns, C., Costa Sousa, M., Maurer, F., and Tang, A. (2012). Eliciting Usable Gestures for Multi-display Environments. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 41–50, New York, NY, USA. ACM. event-place: Cambridge, Massachusetts, USA. (Cited on pages 8, 12, 23, 26, 34, 49, 53, and 58.)

Seyed, T., Costa Sousa, M., Maurer, F., and Tang, A. (2013). SkyHunter: A Multi-surface Environment for Supporting Oil and Gas Exploration. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 15–22, New York, NY, USA. ACM. event-place: St. Andrews, Scotland, United Kingdom. (Cited on pages 24, 25, 28, 29, 54, 59, and 68.)

Seyed, T., Yang, X.-D., and Vogel, D. (2016). Doppio: A Reconfigurable Dual-Face Smartwatch for Tangible Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4675–4686, New York, NY, USA. ACM. (Cited on page 135.)

Shaer, O., Kol, G., Strait, M., Fan, C., Grevet, C., and Elfenbein, S. (2010). G-nome Surfer: A Tabletop Interface for Collaborative Exploration of Genomic Data. In *Proceedings of the*

SIGCHI Conference on Human Factors in Computing Systems, CHI '10, pages 1427–1436, New York, NY, USA. ACM. event-place: Atlanta, Georgia, USA. (Cited on page 37.)

Shaer, O., Strait, M., Valdes, C., Feng, T., Lintz, M., and Wang, H. (2011). Enhancing Genomic Learning Through Tabletop Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2817–2826, New York, NY, USA. ACM. event-place: Vancouver, BC, Canada. (Cited on page 37.)

Shingler, B. (2014). Cops with cameras: Growing number of police armed with recording devices. *canada.com*. (Cited on page 69.)

Siek, K. A., Rogers, Y., and Connelly, K. H. (2005). Fat finger worries: how older and younger users physically interact with PDAs. pages 267–280. Springer-Verlag. (Cited on page 94.)

Silver, N. (2012). *The Signal and the Noise: Why So Many Predictions Fail-but Some Don't*. Penguin Books, 1 edition edition. (Cited on page 70.)

Snibbe, S. S. and Raffle, H. S. (2009). Social Immersive Media: Pursuing Best Practices for Multi-user Interactive Camera/Projector Exhibits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1447–1456, New York, NY, USA. ACM. event-place: Boston, MA, USA. (Cited on page 11.)

Steenson, M. and Donner, J. (2009). Beyond the personal and private: Modes of mobile phone sharing in urban India. *The reconstruction of space and time: Mobile communication practices*, 1:231–250. (Cited on page 133.)

Stefik, M., Foster, G., Bobrow, D. G., Kahn, K., Lanning, S., and Suchman, L. (1987). Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. *Commun. ACM*, 30(1):32–47. (Cited on page 36.)

Streitz, N., Prante, T., Müller-Tomfelde, C., Tandler, P., and Magerkurth, C. (2002). Roomware: The Second Generation. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, pages 506–507, New York, NY, USA. ACM. event-place: Minneapolis, Minnesota, USA. (Cited on page 13.)

Streitz, N. A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R. (1999). i-LAND: An Interactive Landscape for Creativity and Innovation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 120–127, New York, NY, USA. ACM. event-place: Pittsburgh, Pennsylvania, USA. (Cited on pages 13 and 36.)

Sultanum, N., Sharlin, E., Sousa, M. C., Miranda-Filho, D. N., and Eastick, R. (2010). Touching the Depths: Introducing Tabletop Interaction to Reservoir Engineering. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 105–108, New York, NY, USA. ACM. event-place: Saarbrücken, Germany. (Cited on page 34.)

Sultanum, N., Somanath, S., Sharlin, E., and Sousa, M. C. (2011). "Point It, Split It, Peel It, View It": Techniques for Interactive Reservoir Visualization on Tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, pages 192–201, New York, NY, USA. ACM. event-place: Kobe, Japan. (Cited on page 37.)

SUN, C., News, Calgary, Share Calgary police dash cams show crashes, a. T. P. G. P. R. L. E., Tumblr, Pinterest, Plus, G., Reddit, LinkedIn, Email, signs, N. M. d. c. b. g. b. s., addiction, N. r. a. c. a. c. v. t. h. w. o., crimes, C. r. g. a. d. a. t. h. p. t. f. o., and woman, C. d. w. f. t. c. m. a. o. s. a. (2013). Calgary police dash cams show crashes, arrests | Calgary Sun. (Cited on page 69.)

Tani, M., Horita, M., Yamaashi, K., Tanikoshi, K., and Futakawa, M. (1994). Courtyard: Integrating Shared Overview on a Large Screen and Per-user Detail on Individual Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 44–50, New York, NY, USA. ACM. event-place: Boston, Massachusetts, USA. (Cited on page 36.)

Tena, S., Aedo, I., and Díaz, P. (2013). Achieving Shared Understanding in Face-to-face 'Tabletop' Exercises. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*, BCS-HCI '13, pages 61:1–61:2, Swinton, UK, UK. British Computer Society. event-place: London, UK. (Cited on page 55.)

Torán, M. M., Bonillo, Alicia, and Espí, Emilio (2018). Future Trends about Fashion and Technology: A Forward Planning. *Current Trends in Fashion Technology & Textile Engineering*, 2(2). (Cited on page 179.)

Touretzky, D. S. (2014). Teaching Kodu with Physical Manipulatives. *ACM Inroads*, 5(4):44–51. (Cited on page 212.)

Vaccaro, K., Agarwalla, T., Shivakumar, S., and Kumar, R. (2018). Designing the Future of Personal Fashion. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 627:1–627:11, New York, NY, USA. ACM. (Cited on page 179.)

Van Buskirk, R. and LaLomia, M. (1995). A Comparison of Speech and Mouse/Keyboard GUI Navigation. In *Conference Companion on Human Factors in Computing Systems*, CHI '95, pages 96–, New York, NY, USA. ACM. event-place: Denver, Colorado, USA. (Cited on page 94.)

Vogel, D. and Balakrishnan, R. (2004). Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 137–146, New York, NY, USA. ACM. event-place: Santa Fe, NM, USA. (Cited on page 11.)

von Zadow, U., Büschel, W., Langner, R., and Dachselt, R. (2014). SleeD: Using a Sleeve Display to Interact with Touch-sensitive Display Walls. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ITS '14, pages 129–138, New York, NY, USA. ACM. event-place: Dresden, Germany. (Cited on page 13.)

Vossoughi, S. and Bevan, B. (2014). "Making and Tinkering: A Review of the Literature.". *National Research Council Committee on Out of School Time STEM*, pages 1–55. (Cited on pages 192 and 193.)

Wagner, J., Nancel, M., Gustafson, S. G., Huot, S., and Mackay, W. E. (2013). Body-centric Design Space for Multi-surface Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1299–1308, New York, NY, USA. ACM. event-place: Paris, France. (Cited on page 13.)

Walle, B. A. V. D. and Turoff, M. (2007). Emergency Response Information Systems: Emerging Trends and Technologies. 50(3):28–65. Google-Books-ID: RwflSAAACAAJ. (Cited on page 52.)

Wang, Y., Luo, S., Lu, Y., Gong, H., Zhou, Y., Liu, S., and Hansen, P. (2017). AnimSkin: Fabricating Epidermis with Interactive, Functional and Aesthetic Color Animation. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, pages 397–401, Edinburgh, United Kingdom. ACM. (Cited on pages 180 and 183.)

Want, R., Hopper, A., Falcão, V., and Gibbons, J. (1992). The Active Badge Location System. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102. (Cited on page 11.)

Weigel, M., Nittala, A. S., Olwal, A., and Steimle, J. (2017). SkinMarks: Enabling Interactions on Body Landmarks Using Conformal Skin Electronics. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3095–3105, Denver, Colorado, USA. ACM. (Cited on page 183.)

Weiser, M. (1999). The Computer for the 21st Century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11. (Cited on pages 1, 8, 10, and 257.)

Wigdor, D., Jiang, H., Forlines, C., Borkin, M., and Shen, C. (2009). WeSpace: The Design Development and Deployment of a Walk-up and Share Multi-surface Visual Collaboration System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1237–1246, New York, NY, USA. ACM. event-place: Boston, MA, USA. (Cited on page 36.)

Wilson, A. D. and Benko, H. (2010). Combining Multiple Depth Cameras and Projectors for Interactions on, Above and Between Surfaces. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 273–282, New York, NY, USA. ACM. event-place: New York, New York, USA. (Cited on page 13.)

Wyld, J. and Dierking, L. D. (2015). Design, Make, Play: Growing the Next Generation of STEM Innovators, edited by Margaret Honey and David E. Kanter. Routledge, New York, NY, USA, 2013. xvii + 238 pp. ISBN 978-0-415-53920-3. *Science Education*, 99(4):779–782. (Cited on pages 192 and 193.)

Xiao, R., Laput, G., and Harrison, C. (2014). Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. pages 193–196. ACM. (Cited on pages 94, 96, and 121.)

Xu, C. and Lyons, K. (2015). Shimmering Smartwatches: Exploring the Smartwatch Design Space. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, pages 69–76, New York, NY, USA. ACM. event-place: Stanford, California, USA. (Cited on page 97.)

Yang, J. and Wigdor, D. (2014). Panelrama: Enabling Easy Specification of Cross-device Web Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2783–2792, New York, NY, USA. ACM. event-place: Toronto, Ontario, Canada. (Cited on pages 14, 27, and 31.)

Zamel, N., Pita, J., and Dogru, A. (2001). Next-Generation Visualization Technologies for Exploration & Production. In *SPE-68099-MS*, page 5, SPE. Society of Petroleum Engineers. (Cited on page 38.)